Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Analysis and Evaluation of Aware: A Context Awareness Framework.

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Ana Elena Loza Quijada

DIRECTOR: Roc Meseguer

DATE: December 10, 2014

Title: Analysis and Evaluation of Aware: A Context Awareness Framework.

Author: Ana Elena Loza Quijada

Director: Roc Meseguer

Date: December 10, 2014

Overview

This thesis goes into context aware systems, specifically studies the AWARE framework and the information stored by this tool.

The initial objective is the understanding of the term context awareness and the different types of context.

A second objective is the study of the AWARE framework and the parts that compose it, which includes the client application (configuration and sensor information) and the AWARE server (database and web services). This study doesn't include any further programming of the sensors or the creation of new plug-ins.

And as a final objective will be to test whether or not the AWARE framework is able to provide context aware information that allows the user to monitor, study and reconstruct a given activity. For this, a test was performed using three different users to recreate an ongoing activity in Barcelona. Also a test varying the location sensor settings was performed, to help improve the accuracy of the information offered by this sensor.

INDEX

INTRODUCTION

The following Master Thesis is the study of the AWARE Framework and context aware systems.

AWARE is a tool for smart devices that registers and stores mobile context information, which can be related to software, hardware and human based data. The framework consists in a server and a client application.

The client application allows the user to share context information by storing it in the device, processing it to be understandable and then sending it to the AWARE server. The server manages a dashboard that allows the user to interact with other users and request for information. It also contains a database that stores the information to carry on further studies.

Context aware systems like the one AWARE framework offers can be used for multiply applications like the improvement of the transport system, by recognizing blocked highways and reorganizing the traffic flow; or the creation of interactive systems to help improve learning methods, like the audio guides in museums.

The initial objective of this master thesis is the understanding of the term context awareness and the different types of context.

A second objective will be the study of the AWARE framework and the parts that compose it; server, client and sensors. This study doesn't include modifying any of this sensors nor creating a new AWARE plug-in.

And as a final objective will be to test whether or not the AWARE framework is able to provide context aware information that allows the user to monitor, study and reconstruct a given activity.

CHAPTER 1. CONTEXT AWARE SYSTEMS

The following chapter will describe the terms context and context awareness, and will also define some context categories that will be taken into account in the rest of the document.

1.1. Introduction

The introduction of smart devices in the world has facilitated the connection of people and devices in an easier and faster way. The amount of information that can be obtained with the smart devices is high and can be used to improve many aspects of a person regular life and also the services that can be offered to any user. Most of this information can be collected without the user noticing and a way to identify this information can be by using the term context. Many definitions can be found for the word context, but two of them are more relevant for this document and can be easily understood and identified.

Abowd and Mynatt [1]defined context using the five Wôsò of context (Who, What, Where, When, Why) as a good minimal set of necessary context, referring to the many pieces of information that are present in the environment and can be found by answering this five questions.
Dey and Abowd[2] defined the term context as: ñContext is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.ò This definition is more appropriate to this document because it gives a general understanding of context and it complements well with the term context awareness.

1.2. Context awareness

Context awareness is a term that was first introduced in 1994 by Schilit and Theimer[3]. Their definition was related to computer systems and was given as: ñsoftware that ñadapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time.ò After this many researchers have defined it and have failed to give a definition able to identify properly context aware systems, because of being too specific.

The definition that this document will use is the one given by Dey, which is a general definition that can be use to easily identify if a system applies to be a context aware system or not: ñA system is context-aware if it uses context to

---

[1] Gregory D. Abowd and Elizabeth D. Mynatt, ñCharting Past, Present, and Future Research in Ubiquitous Computingò, Georgia Institute of Technology
[2] Dey, A.K. Abowd, G.D. ñTowards a Better Understanding of Context and Context-Awarenessò CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness (2000)
[3] Schilit, B., Theimer, M. ñDisseminating Active Map Information to Mobile HostsòIEEE Network, 8(5) (1994) 22-32

provide relevant information and/or services to the user, where relevancy depends on the userôs taskö.

To complement this definition is important to also mention three categories of features that a context-aware application can support according to Dey:

Å Presentation of information and services to a user.
Å Automatic execution of a service for a user.
Å Tagging of context to information to support later retrieval.

A context-aware system then should be able from the first feature to present to the user the information that the user wants to see in the moment the user wants to see it, according to the user environment. Also from the second feature it should be able to execute an action when this action is required, also by using the user context. And at last, it should be able to maintain all the information organized by using tags on the data recollected by the sensors.
In further chapters, the system that will be used will be also evaluated using these features.

1.3. Context types

Different context types have been defined using different perspectives. Like it was done before with the terms context and context awareness, two of these definitions will be described in order to understand better the concept and to be used later in this document.
Schilit categorized context into three categories using a conceptual categorization based technique on three common questions that can be used to determine the context.

Å Where you are: This includes all location related information.
Å Who you are with: The information about the people present around the user.
Å What resources are nearby: This includes information about resources available in the area where the user is located.

Dey and Abowd defined primary and secondary context types. An approach similar to this one is the one that will be taken into account in this document.

- Primary context is identified as location, identity, time, and activity.
- Secondary context is the one that can be found using primary context.

CHAPTER 2. AWARE FRAMEWORK

The following chapter will briefly describe the software used to develop the project and its characteristics.

2.1. Introduction

AWARE is an Android framework dedicated to instrument, infer, log and share mobile context information. The framework is oriented to application developers, researchers and mobile phone users. AWARE is designed as a plug-in architecture which allows anyone to cooperate, using AWARE Add-ons. AWARE Add-ons can be of two kinds:

- Context Sensors: their function is to collect and abstract data to generate context. Context Sensors do not have an interface for the user and are unobtrusive to the mobile device and the user.
- Context Plug-in: their function is to present and explain context to the user or other researchers. Context Plug-in allow users and researchers to benefit from the added context capabilities and provide a user interface for interaction with or presentation of context information, thus supporting context intelligibility and context accountability.

AWARE can be divided in two main sections, which are the AWARE server and the AWARE CLIENT.

2.2. AWARE Client

AWARE Client allows researchers to control the Context Sensors and Plug-in to different requirements. Users can modify Context Plug-in settings or visualise context data in available user interfaces.
When the user installs the AWARE Client, two new icons are available: AWARE Client and AWARE Add-ons. When AWARE Client is launched, a dashboard will be displayed and provide the possibility to enable/disable the core data sensors. The data that is senses is stored locally on the deviceȭs external storage and it is not shared with remote devices or servers.

2.2.1. Requirements

To install the AWARE Client is necessary to have an Android 2.3+ phone or tablet. The installer can be found at: http://www.awareframework.com/home/.

2.2.2. Sensors

The aware application offers initially several sensors. In addition to these sensors, it also offers some plug-ins that can be downloaded according to the user needs.

- Wi-Fi: logs the mobile device's Wi-Fi sensor and detects surrounding Wi-Fi visible devices with respective RSSI values at specified intervals (default is 60 seconds). A scan session assigns the same timestamp for all the found Wi-Fi devices.

- Temperature: measures the ambient air temperature, in Celsius (°C). This sensor is available according to the device hardware.

- Telephony: provides information on the mobile phone capabilities of the device, connected cell towers and neighbouring towers.

- Screen: monitors the screen statuses, such as turning on and off, locked and unlocked.

- Rotation: measures the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle θ around an axis (x, y, or z).

- Proximity: measures the distance to an object in front of the mobile device. Depending on the hardware, it can be in centimetres or binary.

- Processor: measures the mobile phone's processor load. It provides the average processors load (for multi-core devices) dedicated to the user processes, system processes and idle. It also broadcasts when the processor is under stress or relaxed.

- Barometer: measures the ambient air pressure. Barometer can be leveraged to detect and predict short team changes in weather, for example drops in pressure indicate rain, while raises indicate good weather ahead.

- Network: provides information on the network sensors availability of the device. These include use of airplane mode, Wi-Fi, Bluetooth, GPS, mobile and WIMAX status and internet availability. This sensor can be leveraged to detect the availability of wireless sensors and internet on the device at any time. Moreover, this sensor also provides mobile and Wi-Fi interval traffic, in bytes and packets.

- Magnetometer: measures the geomagnetic field strength around the device. It lets you monitor changes in the Earth's magnetic field. This sensor provides raw field strength data (in μT) for each of the axis.

- Locations: provides the best location estimate for the users' current location, automatically. There is a built-in algorithm that provides the user's location with a minimum battery impact using values like the cached location, the network location and the GPS location.

- Linear Accelerometer: measures the acceleration applied to the sensor built-in into the device, excluding the force of gravity, in m/s^2. The linear acceleration sensor always has an offset, which the user needs to remove. The simplest way to do this is to build a calibration step into the application. During calibration the user can be asked to set the device on a table, and then read the offsets for all three axes. Then that offset can be subtracted from the acceleration sensor’s direct readings to get the actual linear acceleration.

- Light: measures the ambient light. It can be used to detect indoor or outdoor light conditions.

- Gyroscope: measures the rate or rotation in rad/s around a device’s x, y and z axis. Rotation is positive in the counter-clockwise direction; that is, an observer looking from some positive location on the x, y or z axis at a device positioned on the origin would report positive rotation if the device appeared to be rotating counter clockwise. Gyroscope noise and drift errors need to be compensated for.

- Gravity: measures the force of gravity applied to the sensor built-in into the device and provides a three dimensional vector indicating the direction and magnitude of gravity (in m/s^2). When a device is at rest, the gravity sensor should measure equally as the accelerometer.

- ESM: adds support for user-provided context by leveraging mobile Experience Sampling Method (ESM). The ESM questionnaires can be triggered by context, time or on-demand, locally or remotely (with AWARE Server Dashboard). Although user-subjective, this sensor allows crowd sourcing information that is challenging to instrument with sensors.

- Communication: logs communication events such as calls, messages, calling availability and actions. This sensor does not record personal information, such as phone numbers or contact information. Instead, an unique ID is assigned that is SHA-1 encrypted but it is always the same for the same source.

- Bluetooth: logs the mobile device’s Bluetooth sensor and detects surrounding Bluetooth-enabled and visible devices with respective RSSI values at specified intervals (default is 60 seconds). A scan session assigns the same timestamp for all the found Bluetooth devices.

- Battery: monitors battery information and monitors power related events. This sensor provides user-driven contexts, such as initiating a charge and unplugging the device.

- Applications: logs application and notifications usage on the device. It captures every time the user changes application on the foreground and on the background and any new application’s notifications.

- Accelerometer: measures the acceleration applied to the sensor built-in into the device, including the force of gravity. In other words, the force of gravity is always influencing the measured acceleration, thus when the device is sitting on a table, the accelerometer reads the acceleration of gravity: 9.81 m/s². Similarly, if the phone is in free-fall towards the ground, the accelerometer reads: 0 m/s².

2.3. AWARE Server

The AWARE Server offers different purposes: replicate, command, exchange and visualise context data. It is divided in two different parts:

- AWARE Server Web services: Synchronises the local data with a remote database with AWARE Server web services. The synchronisation can be scheduled, triggered locally with an event or requested remotely.
- AWARE Server MQTT.

2.3.1. Requirements

To configure the AWARE Server is required to have a server with PHP and MySQL installed. The configuration files can be found in: https://api.awareframework.com/releases/AWARE-WS-2.1.6.zip .

2.3.2. MQTT

AWARE has a built-in MQTT client for sharing information in real-time to other AWARE Clients and AWARE Servers. It can be used to exchange context with other devices, remotely manage sensors or issue ESMs questionnaires on devices.

MQTT stands for MQ Telemetry Transport. MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

By default, when activating the AWARE's MQTT client, it subscribes to three topics (broadcasts, ESM and configuration), under its own MQTT Device ID. They offer the following functionalities:

- MQTT Device ID/broadcasts: receives broadcast actions and broadcasts on the device.
- MQTT Device ID/esm: queues an ESM.
- MQTT Device ID/configuration: remotely activate/deactivate sensors.

CHAPTER 3. AWARE SETUP

To be able to start using AWARE there are some important steps that must be followed. These steps are divided in two main sections: the server and the client.

3.1. Server Setup

As said previously, to configure the AWARE Server is required to have a server with PHP and MySQL installed. A new database needs to be created in order to be used to store the information obtained from the AWARE client. This database can be created following the instructions given by the server that it is installed and it must have an user for the client with all the permissions to create, delete and modify tables.

After downloading and decompressing the files obtained in the AWARE website, there will be two main folders. The first folder is exclusive for MAC systems and the second one is for the rest of the available systems. Inside both of these folders there will be the files needed for the configuration of both the MQTT server and the Web Services Server.

3.1.1. MQTT

Inside the folder called MQTT there will be the files of the IBM's MQTT Real Small Message Broker (RSMB), which were downloaded from the official IBM site. Depending on the operating system, over which the server is installed, there will be different broker files. The corresponding file must be executed in order to start the MQTT server. The configuration files clients.auth and config.cfg must be copied to the server folder.

Inside the configuration file clients.auth must be all the usernames and passwords that will be allowed by the MQTT server. By default the username and password is ñdemo:passò. This default value should be modified and new users must be present in this file.

The configuration file config.cfg contains the settings for the server. The values in this file can be left as default for the setup of the AWARE server.

3.1.2. Web Services

The folder called AWARE contains the configuration files of the AWARE web services server and needs to be placed inside an HTTP accessible folder on the server that it is already installed. Inside the configuration files, there are three files that need to be modified:

- The first file that needs to be modified itõs located in the "application/config/" and it is called database.php. In this file, the information must be changed according the database that was created. Every ñYOURDATABASEò must be replaced by the name of the MYSQL database assigned to the aware. The text ñYOURDATABASEHOSTò must be replaced by the IP address or hostname of the server. Finally the texts ñYOURDATABASEUSERò and ñYOURDATABASEPASSWORDò must be replaced according to the username and password of the client. The details of these changes are shown in the following image:

```
$active_group = 'YOURDATABASE';
$db['YOURDATABASE']['hostname'] = 'YOURDATABASEHOST';
$db['YOURDATABASE']['port'] = '3306';
$db['YOURDATABASE']['username'] = 'YOURDATABASEUSER';
$db['YOURDATABASE']['password'] = 'YOURDATABASEPASSWORD';
$db['YOURDATABASE']['database'] = 'YOURDATABASE';
```

```
$active_group = 'aware';
$db['aware']['hostname'] = '192.168.1.131';
$db['aware']['port'] = '3306';
$db['aware']['username'] = 'demo';
$db['aware']['password'] = '123456';
$db['aware']['database'] = 'aware';
```

Fig. 3.1.1 AWARE database configuration

As seen in the image, the database is called ñawareò and it is located on a server with an IP address 192.168.1.131 using the port 3306. The username of the client is ñdemoò and its password is ñ123456ò.

- In the same location there is a file called ñconfig.phpò. In this file the variable $config['encryption_key']='YOURENCRYPTIONKEY' must be changed to protect the web services aware dashboard from being access by unauthorized users. In this case the text ñYOURENCRYPTIONKEYò was modified using a encryption key obtained by a random key generator.

- Lastly, in the location "application/controllers/ò, some parameters in the file ñaware_ws.phpò must be also modified according to the MYSQL database and the server configuration. The text ñYOURDATABASEò must be changed to the name of the database. An user with a password must be assigned to grant access to the AWARE dashboard, this is done by modifying ñYOURADMINISTRATIONUSERò and ñYOURADMINISTRATIONPASSWORDò. The information of the MQTT server and its clients should be also modified. The details of these changes are shown in the following image:

```
var $database = 'YOURDATABASE';
var $ws_user = 'YOURADMINISTRATIONUSER';
var $ws_pass = 'YOURADMINISTRATIONPASSWORD';
var $mqtt_server = 'YOURHOST';
var $mqtt_port = '1883';
var $mqtt_user = 'demo';
var $mqtt_pwd = 'pass';
```

```
var $database = 'aware';
var $ws_user = 'root';
var $ws_pass = '123456';
var $mqtt_server = '192.168.1.131';
var $mqtt_port = '1883';
var $mqtt_user = 'demo';
var $mqtt_pwd = '123456';
```

Fig. 3.1.2 AWARE MQTT configuration

As said before, the name of the database is aware. The administration user is called ñrootò and its password to access the AWARE dashboard is ñ123456ò. The MQTT is located in the same server 192.168.1.131 using the port ñ1883ò and the client ñdemoò can access it using the password ñ123456ò.

After modifying these files the Web Services server is ready to work. The dashboard can be accessed using the address http://YOURHOST/AWARE/index.php/aware_ws/dashboard and the administration user. The status of the web server can be seen at http://YOURHOST/AWARE/index.php. According to the configuration done before, the dashboard can be accessed using the URL address http://192.168.1.131/AWARE/index.php/aware_ws/dashboard and the administrator user ñrootò with the password ñ123456ò.

3.2. Client Setup

Once the MQTT server and the Web Services Server have been configured and are working, the client setup can start. For this, the AWARE application must be installed in the device that will be used to gather context aware information.

To setup the AWARE in a device there are two parts inside the application that must be modified, the information about the MQTT Server and the information about the Web Services Server. First, inside the option ñAWARE Server MQTTò, the following parameters must be changed according to the previous server configuration:

- AWARE Server MQTT server
- MQTT port
- MQTT username
- MQTT password

The details of these changes are shown in the following image:

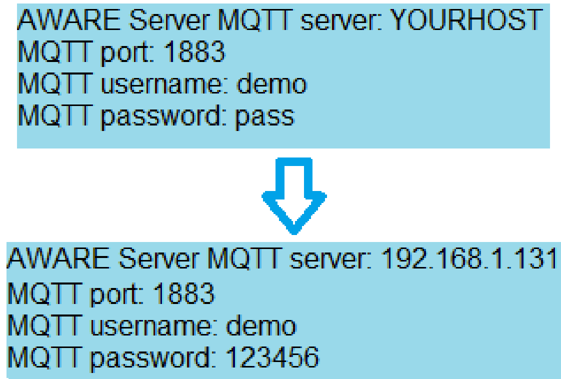


Fig. 3.2.1 AWARE client configuration

After this, the connection with the MQTT server can be established by selecting the option ñActiveò. When the client is connected there will be a message in the broker and a MQTT ID will be assigned to the device.

Secondly, inside the section "AWARE Server Webservices" the parameter with the same name must be modified according to the address of the server. In this case the URL address will be changed by http://192.168.1.131/AWARE/index.php/aware_ws/index.

After this, the connection to the Web Services Server will be established by selecting the option ñActiveò. Once the client is connected, it will be able to create tables and post data to the MYSQL database.

The sensors can be activated by pressing the desired one and selecting the option activate. This can be done before or after the configuration and establishment of the connections.

3.3. Possible Setup Difficulties

When setting up the AWARE server and client, there can be some difficulties according to the version and configuration of the server. There is a few difficulties that were identified independently of which server version was used, this difficulties are:

- In the configuration file ñdatabase.phpò there is a miss typing in the line where the database port is specified, it must be written using the following format: $db['YOURDATABASE']['port'] = ó3306ô.
- The server must allow access to some file locations to the administrator and client, this file locations can be found in the error log of the server. This can be fixed by adding some lines into the configuration file of the server according to the version of it. For example in the WAMP Server 2.2 used, the following lines were added to the configuration file to fix this problem:

  </Directory>
  <Directory "C:/wamp/www/AWARE/application/views/css/">

```
Order allow,deny
Allow from all
</Directory>
<Directory "C:/wamp/www/AWARE/application/views/">
Order allow,deny
Allow from all
</Directory>
```

- In some cases, the firewall of the Windows must be disabled in order to allow the client to push the data into the database.
- The installation of the application in the client has given some problems, to solve this issue the application can be installed from the Android SDK. The instructions and files to perform this installation can be found in the AWARE website.

CHAPTER 4. CONTEXT AWARENESS ORIENTED TO INTERACTIVE ACTIVITIES

As presented before there are many applications for context aware systems, one of them could be related to activities like interactive games in which certain information about the participants can be useful for posterior studies. In this chapter, one of these activities will be introduced along with possible context aware applications.

4.1. Activity: ñBarcelonadaò

In Cataluña there is an event that it is organized with the purpose of helping students from different towns of the region to move easily around the city of Barcelona. This event consists in dividing the students in different groups and providing them with cards that represent the places they must find and visit. After visiting a location they must take a picture and upload so it can be shared with the rest of the people. This is all done with the help of different mobile phones applications that are available to the public.

4.1.1. General Instructions

Students are divided into groups and are given 6 cards that represent the places they must visit. From these 6 cards one can be discarded and also one can be traded with other groups to make the trip more convenient. After this, with the 5 definite sites, a route is made containing at least one metro trip.

When the groups arrive to Barcelona they must follow the already planned itinerary and take a picture in every site. The picture must be posted on Instagram with the corresponding hash tags, place information and location. In one of the sites the students must make a video with an explanation of the site and upload it to Instagram as well. The activity is completed after all the pictures/video is uploaded and the groups go back to the meeting place.

After this, as a post activity task, the students must answer a poll that will be found in the mobile application Socrative or through the website http://www.socrative.com.

4.1.2. Location cards

An important part of the event is the cards used to provide students with information about the sites they have to visit. The cards are divided into two groups: ñordinary cardsò and ñwildcardsò, where the first group represents places of interest to visit like monuments and emblematic places and the second group are restaurants, bars, etc.

The structure of the cards consists in a front side with the information about the place and a back side with a QR code. The QR code provides the location in Google maps of the site to visit. To access this QR code there is a mobile application suggested in the point 1.1.2.

4.1.3. Mobile applications and websites of interest

Since the activity is developed using the help of a mobile phone it is important to know which applications are used for it. The applications used in previous occasions for this activity are the following:

Instagram: Instagram is an application to capture, customize and share photos and videos. The photos and videos can be shared with friends and followers through the photo feed or by sending them directly. They can also be shared publicly using hashtags. A hashtag is word or phrase without spaces with the number sing (#) that can be used to return a set of elements (message, pictures, etc.) containing it.
This application is used in the activity to upload the pictures and the video of the visited sites.

Life 360: Family Locator by Life360 is an application that uses GPS tracking technology and groups called ñCirclesò to allow users to find the location of the members of these ñCirclesò on a private map. It also allows chat with everyone within each of the ñCirclesò as well as alarms for when someone reach a destination and the ability to track lost devices.
This application is used in the activity to keep track of where the students are located at all times, this information is lost at the end of the activity.

TMB virtual: TMB Virtual is an application that allows the user to access all the information on Barcelona metro and bus services. It shows the user locations, where the nearest metro stations are located and the timeline of the buses.
This application is used in the activity as a tool to help the students to arrive to the different sites they need to visit.

Socrative Student: Socrative student is a tool that uses quizzes, quick question polls and other kind of formative assessment questions, to visualize the students understanding. The students join a teacherôs room using an unique code, this way no students accounts are needed.
This application is used as a post task of the activity where the students are asked to answer to a poll inside this app.

QR code reader: A QR code reader is required in this activity to scan and retrieve the information given by a QR code located in the cards.

4.2. AWARE Sensors

In order to incorporate AWARE into this types of events, for example the ñBarcelonadaò, itôs necessary to explore which AWARE sensors can help gather useful information.

4.2.1. Linear Accelerometer

To help identify which transportations means are being taken by the students, the acceleration which they are going through can be used. Aware offers a sensor that measures the linear acceleration of the device excluding the force of gravity. This sensor returns three different values corresponding to the three different axes, with an offset that can be calibrated by measuring the acceleration without any movement and then subtracting this value from posterior measurements. The way the axes are positioned is shown in the following image:

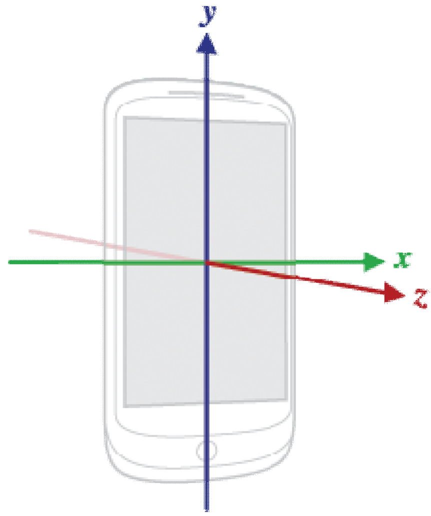


Fig. 4.2.1 Linear Accelerations Axes (http://www.awareframework.com/)

This sensor stores the following information in the database:

Table 1 Linear accelerometer information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.
Device ID	Contains the AWARE device ID that collected this data.
Double Values 0	Contains the acceleration along the x axis (m/s^2).
Double Values 1	Contains the acceleration along the y axis (m/s^2).
Double Values 2	Contains the acceleration along the z axis (m/s^2).
Accuracy	Contains the constant given by the sensor manager that represents the sensor accuracy level.

4.2.2. Application

As a way to identify which applications are being used and at what time, the application sensor can be useful. The application sensor stores different information related to the applications usage. The information retrieved and store can be selected separately by activating the different options the sensor offer, this options are explained briefly following.

Foreground

This sensor register and stores the applications that are in-use, for example the applications the user is interacting with. It contains the following information:

Table 2 Application foreground information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.

Device ID	Contains the AWARE device ID that collected this data.
Package name	Contains the application's package name.
Application name	Contains the application's localized name.
IS System App	Contains whether it is or not a device's pre-installed application.

History

This sensor registers and stores the application usage status, for example foreground or background. It contains the following information:

Table 3 Application history information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.
Device ID	Contains the AWARE device ID that collected this data.
Package name	Contains the application's package name.
Application name	Contains the application's localized name.
Process Importance	Contains one of the Android process importance values.
Process ID	Contains the ID of the application on Android's process stack. It is used to identify ended application sessions.
End Timestamp	Contains the time instance of when the application terminated or a change in the process importance.
IS System App	Contains whether it is or not a device's pre-installed application.

Notifications

This sensor stores the application notifications. It contains the following information:

Table 4 Application notifications information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.
Device ID	Contains the AWARE device ID that collected this data.
Package name	Contains the application's package name.
Application name	Contains the application's localized name.
Text	Contains the notification's header text (not the content).
Sound	Contains the notification's sound source (if applicable).
Vibrate	Contains the notification's vibration pattern (if applicable).
Default	Contains if notification was delivered according to device's default settings.
Flags	Contains a flag obtained from the Android's notification flags.

Crashes

This sensor stores the log of crashed applications. It contains the following information:

Table 5 Application crashes information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.
Device ID	Contains the AWARE device ID that collected this data.
Package name	Contains the application’s package name.
Application name	Contains the application’s localized name.
Application version	Contains the application’s version code.
Error short	Contains a short description of the error.
Error long	Contains a longer version of the error description.
Error condition	Contains the error condition. 1 equals code error and 2 equals non-responsive (ANR error).
IS System App	Contains whether it is or not a device’s pre-installed application.

4.2.3. Location

Since an important part of the activity is the movement and location of the students, the location sensor of Aware can be used. To estimate the location of the user, Aware uses an algorithm with a minimum battery impact. This algorithm can be changed according to the needs of the researchers and users, varying factors like the frequency of the location updates and the minimum accuracy. The algorithm can be explained using the following image:

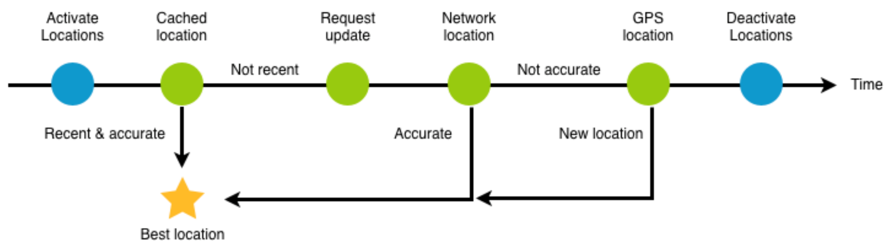


Fig. 4.2.2 Location Algorithm (http://www.awareframework.com/)

As can be seen from the image, a best location is given to the user automatically using the cached location. This location is then updated according to the update frequency using the Network location or the GPS location. These options can be activated separately in the device application.
The following location information can be found in the database:

Table 6 Location information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.
Device ID	Contains the AWARE device ID that collected this data.
Latitude	Contains the location’s latitude (degrees).
Longitude	Contains the location’s longitude (degrees).
Bearing	Contains the location’s bearing (degrees).
Speed	Contains the speed if available (m/s over ground).
Altitude	Contains the altitude in meters above sea level, if available.

Provider	Contains if the information comes from GPS or network location.
Accuracy	Contains the estimated location accuracy.

4.2.4. WIFI

Finally, to help identify the surroundings of the students the WIFI sensor can be used. This sensor gathers information regarding the mobile deviceøs WIFI sensor, the visible WIFI devices and their respective RSSI dB values at specified intervals. It contains the following information:

Table 7 WIFI information

ID	Contains the ID of the database entry.
Timestamp	Contains the time instance of the database entry.
Device ID	Contains the AWARE device ID that collected this data.
BSSID	Contains the deviceøs WIFI MAC address.
SSID	Contains the deviceøs WIFI user assigned name.
Security	Contains the WIFI active security protocol.
Frequency	Contains the WIFI band frequency.
RSSI	Contains the RSSI dB to the scanned device.

4.3. AWARE role inside the ñBarcelonadaò

The next step is to incorporate AWARE into the ñBarcelonadaò. To do this each sensor will play a different role into the recollection of information, this information then we will use to determine or study different factors. The factors that initially are planned to be studied are the following:

- **Tracking of the movement:** One of the weaknesses of the current practices is that the location of the students is being tracked but not being stored. When the activity is done all the information related to the movement of the students during the day of the activity is lost. Using the location sensor of AWARE, this information will be stored in the server so it can be studied and used to improve future activities.

- **Means of transportation:** Another point that could be relevant is which means of transportation the students use during the activity. For this the combination of several sensors can be used. The use of the acceleration can be used to determine the means of transportation with an acceptable accuracy. Combining this method with the location information could also improve the accuracy, as well as using different algorithms with the found information.

  Another alternative could be the use of the WIFI sensor, knowing which networks are available could help recognize if the student is underground (Metro, no WIFI networks visible) or on the surface (BUS, many WIFI networks visible).

- **Development of the tasks:** The tasks related to this activity include taking a picture/video and uploading it to Instagram. Information regarding when the students upload this picture/video and which type of network they were using, could also be relevant to study or improve future activities. Using the application sensor, information regarding to when an application was used will be stored, this information could be used to recognize the parameters previously mentioned.

- The ESM sensor could be used to send information to the students and also to realize the poll that is currently being done using Socrative. This sensor has not been tested in this study given that it doesn't work properly in all the versions of the AWARE framework.

CHAPTER 5. FRAMEWORK PERFORMANCE TEST

To be able to determine if the AWARE framework is able to be used as a tool to monitor and improve the kind of activities previously described, it is important to answer some crucial questions.

5.1. Activity data

One of the key factors of using AWARE into the activity is the data recollected by the sensors of the framework and what are we able to determine using this data: are we able to use this information to reconstruct and monitor the activity?
To answer this question a small recreation of the activity will be carried out in the campus of the university. There will be different users identified from 1-3 and each of them will follow different routes and profiles. The users will be as followed:

Table 8 Activity test profiles

Route	Stages	Speed used	Picture activity
1	SP → 4	Medium	On site.
	4 → 7	Slow	
	7 → 2	Slow	
	2 → 1	Slow	
	1 → 5	Fast	
	5 → SP	Fast	
2	SP → 6	Slow	On the way.
	6→ 4	Slow	
	4 → 5	Medium	
	5 → 1	Slow	
	1→ 3	Medium	
	3 → SP	Slow	
3	SP → 2	Slow	On site.
	2 → 1	Medium	
	1 → 4	Medium	
	4 → 7	Fast	
	7 → 6	Slow	
	6→ SP	Slow	

Where SP indicates a starting point that will be located at the entrance of the C4 building, the numbers in the route correspond to the numbers of the location cards, the speed used corresponds to different speeds on foot, slow: walking, medium: jogging and fast: running; and the picture activity indicates when the picture is being uploaded.
The sensors that will be active during the test will be:

- Applications (usage, notifications and crashes).
- Battery
- Locations
- Linear Accelerometer
- Wi-fi.

5.2. Different location settings

A parameter that can be very critical in the data recollected is the location, if this data is not accurate then it is impossible to reconstruct and monitor where the students are located during the activity. Is there a significant difference in the information gathered if different adjustable settings of the locations sensor are being used?

To answer this question a test will be carried out using 2 points, the starting point and the card #4, varying the locations sensor parameters in the following way:

Table 9 Location settings test profile

Profile	GPS accuracy	Network accuracy	GPS update frequency	Network update frequency	Location expiration time
1	0	1500	180	300	300
2	75	1500	180	300	300
3	150	1500	180	300	300
4	150	0	180	300	300
5	150	750	180	300	300
6	150	1500	0	300	300
7	150	1500	360	300	300
8	150	1500	180	0	300
9	150	1500	180	600	300
10	150	1500	180	300	100
11	150	1500	180	300	200

The only sensor active during this test will be the battery sensor.

5.3. Battery life

Finally, is the battery of the device able to last through the whole activity using the application? To answer this question the battery sensor will be active through all the tests to help estimate if the battery can last through the whole activity.

CHAPTER 6. FRAMEWORK PERFORMANCE RESULTS

In this chapter the results obtained after carrying out the test presented in the previous chapter, will be presented. To do this the results have been divided according to the sensor used to measure them and the route or test that was performed.

6.1. Applications

The applications sensor registers and stores different information related to the usage of the applications. The important data related to this sensor will be the when the camera and Instagram were used during activity, later with this information it should be possible to determine in which parts of the route these applications were used.

6.1.1. Application foreground ï Route 1

The information registered with the application foreground sensor for route 1, can be seen in the following graph:

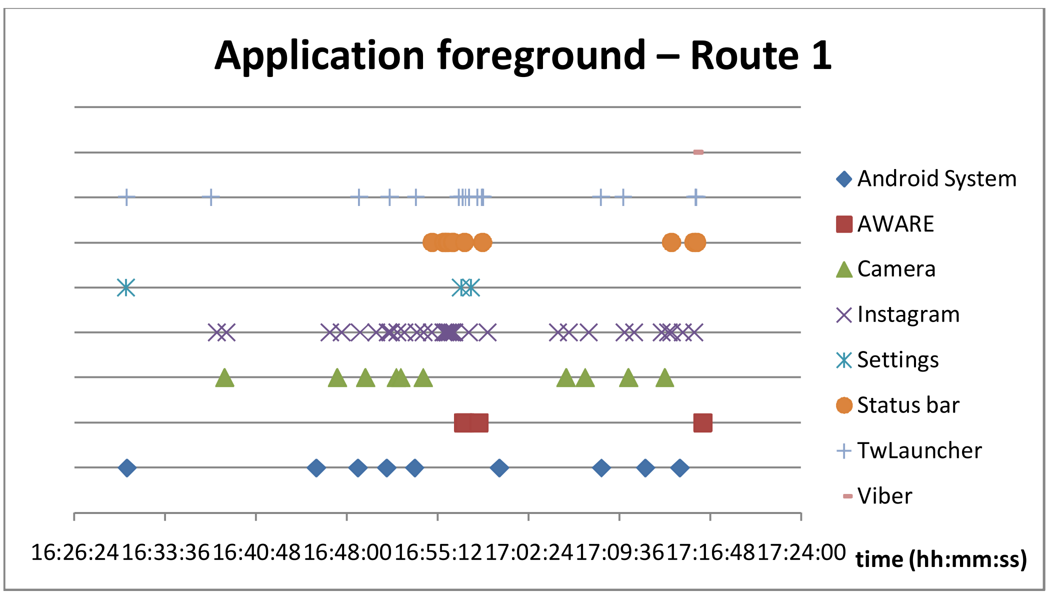


Fig. 6.1.1 Application foreground - Route 1

Where the markers indicate the times when the application was in use.

6.1.2. Application foreground ï Route 2

The information registered with the application foreground sensor for route 2, can be seen in the following graph:

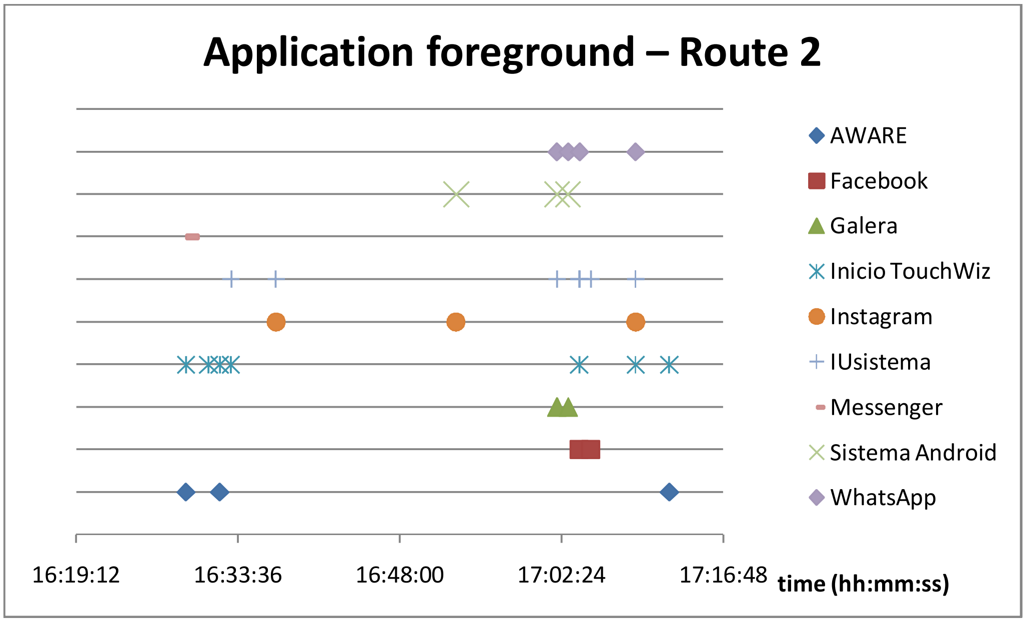


Fig. 6.1.2 Application foreground - Route 2

The markers in the graphs indicate the times when the applications were in use.

6.1.3. Application foreground ï Route 3

The information registered with the application foreground sensor for route 3, can be seen in the following graph:

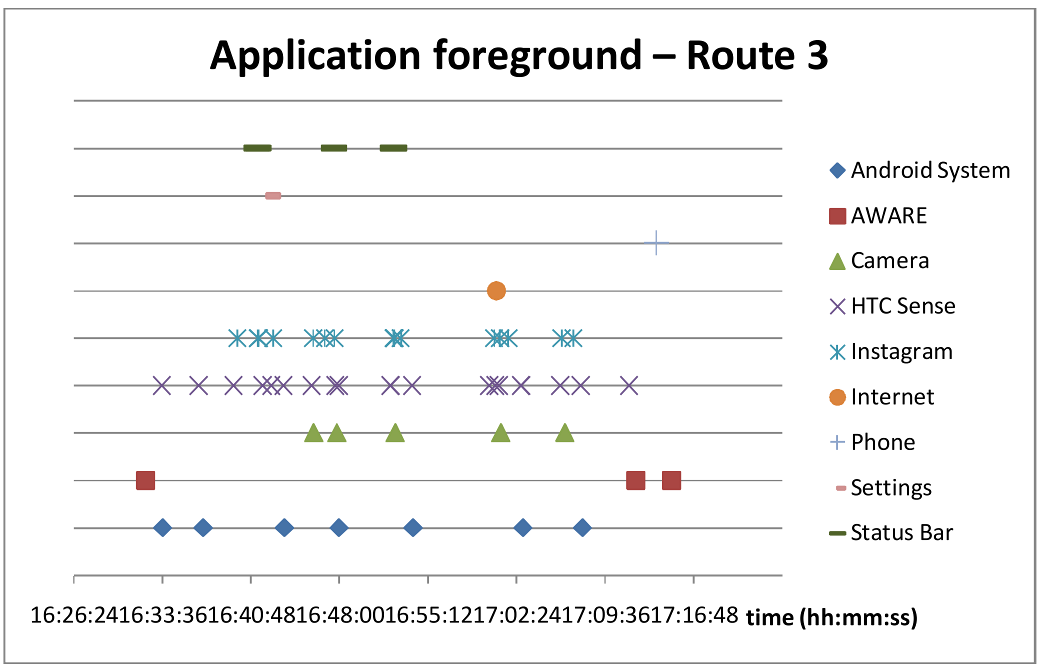


Fig. 6.1.3 Application foreground - Route 3

As in the previous cases, the markers indicates when each application was in use.

6.2. Battery

The battery sensor registers and stores battery information. The data that will be displayed about this sensor will be the battery level; this information will be used later on to determine is the battery is able to last for enough time to perform the activity.

6.2.1. Route 1

The information registered with the battery sensor related to the battery level for route 1, can be seen in the following graph:

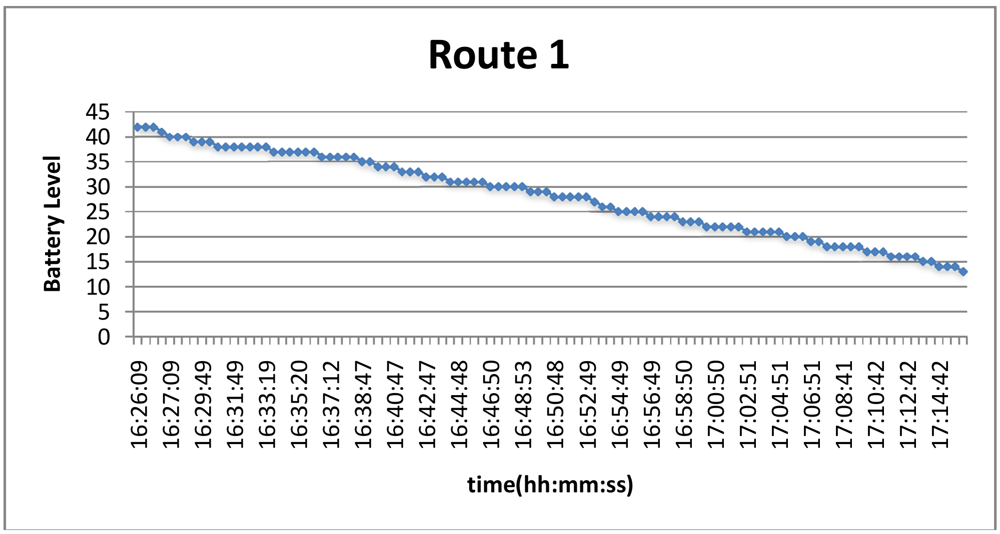


Fig. 6.2.1 Battery level - Route 1

The starting point was different than the full scale (100), because the battery wasn't fully charge when the activity started.

6.2.2. Route 2

The information registered with the battery sensor related to the battery level for route 2, can be seen in the following graph:

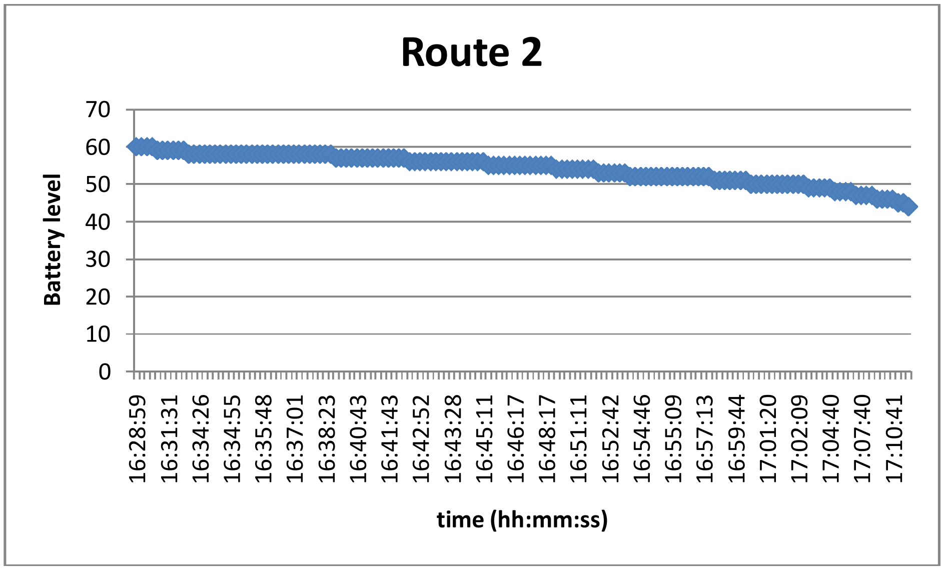


Fig. 6.2.2 Battery level - Route 2

In this case the starting battery level was 60 but the maximum value of the scale is 100 as in all the routes.

6.2.3. Route 3

The information registered with the battery sensor related to the battery level for route 3, can be seen in the following graph:

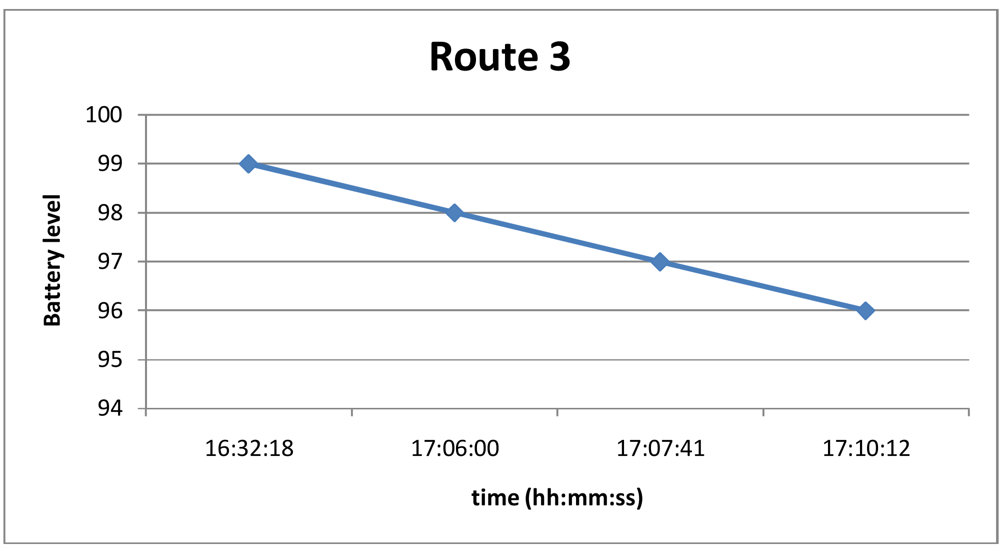


Fig. 6.2.3 Battery level - Route 3

In this case the battery was almost full at the start of the activity, but only 4 registers were stored.

6.3. Locations

The location sensor was used for two different types of tests, the activity data test and the location settings test. To display this information different maps have been created using the latitude and longitude stored by the sensor. To obtain these maps the data registered was saved as a csv file and then imported to an online application called Mapbox to be mapped.

6.3.1. Activity data test

The maps related to the activity data test can be seen in the following table. Each marker in the map represents a geographic location stored by the sensor.

Table 10 Location maps of the activity test

<table>
<tr><td>User 1

</td></tr>
<tr><td>User 2

</td></tr>
<tr><td>User 3

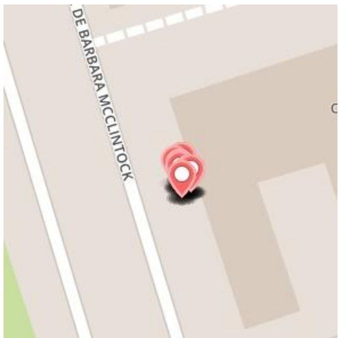
</td></tr>
</table>

6.3.2. Location settings test

The maps related to the location settings test can be seen in the following table. Each marker in the map represents a geographic location stored by the sensor during a 5 minute period for each profile.

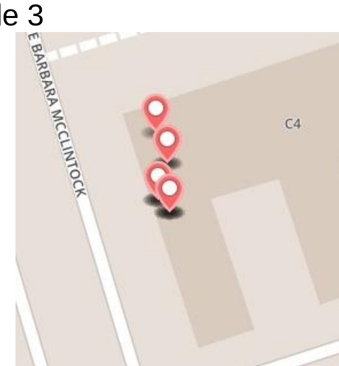
Table 11 Location maps of the settings test

Profile 1



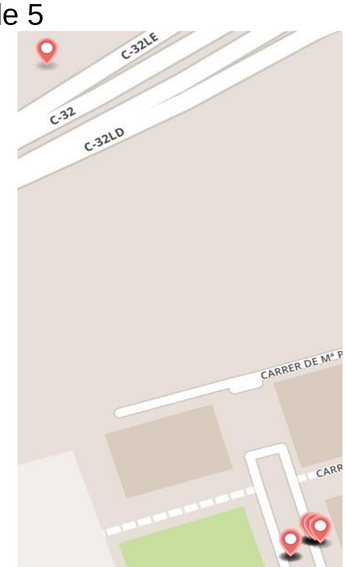
Profile 2

Profile 3

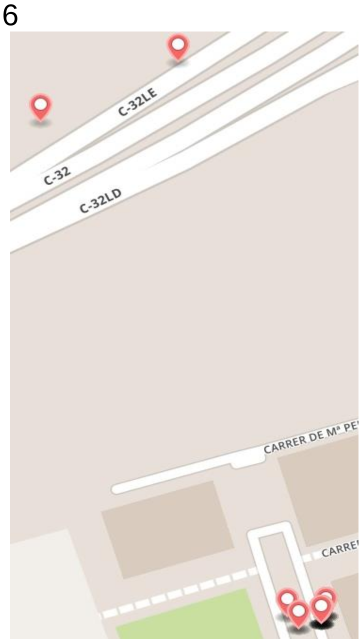


Profile 4

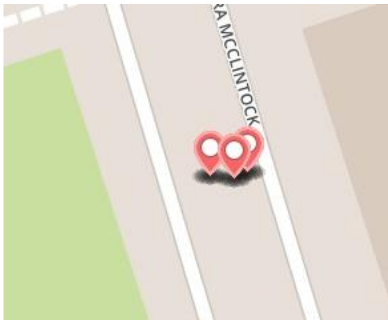
Profile 5



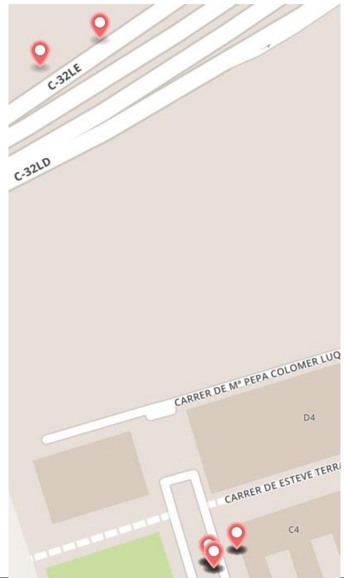
Profile 6



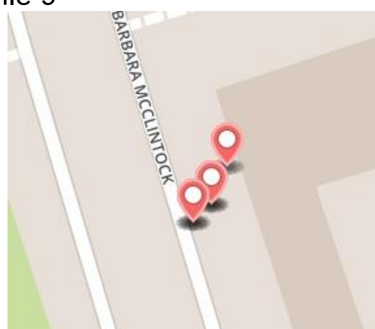
Profile 7



Profile 8

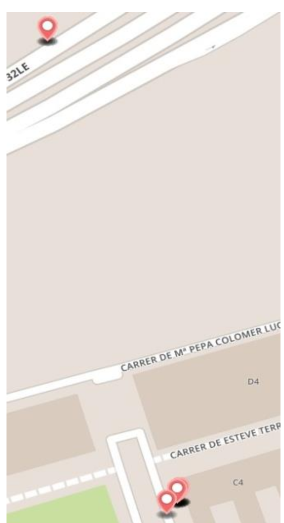


Profile 9



Profile 10

Profile 11



6.4. Linear accelerometer

The linear accelerometer sensor stores the acceleration the device experiments in three different axes: x, y and z. This information can be useful to determine if the user was walking, running or jogging during the activity test and which means of transportation is used on the real activity.
For the results, one of the devices didn't have this sensor (the device used for the route 2) and another one didn't register any values (the device used for the route 1).

The information registered by the linear accelerometer sensor in the route 3 can be seen in the following graphs.

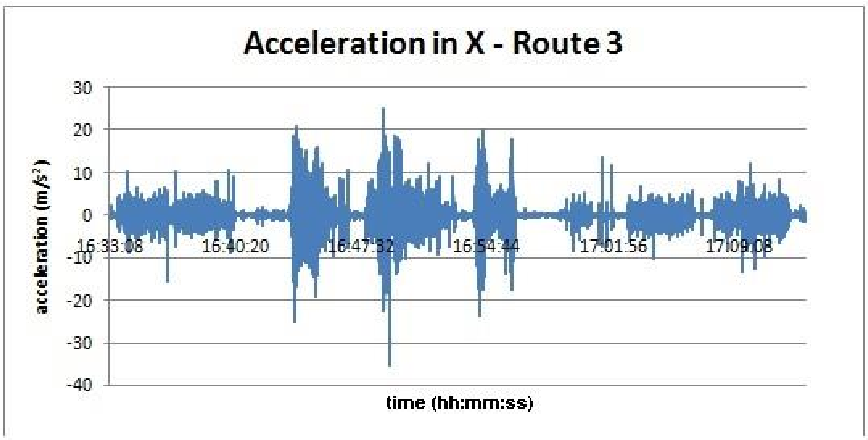


Fig. 6.4.1 Acceleration in X - Route 3

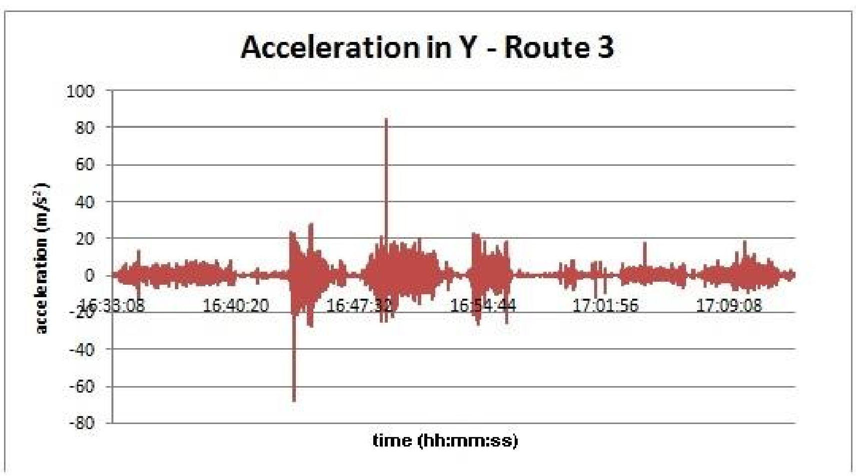


Fig. 6.4.2 Acceleration in Y - Route 3

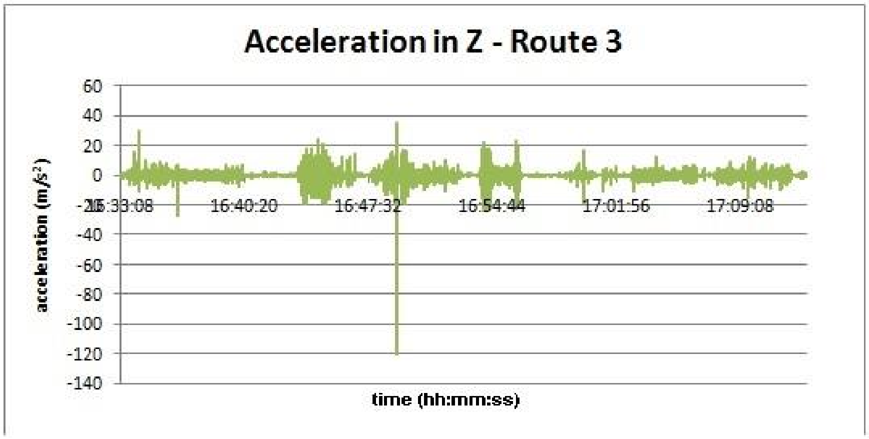


Fig. 6.4.3 Acceleration in Z - Route 3

6.5. WIFI

The WIFI sensor stores the information related to the networks the sensor has in its range. This information comes with an RSSI value to determine the intensity of this signal. This information can be used to help improve the location of the user, as well as the user surroundings.

6.5.1. Route 1

Histograms showing the results of the networks found and their respective RSSI values for route 1 can be seen in the following graphs (the graphs were divided due to the large number of recognized networks):

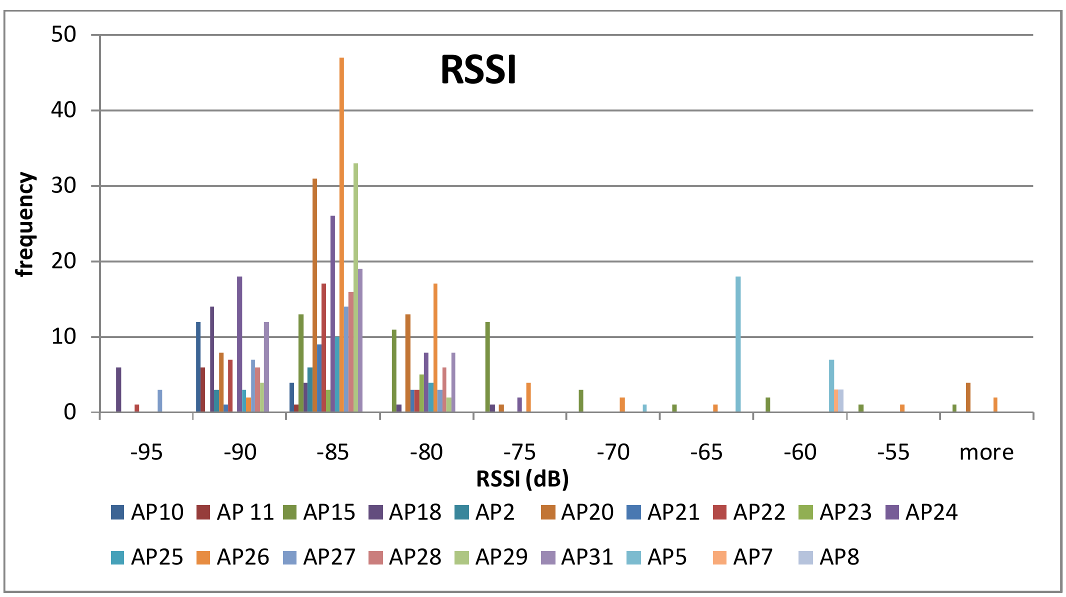


Fig. 6.5.1 RSSI Values - Route 1

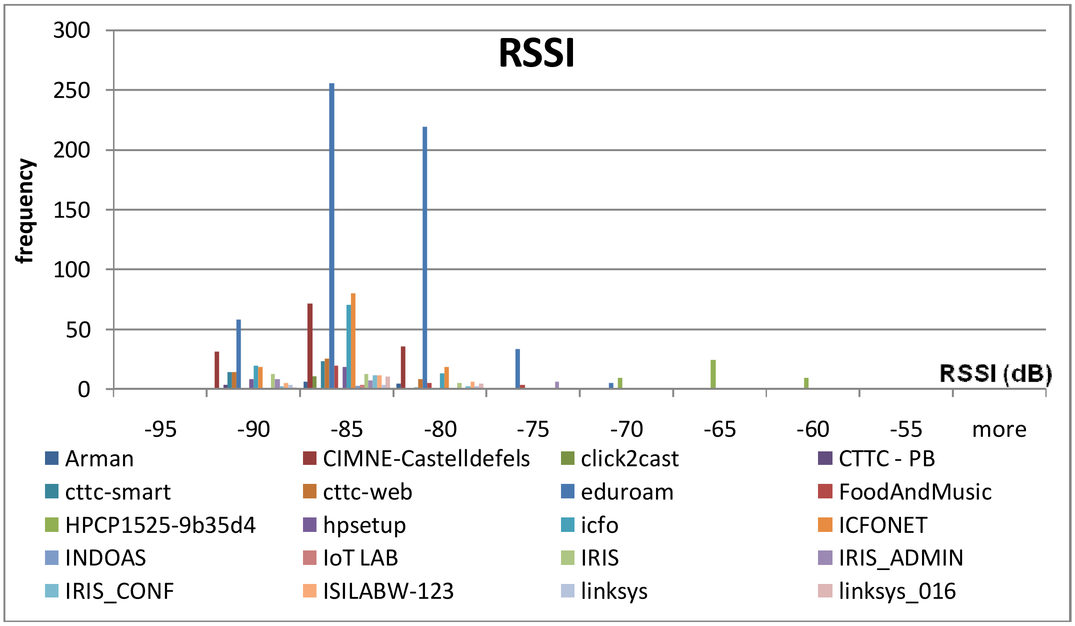


Fig. 6.5.2 RSSI Values - Route 1

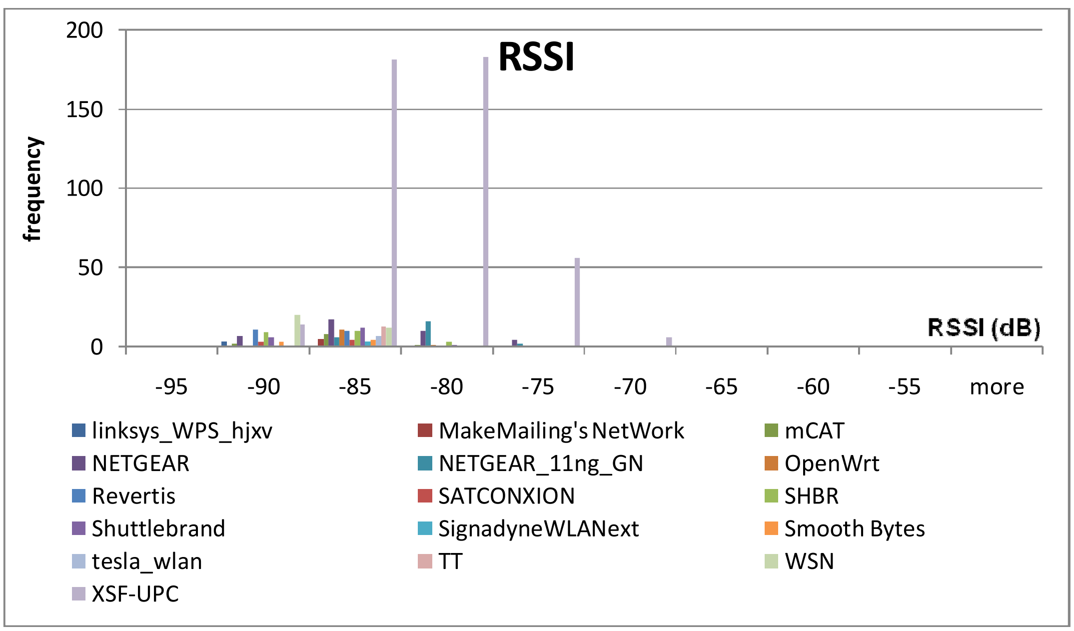


Fig. 6.5.3 RSSI Values - Route 1

As it can be appreciated by the histograms, the majority of the registers were for networks like ñeduroamò and ñXSF-UPCò as it was expected since this are the main networks inside the campus.

6.5.2. Route 2

In this route most of the registers were similar to the case of route 1, ñeduroamò and ñXSF.UPCò. This device registered less variety than the other, which it was expected since the other two routes passed through office areas that this route didnôt have. Histograms showing the results of the networks found and their respective RSSI values for route 2 can be seen in the following graphs:

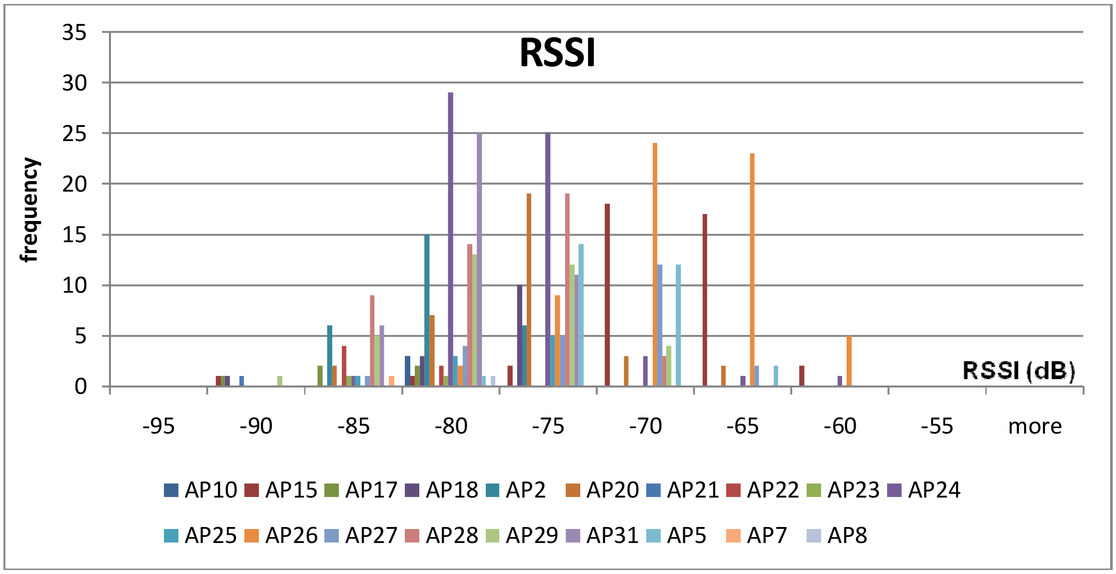


Fig. 6.5.4 RSSI values - Route 2

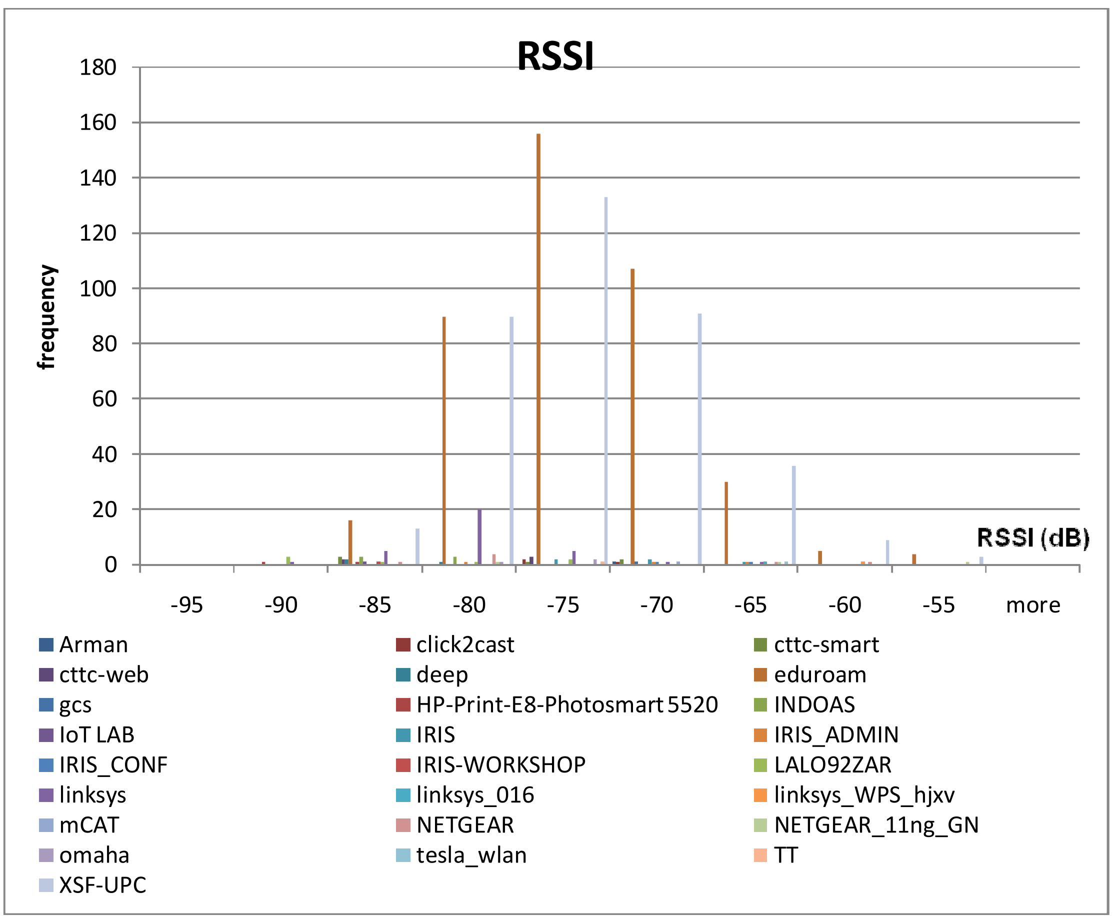


Fig. 6.5.5 RSSI values - Route 2

6.5.3. Route 3

Most of the values were register for "eduroam" and "XFC-UPC" like in the previous routes. New personal networks appeared like "MacBook" networks and some WLANs. Histograms showing the results of the networks found and their respective RSSI value for route 3 can be seen in the following graphs:

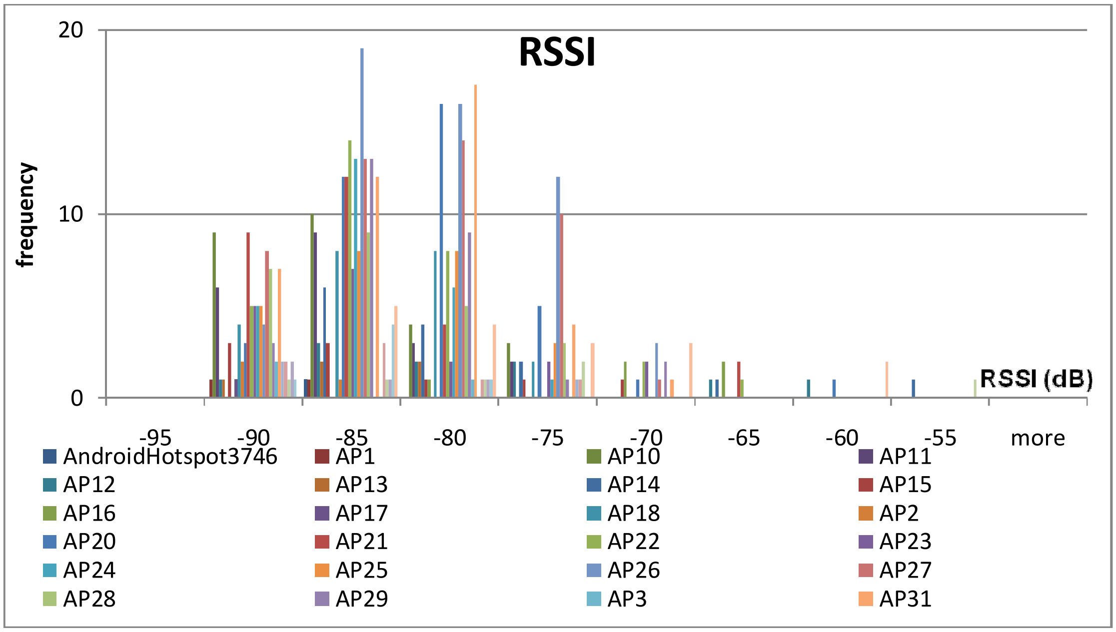


Fig. 6.5.6 RSSI values – Route 3

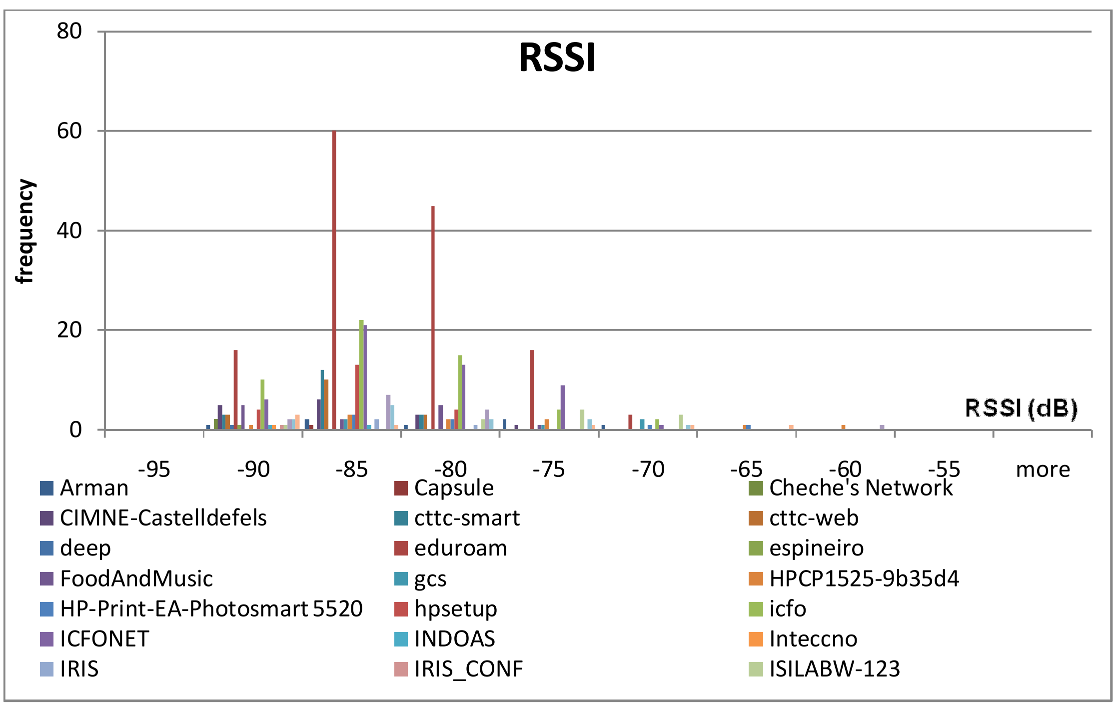


Fig. 6.5.7 RSSI values - Route 3

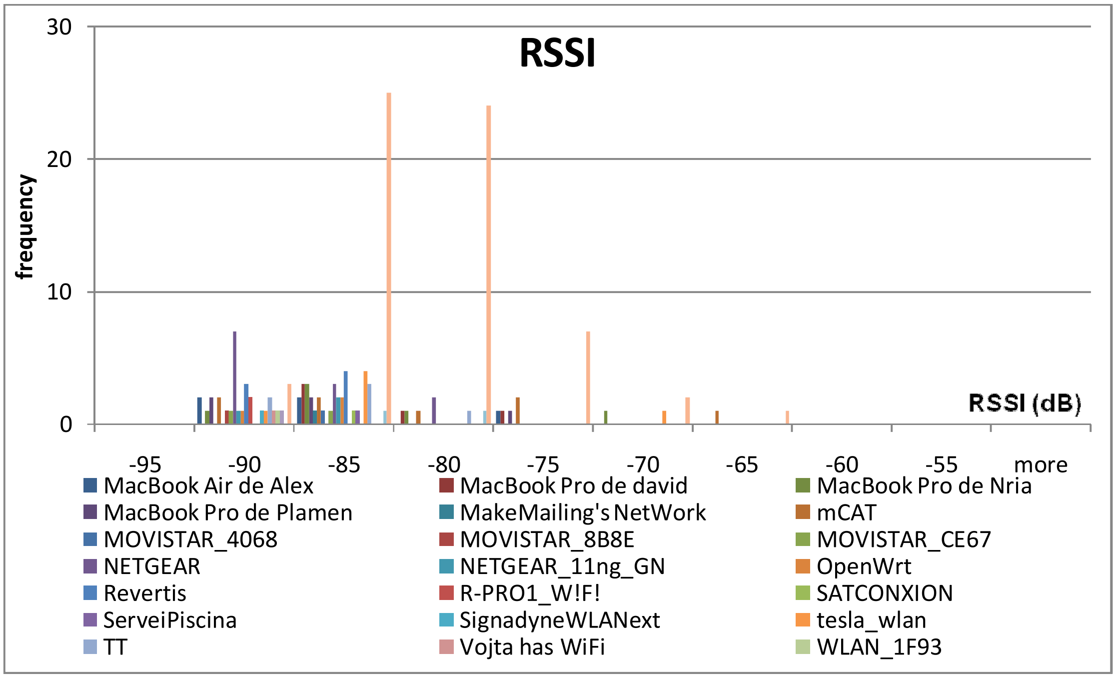


Fig. 6.5.8 RSSI values - Route 3

CHAPTER 7. FRAMEWORK PERFORMANCE ANALYSIS

After obtaining the results is time to determine if the information obtained is useful and consistent. In this chapter different sensors will be related to each other to arrive to different conclusions and some performance tests will be analysed.

7.1. Linear acceleration versus location data

A way to determine the means of transportation during the activity can be the information collected either by the location sensor or by the linear accelerometer sensor.
To evaluate if the sensors of AWARE are able to determine the means of transportation, the information stored during the activity test has been used. For the first case, the distance and time between points have been calculated using the geographical location of the points stored and the maps related to the activity test shown in the previous chapter. For the second case the acceleration registered for the route 3 has been divided into different stages according to the location data.

7.1.1. Route 1

For the route 1, six different stages of the route can be distinguished. These stages are consistent with the route that was made. The data that was calculated using this information can be seen in the following table:

Table 12 Speed calculations - Route 1

Stage	Calculated distance (m)	Time	Average speed (m/s)	Calculated speed (m/s)
1	264,6829691	0:01:39	8,00	2,673565345
2	249,1640316	0:04:09	0,30	1,000658762
3	412,7596301	0:04:57	0,50	1,389763064
4	153,39907	0:04:15	5,75	0,60156498
5	205,3832979	0:03:09	5,75	1,086684116
6	36,28573121	0:01:52	0,25	0,323979743

The calculated distance was found using the Haversine[4] formula; the time was obtained using the timestamp from the different location points; the average speed corresponds to the average of the speeds stored by the sensor; and the calculated speed was calculated using the distance and the time between each point.

Given that the user could have used three different methods to reach its destiny, walking, jogging or running; the different speeds have been divided using two different thresholds. The following graph shows the results.

[4] The Haversine formula gives the shortest distance between two points over the earth's surface, ignoring elevation, hills, etc.

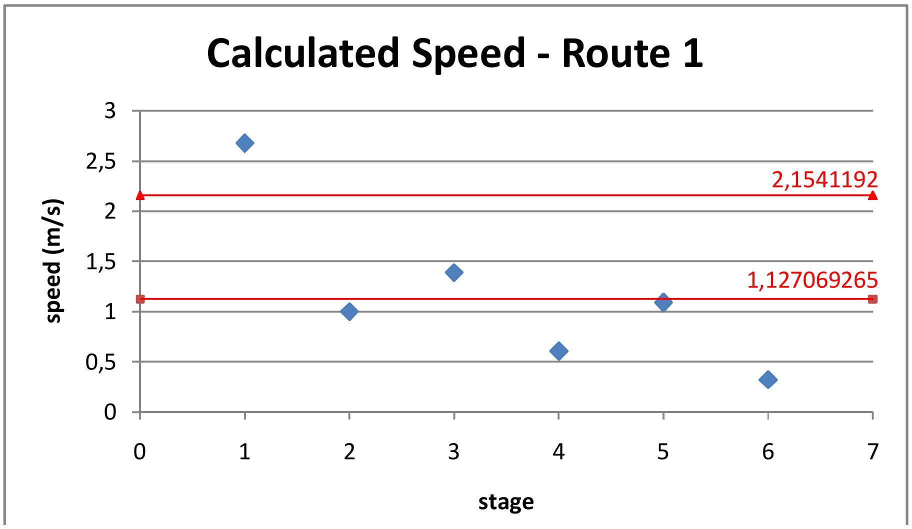


Fig. 7.1.1 Calculate speed - Route 1

As observed in this graph, if the results were to be classified using different thresholds according to the minimum and maximum values calculated, then the results will be not consistent. Only stage 2 and 4 were marked correctly as slow stages, the rest are inconsistent with the test, which means an accuracy of 33,3%.
In the case the average registered speed was used and also classified using this method, it will be 3 out of 6 stages classified correctly, which means an accuracy of 50%.

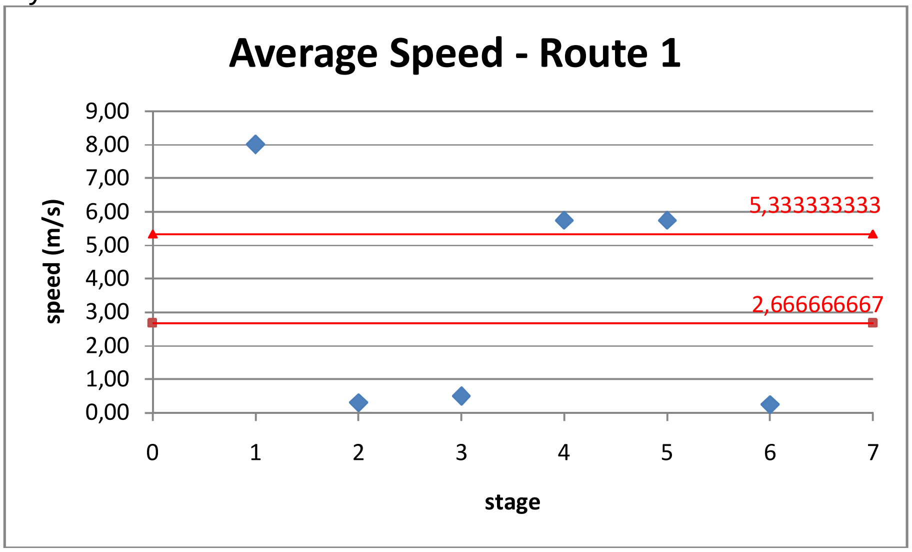


Fig. 7.1.2 Average speed - Route 1

The motive of this results can be related to the algorithm that the location sensor uses to save energy, if the accuracy level is between the desired parameters then the location is not updated until the location expiration time is reached; as a result of this the time between each sample is variable which makes it hard to determine when was the exact change of location made. This can possibly be used by lowering the expiration time or decreasing the accuracy (in meters) parameters, but the battery life will be reduced by this.

7.1.2. Route 2

For the route 2 also six different stages can be distinguished, which is also consistent with the route that was made. The data calculated for this route can be seen in the following table:

Table 13 Speed calculations - Route 2

Stage	Calculated distance (m)	Time	Average speed (m/s)	Calculated speed (m/s)
1	71,2952852	0:03:19	5,44	0,358267765
2	215,650972	0:03:05	6,93	1,165680929
3	199,800031	0:01:22	1,42	2,436585749
4	147,632313	0:03:01	2,39	0,815648139
5	154,983921	0:01:53	3,83	1,371539126
6	11,6022422	0:01:56	0,16	0,10001933

The table was made using the same methods as in the first case, as well as the following graph showing the results.

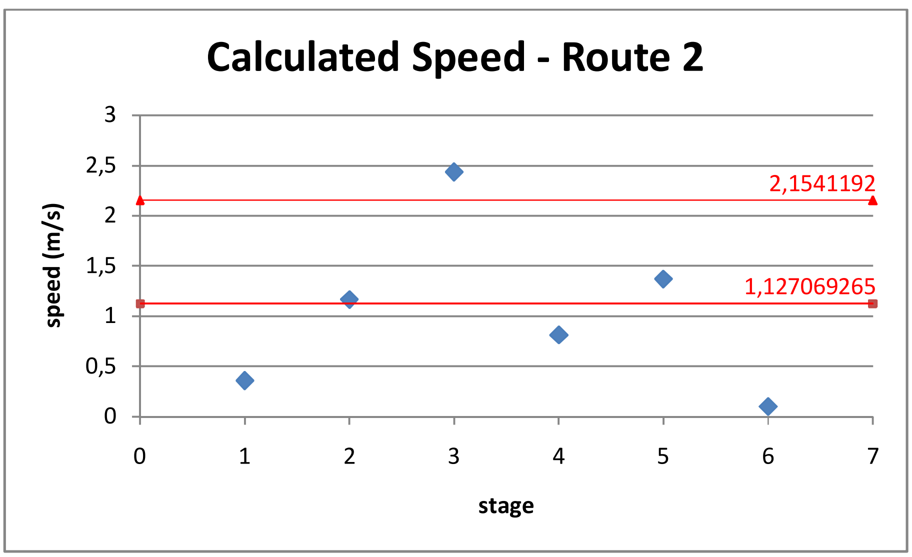


Fig. 7.1.3 Calculated speed - Route 2

In this case, four of the six stages are consistent with the test that was made, which corresponds to an accuracy of 66,7%. While observing the information stored there is more registers for this route than for route 1, the number of samples between one point and another is higher which allows monitoring better the change of one stage to the other.

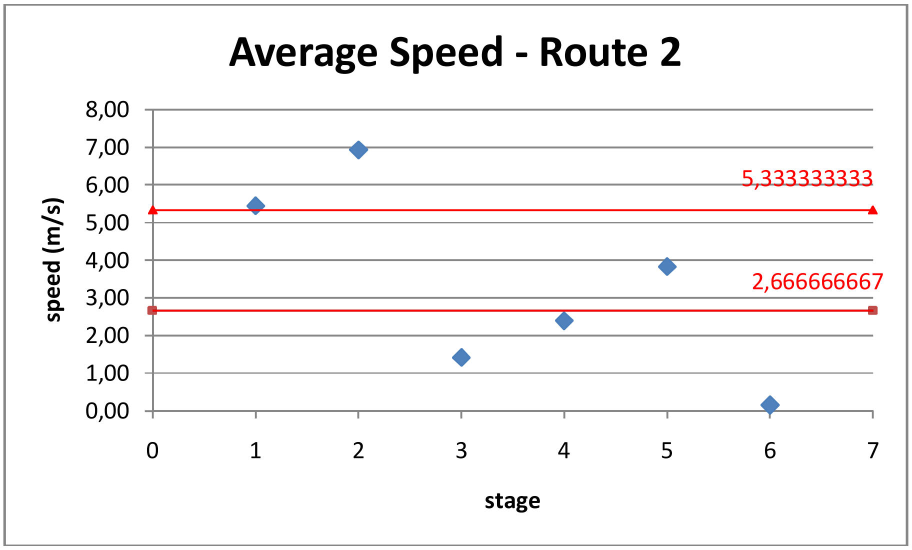


Fig. 7.1.4 Average speed - Route 2

Using the average speed registered in the route 2, 3 of the resulting values are consistent with the route that was developed. This means the results in this case have a 50% accuracy with respect to the original value.

7.1.3. Route 3

The device used for this route registered less values than the other cases and only 5 stages could be distinguished. The last stage of the route was not stored. It also registered more network measurements than the other cases, which are less accurate than the GPS measurements. The average speed was 0 for most of the cases. The results can be seen in the following table:

Table 14 Speed calculations for route 3

Stage	Calculated distance (m)	Time	Average registered speed	Calculated speed
1	231,64879	0:05:54	0,00	0,654375114
2	268,371482	0:04:23	0,00	1,020423885
3	124,090235	0:04:13	0,58	0,490475237
4	204,872536	0:04:54	0,00	0,69684536
5	203,594825	0:01:04	0,00	3,181169134

The same method as in the previous routes was used to make these calculations. The graphs bellow shows the speed values for route 3.

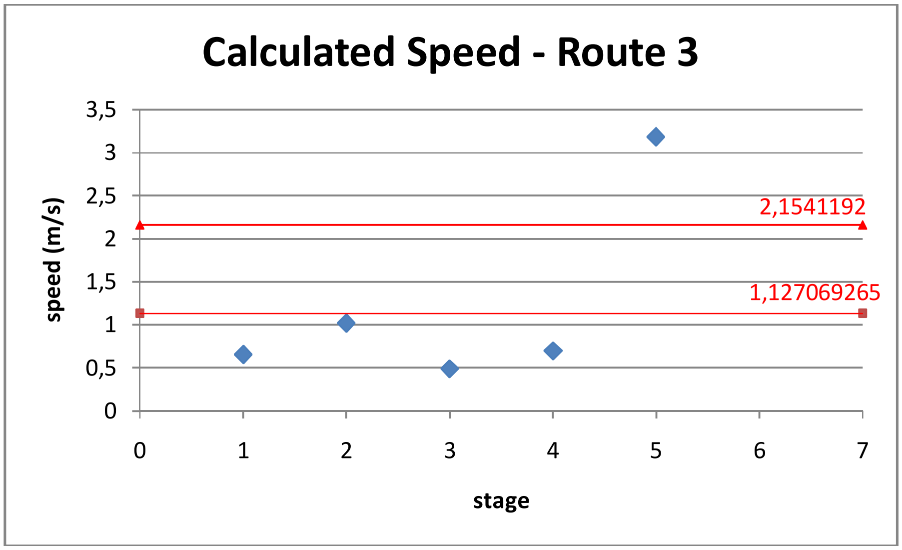


Fig. 7.1.5 Calculated speed - Route 3

In this case only the value of the first stage is consistent to the test, which corresponds to an accuracy of 16,7%. Again the non-consistent results could be related to the low amount of stored in the database and in this case also the lower accuracy in respect with the location given by the higher amount of network measurements.

To improve these results, the linear acceleration values can be help by using as model a study made in the use of acceleration data to predict the means of transportation used[5]. First the results of the acceleration sensor are divided in different stages, the five stages that could be distinguished using the location sensor plus a last stage that can be seen due to the variation of the data. These stages are shown in the following graphs:

[5] Use of acceleration data for transportation mode prediction by Muhammad Awais Shafique and Eiji Hato.

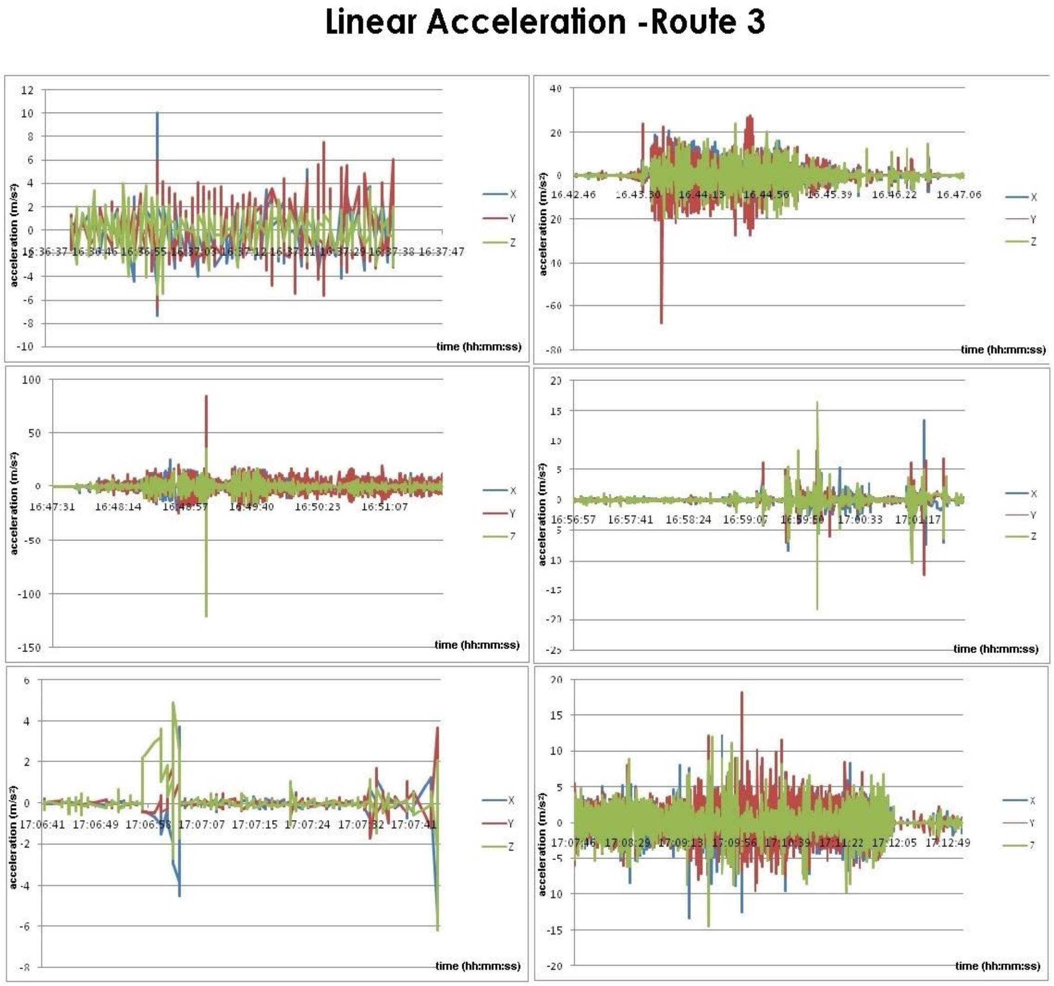


Fig. 7.1.6 Linear accelerometer stages

After just observing the maximum and minimum values of the acceleration, stages 1 and 5 can be classified as slow stages (walking), because unlike the others the acceleration value does not surpass 10 m/s^2. Similar appreciation of these values can be seen in another study by Ekachai Thammasat[6].

This will leave 4 undetermined stages. Following this study another solution could be calculating the standard deviation and the mean of these values. The following table shows these calculations:

Table 15 Standard deviation and mean acceleration of each stage

Stage 1		Stage 2		Stage 3	
Deviation	Mean	Deviation	Mean	Deviation	Mean
2,073	0,000	4,239	-0,021	5,035	0,010
2,216	-0,022	5,876	-0,016	7,249	0,059
2,526	-0,012	4,327	0,004	5,480	-0,071

[6] The Statistical Recognition of Walking, Jogging, and Running Using Smartphone Accelerometers by Ekachai Thammasat

Stage 4		Stage 5		Stage 6	
Deviation	Mean	Deviation	Mean	Deviation	Mean
1,134	-0,007	0,698	-0,087	2,183	-0,006
1,098	0,020	0,533	0,023	2,636	0,004
1,289	-0,009	0,699	0,043	2,200	0,007

Following the same study the mean values aren't so useful because they are too similar to each other, but the standard deviation could be used to classify them into different groups. Using as reference the values of the study the stages 1, 4, 5 and 6 could be classified as slow stages and stages 2 and 3 as medium or fast stages. With this classification stage 4 will be wrong.
Some additional data like the information obtained from the gyroscope sensor could be useful to improve the analysis of the results. Since in the activity the idea is to distinguish movement from other means of transportation, by foot, bus and metro, then in those cases should be clearer the classification without this additional data.

The other devices didn't record any data related to the linear acceleration, it's difficult to reach accurate conclusions due to the lack of data to compare to and to make a model around.

7.2. Different location settings

In the previous chapters the location sensor was activated to register the geographic location of a given point using different configurations of this sensor. The results were displayed in the previous chapter using different maps to localize all the points that were stored.
Using the latitude and longitude of the original point localized in the map and the points that were stored, the distance between these points was calculated with the help of the Haversine formula. The results obtained of these calculations for each configuration are shown following.

7.2.1. GPS accuracy

A histogram with the results of varying the GPS accuracy parameter and calculating the distance between the original point and the points registered is shown following:

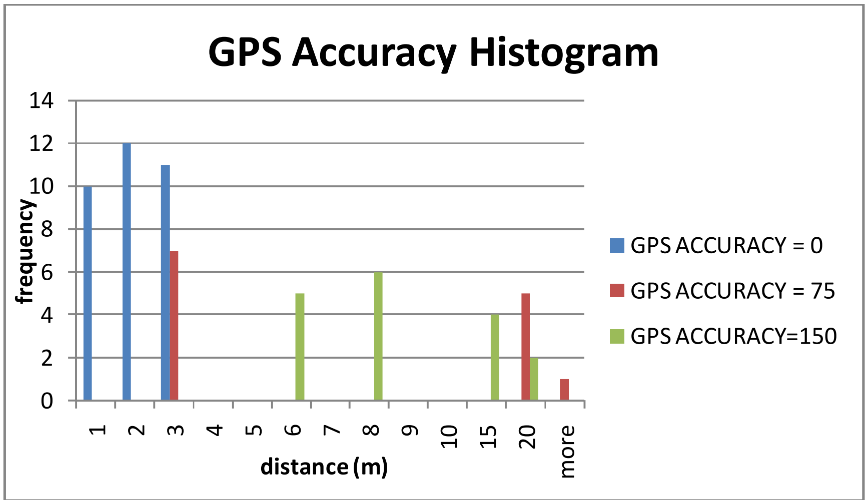


Fig. 7.2.1 GPS Accuracy

The best results were obtained using a GPS accuracy of 0. More than 50% of the results with a GPS accuracy of 75 were bellow 5, while none with an accuracy of 150 were. Also calculating the average of the distances calculated with the measured points by the GPS, the results are: 1,49m, 7,65m and 8.4m respectively. So it can be concluded that changing the GPS accuracy does affect the results, with a lower value for the GPS accuracy the location registered is more accurate.
The highest distance obtained by the calculations was given by a point measured using the network instead of the GPS, so it is not relevant in this case.

7.2.2. GPS update frequency

A histogram with the results of varying the GPS update frequency parameter and calculating the distance between the original point and the points registered is shown following:

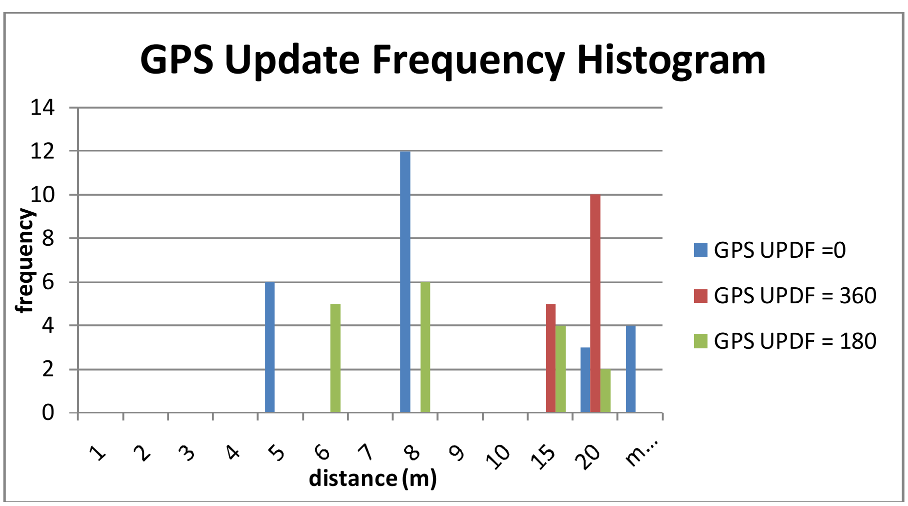


Fig. 7.2.2 GPS update frequency

Best results were obtained using an update frequency of 0 which has the majority of the measurements bellow 8 meters. The worst results were obtained by using an update frequency of 360, all above 15 meters. By calculating the averages of the distances the results are 9,6m for a frequency of 0, 8,4m for a frequency of 180 and 15,5m for a frequency of 360. The highest distance was obtained using a network measurement instead of a GPS measurement like in the previous case, for a variation of the GPS parameter these values aren't relevant and were not taken into account in the average calculations.

A lower frequency did increase the number of registers, but by decreasing the frequency by a coefficient it doesn't mean that the number of registers decreased by the same coefficient.

7.2.3. Location expiration time

A histogram with the results of varying the location expiration time parameter and calculating the distance between the original point and the points registered is shown following:

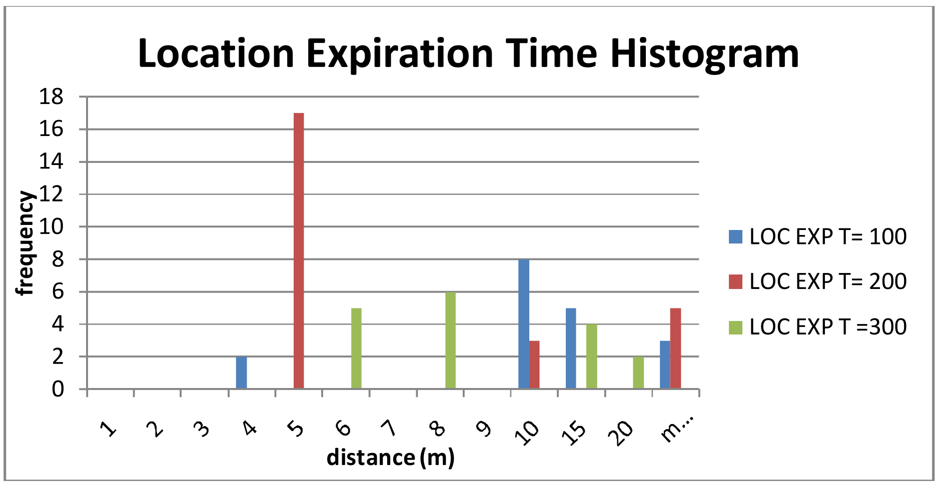


Fig. 7.2.3 Location expiration time

In the case of varying the location expiration time parameter, the best result was obtained using a lower expiration time, but a higher percentage of values bellow 5 meters were obtained using an expiration time of 200. Also the expiration time of 300 registered more values bellow 8 meters compared to the location expiration time of 100. The average values of all the distances calculated using the measurements by GPS and network are: 57,4m for an expiration time of 100, 63m for an expiration time of 200 and 8,4m for an expiration time of 300. They show no clear relation.

While inspecting the data, one of the reasons why the expiration time of 300 has better results is because all the registered points during this period of time were measured with the GPS, which has a better accuracy than the network measurements. If these points are not taken into account the average distances improve greatly for the other cases, being 9,9m and 5m respectively. But still no clear relation can be made.

7.2.4. Network accuracy

A histogram with the results of varying the network accuracy parameter and calculating the distance between the original point and the points registered is shown following:

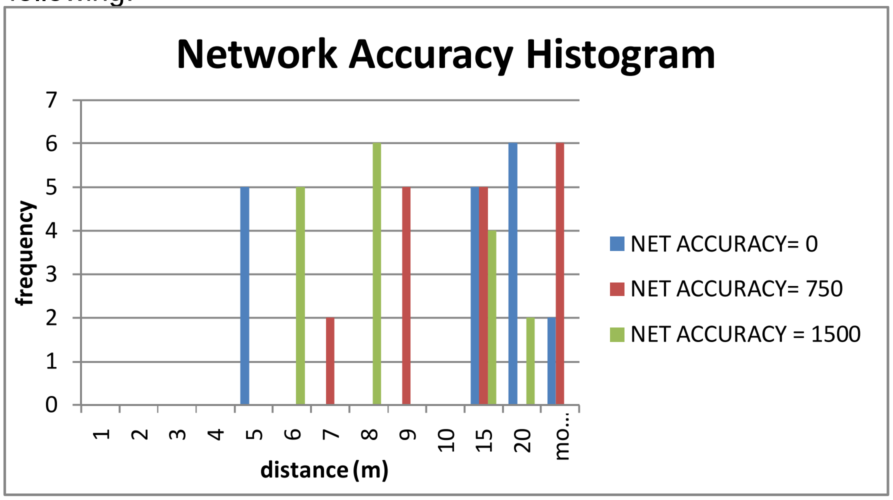


Fig. 7.2.4 Network accuracy

When analyzing this histogram the column of more is taken into account, because those are the values that were measured using the network instead of the GPS. The best results were obtained using a network accuracy of 0, compared to the results obtained using a network accuracy of 750. In the measurements of 1500, all the values were measured using the GPS instead of the network.

7.2.5. Network update frequency

A histogram with the results of varying the network update frequency parameter and calculating the distance between the original point and the points registered is shown following:

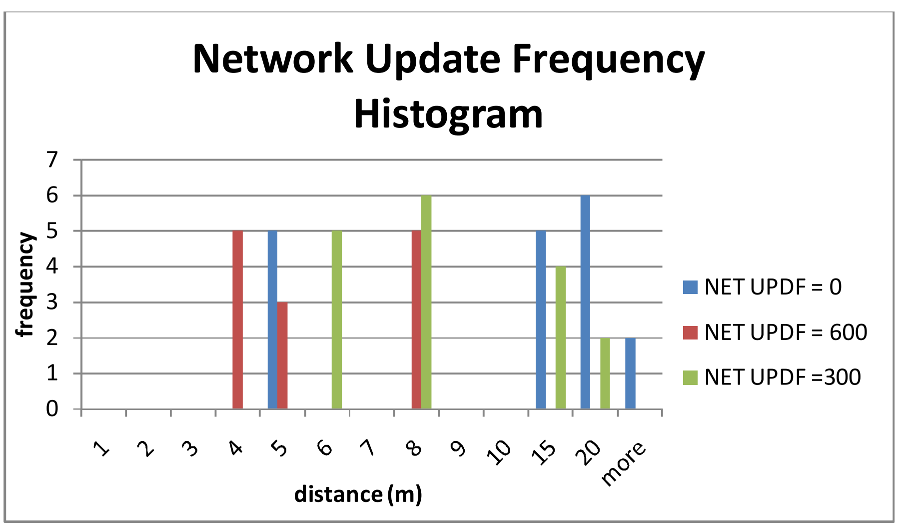


Fig. 7.2.5 Network update

In case of varying the network update frequency the worst results are obtained with a lower update frequency, this happens because more network data is stored and this data is less accurate than GPS data. The information stored and the graph reflects this relation.
The tests with a lower update frequency registered more values as it should be, but by increasing the frequency by a number it doesnꞌt mean that the number of registers decreased by the same number.

7.3. Application usage

To be able to determine in which part of the route Instagram was used, the foreground data of this application has been separated for the rest for each route. This information has been mapped on the time and since the different stages of route has been already defined; it is possible to relate the stages with the use of Instagram. For route 1 the usage of the application is shown in the following figure:

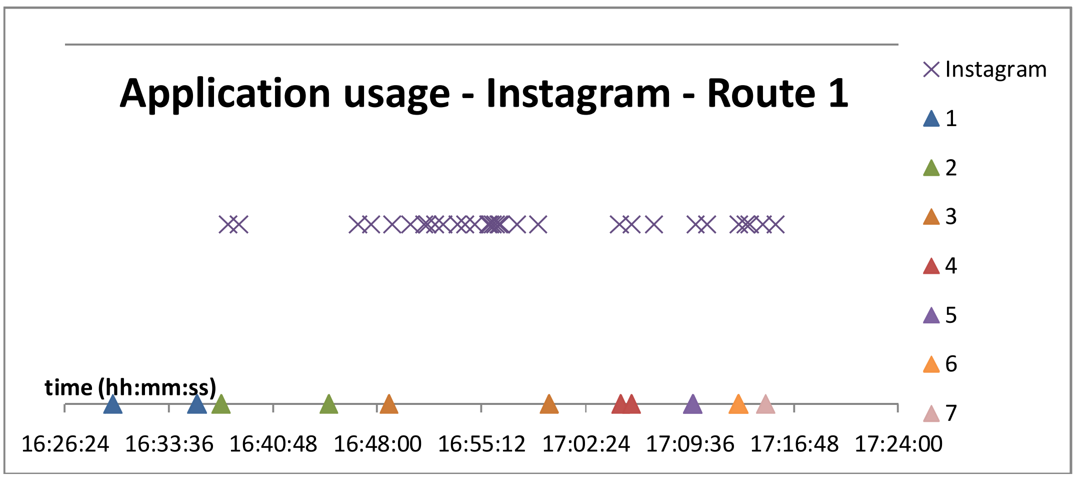


Fig. 7.3.1 Application foreground data for Instagram - Route 1

In the previous graph is shown that this user was probably uploading every picture in the same site where it was taken. The empty spaces correspond mostly to areas where the user was walking from a site to the other. At the stage1 there is no application usage because what is considered as stage 1 is the starting point, where no picture is taken. Also it is possible to see that there was a lot of time expended at stage 3 that is because this user had some problems with the network and had to wait there to solve it and to upload the pictures. These numbers are not related to the cards location number.

In the case of route 2, the sensor didnꞌt register much data about the usage of the application; there were only 3 registers for Instagram. The real reason about this is unknown, but it can be related to the operating system of this phone, which was different than the others. Even though this happened, the results are shown following:

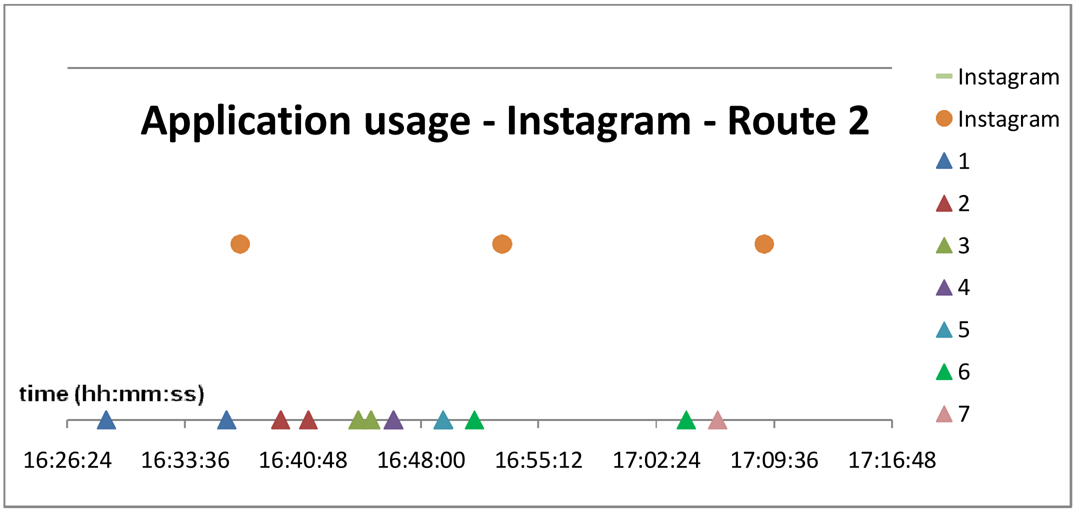


Fig. 7.3.2 Application foreground data for Instagram - Route 2

In this case it is not possible to appreciate clearly what happened during the activity due to insufficient data. In one of the points the application was used while walking and in the other two it was used in the same place. Anyways a conclusion cannot be reached.

Finally for the route 3 the same study was done, in this case as in route 1 there is sufficient data to analyze it and it is possible to distinguish different points where the application was used. The only inconvenient would be the location information, as previously it was concluded that for route 3 the stages were not well defined due to the lack of accuracy of the sensed location. The application usage is shown in the following figure:

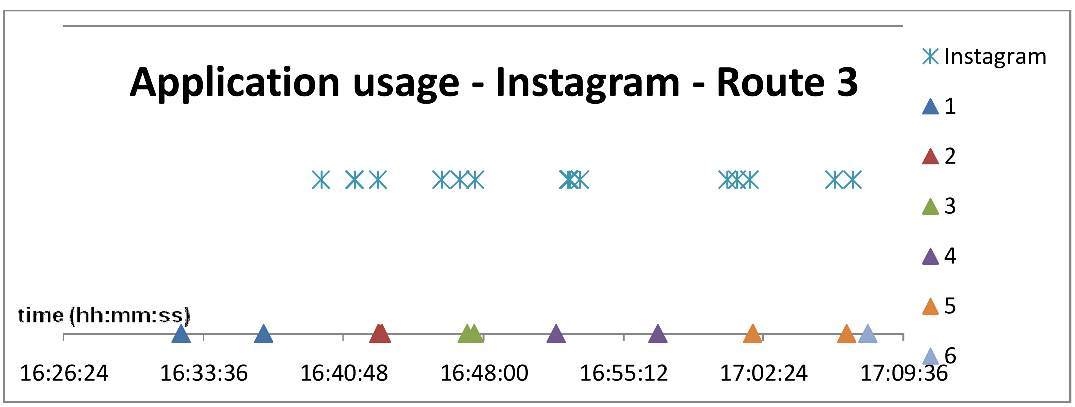


Fig. 7.3.3 Application foreground data for Instagram - Route 3

It can be observed that even thought the application usage is well divided in 5 stages along time, the stages from the location sensor doesnÕt seem to fit properly with the information. It is not clear if the application was being used during one of these stages or not. Analyzing the information probably stages 2 and 3 should last longer, because there is not enough time for the user to take a picture in them. So probably the application usage was during these stages, not partially inside like it looks in the graph.

7.4. WIFI sensor versus location sensor

One of the most important points of the activity is to be able to monitor the position of the participants during the activity. In previous sections, it has been shown that with the location sensor it is possible to determine the different routes the users made. For two of the three cases all the stages of the route were recognized, in the case of route 3 the last stage (the way back to the starting point) was missing.

To improve the location the WIFI sensor might be used along the location sensor. To be able to determine if the information offered by AWARE is useful for this, the routes have been divided in different stages and the WIFI registers of the common points between the routes have been compared to each other.

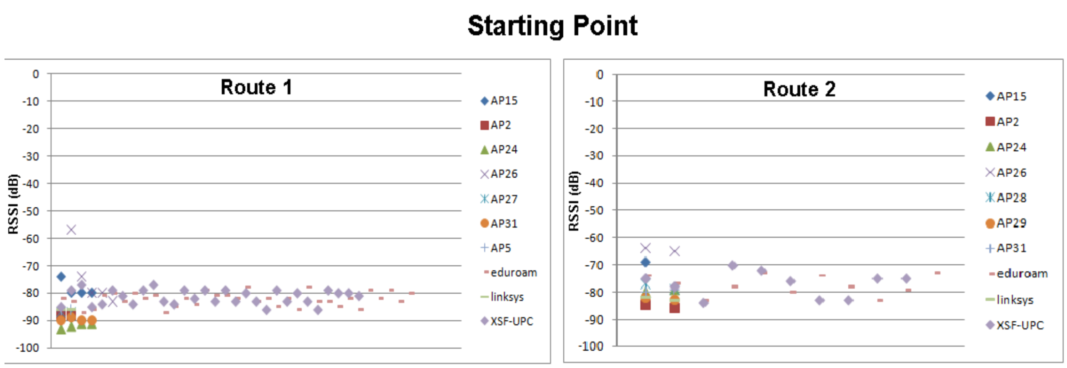


Fig. 7.4.1 Starting point networks for route 1 and 2

For the route 3 is difficult to identify the starting point using the location, because the measurements were made using the network and the accuracy is really high, between 100 and 800 meters. Also at the end of the route the stage that returns to the starting point is missing. So with the help of the WIFI network values given by the devices of route 1 and 2 of the starting point it should be possible to make an estimation of the time the user was at this location and the time the user left.

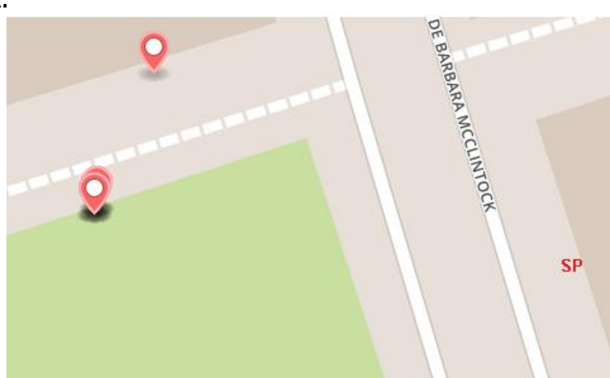


Fig. 7.4.2 Starting point registers for route 3

In the following graph is shown the WIFI networks available at the beginning of the route, when the measurements were inaccurate.

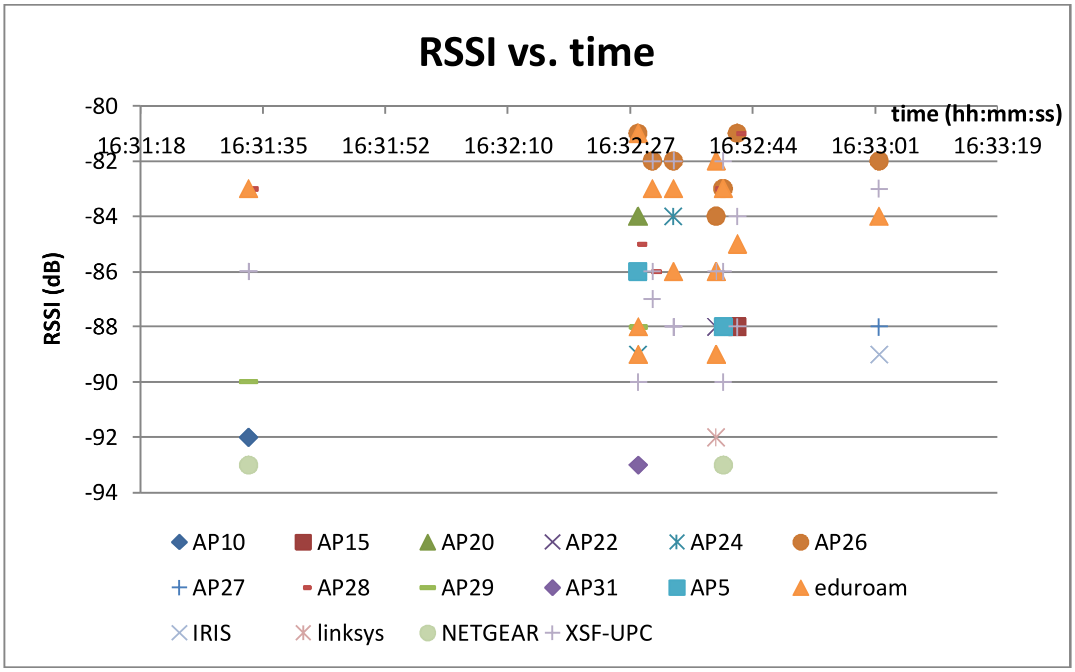


Fig. 7.4.3 WIFI networks at the start of route 3

It is possible with this graph to distinguish points that doesn't fit the profile given before. For example, the presence of the NETGEAR network at 16:31:33 and 16:32:40 and later on after the presence of another new network IRIS, will give the hint that the user is moving already. Also around 16:32:27 there is a presence of networks that fit perfectly with the ones found above.
A way to see it clearer will be to spread this area as it was done for the other routes, this graph can be seen following:

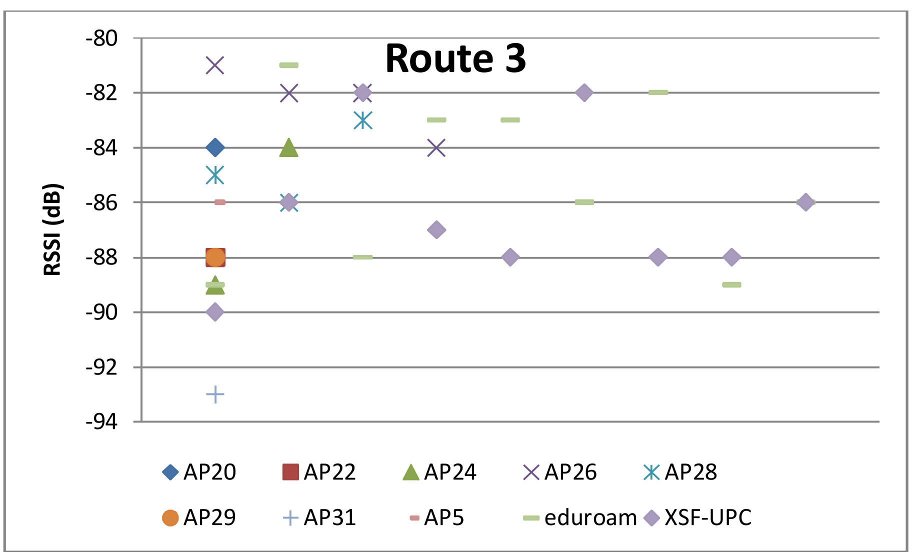


Fig. 7.4.4 RSSI values for the starting point in route 3

As it was done with this case by simple observation, it can also be done in other stages of the route and using more accurate methods. In this case, the information gathered by this sensor is richer so it should be able to work well in many studies.

7.5. Battery performance

The last key factor will be determining if the battery is able to last through the whole activity. As it has been shown before, some results could be improved by modifying certain parameters, like the GPS accuracy, but these improvements in the measurements could affect the battery level.

By observing the behavior of the data it is possible to find a quick estimation of how long the battery will last by approximating the curves to a linear function. Using the slope obtained from this, the battery will be last around 3 hours in the first case and 5 hours in the second case from the full scale value which is 100 in both cases. The case of the route 3 wasn’t taken into consideration, because the data was insufficient to make a more accurate approximation.

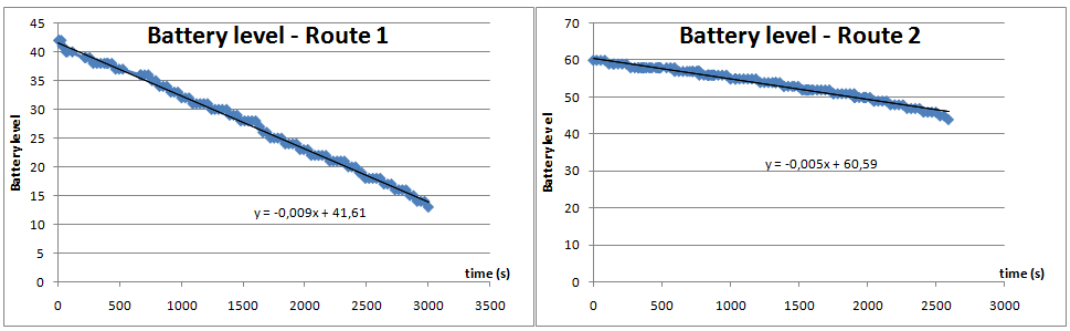


Fig. 7.5.1 Battery level vs. time

These calculations are not accurate, since the discharging of a battery is doesn’t have a linear behavior but it is useful to make a rough calculation. In case it was desired to make a more accurate estimation, AWARE offers other useful information like the battery voltage that could be analyzed. The information about the voltage can be seen following:

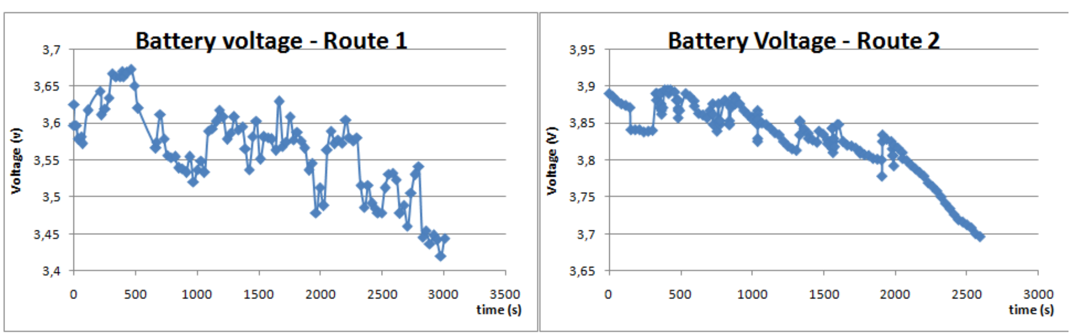


Fig. 7.5.2 Battery voltage vs. time

CHAPTER 8. CONCLUSIONS AND OBSERVATIONS

The AWARE framework offers a variety of tools that could be helpful to monitor and study a wide range of activities like the one used as a model in this thesis. Even thought it works well as a base, it still has a lot of room for improvement. Since it is still being developed several problems can appear, like difficulties in setting up the application or in the developing of a new plug-in or a new setup.
Taking all that into consideration, still with the use of AWARE it is possible to determine several context aware factors.

The location sensor, even though in some cases wasn't so useful, for example to determine time between locations due to the long periods without a measurement; it still was helpful to determine the different routes the user went through. Also by the location settings test it became clear that by adjusting the settings used by the sensor it should be also possible to improve the cases where it wasn't so useful.

There are some sensors that still need some improvement like the applications sensor and the linear accelerometer, because due to the missing information in some of the routes some secondary context could not be found. Also it is strange that even thought the devices were configured with the same settings and the users were recreating the same king of activity, the variation in the number of entries was considerable.

In general, AWARE framework allowed the logging and study of certain context information as it was supposed to and it was useful to determine some of the factors that were expected, but there are few problems with some of the sensors and due to the constant changes in the application and the website, it is hard to find plug-ins to adapt it to a certain activity. Also the new version of the client application is not compatible for some devices and there is no easy way for users to download and install other versions compatible with older devices.
This kind of factors could make it difficult for the application to be used with a large number of users, because it will have to be installed in each phone separately, so to improve this it will need some programming to make it more user-friendly.

ANNEX

A I LOCATION CARDS

Table 16 Cards information

Card Number	Site	Location
1	Buildings Pathway	41.275556, 1.987261
2	Bus Stop	41.275913, 1.990093
3	Coffee Shop	41.275537, 1.985417
4	Duck Bridge	41.273306, 1.983861
5	Great Wall	41.274815, 1.986357
6	Highway Bridge	41.275699, 1.984033
7	Library	41.275522, 1.985280

Card #1

Card #2

Card #3

Card #4

Card #5

Card #6

HIGHWAY BRIDGE

41.275699, 1.984033

Card #7

LIBRARY

41.275522, 1.985280

FIGURES INDEX

TABLES INDEX

BIBLIOGRAPHY

http://www.awareframework.com/

http://www.epochconverter.com/

http://www.movable-type.co.uk/scripts/latlong.html

Gregory D. Abowd and Elizabeth D. Mynatt, ñCharting Past, Present, and Future Research in Ubiquitous Computingò, Georgia Institute of Technology

Dey, A.K. Abowd, G.D. ñTowards a Better Understanding of Context and Context-Awarenessò CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness (2000)

Schilit, B., Theimer, M. ñDisseminating Active Map Information to Mobile Hostsò IEEE Network, 8(5) (1994) 22-32

Muhammad Awais Shafique and Eiji Hato, ñUse of acceleration data for transportation mode predictionò, August 1, 2014

Ekachai Thammasat, ñThe Statistical Recognition of Walking, Jogging, and Running Using Smartphone Accelerometersò, 2013