INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Web portal for continuous Internet access Quality of Service measurement

Ricardo Jorge Francisco Nunes

Dissertação para obtenção de Grau de Mestre em

Engenharia de Redes de Comunicações

Júri

Presidente:	Prof. Paulo Jorge Pires Ferreira
Orientador:	Prof. Rui Jorge Morais Tomaz Valadas
Co-orientador:	Prof. Ricardo Jorge Feliciano Lopes Pereira
Vogais:	Prof. Fernando Henrique Mira da Silva
	Eng. Sandro Miguel Carvalho Parrança

Outubro 2011

Web portal for continuous Internet access Quality of Service measurement

MERC/2011

Ricardo Jorge Francisco Nunes

ricardojfnunes@gmail.com

Abstract. Internet Service Providers advertise their products mainly with the maximum speed that their connections can achieve, but there are many factors that can influence what a user perceives as good Quality of Service (QoS). The most referenced study regarding this subject is the one presented by Portuguese communications regulator ANACOM. However the main focus of this study is to perform country wide ISP evaluation and, in practice, does not contains relevant number of samples from scarcely populated areas where network investments are frequently overlooked. The proposed project presents a measurement platform that could complement these tests. Our approach is to build a comprehensive Internet access evaluation platform, based on an open system where any consumer may participate simply by downloading a software package and using their personal equipment, obtain an estimate of the QoS of their Internet connection. The proposed solution consists in a software agent to be installed on the user's Computer or Smartphone, one or several application servers that will respond to the agent's measuring requests and a web portal/database where all the data is stored and where users can obtain their tests results. Furthermore with the geographical information given by participants or extracted from the mobile measurement agents, we can build an administrator report with region specific aggregated data. Our solution presents several innovations regarding common problems when developing these solutions. Such as the test scheduling, configurations updates, connection identification and location aware agents. We implemented a fully functional prototype with an extensive set of QoS KPI, without resorting to any commercial applications and through system tests proven to be in the same performance class.

Keywords: Network monitor, Quality of Service measurement, heterogeneous networks, monitoring web portal, location aware, ISP evaluation.

Resumo

Actualmente os fornecedores de serviço de Internet publicitam os seus produtos simplesmente com a taxa de transferência máxima que as suas ligações conseguem atingir. No entanto existem muitos outros factores que podem influenciar o que um utilizador sente como sendo uma boa qualidade de serviço. Um dos estudos de referência em Portugal sobre este tema é o apresentado anualmente pelo regulador das comunicações ANACOM. No entanto o foco principal deste estudo é efectuar uma avaliação a nível nacional dos vários fornecedores e, na prática, leva a que sejam consideradas poucas amostras de zonas rurais ou pouco habitadas. Sendo que, pela lógica do negócio, é nestas regiões que os investimentos em infra-estruturas são mais vezes negligenciados. Este projecto pretende complementar este estudo e apresenta uma plataforma de monitorização, tendo por base um sistema de participação aberto. Neste sistema qualquer cliente de um serviço de Internet pode participar simplesmente descarregando uma aplicação e utilizando o seu próprio equipamento obter uma estimativa da qualidade de serviço da sua ligação de banda larga. A solução proposta consiste numa aplicação que será instalada no PC ou telemóvel do utilizador, um ou mais servidores que serão utilizados para obtenção dos vários parâmetros das medidas e um portal onde todos os dados são guardados e os utilizadores podem obter os resultados das suas medidas. Além disso, com a informação geográfica fornecida pelo utilizador ou pelo receptor de GPS contido nos telemóveis, conseguiremos gerar um relatório de administrador agregando os dados por ISP ou regiões. A plataforma apresentada introduz várias inovações, sobre os problemas comuns deste tipo de plataformas, tais como, a calendarização dos testes, actualizações de configurações ou procedimentos dos testes e a obtenção da informação geográfica dos agentes móveis. Foi implementado um protótipo plenamente funcional, com um conjunto extensivo de indicadores sem recorrer a aplicações externas e através dos vários testes à plataforma provamos que se encontram na mesma classe de desempenho.

Palavras Chave: qualidade de serviço, monitorização de redes, redes heterogéneas, detecção de localização, avaliação de serviço Internet.

Contents

List of Figures

List of Tables

List of Equations

1. Introduction

When the Internet was first created, there was no perceived need for Quality of Service (QoS) and the Internet ran on a "best effort" system. Nowadays, it became part of our lives, enabling more than the simple exchange of messages or browsing over static pages. New real time applications such as video streaming, VoIP, online gaming, etc. have service requirements significantly different from previous data oriented applications. Furthermore even between real time applications, several important differences can be found. For example video streaming can tolerate moderate delay but requires a high throughput and low error rate. In contrast, VoIP applications do not need a high throughput but are very sensible to delay and can handle a slightly higher error rate.

Today Internet Service Providers (ISP) in order to gain market share advertise their products mainly with the maximum speed that their connections can achieve, but as stated before there are many factors that can influence what a user can perceive as a good QoS. Furthermore the Service Level Agreements established between customers and ISPs do not have any parameter related to the expected QoS other than the maximum speed and even these values can be misleading. For instance in an Asymmetric Digital Subscriber Line (ADSL) connection, the maximum speed that a user can achieve is directly related to the distance of the local telephone exchange. It is then possible for a user in an unfavorable position to enter in a contracted service that, due to technological limitations, can never achieve its maximum speed [1]. It is then very important to perform periodical evaluations of the ISP's connections, enabling the consumers the possibility to make an informed choice and to deliver to the ISP the knowledge of their position in the market.

A study that is considered a reference in Portugal is the one presented annually by the National Authority of the Communications (ANACOM) [2] that evaluates the QoS from both fixed and wireless access ISPs. This study is performed in a controlled environment with a standard commercial measurement platform and with participants being individually invited according to the regional Internet access penetration rate. This enables the statistical accuracy necessary to perform a national benchmark and comparison between the selected ISP operators.
However these measurement campaigns are extremely expensive and the limited set of samples are not completely representative of the country's effective Internet access network coverage. Given the fact that the recruitment of volunteers is directly proportional to the service's penetration rates, in practice this approach often leads to the exclusion of scarcely populated areas where network investments are frequently overlooked.

The proposed project presents a measurement platform that could complement these tests. Our approach is to build a comprehensive Internet access evaluation platform, based on an open system where any consumer may participate simply by downloading a software package and using their personal equipment to obtain an estimate of the QoS of their Internet connection. When possible, we should be able to complement the measurement results with some geographical information from where these results were obtained. Using this geographical information we will be able to create an administrator's report to assess the ISP/region overall results.

When developing this type of platform the main issue is that the software will perform measurement tests in an unattended manner with different user's equipment. From that, several problems arise and have to be taken into account.

Firstly it has to maintain a low system footprint while running and on standby as some participants might have low end machines and should experience minimal performance degradation. Also it will have to be extremely portable as it will be run in different operating systems. As such no system specific application can be used.

Although running without supervision, we still need to maintain some control when some measurements are taken. So we have to develop some scheduling mechanism to dynamically change these settings.

As expected the participants will have different technological knowledge so the user agent has to have a simple interface and be extremely user friendly. Also, the results have to be presented in a simple manner with easily readable information.

Also, it should account for the fact that a single user may have one or several connections, with different access types and may even use a connection that he/she is not the owner of. For this purpose we have to be able to provide participants some way of identifying from which connection the measurement test was made.

Finally we have to implement some security measures in order to prevent, as much as possible, test adulteration from both friendly and unfriendly users.

This report functions as a basic ground for our goal to develop this project and is organized as follows. Section 2 provides a context by analyzing several aspects dealing with the problematic of QoS measurement and then presenting several similar projects both national as international. The global system architecture and general aspect of the proposed solution are presented in section 3. In section 4 we detail the several implementation options that were used in the elaboration of the prototype. Section 5 presents the system evaluation tests and finally the proposed future work and main conclusions are presented in sections 5 and 6.

2. Related Work

2.1 QoS measurement principles

Internet QoS is fairly recent and complex subject that continues to be the source of several investigation reports by the scientific community. ITU-T as published important recommendations concerning QoS management such as G.1000 [4], E.802 [5], although there is no clear set of standards that are used to perform QoS evaluation.

Measuring QoS is very similar to network traffic measurement as the difference relies mainly on how these results are analyzed. In network traffic analysis the focus is on metrics associated with a given network element, such as network load, queuing performance, flow models etc. In QoS these results are a not so important as they are just a mean to obtain the performance characteristics (delay, throughput, etc.).

Measurement methods can be categorized into passive and active methods, depending on whether the parameters are obtained from simply analyzing the network traffic or if artificial traffic is generated to estimate these values. Passive measurements are usually used to measure metrics as link throughput and packet size statistics. However from the application point of view, particularly when dealing with real-time applications, end-to-end quality of service metrics are more important, and for these the passive approach is inappropriate as the presence of traffic between the end points is not guaranteed. Thus active measurement methods are typically used to obtain end-to-end statistics such as delay, loss and route availability.

2.1.1 Subjective vs. Objective measurements

It is generally accepted that quality of service spans beyond the strict parameters of network performance, but also some subjective parameters such as client satisfaction. ITU-T Recommendation E.800 [6] defines QoS as “the collective effect of service performance which determines the degree of satisfaction of a user of the service.” Therefore, it is not sufficient to address QoS evaluation solely based on network measurements, but is also important to analyze the other factors that influence the customer’s service experience and the resulting customer satisfaction.
QoS can be classified as subjective and objective. Objective QoS means usually something concrete and quantitative, i.e., something that can be measured directly. Subjective QoS, on the other hand, corresponds to the service quality from the user perspective that is, how does the user feel about the quality. Naturally, subjective QoS is much more difficult to measure than objective QoS, since the user experienced quality does not necessarily follow technical solutions that can have direct impact to the objective QoS. [8]

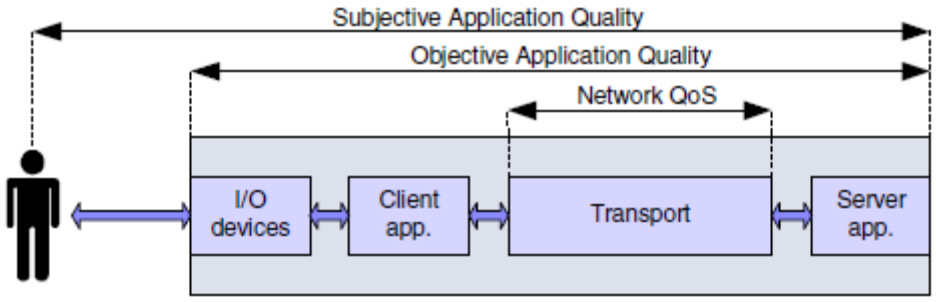

Figure 1 - Different scopes of QoS in a client-server communication. From [8]

Therefore in a network evaluation tool it is also important to obtain the user's input of the performance of the ISP in several types of uses (web browsing, P2P, "download/upload speed", availability …) and perform a comparison with the actual measurements obtained by the software agents.

2.1.2 Test environment

When developing a QoS evaluation system the environment settings in where the system is performed are determined by the main goal of the project [5]. In the tests promoted by ANACOM the main goal is to evaluate the overall performance of ISPs and to have as much accuracy as possible in every individual result obtained. This demands a comprehensive control of the environment settings in terms of user selection and the systems where the tests are performed. As such users are selected in a closed system (only accessible by invitation) according to the region's Internet access penetration rate. To control the environment where the tests are performed the same set of hardware is installed in each user's location to obtain a standard evaluation process. To have complete control of the test environment the users are deprived from accessing the Internet for the whole duration of the test (approximately 2 weeks). Even with a financial incentive to the user this discourages volunteer enrollment and the measurement campaigns are extremely expensive.

This project implement an open system where any customer is able to participate using their own hardware and network measurements are performed without restricting their Internet access. With this type of uncontrolled environment it is expected that some metrics can be degraded due to non-network related problems such as:

- Insufficient processing power from the client's hardware
- Existence of other traffic sharing the same broadband connection
- Simultaneous execution of other applications that may reduce software performance.

However is our belief that with sufficient measurements and proper data analysis it is possible to overcome some erroneous samples and obtain an estimate of the perceived QoS provided by the ISP with a fairly good degree of confidence. Most

importantly with this type of project we should be able to obtain more data from scarcely populated areas and thus obtaining a better regional performance rating.

2.2 Related studies and platforms

In this chapter we present some similar projects both national and international. For each project we shall analyze the environment conditions, project goals, system architecture, test methodology (if available) and the selected metrics.

2.2.1 ANACOM – IxChariot

The Portuguese communication policy regulator ANACOM has been a pioneer in terms of conducting QoS and ISP benchmarking and keeps presenting annual reports since 2005. As previously stated in section 2.1.2 the tests are performed in a closed system where users are individually invited to participate and in a controlled environment where hardware capacity and simultaneous access are accounted and controlled. For the test duration the participants were restricted from accessing their Internet access service. As of 2009 [9] ANACOM has broadened the test scope and conducted the evaluation of ISP for fixed and mobile access.
For the 2010 [10] study in fixed access ISP a proprietary system IxChariot [11] (developed by Ixia [12]) was used to perform network measurements. The test lasted for 7 days and in each test site 61 measurements samples were performed. The distribution was such that 9 daily samples were performed if it was a weekday and 8 if it was during a weekend. After quality control 258 tests equally divided between all operators were considered valid and were the basis for the study. Furthermore results were divided and analyzed according to their hourly and geographical distribution.

IxChariot is an active measurement system that generates traffic designed specifically for the test purposes between relevant networks endpoint.

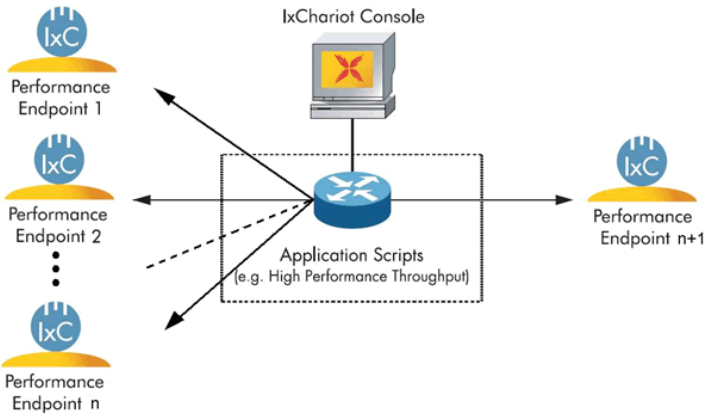

Figure 2 - IxChariot system architecture

The performed tests were end-to-end based between Performance Endpoints. The endpoints ranged 1 to n are software agents installed in the client's premises and Performance Endpoint n+1 are also software agents but installed in dedicated servers to simulate a client's connection when using the Internet. ANACOM selected 3 different "Performance Endpoint n+1": 1 national and 2 international. The national endpoint was installed in Portuguese "Fundação para a Computação Científica Nacional" (FCCN) [13] a non-profitable organization that maintains several public services as, the planning of several public institute's networks and management of the top-level domain .pt. The 2 other endpoints were selected to simulate an Internet connection to a European country (United Kingdom – London) and an overseas connection (United States of America – Houston).
Another key element in the system architecture is the "IxChariot Console" that performs the management, scheduling of the tests and is also responsible for collection and storage of the measurement results in coordination with the "Aptixia Test Conductor".

For the mobile access ISP all tests were performed in Portugal's 2 main cities (Lisboa and Porto) and with both public (schools, airports…) and private (residential) environment conditions. The test's locations were selected specifically to simulate as closely as possible the major locations were these services are used.

ANACOM selected 180 different locations to perform the measurements and in each location the 3 major Portuguese mobile access ISPs were tested. The test's frequency was the same has with the fixed access: 61 measurements per location, 9 per day if a weekday and 8 per day if a weekend.
The selected platform for the tests was a proprietary solution currently maintained by Ascom [14] TEMS Automatic [15].

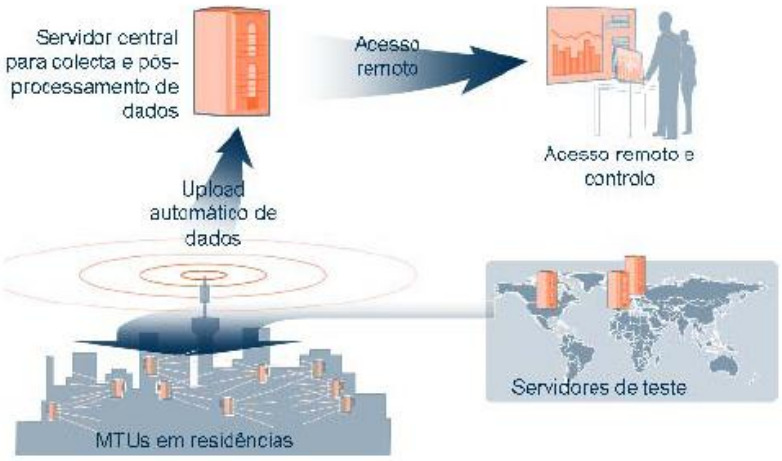

Figure 3 - System architecture used by ANACOM mobile tests, from [10]

This system is comprised by three main elements:

- Mobile Test Unit (MTU) – mobile equipment's that hold the SIM cards from the analyzed mobile access ISP and are where the performance test are initialized.
- Test servers – 3 servers to act as endpoints for the MTU's tests. ANACOM selected the same set of servers that had been used in the fixed ISP

benchmarking: 1 national server installed in FCCN's server, 2 international in London - UK and Houston – USA.
- Main server – repository database where all data is kept, processed and presented

The same set of parameters was selected for both types of Internet access.

- Service availability – percentage of tests that failed due to a network access failure

- Causes for network failure – this indicator establishes the cause of the network failure. This is done by analyzing the PDP Context in mobile networks or the IP connection in fixed networks.

- Activation time – represents the necessary time to establish a network connection. More relevant in mobile networks, when PDP context activation is required.

- Latency – measured as the RTT that a 256 byte ICMP Echo message takes between the client and the server.

- Loading time of a web page – the time that a user should experience when loading a single webpage.

- File transfer speed download/upload – for this test the download speed of a single file through FTP protocol was performed. The size of the file attended the rule that it should be at least twice of the connection's speed in kbps. For the wireless networks 4MB and 1,6MB files were used for download and upload respectively.

- Packet loss – for the estimation of this metric a streaming script was used as it is one of the services that suffers a greater impact in presence of packet loss.

- Jitter – This test was performed by sending a datagram with a timestamp. Upon reception the receiver adds another timestamp marking its intervention. Using these two timestamps a transmission time is determined. If there is a variance in these values the user is experiencing jitter. No information was described on how clock synchronization was performed.

- DNS lookup time – this parameter was obtained through successive queries to the DNS servers provided by the network's DHCP configuration. For each test 100 queries were performed using the Top 100 WebPages accessed by the Portuguese users.

2.2.2 LIRNEasia – AT-Tester

This tool was developed by the TeNet Group of IIT-Madres [16] and LIRNEasia [17] in order to perform QoS benchmarks for selected countries in South and

Southeast Asia regions. It is freely available for download but to be used it is necessary to register the user's ISP connection. The tool is written in VBScript and both the client software and test methodology are open-source. The measurement tests are performed against 3 different servers:

- ISP Level Test – using a server in the subscriber's ISP
- Country Level Test – using a server in subscriber's country but hosted in a different ISP
- Global Level Test – using a server outside the subscriber's country.

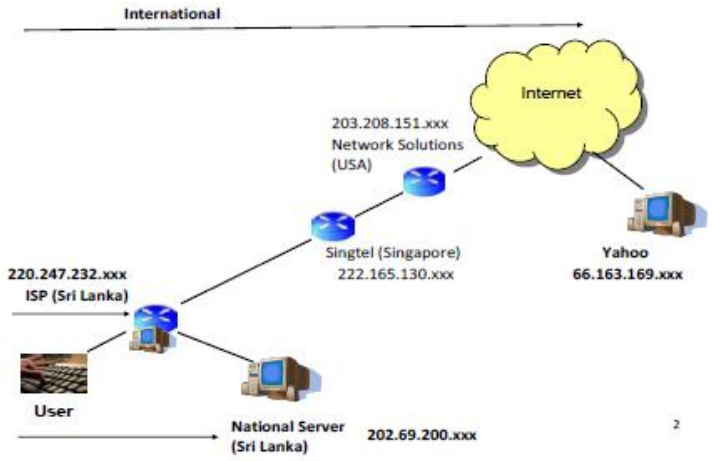

Figure 4 - AT-Tester system architecture, from LIRNEasia, from [18]

The tool supports both scheduled and on-demand tests and the results are stored and can be accessed locally. At any given moment the user may choose to upload their results to a central database where they are aggregated with other results from the same ISP. Scheduled tests are performed 6 times per day between 8:00 AM and 11:00 PM trying to cover peak/non-peak for business and residential hours. The total time taken for each test is about 5 minutes and the complete test suite requires 6 tests a day on at least 2 days.

The selected metrics that are evaluated in each individual test are:

- Download speed – this metric is obtained through a download of a 5MB file with the GNU wget [42] tool from the three servers previously described. The average speed is obtained by dividing the process duration by the file size.

- Upload speed – for the upload test a 1MB text file is generated by the client's software agent and uploaded to the server through a HTTP POST command to the endpoint servers.

- RTT/Llatency – using the standard ping application, this metric is obtained by estimating the average of 100 ICMP echo requests sent to the servers.

- Jitter – Using the RTT test Jitter is estimated through the variance of the result set

- Packet loss – also using the RTT results, packet loss is obtained through the percentage of messages that are not returned.

- Availability – The network availability is measured by checking the accessibility to the Internet. As previously stated, tests are scheduled to run 6 times a day. Hence if T is the total number of times the experiment is done for and F is the number of times Internet is unreachable, then the Availability for that user is calculated as:

$$(1–F/T) \times 100\%$$

Equation 1 – LIRNEasia definition of service availability

An attempt is declared as failed, if no connection is possible on any of the local, national or International site within 30 seconds. On the contrary if any one of the site is reachable within 30 seconds, the attempt is declared success.

2.2.3 FCC – Ookla Net Metrics

In the beginning of March 2010 the United States Federal Communications Commission (FCC) started a national enquiry and broadband assessment [22] to give consumers additional information about the quality of their connections and how it affects different content over the Internet [29]. Two different measurement tools are made available: Ookla [23] and M-Lab [24].

Ookla has been developing measurement tools for almost a decade and is a recognized global leader in web-based network diagnostic applications. Their sites [25][26] provide one of the most widely used Internet based measurement tool for private consumers. The tool is browser-based so there is no application to be installed on the user's computer. The only requirement is that the browser supports Adobe Flash. There are about 100 servers in North America and Europe and another 100 in the rest of the world. In Portugal there are 2 servers in major cities Lisboa and Porto. The test is performed against a single server that may be chosen freely by the user. The test duration is about 1 minute and the selected metrics are:

- Latency – The average of 10 samples of the RTT time for TCP packets using port 5060

- Jitter – This result is obtained by estimating the variance of all 10 samples.

- Packet Loss – The client sends 100 UDP packets to the selected server and the result is obtained by the percentage of packets that are not received.

- Download speed – this test starts by downloading several small binary files from the web server to establish an estimate for the connection speed. Based on this result one or several file sizes are selected to use for the real download test. Up to 8 parallel HTTP threads can be used for the test and the results are aggregated into 20 slices. The fastest 10% and the slowest 30% are discarded. This is done because the measured data is being sent over HTTP via Flash which encompasses some protocol overhead, buffering and throughput

bursting. Also because the test duration is very short the initial delay associated with the TCP congestion window may influence the final result. To prevent cache acceleration a random string is appended to each file.

- Upload speed – The upload speed test starts to generate a small amount of data in the client and sent to the web server to estimate the connection speed. Based on this result an appropriately sized set of randomly generated data is selected for upload and pushed to the server via a POST command. Up to 8 parallel connections can be performed and the 50% fastest samples are averaged to obtain the final result.

Ookla also provides similar application for the mobile operating systems: Android and iPhone, applying the same metrics and methodology.

2.2.4 FCC – M-LAB Network Diagnostic Tool

The other tool supported by the FCC broadband test is the Network Diagnostic Tool (NDT) [27] developed by measurement Labs (M-Lab) [24]. This is an open source project that is under active development by Internet2 [28]. Being a consortium of American universities their core network is implemented to cover only North American continent.

Two versions of this tool are available: a downloadable command line client application and a web applet to be ran from the browser. Both versions make use of the same performance metrics and methodology. The FCC's broadband test recommends the users to perform the tests with the web applet for the ease of use. When the user activates the M-Lab NDT application a Java Applet is downloaded onto the client's computer. The applet communicates with the server to run a series of short tests, as follows:

- Download speed - A block of pseudo random data is generated on the server and stored in memory. The server repeatedly transmits this data to the Java client for 10 seconds which is discarded upon acknowledgement by the client. The server calculates the average throughput by dividing the number of bytes sent by the test time. The server also makes measurements on a per-packet basis, calculating the capacity of the end to end path. The server measures the number of lost and a delayed packet, the instantaneous round trip time, and examines the client's network configuration to determine what, if anything caused a drop in the expected throughput.

- Upload speed - A block of pseudo random data is generated on the client and stored in memory. The client repeatedly transmits this data to the server for 10 seconds. The server acknowledges receipt of this data and then discards it. The server calculates the average throughput by dividing the number of bytes sent by the test time. The server also makes measurements on a per-packet basis, calculating the capacity of the end to end path. Results from the test are returned to the client at the conclusion of the test.

- Latency - Multiple measurements of the round trip delay and per-packet transmission/arrival times are collected. The delay for each measurement is summed and divided by the total number of measurements to generate the average round trip delay.

- Jitter – This parameter is obtained by estimating the variance of all latency samples.

2.2.5 Osservatorio della Banda Larga – Isposure

This tool developed by Epitiro [32] and Between [33] was the tool selected by the Osservatorio della Banda Larga [34] to conduct a national broadband performance test in Italy [36]. This project was performed on an uncontrolled environment where every ISP client may participate by simply installing a client software that performs scheduled and on demand performance tests. The test duration is a little over 1 minute and the results are available to the user in a tabular form for the summary of the test or in a graphical display to show the behavior of each individual metric. To be able to see their results the user must register their connection by providing the connection Id, whether it is a mobile broadband connection, the ISP identification and the package name, price and maximum speed.

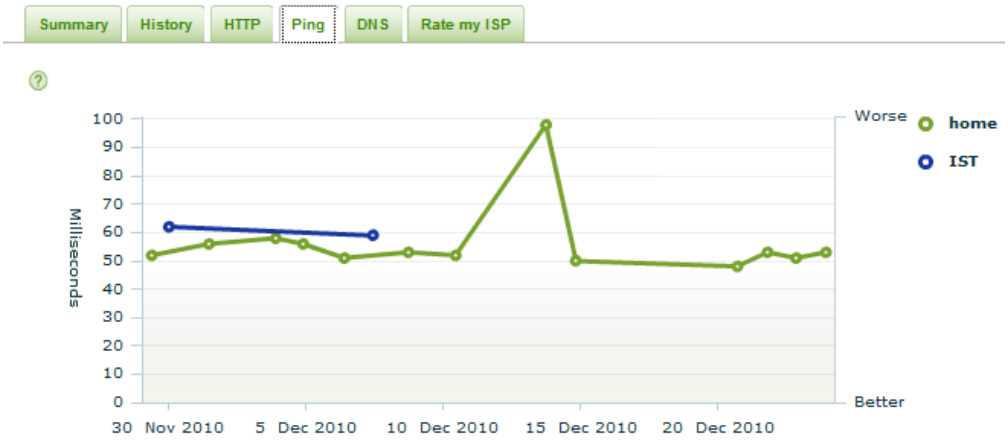

Figure 5 - Presentation of one of the Isposure's [36] test results

The metrics used to evaluate the ISP connection are:

- TCP throughput speeds (download / upload) – this metric is obtained by amount of data received in a determined time interval through a TCP connection. No other specification of the duration of the interval was made available.

- DNS Lookup – this test obtains the time in milliseconds that takes to perform a DNS Lookup in the ISP determined DNS servers.

- Ping time – referred previously as RTT/latency measures the time that an ICMP message takes to reach the server and return to the client.

- HTTP Download speed – this metric measures the time to download several web pages both national and international and returns the "web browsing speed". Although being directly correlated with the TCP throughput encompasses also the DNS Lookup, network control messages (more relevant in wireless access ISP) and aims to be an objective indication of the user's experience when surfing the web.

2.2.6 FCCN – Speedmeter

The "Fundação para a Computação Científica Nacional" (FCCN) [13] is a Portuguese non-profitable organization that maintains several public services as the planning of several public institute's networks and management of the top-level domain .pt. Its goals are to improve the connection between all IP networks implemented by Portuguese ISP and to avoid the use of international resources to forward IP packets with source and destination in Portugal.
One of the managed networks has a particular relevance for this project: the GigaPIX is a network that connects all major ISP that are currently operating in Portugal. This makes FCCN a privileged position to place a measurement station as it can be considered a neutral point for all referred ISPs.

Figure 6 - List of ISP connected to the GigaPIX network

FCCN has developed and currently hosts a web-based application [38] to perform on-demand bandwidth measurements. The system is built as an open system and is available to any user without registration or prior validation. Its main goal is to provide a tool for users to evaluate their connections and to the best of our knowledge its results do not serve any other purpose. As no registration is possible it is not possible to keep track of the user's measurement result history.
Currently on its second version, it is flash based application and performs the following measurements:

- Download/upload speed – consists in downloading/uploading a set of randomly generated data, between the FCCN server and the user's equipment. The test duration is exactly 10 seconds and the metric is obtained by the amount of data that can be transferred during the test duration. Prior to the test there is a procedure to stabilize the server connection and to avoid traffic peaks and thus influence the results. The result is the maximum, average and minimum bandwidth in Mbit/s for both upload and downloads.

- Ping / Latency – this metric is obtained by the observation of the RTT of 20 TCP packet. The available results are the minimum, maximum and average RTT, the variance of the RTT (jitter) and connection establishment time.

2.3 Evaluation

We have presented several projects representing different test methodologies and environment settings. This chapter summarizes these projects and presents the main lessons learned used to evolve from the base ground of our project.

The following table summarizes the main differences of all projects in terms of test environment and methodology.

	Participant selection	**Environment**	**Method**	**User convenience**	**Frequency**
ANACOM	Closed	Controlled	SW Agent/ Default HW	Bad	8/day – 7days
Epitiro	Open	Uncontrolled	SW Agent	Good	N/A
LIRNEasia	Open	Uncontrolled	SW Agent	Good	6/day – 2 days
FCCN Speedmeter	Open	Uncontrolled	Web Applet	Optimal	User dependent
FCC Ookla	Open	Uncontrolled	Web Applet	Optimal	User dependent
FCC MLab	Open	Uncontrolled	Web Applet	Optimal	User dependent

Table 1 – Summary of test environment and methodology characteristics

- In terms of participant selection, ANACOM's project differs from all others by using a closed system where participants are individually invited. All other projects do not regard this as a key point and used an open system where participants are free to join or leave the project at any given moment. We believe ANACOM's method is the correct procedure and allows a statistical precision necessary for an overall country wide ISP evaluation. However our project does not aim to perform overall ISP comparison and thus we do not need to restrict participant access to our project. In fact our

project encourages user participation in order to gather sufficient data from scarcely populated areas.

- In terms of test environment ANACOM's also differs from all other projects by using a controlled setting. This is done by installing a standard set hardware in the participant's premises and by restricting participants from using their network access during the whole process. All the other projects perform the measurements using the participant's own equipment. Also they do not attend to the fact that during the measurements some other process or user is sharing the same network access. Although ANACOM's method provides more accurate KPI, it has serious impact in user's convenience and discourages participants to enter the process. As previously stated our project relies on the gathering of significant data, so it is unfeasible to impose any restrictions.

- For the method of collecting measurements two possibilities are presented: using an online web applet or a dedicated software agent. The online web applet has the advantage of having optimal user convenience, as no installation is needed and participants willingly perform on-demand tests. Also we believe that it should return more accurate measurement as participants are aware that they are conducting a measurement test and, in principle, voluntarily turn off all other applications that might share the network access. However the software agent has one key advantage. By using an automatic scheduler we can have some control over when the measurements are performed and allow for more samples from each participant to be obtained. For our project we favored the advantages of having a software agent and tried to minimize the impact of having an installed software by developing a light application with minimal user interaction and a user-friendly interface.

- For the test frequency each ANACOM's test lasted for 7 days and performed 8 or 9 daily tests depending if it is a weekday or weekend. LIRNEasia method differs in the fact that for the measurements of any participant to be considered valid and enter the estimation of the overall ISP results they should have at least 6 measurements from at least 2 different days. For our project we do not think it is relevant to impose any minimum number of samples to consider the measurement valid. Also because this is to be a continuous system and as expected have different levels of user participation, we should be able to dynamically change the frequency of the measurement tests in order to prevent overloading our servers. This decision should be left to the system administrator and requires some human supervision.

In terms of key performance indicators (KPI), although there are no standards for network measurements 5 types of KPI that are somewhat common in these projects: Round Trip Time, Packet Loss, DNS resolution time, Web page loading time,

Throughput (download/upload). In some projects it is also considered the service availability. In the following table we present a summary of the several project choices of KPI and some technical information of their differences.

	RTT	Packet Loss	DNS	Web load time	Download/ Upload	Service Availability
ANACOM	+ ICMP	+ (UDP Streaming)	+	+	+ (FTP Single file)	+
Epitiro	+ ICMP	–	+	+	+ (HTTP)	–
LIRNEasia	+ ICMP	+ (from RTT)	+	+	+ (HTTP Single thread)	+
FCCN Speedmeter	+ TCP	–	+	+	+ HTTP Single thread	–
FCC Ookla	+ UDP	–	+	+	+ (HTTP 8 threads)	–
FCC MLab	+ UDP	–	+	+	+ HTTP	–

Table 2 – Summary of studied KPI

- In terms of estimating the RTT there are several valid possibilities. As expected ICMP Echo/Reply message is the preferred method, as it is the most known and there are built-in applications (ping) available in all operating systems.

- Only ANACOM and LIRNEasia chose to measure the packet loss. ANACOM chose to send a burst of UDP packets and LIRNEasia used the number of ICMP failed messages used to estimate RTT.

- All projects support and measure the DNS resolution time and web page loading time. However, only ANACOM can assure that the measurements are performed with the ISP default server. All other projects cannot predict the fact that the participant could manually change the server.

- When analyzing the throughput most projects selected the HTTP protocol as it is commonly the preferred method by internet users when download files. FCC project using Ookla applet presented the novelty of using several threads in the download/upload process. We considered being the correct approach as some ISP may have some traffic shaper that could limit the bandwidth per TCP session.

For our project we wanted to present the most comprehensive set of measurements, so RTT, packet loss, DNS, web loading time and throughput were selected.
We chose not to include the service availability KPI as in an open system there could be several non-network related factors that could influence the result. For instance the user might deliberately remove the WAN cable, or is simply because all interfaces are disabled. The main advantage would be test the DNS availability, but as previously stated the participant might have manually selected a DNS server other than the ISP default obtained by DHCP.

3. System Architecture

The goal of this project is to build a platform for the evaluation of the quality of service of the internet access. The results are obtained by analyzing the KPI obtained from active measurements in a client-server topology. In this chapter we will present the logical and physical system architecture that formed the basis for implementing the current project.

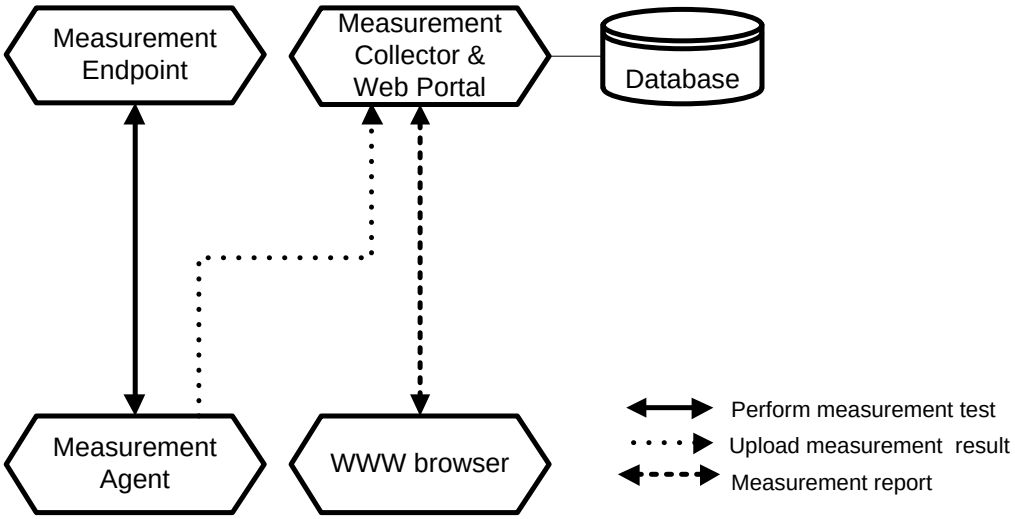

Figure 7 – Logical system overview

In Figure 7 we present the logical overview of the proposed platform. It is comprised of 5 main elements: a measurement agent, an endpoint a collector with an associated database and a standard browser.

The "Measurement Agent" will start scheduled or on-demand measurement tests. This test will gather the KPI that allow us to evaluate the QoS of the Internet access service. The "Measurement Endpoint" is another key element for the measurement process as it will accept and respond to the "Measurement Agent's" requests in order for the several KPI to be estimated. After the measurement process is over, the "Measurement Agent" will upload the results to a central collector that will aggregate measurement data from all agents and persistently store them in a database. The "Measurement Collector" also has the function of displaying the user's or administrator's report from the aggregated measurement data. These reports are made available through the web portal and therefore should be accessible through any standard browser.

In terms of physical architecture we propose the following structure:

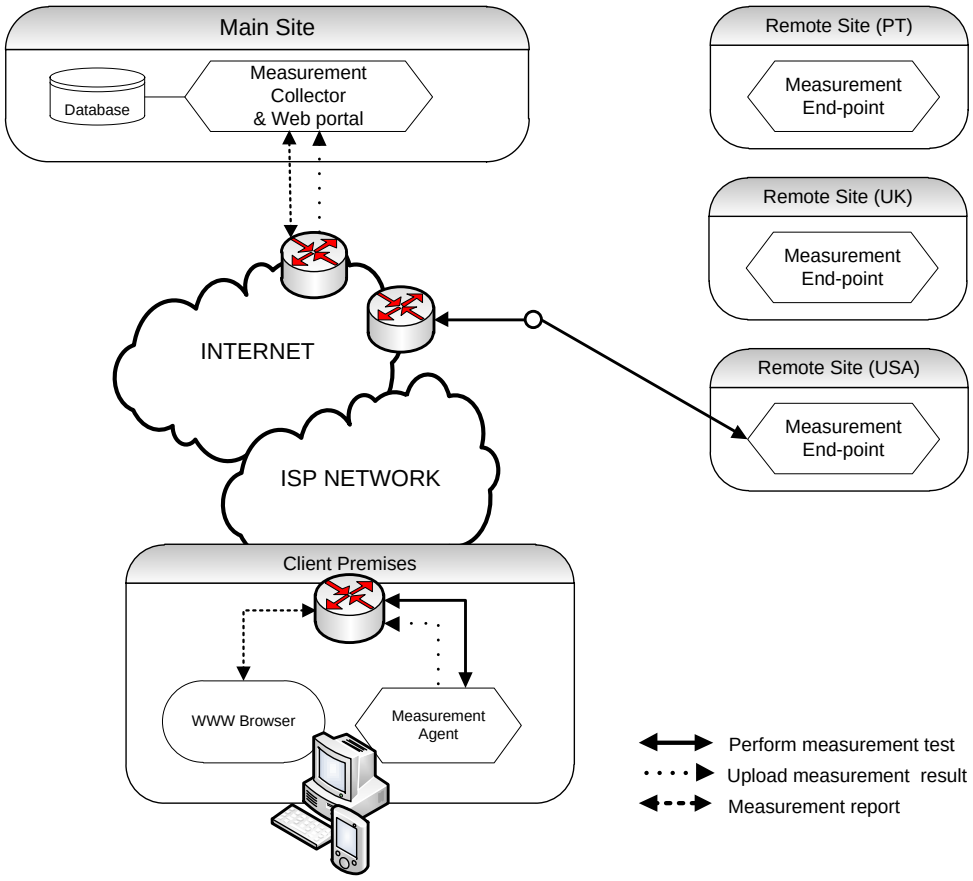

Figure 8 – Physical system architecture

The proposed system infrastructure is composed of 3 elements.

The "Main Site" holds the infrastructure to run the measurement collector and should be physically separated from any measurement endpoint. This is to prevent functions with different classes of services sharing the same bandwidth. For instance we do not want to affect the measurement test results by having other users accessing their reports or an administrator running several reports.

In the "Remote Site" we have the measurement endpoint logical element. Although being one logical entity, it is considered a benefit to have several sites with instances of measurement endpoint. This allows us not only to place endpoint sites in different countries and thus evaluate international connections but also to enable us to spread the measurement test through more sites and prevent network congestion. The number of endpoints should vary throughout the lifetime of the platform and will be directly connected to the number of active participants. The administrator's report provides an estimate of the number of measurements that were made in a determined time frame and along with the measurement results for each endpoint provides sufficient information for the system administrator to assess the necessity of installing or removing endpoints.

The "Client Premises" will hold the client side application that, as previously stated, we have chosen to be an installed software running in the participant's equipment. As previously stated we devised two versions of the measurement agent: one for desktop and another for smartphone. This site also contains the router

that acts as a gateway to the ISP network and the Internet, from which our remaining sites can be accessed.

3.1 System utilization

In terms of system utilization we present the following use case diagram:

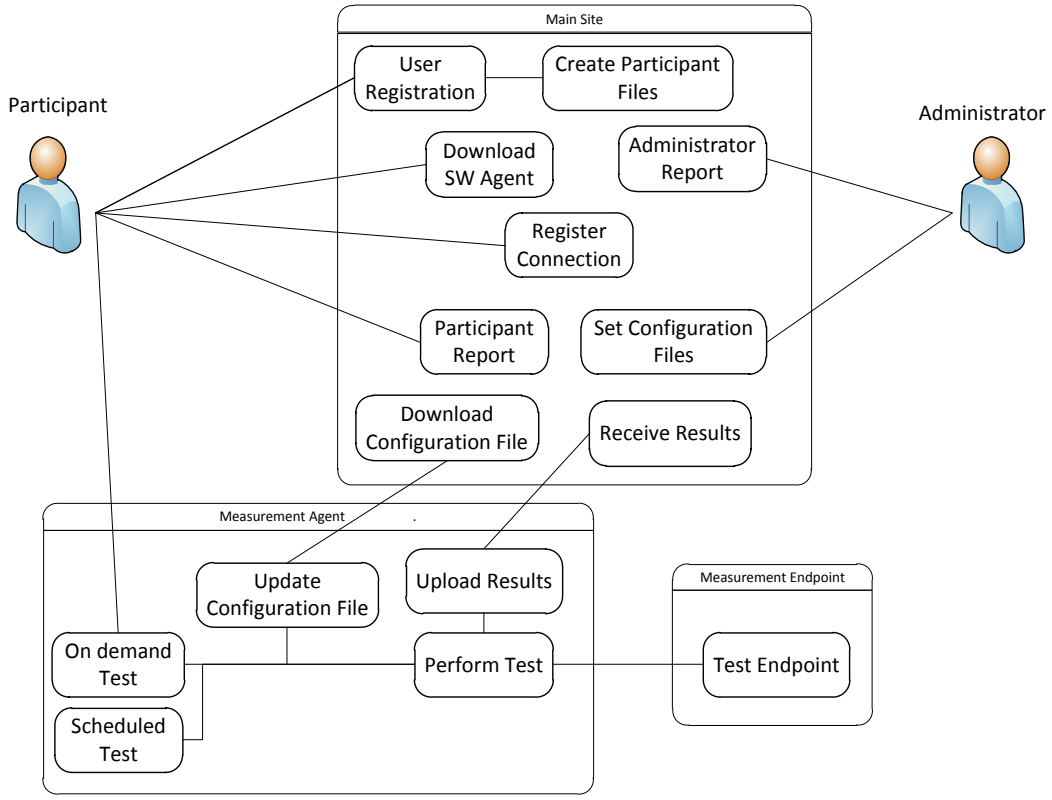

Figure 9 – System use case diagram

The system starts as users enter the web portal and performs the registration. This process triggers the creation of a participant directory that will hold their custom configuration files and the creation of a measurement agent that will be saved inside the database.

After this procedure the participant will perform the download of the measurement agent. In case of the Desktop version it will be distributed from the web portal. In case of the mobile version it should be distributed through the OS application market.

The next step would be to perform a measurement test. This test could be triggered either by a scheduled test or by a participant originated on demand test.

Before the test starts we proceed to check if any updates have been made to the participant's configuration file. This allows us to have a centralized architecture that enables us to control several features related to the measurement process. This file contains several important definitions such as the main site and endpoint

addresses, the test scheduler and other information regarding the estimation of the KPI.
The test is then performed measuring the selected KPI. Most of the measurement tests are performed using the endpoint as the target of the communication, although there are also some tests that will be performed against public servers. The measurement process will be further detailed in the following chapters. After a measurement test has been successfully completed the results will be automatically uploaded to the main site.

At this point the test results will be categorized as being performed through an unregistered connection and will not be included in the several measurement reports available at the main site. For the results to be available for display, the participant must access the web portal and perform the connection registration. This process will ask the user to enter the ISP, the contracted download speed and in case of a residential connection the postal code of the location of the client's premises. In the mobile version we will extract the location from the GPS coordinates if available.

The administrator has privileged access to a special report that aggregates the results from all registered connections. This report allows the generation of aggregated data through multiple variables such as connection's location, contracted throughput speed, ISP, etc. Another function that is reserved for the administrator is to establish the configuration files for each participant.

3.2 Measurement process

The measurement process is executed by the measurement agent and with the aid of the endpoint will estimate the KPI that will be uploaded to the Measurement Collector.

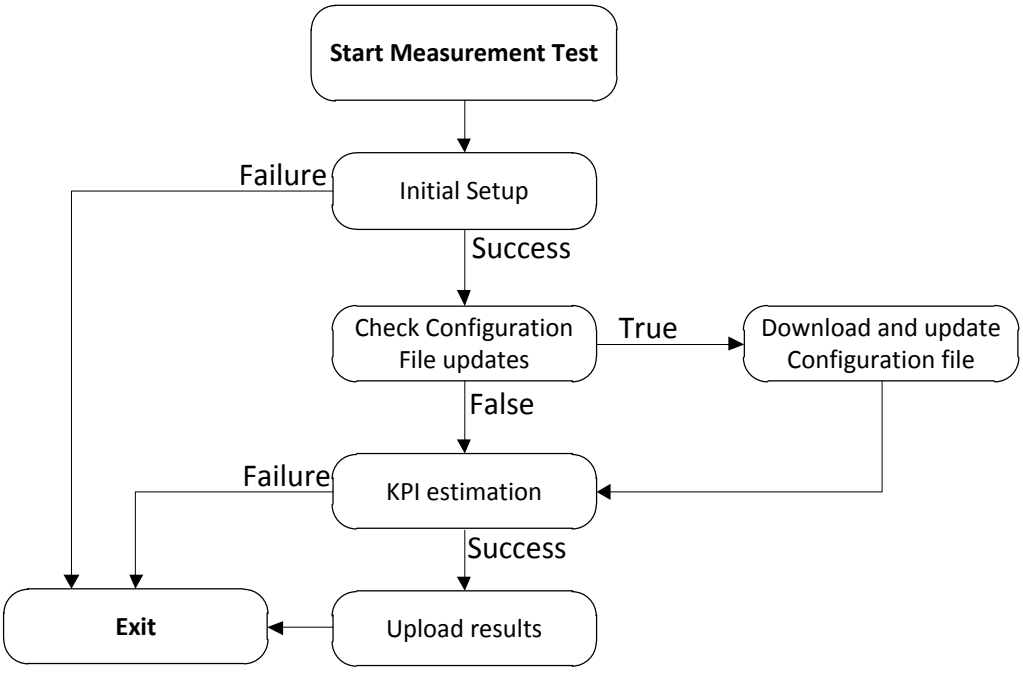

Figure 10 – Measurement process high level design

The measurement process is composed of two main elements: the initial setup and the KPI estimation.

In the initial setup we gather information regarding the network connection. This is an important challenge when developing this type of applications because we need to differentiate between ISPs or even between two different connections form the same ISP. This is an essential question because with the massification of the mobile broadband nowadays it is fairly frequent to find users that have a fixed and a mobile access connection and usually from the same provider. This will be relevant not only to allow participants to distinguish their measurement results in their personal report but also to build an accurate administrator report. These network characteristics include identifying the network interface that is being used, public IP address, and ISP identification.

After the successfully completion of this step we proceed to communicate with the Main Site and assess if there is a new configuration file available. This configuration file holds general information about the scheduling of new tasks, address of the Main Site and Remote Endpoints, and other parameters concerning the estimation of the KPI. The configuration file will be detailed in chapter 4.1.1. For the KPI estimation we will select a remote endpoint and perform the measurement test that will be uploaded to the Main Site.

3.2.1 Initial Setup

As previously stated, the Initial Setup procedure consists on gathering system and network specific characteristics that will allow to uniquely identify the connection. For the Initial Setup procedure we have developed the following high level design.

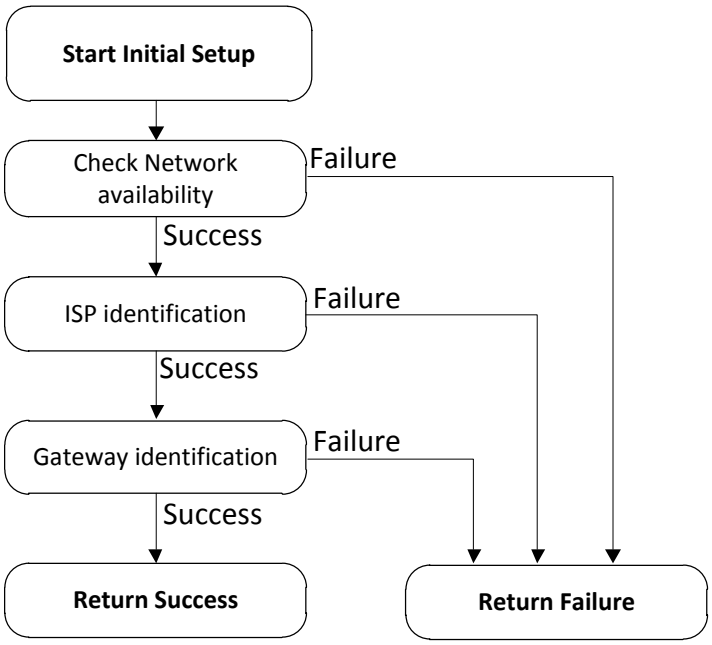

Figure 11 – Initial setup procedure high level design

The first elemental step of the measurement procedure is to asses if the network access is available. After this step we then proceed to identify the connection. Our approach to perform this task is the combination of the ISP identification and a connection characteristic that is unique to every participant.

ISP identification

To be able to distinguish between different ISP we first studied the possibility of building a list of all public IPs from Europe's Regional Internet Registries (RIPE) database and map them to the Portuguese ISP available in the current project. This information would be inserted in our database and would allows us to uniquely identifying the ISP. This soon proved to be an unfeasible task as the ownership of the IP address are constantly changing and the mapping process revealed to be extremely difficult. The next step was to perform the query interactively to the RIPEs database. Our approach is to have the measurement agent perform an online query to the RIPE database through their website, parse the resulting HTML text and return the netname and description fields. As an example we present the identification of the IST University RIPE identification.

```
netname:        UTL-8
descr:          Universidade Tecnica de Lisboa
```

This has the advantage of always having updated information but relies on the content of the RIPE's webpage. If the webpage administrators decided to change the labels to any other description, it would fail to obtain the ISP identification. A possible solution would be to build a whois client and perform the query directly to the RIPE server. This was not implemented and will be addressed in the Future Work chapter.

Gateway identification

However, the ISP information is not sufficient to uniquely identify a connection as one user might have several connections from the same ISP. For that we need to save a unique characteristic from the each ISP connection. This is not a simple task as there are several connection possibilities that use different protocols.

For the residential connection our approach to implement this task is to extract the MAC address of the router present in the Client's premises that acts as the gateway between the user and the ISP network. The Media Access Control (MAC) address is obtained by sending some packets to the endpoint site and through the help of a network listener, read the MAC address of the packet that was returned by the endpoint. A simpler process would be to issue the route system command (in Windows "ROUTE PRINT" and parse the resulting string in order to find the default active route. From that we would find the gateway address and through and ARP command find the gateway's MAC address.

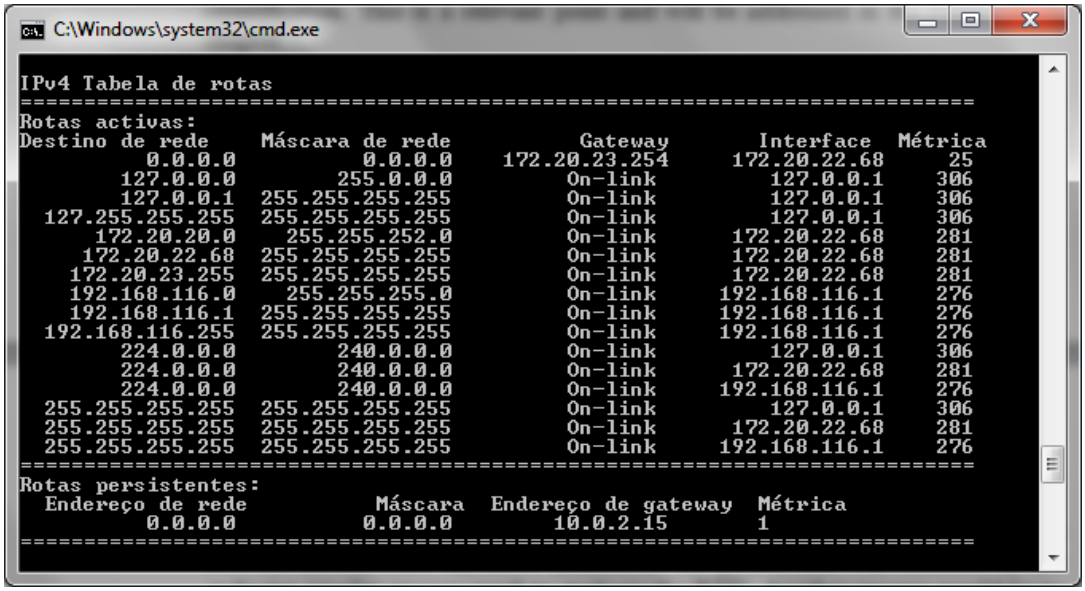

Figure 12 – Example of a route system command in Windows

However as can be seen in Figure 12, implementing the parser to extract these elements from the result of a route command would not be a simple task. Having a virtualization system installed (such as VMWare) will add several additional routes both active and persistent. Furthermore, the route system command is very system dependent and even with the same OS version it will significantly differ according to the installed language pack. In order to accomplish this task we would have to perform extensive test set using several OS and with different language packs.

In the mobile version of the software we cannot get the MAC address because mobile ISP usually have a pool of gateways GGSNs that are chosen randomly and certainly are not willing to distribute this type of information. Thus our approach was to save a hash of the user's International Mobile Subscriber Identity (IMSI) (present in the SIM card) and the International Mobile Equipment Identity (IMEI) (from the mobile phone) used during the measurement test. Although it would be more meaningful to the user to have these elements in clear text, we considered it to be a security threat and, in line with several previously described options, we chose to have this information encoded.

Another exception occurs when a participant using a desktop/laptop connects to a wireless ISP using a 3G USB modem. These connections establish a PPP connection with the ISP network and therefore there are no associated Ethernet frames. Also, in this case, we were not able to extract the previous parameters without specific modem drivers. Our approach was to extract the connection name as present in the OS. This parameter is a connection specific characteristic and usually contains the name of the ISP.

The last case we identified is when a participant is connected to the internet but through a Layer 3 VPN connection. Although underneath the participant may use any of its connections the internet communication would first be made through a remote site and probably will use a different ISP. Therefore we concluded that it would be unacceptable to aggregate the results from these types of connections.

Our approach for this case was to proceed as the previous case and extract the connection name as seen by the OS.

As previously stated, this information will be present in the participant's report and will allow users to distinguish between their ISP connections.

3.2.2 Key Performance Indicators

The parameters to be evaluated should be directly related to the most common utilization that consumers perform and it is generally accepted that the main uses are web browsing, file transfer, audio/video streaming, peer-to-peer (P2P) applications and online gaming.
However given the wide range of participants it is expected that some or most participants (especially in wireless access ISP customers) may have some kind of network constraint whether it may be in terms of amount of traffic generated or through some time limitation. Therefore the measurement process should also encompass this constraint and have a fairly light impact in the consumer's contracted plan rate.

RTT and jitter

When dealing with real-time applications such as VoIP or video streaming, network delay plays a central role in determining the quality that a user experiences. As our analysis focus is on the user's point of view, an efficient performance indicator is to estimate the transport connection's round-trip time (RTT) and how these values change through the connection's duration. An extended analysis of the network delay in each direction (server-client and client-server) could provide additional information of the network characteristics, but as stated, our focus is centered on the user's perception of the offered service. Also, by measuring in a single measurement point one can avoid the need to have absolute clock synchronization of the measurement points.

The behavior of the RTT throughout the connection's duration is estimated through jitter, which is regarded as the variation of the delay. Jitter has several definitions [43], e.g., the maximum variation of the delay, but the most common one is probably the standard deviation of the delay. It's estimated through the analysis of the delay difference between sequential packets. The impact of jitter can be limited with buffers, but this is done at the expense of delay. [43]

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Equation 2 – Definition of jitter

Several options are available to obtain these measurements. During the development of this project our first approach was to use the default ICMP Echo/Reply message. These statistics could be obtained through the common ping application present in all operating systems. However the resulting string that we would need to parse is system dependent. Even the same operating system, using a different language pack, will have a different resulting string format. We would have to conduct tests and have a multitude of options contemplating all possibilities. Furthermore during our tests we found that some ISPs are currently blocking ICMP messages, thus invalidating this option.

Our approach was to use a standard Java UDP Datagram socket connection and measure the RTT of a 32 byte packet. In the measurement point we developed a simple Java application that permanently listens in a designated port and simply returns the received packet. From these measurements we obtained an average, maximum and minimum RTT and associated jitter.

Packet Loss

Packet loss is an important metric to assess the quality of real-time applications such as VoIP and video streaming. Packet loss has a direct effect on other metrics. Consider for instance, an application layer jitter measurement, where jitter is measured directly from sequentially arriving messages. If lower layers fail to deliver some packets correctly, jitter is increased because of the gaps of missing packets. It might also be the case that the application layer packets are all delivered correctly, but jitter is still increased because of erroneous lower layer packets and retransmission. In this case the delay is also increased. To estimate this metric we shall use the previous RTT test and record the number of unanswered messages.

Loading time of a web page

One of the fundamental aspects when evaluating an Internet connection is determined by the time that a webpage takes to load. Although being related to the download speed, this process encompasses also DNS lookup time and additional delay related to network's control message TCP and HTTP protocol. For each test the software agent will look in the configuration file for a list of the 100 most visited websites by Portuguese Internet users. From that list only 20 random sites will be selected and accessed.

For performance reasons we only obtain the interpreted HTML text. No images, videos or animations are downloaded. The returned measurements are the average, maximum and minimum web page loading time and the associated jitter.

Domain Name System lookup time

One parameter that has a great influence on the performance of the loading time of a webpage is the response time of the Domain Name System (DNS) server. In order to assess the average response time of the ISP's DNS server a battery of lookup operations should be performed. From the same list of the 100 most visited

websites (extracted from [47]), we take a subset of 30 samples and perform DNS queries. For this purpose we used the standard java.net package that automatically selects the configured DNS server and performs the queries.

Before conducting this measurement we issue a system command to flush the cached DNS resolution queries from the system. This prevents obtaining false measurements due to local resolution but does not prevent the home gateway router or ISP DNS server to contain some cached queries. We did not find any way of preventing this situation and regarded it as a system limitation.

Also, if the user manually configures a different DNS server other than the ISP's default server, some misleading results will be obtained. A possible solution would be to obtain a list of every ISP DNS server and disregard any measurement test that did not include one of those servers. For security reasons, this information is not publicly available and this issue was also considered as a system limitation

The standard average, maximum, minimum and jitter resolution time are taken from this measurement test.

Download/Upload speed

This metric is defined as the rate obtained during the simultaneous download of 3 files from the remote endpoint through TCP connections. Some attention should be paid when defining the file size of the transferred file, as the TCP congestion avoidance mechanism causes the transfer rate not to be continuous for the duration of the connection. This rate increases as the window size also increases, stabling in long term in a value determined by the network congestion and by traffic management mechanisms. Therefore, it is important that the size of the file should be sufficient to overcome the initial slow start period of the TCP congestion window. On the other hand some ISP's plan's rates may restrict the downloaded traffic to a specified amount, so if the measurement process takes a noticeable toll on that volume, we might expect some users to drop out of the process due to significant consumed traffic. A recent tendency implemented by ISPs is to provide network connections without traffic restrains but limit the connection's speed after a determined amount of traffic. This cannot be determined by the agent, so our best hope is to constrain the amount of transferred traffic and disregard results that are abnormally lower than the average. For the current project we do not prevent such deviations and it will be addressed in future work.

We have chosen to include the file size as part of the configuration file present in the software agent. In this way the administrator may freely change this parameter based on the load of the endpoint or perhaps on user's feedback reporting excessive download traffic generated by our application.

For this KPI we generate two types of results: the average and maximum speed.

To obtain a maximum throughput speed, right after the file transfer process begins, we start a network packet capture (e.g. Wireshark). This listener counts the total transferred bytes from/to the selected endpoint and every 2 seconds estimates the

download throughput. From that list of measurements we will select and return maximum throughput observed by the listener.

The average speed is estimated by the ratio of the sum of all files with the overall time that all threads took during the download procedure.

A different measurement method was implemented in the case of mobile version of the measurement agent or in case of a PPP connection (such as 3G USB modems). Because these connections do not use the Ethernet frames, the network listener is not able to listen to these interfaces. Also these types of connections usually impose more restrictive download traffic limits. In these cases we only perform a single thread download and to estimate only the average throughput speed.

These processes are identical for both download and upload measurements.

3.2.3 Scheduler

For the process of running scheduled measurement tests we have devised a custom scheduler that allows us to finely set different hours depending on the current week number.

Because some users might not have a flat traffic plan rate, this scheduler allows us to define different test frequencies for different weeks. This type of plan resets in the beginning of each month and we considered sufficient to set the week number as maximum granularity that we can achieve. The configuration file that contains the scheduler is included in the measurement agent and every time a new measurement test begins is checked for updates.

In every month we will have at most 6 weeks, depending on the weekday the month starts. However we found sufficient to group the 5^{th} and 6^{th} week, as the 6^{th} week will have at most 2 days. In each week we differentiate between weekdays and weekends.

Week 1	Weekday 1	Weekend 1
Week 2	Weekday 2	Weekend 2
Week 3	Weekday 3	Weekend 3
Week 4	Weekday 4	Weekend 4
Week 5	Weekday 5	Weekend 5

Table 3 – Definition of the scheduler division in a moth

Then we define a list of test frequencies per day.

Frequency 1 – 8:00 (30m); 12:00 (15m); 16:00 (20m); 20:00 (1m)
Frequency 2 – 8:00 (15m); 14:00 (10m); 18:00 (1m)
Frequency 3 – 2:00 (120m);

To prevent users from starting their measurement test simultaneously we added another element in each hour that defines a maximum number of minutes to wait before starting the test. When a scheduled test ends, the measurement agent estimates the time to sleep until the next scheduled test and additionally adds a random minutes that range from 0 to the maximum minutes defined in the frequency.

Then we assign each weekday/weekend a frequency.

Week 1	Weekday 1	Frequency 1	Weekend 1	Frequency 2
Week 2	Weekday 2	Frequency 1	Weekend 2	Frequency 2
Week 3	Weekday 3	Frequency 1	Weekend 3	Frequency 2
Week 4	Weekday 4	Frequency 2	Weekend 4	Frequency 3
Week 5	Weekday 5	Frequency 3	Weekend 5	Frequency 3

Table 4 – Example of a frequency assignment

As previously stated, plan rates that do not have unlimited traffic reset counters at the beginning of each month. So to minimize this impact on the user's bills, we will want to set a frequency with a higher number of measurements in the first weeks and diminish the frequency by the end of the month.

Finally, we can also set some exception days that could have their own test frequency. These could be holidays or perhaps a scheduled day where we would want to perform extensive server maintenance. In this case we could set a frequency type with 0 scheduled tests and assign this repetition to that particular day.

The full scheduler will have the following settings:

Frequency 1 – 8:00 (30m); 12:00 (15m); 16:00 (20m); 20:00 (1m)
Frequency 2 – 8:00 (15m); 14:00 (10m); 18:00 (1m)
Frequency 3 – 2:00 (120m);
Frequency 4 –

Weekday 1	Frequency 1	Weekend 1	Frequency 2
Weekday 2	Frequency 1	Weekend 2	Frequency 2
Weekday 3	Frequency 1	Weekend 3	Frequency 2
Weekday 4	Frequency 2	Weekend 4	Frequency 3
Weekday 5	Frequency 3	Weekend 5	Frequency 3

1 January	Frequency 4
14 October	Frequency 2
25 December	Frequency 4

Table 5 – Example of a full scheduler implementation

3.2.4 User identification

During all communications between the measurement agent's and the web portal we have to provide a method to check the confidentiality, authorization and authenticity of the message.

By using a standard X509 public certificate we can assure that the communication is encrypted and prevent any eavesdropping. However to prove that the user is registered in our project we have to include in the measurement agent the username and password of the participant. Our first approach was to ask the user to perform the login after the installation of the software. Then we save the participant's credentials in an encrypted file using a predefined key that would be common to all measurement agents. We considered this method flawed because having a separated encrypted file even with a strong encryption scheme would make an easy target for dictionary attacks. Furthermore if the common password got revealed it could risk the entire project. To restrain some of these security flaws we devised a custom process that includes the username and password in the measurement agent source code. This is done during the web portal's registration process. We take the participant's credentials create the and before registering them in our database we run a simple bash script that modifies the source code and includes the credentials. The username is set in clear text, but the password is firstly hashed using a SHA-256 digest. Next we recompile the program, create the installation package and save it in our database as a binary object. Every participant will then have their own version of the measurement agent and no registration is necessary other than the one provided by the web portal.

To provide authenticity, the measurement agent includes in the URL the participant's username, a timestamp and a digest. This digest will be result of a hash process using the SHA-256 function using the referred username, timestamp (to provide variability) and the hashed password. The web portal will read the username and timestamp, create its own digest with the hashed password present in the database. If both digest match the measurement results are accepted and saved. Otherwise the message is discarded.

We do not consider this to be an unflawed solution as there are several programs that can take the compiled program and reconstruct the source code. To prevent this we should use a program obfuscator that prevent these attacks. This was not included in the current version of the project and will be addressed in the future work chapter.

For the mobile version of the software we had to revert to our first process as the main distribution channel will be the Android Market and therefore are not able to have individual software for each user. We devised an initial login form in the measurement agent and a web portal page to validate the participant's credentials. If accepted these credentials are kept in the application's Shared Preferences file. This is a special file in the Android OS that allows a single application to share information between its several elements. In the Android OS philosophy each application is regarded as a different user and has its own set of files that are private to that particular application. This way no other application or the user may access that file and read its contents.

However, this method is not perfect as there are several available hacked versions of the Android OS versions that provide root access to the system and therefore to all application's Shared Preferences files. A possible method to minimize this threat would be to encode the credentials using a predefined password common to all software agents and included in the source code. This was not implemented and will be addressed in the Future Work chapter.

3.3 Data Security

Because the system is to run unattended, some consideration should be performed when dealing with the transfer of the results and the storage of the user's credentials. Our approach to this matter includes several interesting options in order to achieve the best tradeoff between data security and usability.

Firstly, all communications between to the Main site web portal should be encrypted through an X509 standard for certificate's Public Key Infrastructure and signed by a trusted certification authority. This prevents eavesdropping or man-in-the-middle attacks. For the prototype version that we have developed, we generated a self-signed certificate that although do not enables the same security strength as standard X509 signed certificates, enables us to encrypt all messages and during our tests predict the computational load on our web server. Besides the encrypted connection, when we upload an test result to the main site we will create an digest of the message combining the sent message, the user's hashed password, and timestamp . This way we can make sure that the message was not intercepted on route and the source is in fact one of our measurement agents.

The code in the web server is also prepared to resist any SQL injection attacks where a badly intentioned user enters SQL commands in HTML/PHP forms in order to gain access to relevant data in our database. This was done using PHP function code to get the real escape string instead of the clear string presented by the user. Furthermore we disabled all directory listing and prevent users from accessing any page without previous authentication.

Another important security feature is not to keep any clear passwords in the database. All user passwords are hashed using a cryptographic hashing function (SHA-2) using an output of 256 bit word. During the process we also include a 10 digit salt (a random generated number) to be included in the password before the hashing function. In this way two users with the same password generate different hashes. This element is also kept in the database next to the complete hashed password.

In the desktop version of the measurement agent, the user and password is kept coded inside the program. Upon a user's registration, the web portal runs a script that takes the entered username and password changing a source file, recompiles the program and saves the program into the database as a binary object. When the user downloads the program it will reference the recently created program and no other credentials are exchanged.

For the mobile version (Android OS) is not feasible to only allow the distribution of the program to be made in the web portal and disregard the official distribution channel of the mobile operating systems. Therefore, we cannot dynamically recompile the source code. As such, in this version we developed an initial login process to be performed only once by the user after registration in our web portal. After a successfully login, we store the username and hashed password in the private shared settings of the Android operating system. Although not being a perfect security process, the shared settings file is a private file only accessible by the application that created it.

Another pertinent aspect is the privacy of data that is kept with the test results. No personal information of the user should be recorded other than the geographical area where the access is installed. Ideally when evaluating a user's connection with a wireless access ISP, no telephone number should be recorded or associated with the GPS position obtained from the Smartphone. However, given the fact that we need to distinguish between different connections we will have to create an association between the SIM card and the phone. Our approach was to get the IMSI, IMEI and username and save a hash of the combination of these elements. Although it would be more meaningful to the user when trying to distinguish between different connections to simply have their phone number, it would a considerable security risk to store clear unencrypted information available when visiting our web portal.

4. System Implementation

In this chapter we will describe in more detail our approach to obtain the processes and results referred in previous chapters.

4.1 Measurement agent

The measurement agent application was built entirely using the Java programming language and was developed to be as portable as possible. The only dependency is to have installed the libpcap library (for Microsoft Windows OS, the correspondent WinPcap library). All other required libraries are fully portable and are included in the jar package. The external libraries included in the software are:

Apache HttpComponents HTTPClient

The Apache HttpComponents project is responsible for creating and maintaining a toolset of low level Java components focused on HTTP and associated protocols. Licensed though the Apache v2 License is free of charge and royalties. In our project it is responsible for handling all communications between the agent and the web portal and in the measurement tests to estimate the throughput speed and web browsing KPI.

Although the standard java.net package provides basic functionality for accessing resources via HTTP, it doesn't provide the full and flexibility and configuration needed for this application. Some advantages were found when trying to accept self-signed certificates, managing multi-threaded downloads and the handling of some HTTP 300 messages that were being incorrectly processed.

jNetPcap

The jNetPcap is an open-source java library that provides a wrapper for nearly all libpcap native calls. Licensed under the LGPL license is also free of charge and royalties. It allows the capture of network traffic and performs protocol analysis. It uses a mixture of native and Java implementation allowing good packet decoding performance needed for real time utilization. This library requires

In our project, this library is used to obtain the gateway's MAC address, find the current active interface (during the initial setup procedure) and to obtain the maximum throughput in the download/upload speed KPI.

JSON (Java Script Object Notation)

JSON open standard is designed for text-based data interchange. Posing as an alternative to XML it allows the serialization and transmission of structured data over a network. In our project it is used primarily to hold the results between the measurements of the several KPI and to transmit these results from the agent to the web portal.

```
{     "Device"    : "PC",
      "OS"        : "Windows 7",
      "ISP"       : "TELEPAC-DSL-RES",
      "descr"     : "PT Telepac - Comunicacoes Interactivas",
      "country"   : "PT",
      "gwmac"     : "00:24:17:A3:FB:87"
}
```

Figure 13 – Example of a JSON object after the initial setup procedure

The source code is organized as follows:

- Default package
  - Monitor.java – Main class
  - GUI.java – User interface
  - User.java – class that holds the participant's username and password
- Config
  - Controller.java – measurement test manager/handler
  - Setup.java – parses and holds information from the configuration file
  - Sites.java – contains the list of measurement endpoints
  - Scheduler.java – implements our custom scheduler process
- Interfaces
  - Interface.java – represents a network interface with characteristics from both Java and JNetPcap.
  - InterfaceList.java – List of interfaces classes
  - ListenInterface.java – implements the JNetPcap network listener
- HttpClient
  - Client.java – implements several shared HTTP methods using Apache HTTPClient
  - SHA256.java – provides the SHA-256 hashing function
- Measurement
  - MeasurementTest.java – Main thread of the measurement test
  - DNS.java – implementation of the DNS measurement test
  - RTT.java – implementation of the RTT measurement test
  - Webbrowsing.java – implementation of the web browsing measurement test
  - MultipleDwnld.java – implementation of the download measurement test
  - MultipleUpld.java – implementation of the upload measurement test

The main flow of the application is composed of three main elements: the "Scheduler", the GUI and the "Controller". The "Scheduler" is responsible for the timing operations used in the scheduling of measurement tests. The GUI is the visible part of our measurement agent and is where participants interact with our program. The "Controller" is a handler and incoming measurement tests requests from both the "Scheduler" and the GUI. In Figure 14 we present the high level design of the software agent.

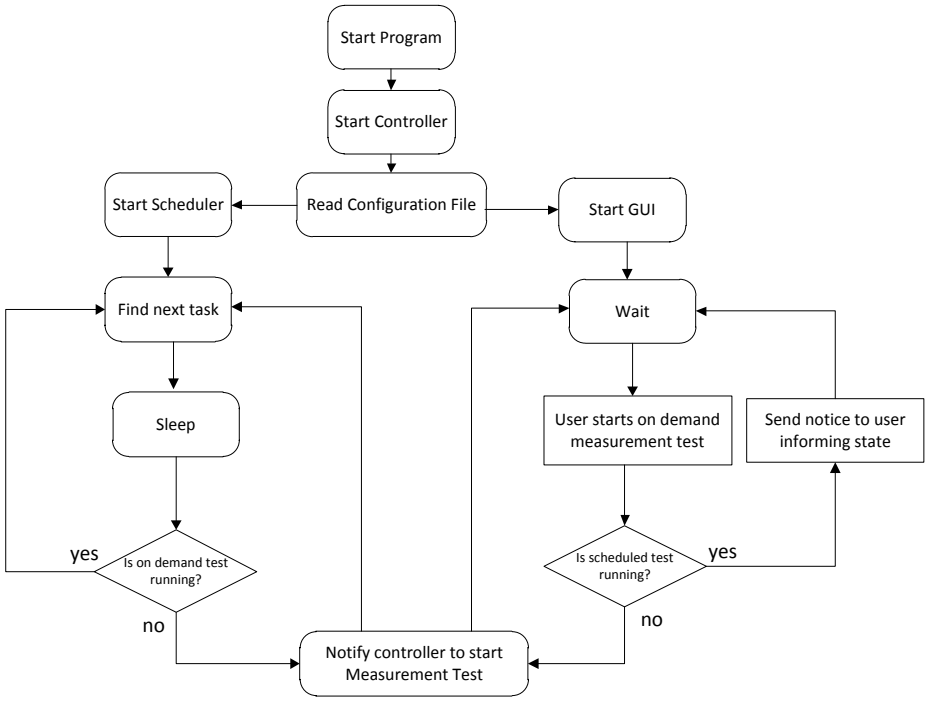

Figure 14 – Main application high level design

The design was developed to successfully handle concurrent test requests from the scheduler and the GUI, or if, due to a human error, two overlapping tests were placed in the Configuration File, generating two simultaneous scheduled tests.

As presented, the program starts by creating the "Controller" element that will handle incoming request for measurement tests. Next we parse the configuration file and extract all contained information into our application. The next step will be to create two threads. One will run the scheduler and the other that will start the GUI. Both of them have similar diagrams. The flows consist in waiting for an event (either by user input or timed activity) and query the "Controller". If there is another concurrent process running a measurement test then we either inform the user or skip to the next scheduled test. If false, we notify the "Controller" to start a new measurement test and wait for the next event.

4.1.1 Configuration File

As previously stated we have devised a configuration file that is included in the measurement agent. This file contains information about the address of the "Main Site", a list of available endpoints, and the implementation of the scheduler presented in chapter 3.2.3. Each participant has their own personal directory inside the web portal where his configuration file will reside. This allows us to set a different configuration file for each individual participant.

We have chosen to implement the configuration file as a XML document, and with the aid of the Java DOM parser we extract all necessary information.

The first element refers to the version of the configuration file. It will be sent to the "Main Site" during the configuration file update process. If a different version is found it will download and update the newest version. This process runs every time a new measurement process starts.

The next elements contain information about the IP address and port of the main site and available endpoints. For the endpoints an extra property is set, indicating their physical location.

The next element will implement our custom scheduler. Firstly we define several different test frequencies. Then assign a frequency for each week in a month, differentiating between weekdays and weekends. Finally we define some extraordinary days where we want to override the defined frequency for that week.

Following we can find the list of the 100 most visited sites to be used when evaluating the DNS and web page loading time measurement tests.

The last element will be the size of the file that will be downloaded in the throughput test.

The configuration file is the same for both desktop and mobile versions and an example is presented in Annex 2.

4.2 User location

To obtain the user's location when using the mobile version we will try to obtain the cached coordinates that are constantly kept by the Android's operating system. If these coordinates are older than 10 minutes, we will state them as outdated and start a new location listener for both GPS and network estimates. For the purpose of our system, we do not need an accurate estimate, so any source will suffice to record the user's location.

After this step, the measurement test starts, evaluating the referred KPI. After the measurement process has finished, we will check if any location updates have been made. If an updated location estimate has been considered valid, we will map the GPS coordinates to the Portuguese postal code of the area and include both GPS and postal code information in the uploaded results.

For this mapping process we used Google's GeoLocator web service. This service reads the GPS coordinates (WGS84 format in decimal degrees) and returns a list of addresses that are in the vicinity of that location. For this list of addresses we parse the text and try to extract the postal code of one of those addresses. If no postal code is found only the GPS coordinates are sent. This location will be used in the administrator's report to restrict the analysis to a particular region based on the area's postal code. If only GPS coordinates are sent it will not be possible to use this information in the current implementation.

4.2.1 Measurement process

The measurement test runs in a separated thread from our main program. It has a streamlined design as all tests run sequentially. If any of them fails the process is considered faulty and will not be uploaded. This way we assure that no inconsistent data is entered into the database.

4.2.1.1 Initial Setup

In terms of high level design both desktop and mobile versions share the same basic characteristics. The initial step is to check if the network connection is active. Next we perform an initial setup procedure, where we try to identify the connection. In the desktop version it will be to find the public IP address, the ISP identification and the gateway's MAC address. In the mobile version will be to get the public IP address, ISP identification, user's location and the IMEI+IMSI values. Following the setup procedure we will contact the "Main Site" in order to check any Configuration File updates. If available it will update the "Configuration File" and obtain the new information. Only then can the measurement test start. We run all tests sequentially and in case of any of them fails, the whole test is cancelled and considered invalid.

4.2.1.2 KPI estimation

The process starts in the client's measurement agent as a scheduled or on-demand process that contacts the Main Site to obtain any available configuration file updates. These updates should contain information such as new remote endpoints, changes in the default scheduled time of the tests and list of websites to be used in the measurement process.

From the available list of measurement endpoints, one will be selected to perform a national or international measurement. We have chosen this selection to be random because, if the project scales to a large panel of participants, some congestion might be encountered.

After this selection, we will perform an initial setup process where we identify which interface is being used, what is our public IP address and a unique identifier for that connection. In case of the desktop version, it will be the MAC address of the gateway and for the mobile version, a combination of the IMEI and IMSI of the smartphone. From the IP address we perform a query to the RIPE's database in order to obtain some information about the ISP. This collected information will permit the users to distinguish between their network access connections in the participant's report. In the mobile version there is an extra step where we will start a location listener so that an estimate of the GPS location coordinates can be obtained. This process will be further explained in chapter 4.2

Only now can the measurement process start, evaluating the selected KPI. After successfully conclusion we will send the result data to the main site. In the mobile version before sending the information we will check if the location listener obtained relevant information and add to our result set. If not the result set will be uploaded without geographical information. After this process the measurement agent will sleep until the next scheduled or on demand test.

4.2.2 Measurement test

After the initial setup has completed successfully we can then step into the measurement tests.

RTT

For this test we will start an UDP datagram socket with a timeout of 1 second. Then we create a UDP packet with the destination IP address and port retrieved from the previously described endpoint selection method. To obtain the RTT we set a timestamp and send a 32 byte UDP packet. After we receive the endpoint's answer we measure the difference between the current and the previous timestamp. If the socket connection throws a timeout exception we consider the packet lost. Each result will be saved into a vector and from that we extract the average, maximum, minimum, jitter and failed values. If all packets are lost we consider the test as failed and quit the measurement process.

DNS

For the DNS measurement test we first perform a DNS flush command in order to prevent any local cached queries. This command is system dependent and there is no class in the Java framework to accomplish this. However there is a class that accesses the OS command line and allows the Java application to issue system commands. Our approach is to issue several system commands from the 3 main OS supported by this project (Windows, Linux and MacOS).

For the Windows and MacOS there are well known system commands to perform this action:

Windows:	`"ipconfig /flushdns"`
MacOSX Leopard:	`"dscacheutil -flushcache"`
MacOS previous versions**:**	`"lookupd –flushcache"`

In the Linux (UNIX) OS this operation is more complex as compared to previous referred OS. Main Linux distribution can use one or several name service caching daemons. However main distributions for end-users often use one of the following: NSCD, BIND or DNSMASQ. As such we have simply performed a command for each one of them. As expected some will fail as the service will not be installed. The issued system commands are:

NSCD:	`"nscd restart"`

DNSMASQ: `"dnsmasq restart"`
BIND Server: `"named restart"`

Then we will copy the list of the 100 most visited websites (extracted from the Alexa website [47]) contained in the "Configuration File". From this list we will randomly select one and save the current timestamp. Next we will perform a DNS query using the Java.net getByName function that will return the website's IP address. After the query we remove the site from the list and save the time gap between the beginning of the function. If the function throws an UnknownHostException we considered as a failed query. If all queries have failed we consider this test as failed and quit the measurement process. This process will be repeated 30 times.

Webpage loading time

This test has the same principle as the previous DNS test. We start to perform a DNS flush to delete all saved queries. Then we get the same 100 website list and create an instance of the Apache HTTPClient. From that instance we will perform the designated number of cycles and extract the average, maximum, minimum and jitter. We do not consider the failed attempts as the reason might not be network related and therefore should not be accounted.

Throughput

The throughput test starts by reading from the "Configuration File" the designated file size to be downloaded. Then we start a multi thread HTTP connection using 3 threads. Each thread saves the timestamp and starts to download the same file. The files are saved into a memory buffer and never written into the hard disk.

Immediately after the connections are created we start a network listener using the JNetPcap library that will add the amount of traffic going through the interface (filtering the endpoint's source IP address). Every 2 seconds we will save the accumulated value and reset the counter. The highest value from those samples will be considered the maximum download speed. The average download speed will be estimated as the ratio between the total file size (from the 3 threads) and the overall time taken by the download procedure.

In presence of a PPP connection or in the Android version, we are not able to perform a network capture and we only estimate the average speed from a single thread.

The equivalent procedure is performed by the upload measurement test, using the files that were saved into the hard disk.

4.2.3 User Interface

4.2.3.1 Desktop version

The first contact with the program starts with a simple installation process that was developed using the open source software IzPack. This is an open source project that only requires a Java virtual machine to run. It is fully cross-platform and generates a single installer. As such, it is an alternative to native solutions such as platform-specific installers and package managers.

The Izpack project is distributed under the Apache 2.0 License, whose copyrights are included in the next installation step.

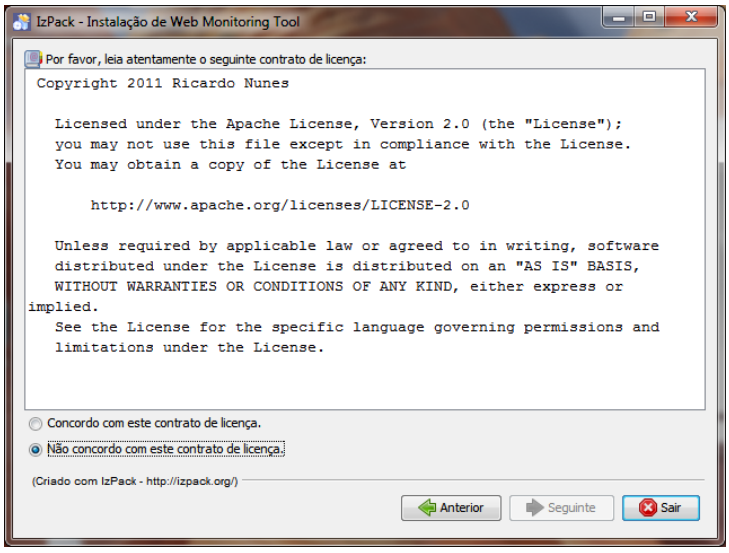

Figure 15 – Application's license agreement

After selecting the destination directory the installer unpacks the program, required libraries and the configuration file to the referred directory. No Desktop or Start Menu shortcut will be created.

For the Desktop version the user interface was designed and implemented to have minimal visual impact on the user and is composed of a single icon in the system tray.

If the user hovers with the mouse, a tooltip is displayed informing that the options are made available through a right mouse click. Identically if the user double clicks the icon a pop up message appears displaying the same information.

From the right click action 4 options are made available to the user:

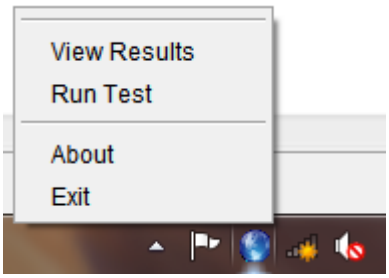

Figure 16 – Application's main options

View Results
This option opens the default browser set by the operating system and redirects the URL to the project's main page defined in the configuration file.

Run Test
This will start an on-demand measurement test. If another scheduled or on demand measurement test is currently running, a pop up message appears informing the user that a measurement test is already being performed.

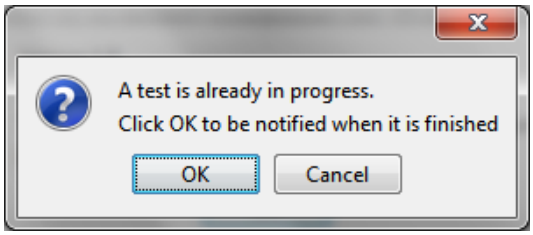

Figure 17 – Notification running measurement test

In this pop up an option is given to the user. If the user enters OK, the user can resume other functions and as soon as the process terminates a new pop up message will appear. On the other hand if the user does not wish to have a new pop up message he may Cancel the pop up and try again later.

About
This is a simple message containing the credits, identification of the author and scope of the program.

Exit
The exit button will terminate the User interface and also the scheduler. No process is left running after the user exits the program. This was an intentional development choice as users tend to not approve having processes running after an intentional exit command.

4.2.3.2 Mobile version

For the mobile version we have developed a login screen that is not present in the Desktop version. As previously stated this is because, mobile applications usually have their own distribution channel (Android Market, App Store, etc...) and it is not possible to perform real time compilation of the software.

Figure 18 – Measurement agent login page

In the login page the user may perform a login using the same credentials entered when registering with our web portal or may perform a new registration. If so the software will open a new web browser and redirects the user to our main project web page.

If the user chooses to perform login but only enters one or none of the available fields the system generates an error message without trying to validate the credentials against the Main Site. If both user and password are entered the system tries to validate the user. If the credentials are incorrect it generates an error message

Upon correct validation the user is redirected to the program's main page and the credentials are stored and never asked again as long as they remain valid.

Figure 19 – Measurement agent main page

From here the user is presented with 3 choices:

Run Test
This option starts a new on demand measurement test that will run in the background and after the process is over it displays a pop up message informing the user that the test is done.

See Results
This opens a new browser and redirects the user to our main login page where the user

Exit
The exit option hides all open windows and redirects the user to the system's home page. Unlike the desktop version it does not stop all scheduling processes. This is done because of the Android's OS operating philosophy where all applications should never be terminated, just sent to background processes.

4.3 Measurement Endpoint

The measurement endpoint consists in a web server and a Java application that will respond to the software agent measurement tests. It provides the following functions:

- Return the public IP address used by the agent
- Container for binary files, used for download KPI test
- Function to accept uploaded files, used for upload KPI test
- UDP server to act as endpoint in the RTT KPI test

In terms of system deployment we implemented the measurement endpoint in the same hardware as the Main Site, using the Apache Web Server virtual hosts configuration. This enables to have multiple domains using the same IP address.

This option was intended only for the prototype deployment because, as previously stated, we recommend having the endpoint and Main Site in different sites.

For the main site we built a standard LAMP (Linux + Apache + PHP + MySQL). The list of software used for the implementation is as follows:

- Linux 2.6.32-32 kernel version
- Ubuntu 11.04 LTS (Long Term Support)
- Apache 2.2 web server
- PHP 5.3.2-1
- php5-mysql (MySQL library for PHP integration)
- libapache2-mod-php5 (Apache library for PHP integration)

In terms of system architecture we have the following conceptual design:

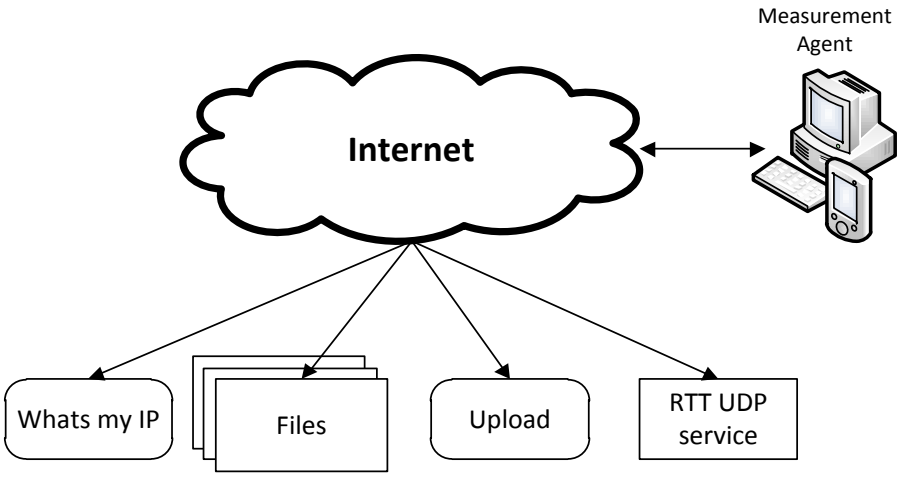

Figure 20 – Measurement endpoint conceptual design

The first interaction between the measurement agent and the endpoint is performed during the initial setup process. In order to perform a query to the RIPE database

and obtain the ISP identification, we previously have to obtain the public IP address. As such, we have developed a simple HTML page using PHP whose only function is to return the public IP address of the agent that accessed the service.

The next elements present in the endpoint are the files to be downloaded in the throughput measurement test. For that purpose we have a directory that holds several files, each with a different size. Each file contains in the filename its size (e.g. a 1MB file will be called "1.file"). In the current version of this project we have built 50 files ranging from 1 MB to 50 MB. The choice of which file to download is present in the configuration file, included in all measurement agents. Based on this parameter the measurement agent can easily redirect the URL to the designated file.

The next element "Upload" will be used in the throughput measurement test. In this test we attach a file in a HTTP POST message and upload it to the endpoint. In the endpoint side we have a HTML page built in PHP that accepts all HTTP POST messages and saves into memory its contents. When finished the memory is cleared and the connection is closed.

The last element is a Java application that listens for UDP connections in a determined port (present in the configuration file). Upon reception of a UDP packet the application simply returns the same packet to the sender, so the RTT may be estimated in the client side.

There are some missing features that could greatly enhance the performance of the measurement endpoint. In terms of security we should only allow the interaction with registered users. This could be accomplished for instance by setting the same procedure as in the "Update Configuration File" present in the "Main Site" element. The lack of this feature allows any badly intentioned internet user to perform a custom script to automatically perform repeated connections to the endpoint and thus degrading its performance or even generate some memory overflow DoS attack. Another key performance issue that could enhance the endpoint performance would be to have stored in memory the files to be downloaded. For this purpose we should limit the number of files and save them in memory (the current 50 files have a combined size of 1,3GB). This would prevent repeated hard disk accesses that could impact in the measurements.

4.4 Main Site

The Main Site is constituted of a webserver and a database to hold all information. It provides the following functions.

- Secure user registration/login
- Allow participants to download the measurement agent
- Allow participants to manage their connections (registration and delete operations)
- Receive measurement results from client's software agents
- Provide updates for configuration file
- Participant's report
- Administrator's report

As previously stated, the Main Site shares the same hardware as the Endpoint. For the implementation we used the following software:

- Apache 2.2 web server
- PHP 5.3.2-1
- php5-mysql (MySQL library for PHP integration)
- libapache2-mod-php5 (Apache library for PHP integration)
- MySQL server 5.1.41-3
- MySQL client (GUI for queries – optional)
- MySQL admin (GUI for database creation – optional)
- php5-mysql (MySQL library for PHP integration)
- openssl 0.9.8k (for creating self-signed certificates)
- open flash chart2 (for graph creation)

4.5 Web portal

As previously stated the web portal provides several services. During this section we will describe our approach for implementing them. The web portal's conceptual design is presented in Figure 21.

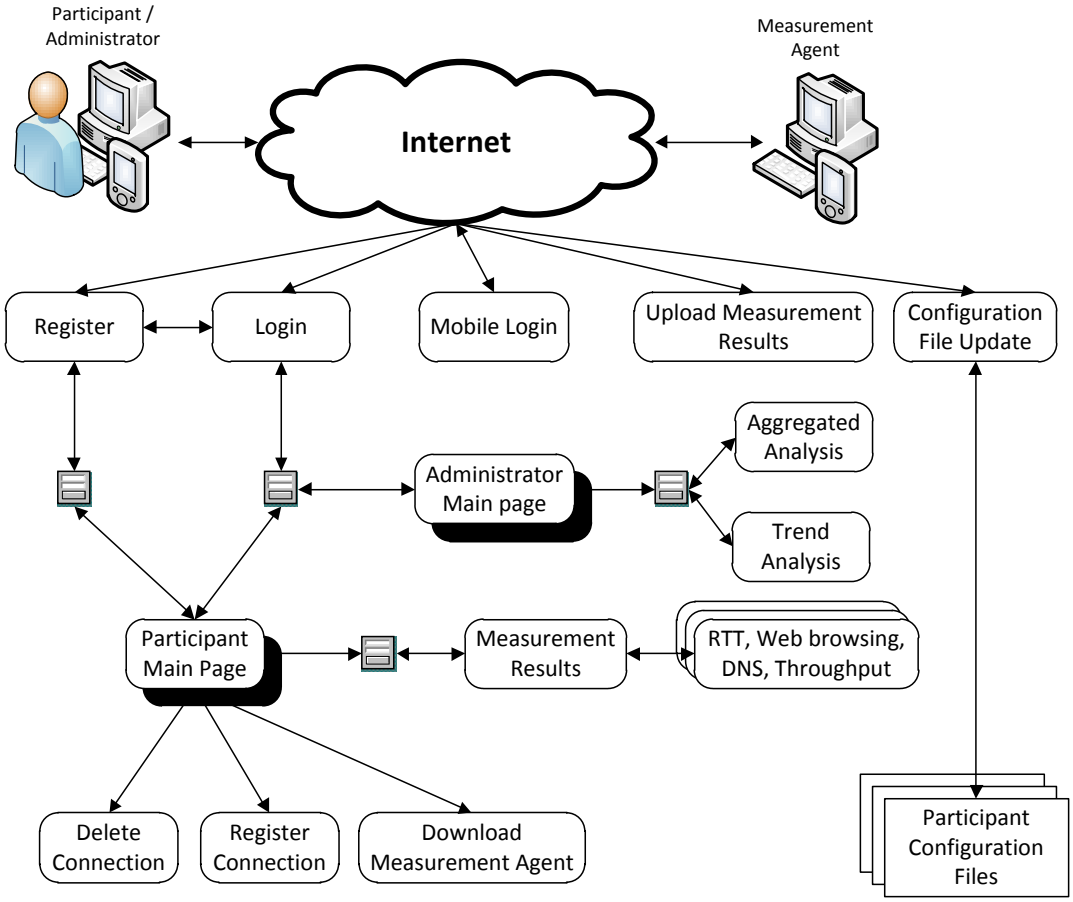

Figure 21 – Web portal's design

The main entry point is the Login/Register page. Here we present a form that will validate or create a user. The administrator login is created statically into the database and due to security reasons it is not possible to register a user with administrator's rights.

Upon login if the user is an administrator he will be redirected to the "Administrator Main Page". In here he will access the project's dashboard containing information about general aspects of the platform, such as number of users, connections, how many connections performed measurement tests in the last 3 months, how many times the measurement agent were downloaded, etc. From this page he will access the specific administrator's report described in section 4.5.4.
If the login has user level rights, he will be redirected to the "Participant's Main Page". From here he will access the list of all connections that were used to perform measurement tests. If the connection has not been registered he can access the "Register Connection" page where he can assign a ISP, plan rate, download/upload speed and take a simple survey to assess the participant's subjective evaluation of that connection. If the participant does not wish to include that particular connection, he may choose to delete it and all related data (including previous measurement tests). From this page the user may also download the measurement agent that is kept in the project's database. Finally the participant can select any registered connection and obtain the measurement results from its 4 main classes: RTT, DNS, Web loading time and Throughput.

There are three webpages that can be accessed without going through the login form. These pages are to be used by measurement agents and will perform the services of uploading the measurement results, updating the participant's configuration files and performing initial login in the mobile version of the measurement agent software. Although the login process is not necessary we still perform a validation, because all agents must include a digest in the URL using its personal hashed password. This way we can check for authenticity and assure that the message hasn't been compromised.

4.5.1 Database

Our database built in MySQL and was designed around the main relations between our 3 main elements: the event – which is generated by the measurement result; a connection - that relates the event to a user; the ISP connections – that relate the ISP plan rates to connections. In Figure 22 we present the implemented database architecture.

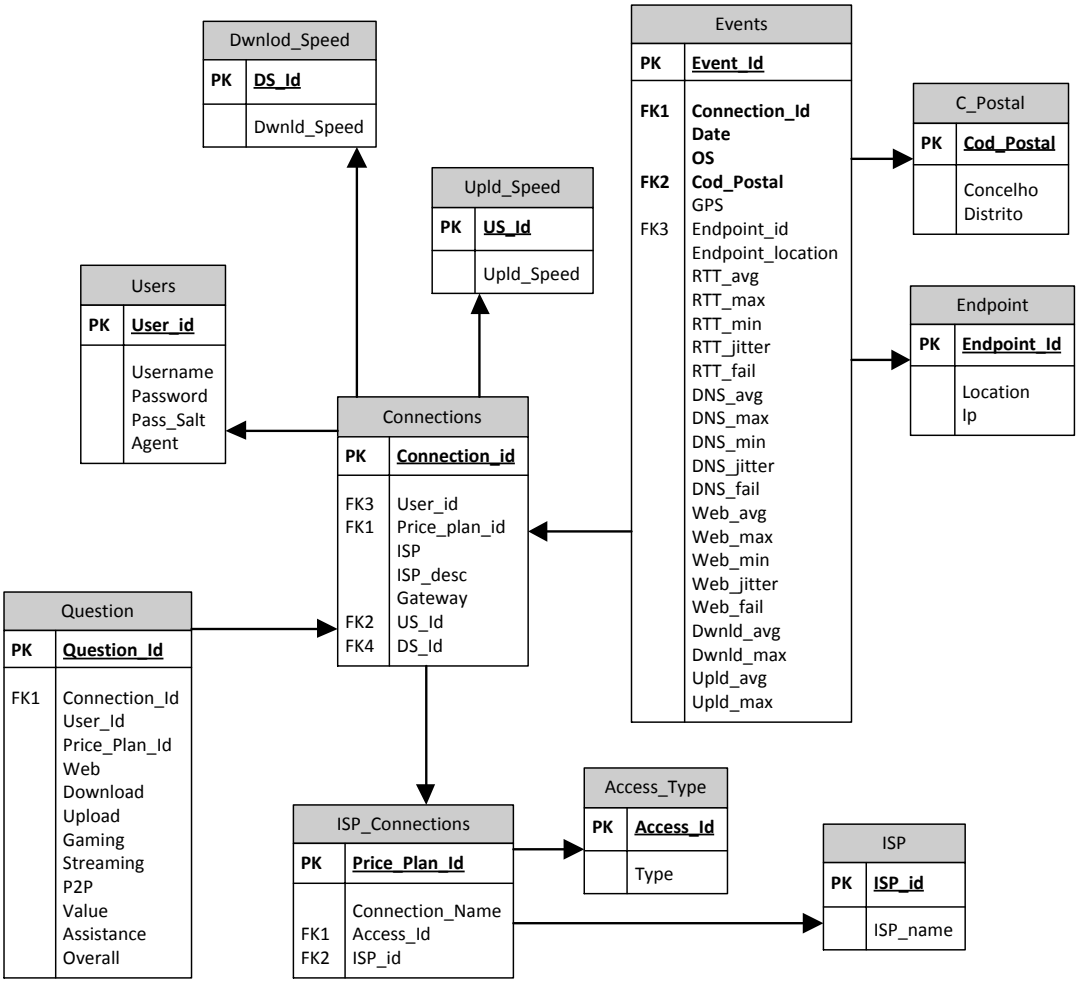

Figure 22 – System's database architecture

Our main table "Events" contains all measurement reports and geographical location both from the postal code and GPS (if available). Supporting this table we have the table "Endpoint" that holds information about Measurement Endpoint (IP address and location) and the "C_Postal". This table was built from the actual Portuguese Postal Office (CTT) codes and contains a list all existing codes and for each one has the corresponding district and county. This will be very helpful in the administrator's report, when we want to set a restriction based on a certain region. Instead of browsing through several meaningless codes we can easily select several regions by their name.

The "Connections" table holds all Internet connections that were used by participants to perform measurement tests. Before the participant registers the connection it will only have relations with the "Users" and "Events" table. Upon registration the user associates that connection to a determined Internet access plan (present in the "ISP_Connections" table), a download/upload speed and the survey that stores the rating given by the participant to that specific connection.

The "ISP_Connections" table contains a list of all ISP plan rates, which will belong to an ISP (table "ISP") and have a determined access type (ADSL, Cable, Fiber, UMTS if mobile, etc.).

4.5.2 Participant Registration/Login

This initial step is taken by the user upon entering the web portal entry page.

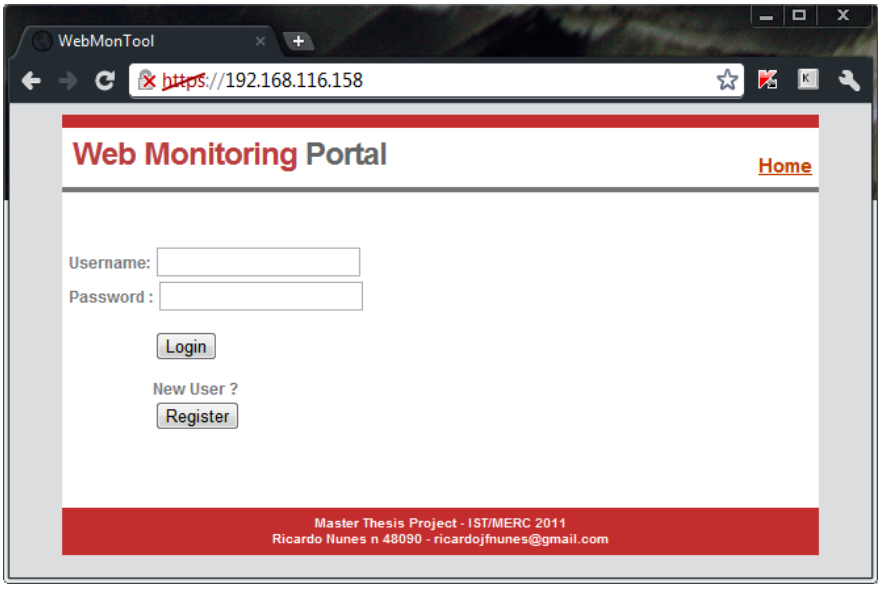

Figure 23 – Web portal's Login page

As can be seen, the browser detected and is displaying an error because we are using a self-signed certificate. Although the connection used is being encrypted, the certificate was not signed by any of the browsers built-in Root Certification Authorities. Because we are merely stating a proof of concept by building a simple prototype and this type of service is usually paid, we did not send our certificate for signing.

If this is the first contact with the web portal, the user should first enter the registration page and define a username and password.

Some usual features associated with this type of registration process were implemented. Username must be unique; passwords must contain between 3 and 10 characters, and should contain at least one lower case character, one uppercase character and one number.

Upon successful username and password validation we run a script that will create this particular user data. This procedure is constituted by the following steps:

- Get the participant's username and password from the HTML form
- Create a timestamp
- Create the hashed password using SHA-256 function:
    - SHA-256(timestamp+password)

- Create a directory with the participant's username
- Copy into the directory:
  - measurement agent jar file
  - current configuration file
  - source code of the Java class that provides username and password
- Change the source of the Java class to include the participant's username and hashed password.
- Compile this class through the standard java compiler
- Update the measurement agent jar file with the previously created java class
- Create the installation package using Izpack software script, including the following files:
  - Measurement agent jar file
  - Configuration file
- Upload to the data base the username, timestamp, hashed password and installation package
- Delete from the users directory all files except the configuration file

After this process the user is redirected to the Participant's Main Page. The next step would be to download the measurement agent software and perform measurement tests.

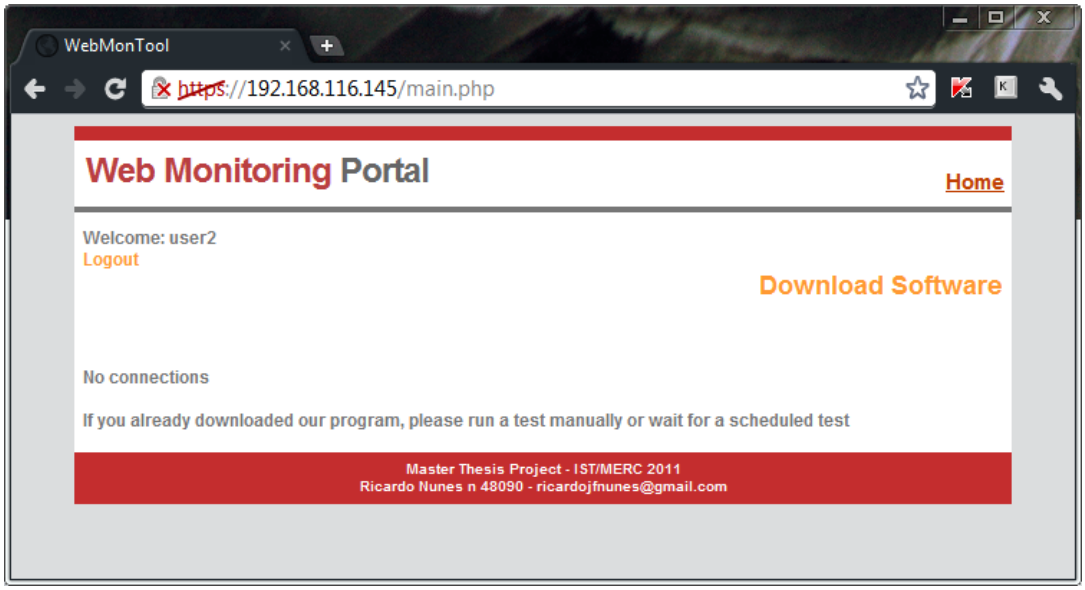

Figure 24 – Participant's main page

4.5.3 Participant's report

The participant's report is comprised of two main elements: a main dashboard where participants manage their connection and a measurement analysis page where the test results are displayed.

4.5.3.1 Dashboard

The dashboard is the entry point of the participant's report. It allows users to download the desktop version of the measurement software and manage their network access connections.

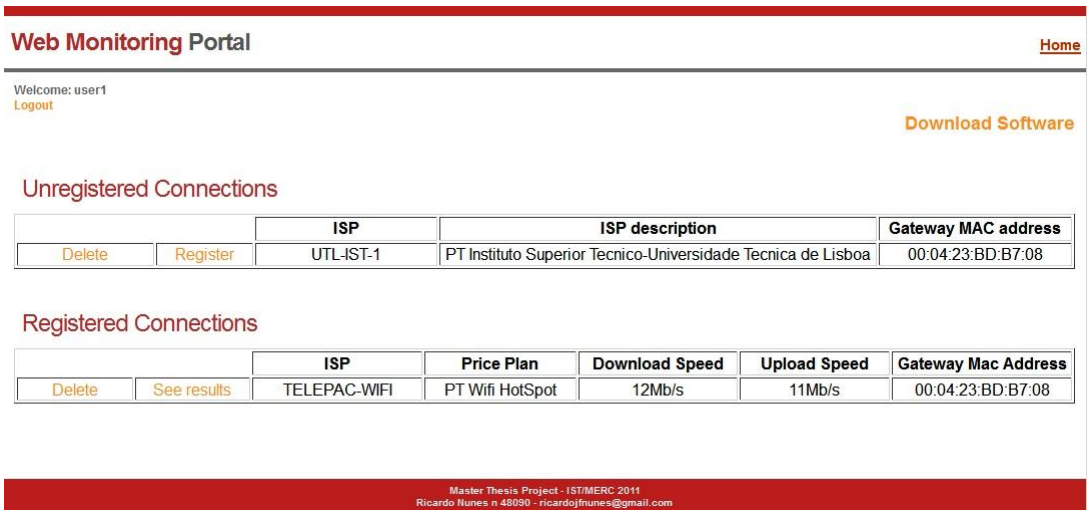

Figure 25 – Participant's dashboard

On the top of the page we have the identification of the current user, the link to perform the logout operation and the link to download the measurement agent.

Next we have the list of connections. In each connection we have the previously stated elements that are obtained during the measurement process in order to allow the user to distinguish their connections. These are the ISP name and description (according to the RIPE database) and the MAC gateway (or the IMSI+IMEI in case of the mobile connection.).

When a connection is first used it goes to the unregistered section. This section does not allow the participant to obtain any measurement results from it. It requires a registration process to confirm that the participant wishes to accept that connection as valid to be entered in our platform. The registration page can be accessed in the "Register" link, present in each connection. If the participant decides that that connection does not belong to him or that it is seldom used he may delete that connection, thereby erasing all information related to it.

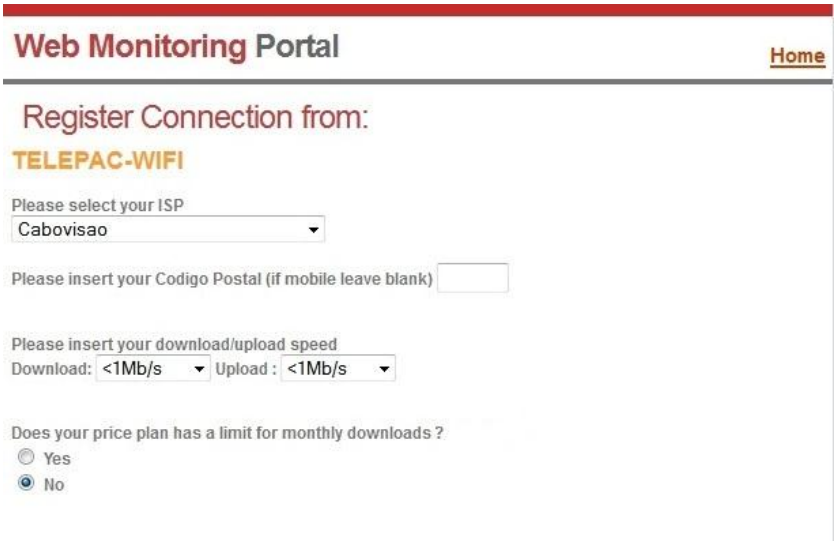

Figure 26 – Connection registration web page

The connection registration process page consists of 2 forms. The first form asks the participant to identify the ISP connection. This identification consists not only in the ISP identification, but also the type of connection (mobile, residential, etc.), and the access type (ADSL, fiber, UMTS, etc.). Besides allowing us to differentiate between several connections from the same ISP, it allows participants to easily identify the connection. To accomplish that, we have performed a market search and gathered a set of network access connections from all major ISP. This list (presented in Table 6) is kept in our database in the "ISP_Connections" table. As new connections are developed, the administrator should add them to our database and the webpage is automatically updated.

ZON Fibra	TMN Banda Larga Movel	Vodafone (Telemovel)	Sapo ADSL
ZON Cabo	TMN (Telemovel)	Meo Fibra	Sapo Banda Larga Movel
ZON Satelite	Vodafone Net Fibra	Meo ADSL	Clix Fibra
ZON Banda Larga Movel	Vodafone Banda Larga Movel	Sapo Fibra	Clix ADSL
Cabovisao	Optimus (Telemovel)	PT Wifi HotSpot	Kanguru Banda Larga Movel

Table 6 – List of actual ISP connections

Secondly, we ask the participant to enter their postal code address. This step is only necessary for residential connections as there is no method to obtain this information through our measurement agent. The next step asks the participant to enter the contracted download and upload speed. Finally, we ask if the ISP connection contains a monthly limit for downloaded traffic. This will allow the system administrator to obtain a list of participants that might be affected by the consumed traffic of our application and perhaps set a lighter test frequency using their configuration files.

There is no guarantee that the participant will enter the correct information. To validate this information we have studied two possibilities. The first would be to have another element in our form that allowed the participant to enter their client's identification number. Upon performing the registration we could develop a method to access an external web service that would contain information from all customers from all ISPs. Or ask the ISP to facilitate a web service to access their customer's database. This would require an agreement between all operators and therefore should not be a real possibility. The second method would be to ask the user to upload a digital invoice received from the ISP. Next we could perform an automatic invoice checking to assess the date, ISP and plan rate. We considered this to be out of the project's scope and did not include it in our registration process. After the registration process is completed the connection transits to the "Registered Connections" section where the link to obtain the measurement results is available.

4.5.3.2 Measurement analysis

The measurement results are presented in a web page as described in Figure 27. On the top of the page we have a list of the registered connections. Next to the list of connections we have a link that allows the participant to restrict the data from a single connection. From there we have a list of the main categories included in the measurement tests. These are DNS, RTT, Web Browsing and Throughput. After selecting one category the participant will be able to see the measurement results in a table format.

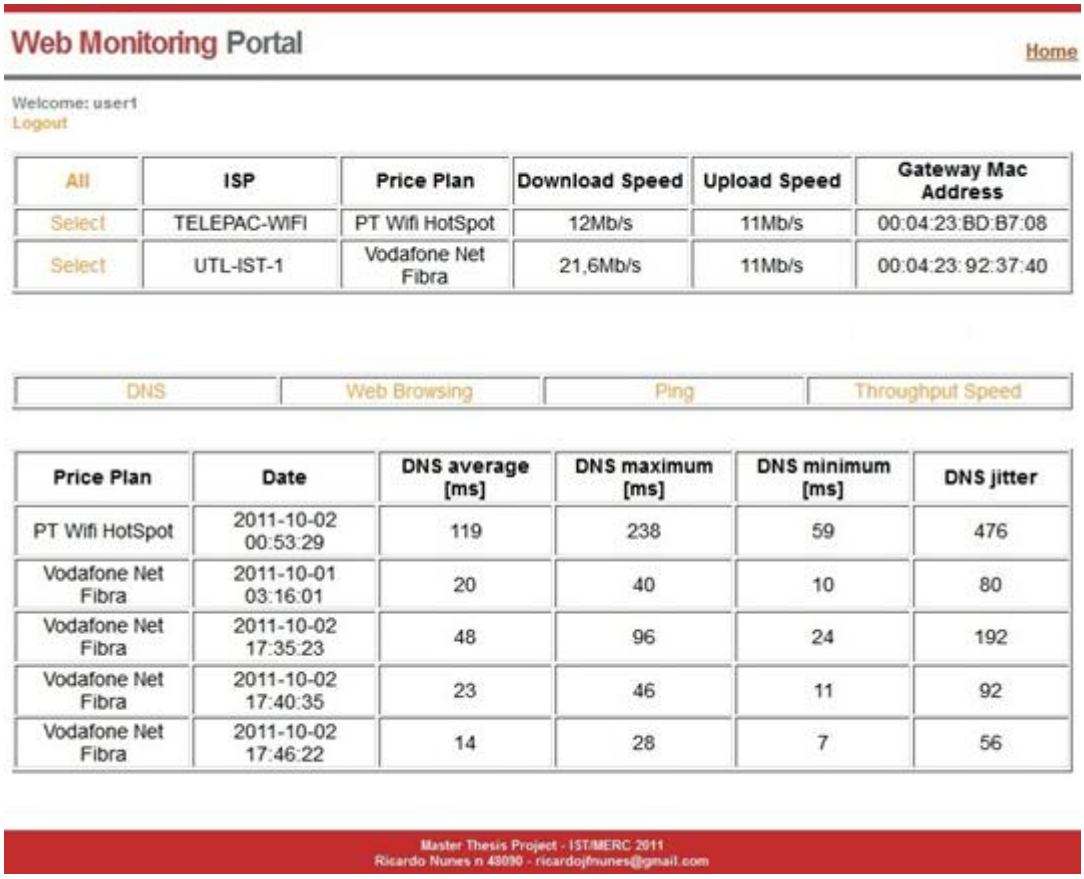

Figure 27 – Example of a participant's measurement result web page

Due to time limitation we were not able to implement a graphical display of the participant's results. No additional technology would have been employed as all the tools are already used in the administrator's report.

4.5.4 Administrator's report

This report provides the system administrator a tool to assess the overall project evolution in terms of participant adherence, and to perform analysis of subsets of measurement results based on several restrictions. The report contains three elements: the dashboard (provides the system overview), the aggregated analysis (where we will obtain the aggregated results in a determined time frame distributed by ISP) and the trend analysis (where we will assess the evolution of the measured results). In the next section we will present these reports and the benefits that they provide.

4.5.4.1 Dashboard

The "Administrators report" starts in the "Dashboard" where we can obtain the overall participant adherence to our measurement platform.

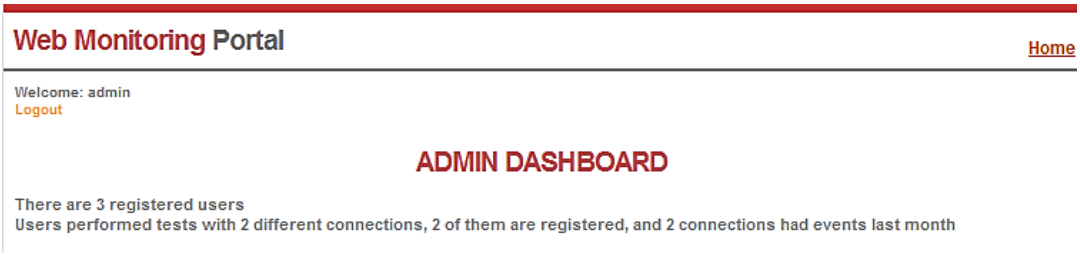

Figure 28 – Example of an administrator's dashboard

In the first section we have a small summary with the following elements:

- Number of registered users since the beginning of our system
- Number of connections
- Number of registered connections
- Number of active registered connections (with events in the last month)

Following in the dashboard we have an assessment of the number of connections distributed between ISP.

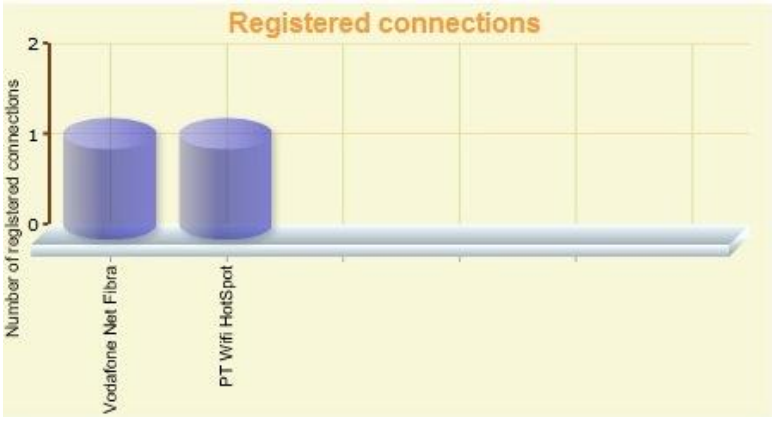

Figure 29 – Dashboard "Registered connections" graph

The first graph presented in Figure 29 allows us to determine the distribution of connections between ISP since the beginning of the project.

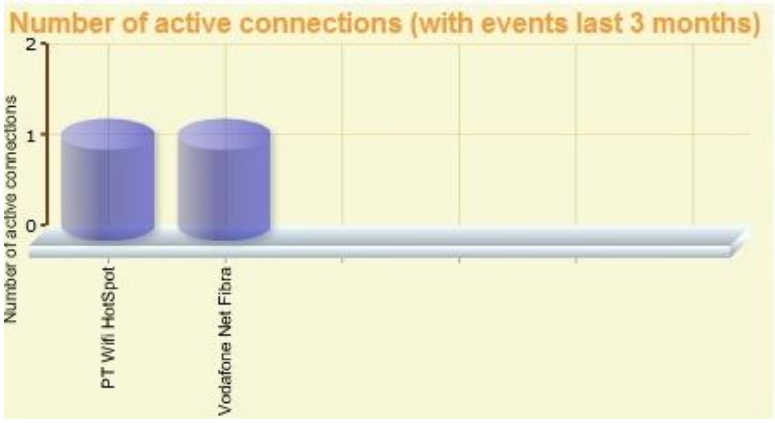

Figure 30 – Dashboard "Active connections" graph

The next graph tells us how many of the previous connections are still considered active. For this selection we have established a date of 3 months (the recommended periodicity for the publication of this report).

Figure 31 – Dashboard "Number of events in last 3 months"

The last graph allows us to determine how many measurement results were performed in each ISP connection in the last 3 months. This will allow us to determine the accuracy of the aggregated results. If we have insufficient data from a certain ISP the system administrator may want to exclude that ISP from the report.

The last elements in the dashboard are two links that direct the user to the measurement reports. We have devised two types of reports: one showing aggregated analysis distributed by several ISP and the other showing the evolution over time for a single (or combined) ISP connection.

4.5.4.2 Aggregated Analysis

The aggregate analysis will assess the measured results distributed by ISP. To setup the report we present a comprehensive form that allows the administrator to define several variables and perform restrictions in the data set.

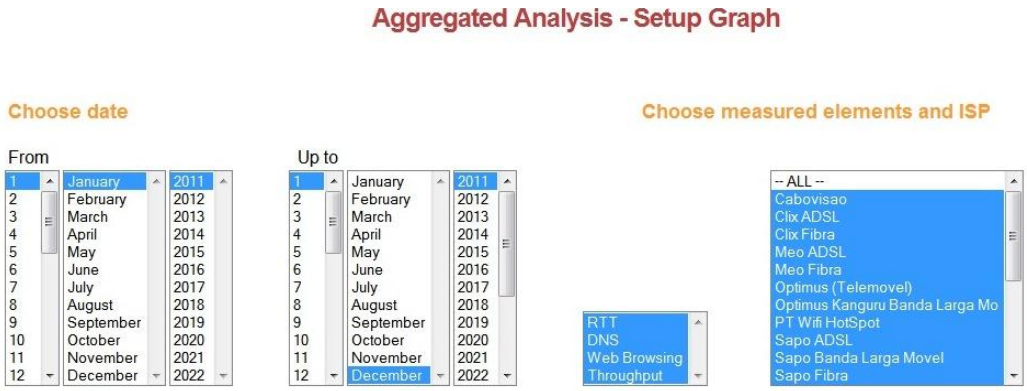

Figure 32 – Setup aggregated analysis graph - section 1

The first section starts by defining the start and end date that are to be included in the report. Following we will set which measured elements are to be analyzed and finally which ISP connections are to be featured.

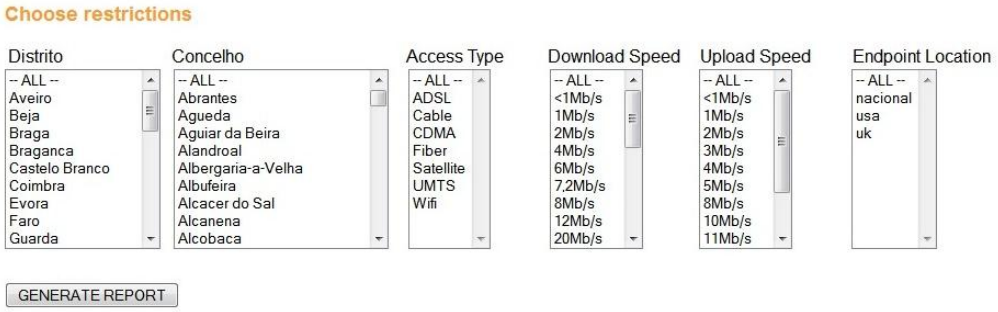

Figure 33 – Setup aggregated analysis graph – section 2

In the next step we will set a second set of restrictions. Here we will be able to constrain the result to a determined set of regions. These regions could be defined by their district or the council. As previously stated although our geographical reference will be the postal code we have devised a method to map the postal codes to real district and council names, using the official postal office database.

The next restriction we may want to impose is to only analyze a determined set of access type. This will allow us to differentiate between mobile and residential network accesses.

The last 3 restrictions are the contracted network speed and the endpoint location. The last will allow us to assess the overall performance of the international ISP core connections.

Upon setting the submit button we will obtain one graph for each one of the previously defined measurement elements. These graphs will show the averaged measurement results for each class distributed by ISP.

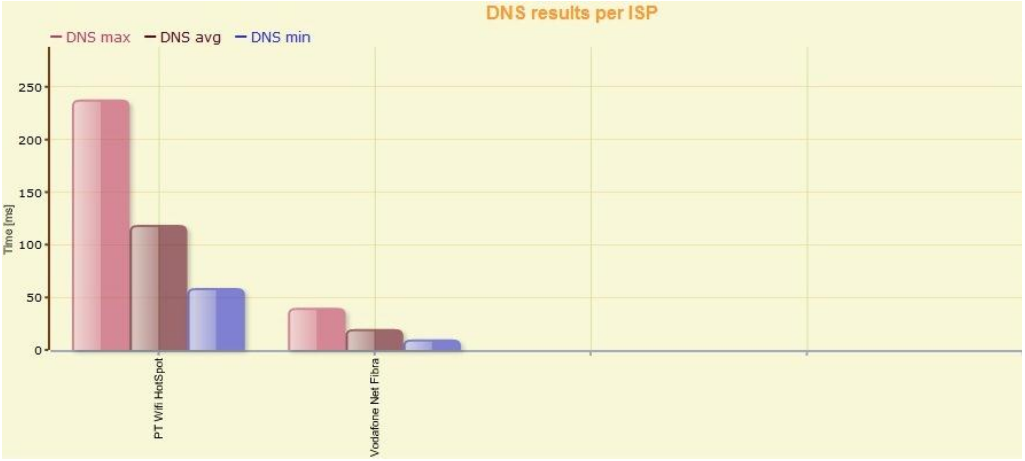

Figure 34 – Example of an aggregated analysis graph

These graphs allow us to perform an ISP comparison for a determined region or set of regions. This type of information could be presented to the user (for instance in the dashboard) has an incentive to the enrollment of new participants.

4.5.4.3 Trend Analysis

This analysis report presents a different approach from the previous aggregated analysis. Instead of comparing results from each of the selected ISP connections, we will select a group of ISP connections, consider it as a single target element and present how these measurements evolve over time.

When building a trend analysis graph we will have the same restrictions as in the previous report. We are able to restrict data based on the date, ISP connections, measured elements, regional location, network access type, throughput speed and endpoint location.

However in the first section of the setup process we have an additional element that allows us to set the granularity for the time axis.

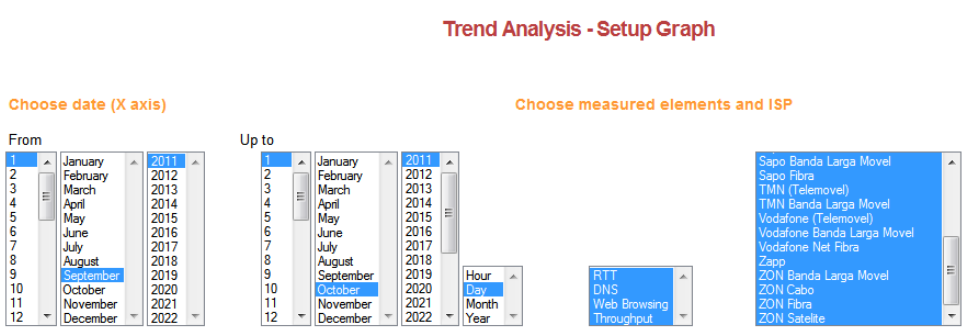

Figure 35 - Setup trend analysis graph - section 1

As can be seen in Figure 35 we are able to define the aggregation level of the time axis in hours, days, month or years. Some caution should be attended by the system administrator as setting a high granularity in a long period of time (ex. several months) may cause performance issues that could impact the normal behavior of the system.

After setting the generate report, 4 different graphs (one for each of the KPI class) will be presented.

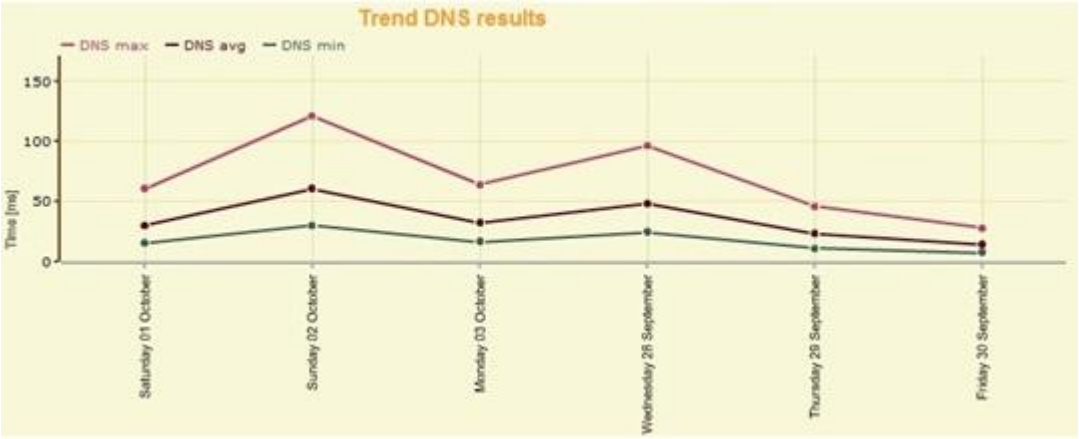

Figure 36 - Example of a trend analysis graph

As can be seen in figure 36, through this analysis we can easily estimate the evolution of the network KPI.

5. System Testing

The performed system tests are divided between three main classes: the web portal, measurement agents (desktop and mobile) and measurement test performance.

In the first series of tests we will test functionalities in the web portal user interaction, such as registration, login, connection management and report presentation (both user and administrator). The second we will test the measurement agent user interface, configuration file update, scheduler, ISP and interface identification and the measurement test upload procedure. The final series of test will focus on the validation of each one of our measurement test against commercial or third party applications.

We used two systems to perform the acceptance tests.

	System 1	**System 2**
CPU	Intel Core 2 Duo P8600 @ 2,4GHz	Intel Pentium M @ 1,5 GHz
RAM	4GB	512MB
Hard disk	320GB	80GB
Operating System	Windows 7 32 bit Professional	Linux Ubuntu 32bit 11.04

Table 7 – Hardware set description

In terms of test setup we installed the main site and endpoint in the system 2. Then with the system 1 we downloaded the measurement agent and performed the tests. To test the Linux compatibility we used a VMware image with a Linux 10.04 LTS version installed in system 1.

During this chapter we will only present the test summary results. The extended list of all test procedures and setup is presented in Annex 3 – Web portal acceptance tests, Annex 4 – Measurement agent acceptance tests and Annex 5 – Measurement process acceptance tests.

5.1 Web portal acceptance test

During these test procedures, we have successfully tested the following aspects:

- **Registration procedure**

  During the registration procedure we require the user to enter a password that should be more than 4 characters long, and must contain small case letters, upper case letter and numbers. Furthermore the registration procedure must return error when the presented username is already registered in the database. All tests where successful.

The next test we validated the registration process. We manually checked that a new tuple was entered in the database, and that it had a username, timestamp, hashed password and a MySQL blob object (measurement agent). Furthermore we checked using an external application that the hashing function was built correctly.

The last step was to check that a new directory was created in the web portal with the entered username and that the directory only contains the user's configuration file.

- **Login procedure**

Firstly we tested the login form and correct database query when entering wrong username / passwords. Using correct credentials from a simple registered user we verified that the web portal correctly redirects to the correspondent dashboard. Then with the administrator password we verified that the login procedure redirects to the administrator dashboard. Finally we have tested the logout link and verified that it successfully destroyed all session variables and cookies. After this test we tried to directly access any of the previous webpages and verified the denied access.

- **Participant's dashboard**

During this test we have checked basic dashboard functionality. We performed a measurement agent download, checked if registered and unregistered connections are successfully distinguished and tested the homepage link from all web pages.

- **Connection management**

The next step was to test the connection management. The first test was to check if the delete procedure successfully removed the connection from the database and subsequent data from the database. This was done with a registered and unregistered connection. The following test we verified the registration process. We checked the form validation (all fields except "Postal Code" must be selected) and verified that the connection has moved to the registered connections section.

- **Participant's report**

In the participant's report we tested the correct presentation of the measurement results. We verified that only registered connections are presented and cycled through all classes of measurement results (RTT, DNS, Throughput and web browsing) and verified their correct presentation. The final test that we performed focused on the filter function that allows participant's to get measurement results from a single connection.

- **Administrator's dashboard**

In the Administrator's dashboard we simply checked for the correct presentation of the graphs and all contained data queries such as number of registered users, active connections, etc.

- **Administrator's report**

For this test we manually created two connections (from the same user) but with different characteristics (ISP, District, County, download/upload speed,). Next we manually created two events for each connection form different endpoints. Next we generated multiple reports, each time imposing a new restriction (date, ISP, district, county, download/upload speed, endpoint location) and verified the presentation of the graphs and compared the included results. This was done in the trend and aggregate analysis reports.

- **General security aspects**

The first security aspect that we checked was the communication encryption. Using a standard browser and network listener/protocol analyzer (Wireshark) we followed every step of the Transport Layer Security (TLS) key exchange process.

Next we check that by entering the web portal login page using an unencrypted HTTP connection the browser is automatically redirected to the secure HTTPS page. Furthermore after the login process we checked that even if the user changes the URL to an HTTP connection, access is denied.

Another security test was to see if any of the webpages returned any results when accessed by an unregistered user (using direct URL access).

Also using a user login we tried to directly access any of the administrator's report pages and verified the denied access.

Finally we tested the web portal resistance for SQL injection attacks using all forms from all developed pages. The attacks were made using the "SQL Inject me" Firefox add-on developed by "Security Compass". This software automatically detects any form in the current webpage and allows the automation of a battery of tests using several SQL Injection strings. All reports returned no vulnerabilities.

5.2 Measurement agent acceptance test

These test focus on general aspects concerning the measurement agent GUI, configuration file, ISP identification and upload procedure.

- **Download and installation procedure**

During these tests we downloaded a version of the measurement agent from the user's dashboard. Started an installation procedure, reviewing every step of the process. After the installation we successfully verified that the installation directory contained all necessary files (Configuration file, measurement application executable, icon image, uninstall executable). This test was performed using both test equipment sets.

- **Graphical User Interface**

With the previous installation we started the measurement agent and tested several GUI elements. We started by successfully testing all the main menu options. Then with a measurement test running we tried to start another on demand test. The application successfully verified that a previous test was being performed and presented a message to the user. From this message we asked the application to inform when the test was completed. After the measurement test ended a new popup message appeared informing the user of its state update. This test was performed using both test equipment sets.

- **Configuration file**

Using the Eclipse IDE we changed the source code to include a function that prints all information contained in the class that holds the configuration file. Using the Eclipse IDE we started a new measurement agent instance and compared the printed information present in the Eclipse console with the configuration file to verify that all information was successfully read.

Locally accessing the main site we edited the configuration file from that specific user and changed the version tag and several other elements. Next we started an on demand test and verified that the application successfully detected that a new version was available. After the update procedure we checked the local configuration file and verified that it contained the recent changes.

- **Scheduler**

The first test was to check that a measurement test successfully started. For that scenario we manually changed the configuration file so that the next test would be in a few minutes and waited for it to start.

For the following scheduler tests we used the Eclipse IDE and added some source code that at startup prints the time that the program will wait until the next test. During a weekday we changed the configuration file so that the next test would be in another weekday, a weekend and a holiday. Then we repeated the process but during a weekend. Finally we changed the configuration file so that the current day was a holiday and performed the same tests.

The last test was to verify the correct estimation of the random time that the application will wait before starting the scheduled test. We changed the configuration file and ran several instances of the application and verified that the estimated time was always within the specified maximum time range.

- **Mobile version – Login**

For the mobile version tests, we used the Android's SDK which has a built in emulator. Then with the Eclipse plugin with installed our application in the emulator and performed the acceptance tests. The first test was to perform a login procedure that would not meet with the username and password requirements. Then we proceeded to perform the initial login (only included in the mobile version). The login procedure was successful and we were directed to the application main page. Using Eclipse plugin we read the Shared Preferences File and confirmed that the included credentials matched the user that performed the login.

- **Mobile version – GUI**

For this test we checked all of the available user interfaces. In the login page we checked that the "Register" button opens the browser and redirects to the project's main page. From the previous test we already concluded over the login button. In the main page we confirmed that the exit button exits the application, the "See Results" button open a new browser directed at the project's home page. Finally we tested that the "Run Test" button starts a new measurement test.

- **Mobile version – User Location**

For the user location we used a feature present in the Eclipse plugin where we are able to set a new GPS event with a determined WGS84 coordinates. For this test we proceeded to perform a measurement test and did not provide any update. We successfully verified that the uploaded result did not contain any GPS or postal code coordinates. Then during the measurement test we provided a GPS update and successfully verified that the resulting measurement test contained the correct information regarding our updated GPS coordinates and that the contained postal code belongs to the vicinity of the GPS location.

5.3 Measurement process acceptance test

The setup for this test was the main site and endpoint installed in system 2 connected to a residential connection from the Portuguese ISP Portugal Telecom, with a fiber connection, using the plan rate "Meo Fibra" and with a contracted speed of 100Mbps. In system 1 we had the measurement agent and went to several ISP clients premises and validated our measurement process using different environment settings and equipments.

- **Network identification**

In the first test we performed, we checked that by disabling all interfaces the application successfully detected that no connection was available and stopped the measurement test.

Next we ran tests using different connections. For these test we physically went to several ISP clients that and performed measurement tests. We tested the following ISP connections:

ISP	Maximum Speed	Interface
MEO Fibra	100Mbs	802.3 Ethernet
IST eduroam	Unknown	802.11 wireless
PT Wifi HotSpot	Unknown	802.11 wireless
ZON TvCabo	30 Mbs	802.3 Ethernet
VPN IST	Unknown	PPTP
TMN Banda Larga Móvel	7,2 Mbs	3G USB modem

Table 8 – List of selected ISP connection used to perform tests

In each test we manually obtained the public IP address (using the *www.whatsmyip.org* website) and recorded the gateway MAC address (by performing a network capture with Wireshark). In the PPTP and 3G USB modem we do not have a MAC address so the dialup connection name was recorded as present in the OS. Then we manually ran several queries in the RIPE's webpage, using the previously recorded public IP address. Finally we accessed the web portal and in the user's dashboard compared the presented connections with the information recorded at the test sites.

The following results were obtained:

MEO FIBRA

Public IP : 85.243.139.185
ISP : TELEPAC-DSL-RES
Description : Telepac - Comunicacoes Interactivas, SA-DSL Service Network
Gateway : 00:24:17:A3:FB:87

IST eduroam

Public IP : 193.136.166.125
ISP : UTL-8
Description : Universidade Tecnica de Lisboa
Gateway : 00:04:23:BD:B7:08

PT Wifi hotspot

Public IP : 193.136.166.125
ISP : UTL-8
Description : PT Telepac II - Comunicacoes Interactivas, SA-WIFI HotSpots
Gateway : 00:1A:6D:D3:ED:43

ZON TVCabo

Public IP : 85.138.194.33
ISP : TVCABO
Description : TVCABO-Portugal Cable Modem Network
Gateway : 00:05:CA:70:78:25

VPN IST

Public IP : 193.136.132.10
ISP : UTL-IST-1
Description : Instituto Superior Tecnico Universidade Tecnica de Lisboa
Gateway : VPN IST

TMN Banda larga movel

Public IP : 89.214.236.172
ISP : PT-TMN-200711
Description : GPRS Costumers
Gateway : TMN

All of the previously listed ISP connections were correctly identified and stored into the web portal's database. Furthermore the used public IP address and gateway identification was also correctly identified by our measurement agent.

- **RTT**

To evaluate the RTT we compared our results with the results obtained through the common ping application. For each test we estimated the average maximum, minimum and jitter of a series of 30 packets.

ISP	Average [ms]		Maximum [ms]		Minimum [ms]		Jitter	
	UDP	ping	UDP	ping	UDP	ping	UDP	ping
IST eduroam	11	11	16	14	10	10	2	2
PT Wifi HotSpot	123	129	234	250	16	23	3353	4899
ZON TvCabo	13	11	18	15	12	9	2	1
TMN Banda Larga Móvel	61	57	70	89	55	45	24	130

Table 9 – Results from RTT system test

We can conclude that both methods returned similar results, thus proving the validity of the process. The PT Wifi HotSpot connection reported the worst results. This unexpected event was due to the fact that the hotspot site was located in a business center and the publicly available account was being shared by several users.

- **DNS resolution time**

To evaluate this KPI we compared our method of performing DNS queries through the Java getbyname function with the results obtained by performing a nslookup system command. We performed 30 queries randomly chosen from our list of 100 most visited websites.

The procedure consists on the following steps:

- Select a random site
- Perform the query using our method
- Perform a DNS flush system command
- Repeat the query but with the nslookup command

This way we can evaluate both systems with the same set of websites. The results were as follows:

ISP	Average [ms]		Maximum [ms]		Minimum [ms]		Jitter	
	Java	nslookup	Java	nslookup	Java	nslookup	Java	nslookup
IST eduroam	38	106	153	470	13	72	2339	4958
PT Wifi HotSpot	257	301	1377	1024	27	20	72098	60227
ZON TvCabo	12	14	39	24	10	13	16	4
TMN Banda Larga Móvel	133	324	214	394	117	229	326	684

Table 10 – Results from the DNS resolution time tests

The presented results show that both processes have similar results and are consistent with the previous stated conclusion of the high network usage on the PT Wifi HotSpot.

- **Web browsing**

To evaluate our method of estimating this KPI we compared with the command line application CURL. This command line application is a free, client-side URL transfer library, supporting several types of protocols. As our method, the application was set to only save static HTML content. An example of one of the command lines would be:

```
curl.exe --silent -L www.google.pt
```

We performed a battery of tests using 30 of the top 100 list of the most visited websites and measured the average, maximum, minimum and jitter. The test procedure was identical to the presented for the DNS resolution time KPI. As in the previous test, between each site we performed a DNS flush to prevent local caching of the DNS query.

ISP	Average [ms]		Maximum [ms]		Minimum [ms]		Jitter	
	Java	Curl	Java	Curl	Java	Curl	Java	Curl
IST eduroam	1363	1316	4742	4099	34	88	1749687	1705274
PT Wifi HotSpot	4153	4469	16577	28283	422	503	1289579	3358569
ZON TvCabo	816	1142	2955	6395	93	103	532044	2024442
TMN Banda Larga Móvel	1518	1394	4188	4322	493	383	535135	617496

Table 11 – Results from the web browsing tests

From the results we proved that our method presents the same performance as the proven CURL application. Once more the network occupancy state of the WifiHotSpot was the reason for some unexpected values.

- **Download speed**

For this KPI we performed a comparison with the results obtained from two applications. The first is the GNU WGET command line application. This is a free software package (present in almost all Linux distributions) for retrieving files using HTTP, HTTPS and FTP protocols. During this test the WGET application was used to perform a single threaded download. An example of one of the command line would be:

```
"wget http://webmontool.dyndns.org/endpoint/files/25.file"
```

The second is a well-known commercial download manager "Download Accelerator Plus" (DAP), used by over 200M users. In our tests we set the maximum simultaneous downloads to three threads (same used by our application).

For the IST eduroam and ZON TvCabo connections we selected a 25MB file, for the remaining we selected a 5MB file size.

ISP	Average [KB]			Maximum [KB]		
	Java	Wget	DAP	Java	Wget	DAP
IST eduroam	2030	1510	1219	2069	1631	1622
PT Wifi HotSpot	103	152	441	823	382	845
ZON TvCabo	2778	2670	2560	2967	2780	2886
TMN Banda Larga Móvel	327	227	256	–	301	328

Table 12 – Results from the download speed tests

Although the results are not completely consistent we can still conclude that our application as the same performance as other well established solutions. In fact we can expect better performance from our method, because unlike the other applications, we do not save the file to the hard disk. The variations can be explained by the fact that, although the tests were performed sequentially, there are still some network usage variations. The maximum throughput from the "TMN Banda Larga Movel" connection was not performed because, as previously stated, we do not implement this method when in presence of a PPP connection.

- **Upload speed**

For this test we compared our method with the CURL application, by performing an HTTP POST with an attached file. For the IST eduroam and ZON TvCabo connections we selected a 25MB file, for the remaining we selected a 5MB file size. The following results were obtained:
For this test the following CURL command was used:

```
"CURL -d "file=@5.file" http://webmontool.dyndns.org/endpoint/
files/25.file"
```

ISP	Average [KB]		Maximum [KB]	
	Java	CURL	Java	CURL
IST eduroam	2133	1861	2323	2220
PT Wifi HotSpot	65	79	101	116
ZON TvCabo	2170	2021	2342	2153
TMN Banda Larga Móvel	93	116	-	137

Table 13 – Results from the download speed tests

The tests results prove that our application has the same performance as the CURL application. Although not completely proven we believe that having 3simultaneous upload threads will benefit more accurate results for the maximum throughput KPI.

6. Future Work

In the technical aspects of the current application, we would like not to have the ISP identification procedure based on the parsing of the RIPE's webpage and to implement a fully functional *whois* client directly in the source code. This would allow the benefit of not relying on current presentation format of the webpage.

In terms of security we would like to have available to the user the possibility of dynamically changing the password. In the current implementation these credentials are set in the registration process and to be changed would require the manual intervention of the system administrator. Also we would like to have the application to be run through a code obfuscation program, performed in real time, during the registration procedure. Although not preventing completely, it would difficult any reverse engineering attack. In the Android version of the "Measurement Agent" we would like to have the extra security provided by the encryption of the username and hashed password present in the Shared Preferences file. Finally we should add some user validation before allowing any of the functions provided by the Endpoint

From the QoS measurement perspective we have missed to implement a report to analyze the subjective QoS evaluation provided by users during the connection registration. It would be interesting to aggregate these answers in the same manner as the measurement results and generate a "per region" analysis report comparing side-by-side the user's perspective with the measured results. To implement this report there should be no fundamental change to the platform as all elements are currently kept in the database.

Another valued aspect would be to have a classification of the user's network bandwidth usage before the measurement test. It would enable us to asses if during the test another application was sharing the network access. This element would be included in the measurement results and could enable a more selective filtering in the administrator's reports. The main challenge would be to develop a global classification system that could encompass the several heterogeneous results returned by the several network access types.

Also we would like to have implemented a multithreaded throughput test to be used when in presence of a PPP connection (used by the 3G USB modems). We believe that currently it is not an issue, but with the imminent evolution to the 4^{th} generation of the cellular wireless standard – Long Term Evolution (LTE), and respective bandwidth increase, it could reveal significant differences when compared with our current method.

Other interesting element would be to try to assess if the ISP is performing any form of traffic shaping. One possible solution would be to implement in the measurement endpoint a Glasnost [44] test server. Also it would be interesting to emulate the behavior of a VoIP call or streaming video. In a more distant perspective it would be an interesting feature to implement a more interactive report, using both the postal codes and the GPS coordinates on a map API (for instance Google Maps). Our vision is to have a multi-tiered analysis, where depending on the zoom level, we could have several different region aggregations up to the street level.

7. Conclusion

Measuring QoS is a complex and relatively new subject that is the target of several recent studies. Portuguese communications regulator ANACOM was one of the pioneers in QoS and network access benchmarking by providing an annual report since 2005. The experience in this subject has translated in a methodology and test environments that are the most accurate and statistically precise of all similar international projects. However, in practice, this approach often leads to the exclusion of scarcely populated areas where network investments are frequently overlooked. Our approach was to build a comprehensive Internet access evaluation platform, based on an open system where any consumer may participate and, using their personal equipment, obtain an estimate of the QoS of their Internet connection. Furthermore, by gathering geographical information we are able to generate multiple region specific reports.

We developed a fully functional prototype with two versions of the measurement agent: one designed for desktop/laptop equipment and the other to use with Android smartphone. No external applications were used to estimate the KPI and through the system tests we have proven that they are in the same performance class as other commercial or well-known applications. The web portal provides some interesting elements, such as the real-time program compilation and allowing the participants to dynamically manage their connections. The combination of the trend and aggregate analysis provide an interesting method to assess the current state and evolution of the Internet access.

Several innovative features have been implemented, such as: the custom scheduler, that enables a flexible implementation of the test frequencies; the configuration file that, in combination with the directory structure in the web portal, enables a centralized architecture simplifying the deployment of new endpoints and enabling a "per user" granularity; the connection identification, combining both the ISP and the gateway elements. Also the combination of the Google's GeoLocator web service with the regional postal code enables us to easily perform a location aware agent and add to the geographical reports data from mobile agents.

All of the project's goals have been approached and we considered all of them as successfully implemented. Although the number of implemented features, there are still several possibilities to build from. The proposed system could enable the development of new interesting features and still maintain the same proposed architecture.

References

1. K. Maxwell, “Asymmetric Digital Subscriber Line: interim technology for the next forty years” IEEE Communications Magazine vol. 34 issue 10, October 1996.
2. http://www.anacom.pt/
3. http://www.planet-lab.org/
4. ITU-T Recommendation G.1000, “Communications quality of service: A framework and definitions”, November 2001.
5. ITU-T Recommendation E.802, “Framework and methodologies for the determination and application of QoS parameters”, February 2007.
6. ITU-T Recommendation E.800, “Terms and definitions related to quality of service and network performance including dependability”, August 1994.
7. A. Bouch, A. Kuchinsky and N. Bhatti, "Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service", in Proc. of the Special Interest Group Computer-Human Interaction conference on Human factors in computing systems, Computer Human Interaction (CHI) Letters, Vol. 2, Issue 1, April 2000.
8. T. Sutinen, T. Ojala, “Case Study in Assessing Subjective QoS of a Mobile Multimedia Web Service in a Real Multi-access Network”, in Proc. Thirteen International Workshop on Quality of Service, Passau, Germany, June 2005.
9. ICP-ANACOM, “Estudo de aferição da qualidade do serviço de acesso à internet Banda Larga”, March 2009.
10. ICP-ANACOM, “Estudo de aferição do serviço de acesso à internet Banda Larga, Relatório Metodológico”, July 2010.
11. http://www.ixchariot.com/products/datasheets/ixchariot.html
12. http://www.ixiacom.com/
13. http://www.fccn.pt/
14. http://www.ascom.com/en
15. http://www.ascom.com/en/tems-automatic-3
16. http://www.tenet.res.in/
17. http://lirneasia.net/
18. W. Chanuka, N. Kapugama, “Prospects of Volunteer Computing model in performance data gathering for Broadband Policy Formulation: A Case study from South Asia”, presented at the Experts Workshop, Washington DC, September 2009.
19. LIRNEasia, “Broadband Quality Test Plan”, version 1.2, December 2007
20. T. Gonsalves, “Broadband Quality of Service Experience Test Results”, November 2009
21. T. Gonsalves, A. Bharadwaj, “Comparison of AT-Tester with Other Popular Testers for Quality of Service Experience (QoSE) of an Internet Connection”, TeNeT Group, Dept. of Computer Science & Eng., IT-Madras August 2009
22. http://www.broadband.gov/qualitytest/
23. http://www.ookla.com/
24. http://www.measurementlab.net/
25. http://www.speedtest.net/
26. http://www.pingtest.net/
27. http://www.internet2.edu/performance/ndt/
28. http://www.internet2.edu/
29. Federal Communications Commission, “Connecting America: The National Broadband Plan”, March 2010.
30. Federal Communications Commission, “Broadband Assessment Model”, March 2010.
31. Federal Communications Commission, “Broadband Performance”, Omnibus Broadband Initiative Technical Paper nº 4, August 2010.
32. http://www.epitiro.com/
33. http://www.between.it/
34. http://www.osservatoriobandalarga.it/
35. http://www.isposure.com/index.htm
36. Osservatorio della Banda Larga, “Italian Broadband Quality Index: Rapporto Preliminare”, October 2009.
37. Epitiro, IDC, “Report for the Commerce Commission on New Zeland Broadband Quality”, December 2008,
38. http://speedmeter.fccn.pt/
39. S. Jha, M. Hassan, “Engineering Internet QoS”, 1st edition, Artech House Publishers, August 2002

40. http://blog.ioshints.info/2007/04/why-is-firstping-lost.html.
41. http://www.gnu.org/software/wget/
42. Y. Jiang, C. Tham, C. Ko, "Challenges and approaches in providing QoS monitoring", International Journal of Network Management vol. 10 number 2, Wiley, April 2000.
43. A. Tannenbaum, "Computer Networks", 4th edition, Prentice-Hall, August 2002.
44. M. Dischinger, M. Marcon, S. Guha, K. Gummadi, R. Mahajan, S. Saroiu, "Glasnost: Enabling End Users to Detect Traffic Differentiation", in Proc. of the 7th USENIX Symposium on Networked Systems Design and Implementation, April 2010.
45. C. Shiflett, "Essential PHP Security", 1st edition, O'Reilly, October 2005.
46. R. Rogers, J. Lombardo, Z. Mednieks, G. Meike, "Android Application Development: Programming with the Google SDK", 1st edition, O'Reilly, May 2009.
47. http://www.alexa.com/topsites/countries/PT
48. M. Marchese, "QoS Over Heterogeneous Networks", 1st edition, Wiley, June 2007.

Annex 1 – Subjective ISP evaluation

ISP user evaluation: How do you rate your ISP in terms of:

Web browsing
a) Don't use/ Don't know
b) Bad
c) Poor
d) Fair
e) Good
f) Excellent

Download speed
a) Don't use/ Don't know
b) Bad
c) Poor
d) Fair
e) Good
f) Excellent

Upload speed
a) Don't use/ Don't know
b) Bad
c) Poor
d) Fair
e) Good
f) Excellent

Online gaming
a) Don't use/ Don't know
b) Bad
c) Poor
d) Fair
e) Good
f) Excellent

Audio (VoIP) / Video streaming
a) Don't use/ Don't know
b) Bad
c) Poor
d) Fair
e) Good
f) Excellent

P2P
a) Don't use/ Don't know
b) Bad
c) Poor
d) Fair
e) Good
f) Excellent

Price/performance
a) Bad
b) Poor
c) Fair
d) Good
e) Excellent

Technical assistance
a) Bad
b) Poor
c) Fair
d) Good
e) Excellent

Overall
a) Bad
b) Poor
c) Fair
d) Good
e) Excellent

Annex 2 – Configuration File example

```
<?xml version="1.0"?>
<config>
	<version>1</version>
	<mmsite ID="0">
		<ip>ricrit.dyndns.info</ip>
		<port>2222</port>
		<location>http://www.netmon2l.com</location>
	</mmsite>
	<rmsite ID="0">
		<ip>ricrit.dyndns.info</ip>
		<port>5001</port>
		<location>pt</location>
	</rmsite>
	<scheduler>
		<repetition ID="0">
			<hour>null</hour>
		</repetition>
		<repetition ID="1">
			<hour DURATION="10">09:30</hour>
			<hour DURATION="10">11:00</hour>
			<hour DURATION="10">13:25</hour>
			<hour DURATION="10">15:00</hour>
			<hour DURATION="30">17:00</hour>
			<hour DURATION="30">19:45</hour>
			<hour DURATION="30">21:00</hour>
			<hour DURATION=”60">22:00</hour>
		</repetition>
		<repetition ID="2">
			<hour DURATION="120">04:00</hour>
		</repetition>
		<weekday>
			<week ID="1">1</week>
			<week ID="2">1</week>
			<week ID="3">1</week>
			<week ID="4">1</week>
			<week ID="5">1</week>
		</weekday>
		<weekend>
			<week ID="1">2</week>
			<week ID="2">2</week>
			<week ID="3">1</week>
			<week ID="4">1</week>
			<week ID="5">1</week>
		</weekend>
		<holydays>
			<day DATE="2/4">2</day>
			<day DATE="4/4">2</day>
			<day DATE="1/5">2</day>
			<day DATE="10/6">2</day>
			<day DATE="15/8">2</day>
			<day DATE="5/10">2</day>
			<day DATE="1/11">2</day>
			<day DATE="1/12">2</day>
			<day DATE="8/12">2</day>
			<day DATE="25/12">2</day>
		</holydays>
	</scheduler>
	<top100>
		<site ID="1">google.pt</site>
		<site ID="2">facebook.com</site>
		<site ID="3">google.com</site>
		<site ID="4">youtube.com</site>
		<site ID="5">sapo.pt</site>
		<site ID="6">pokerstrategy.com</site>
		<site ID="7">live.com</site>
		<site ID="8">wikipedia.org</site>
		<site ID="9">xl.pt</site>
		<site ID="10">yahoo.com</site>
		<site ID="11">abola.pt</site>
		<site ID="12">iol.pt</site>
		<site ID="13">msn.com</site>
		<site ID="14">twitter.com</site>
		<site ID="15">wordpress.com</site>
		<site ID="16">olx.pt</site>
		<site ID="17">linkedin.com</site>
		<site ID="18">publico.pt</site>
		<site ID="19">imdb.com</site>
		<site ID="20">cgd.pt</site>
		<site ID="21">megaupload.com</site>
		<site ID="22">myspace.com</site>
		<site ID="23">hi5.com</site>
		<site ID="24">millenniumbcp.pt</site>
		<site ID="25">www.portaldasfinancas.gov.pt</site>
		<site ID="26">microsoft.com</site>
		<site ID="27">clix.pt</site>
		<site ID="28">meteo.pt</site>
		<site ID="29">standvirtual.com</site>
		<site ID="30">ojogo.pt</site>
		<site ID="31">aeiou.pt</site>
		<site ID="32">fileserve.com</site>
		<site ID="33">uol.com.br</site>
		<site ID="34">xvideos.com</site>
		<site ID="35">bing.com</site>
		<site ID="36">flickr.com</site>
		<site ID="37">zerozero.pt</site>
		<site ID="38">custojusto.pt</site>
		<site ID="39">paypal.com</site>
		<site ID="40">livejasmin.com</site>
		<site ID="41">thepiratebay.org</site>
		<site ID="42">badoo.com</site>
		<site ID="43">www.googleusercontent.com</site>
		<site ID="44">expresso.pt</site>
		<site ID="45">pai.pt</site>
		<site ID="46">rtp.pt</site>
		<site ID="47">ask.com</site>
		<site ID="48">youporn.com</site>
		<site ID="49">jornaldenegocios.pt</site>
		<site ID="50">dn.pt</site>
		…
		…
		<site ID="85">booking.com</site>
		<site ID="86">www.bpinet.pt</site>
		<site ID="87">maistrafego.pt</site>
		<site ID="88">xnxx.com</site>
		<site ID="89">neobux.com</site>
		<site ID="90">btnext.com</site>
		<site ID="91">pixmania.com</site>
		<site ID="92">globo.com</site>
		<site ID="93">adultfriendfinder.com</site>
		<site ID="94">torrentz.eu</site>
		<site ID="95">vodafone.pt</site>
		<site ID="96">adobe.com</site>
		<site ID="97">serbenfiquista.com</site>
		<site ID="98">mediafire.com</site>
		<site ID="99">netaffiliation.com</site>
		<site ID="100">ainanas.com</site>
	</top100>
	<downloadsize>4</downloadsize>
	<uploadsize>5</uploadsize>
</config>
```

Annex 3 – Web portal acceptance tests

Test No.	WEB.01
Title:	Registration security
Purpose:	Verify the registration procedure's security
Procedure:	1. In a standard browser (Firefox) access to the website and access the registration link 2. Enter any username and enter two different passwords. Verify the denial of registration. 3. Enter any username and equal passwords with only small case letters, both small and upper case letters, only small case letters and number, upper case letters and numbers. Verify the web portal denies registration in all instances. 4. Enter any username with correct casing password but with less than 4 characters. Verify the denial of registration. 5. Enter a username that already is present in the database. Verify the denial of registration. 6. Enter a new username and 2 correct casing passwords over 4 characters. Verify that registration is accepted.
Checks:	
Different passwords	Pass: ☒ Fail: ☐
Different case letters and numbers	Pass: ☒ Fail: ☐
Password less than 4 characters	Pass: ☒ Fail: ☐
Username already registered	Pass: ☒ Fail: ☐
Registration acceptance	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.02
Title:	Registration procedure
Purpose:	Verify the correct user registration procedure and database insertion
Procedure:	1. Verify that in the database (Users table) exists the previously created user. There should be a username, timestamp, hashed password and blob object (measurement agent). 2. From the database copy timestamp and hashed password. In a Java application compute the SHA-256 function using entered password and timestamp. Verify equal hashed strings 3. In the web portal verify that a new directory was created with the username inside the "config" directory. Verify that it only contains the Configuration file.
Checks:	
User table tuple creation	Pass: ☒ Fail: ☐
SHA-256 hash function	Pass: ☒ Fail: ☐
User directory creation	Pass: ☒ Fail: ☐

Test No.	WEB.03

Title:	Login security
Purpose:	Verify the login procedure's security
Procedure:	1. Enter invalid username and password, valid username invalid password. Verify denied access and correct redirection of the "Try again" link. 2. Enter a valid username and password with user privileges and verify that the browser is redirected to the user's dashboard. 3. Enter a valid username and password with administrator privileges and verify that the browser is redirected to the administrator's dashboard.

Checks:	
User table tuple creation	Pass: ☒ Fail: ☐
User redirection	Pass: ☒ Fail: ☐
Administrator redirection	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.04
Title:	Web portal security
Purpose:	Verify overall web portal security
Procedure:	1. Access the web portal main page using unencrypted HTTP connection. Verify that the browser is redirected to the secure HTTPS link (in case home page or error otherwise). 2. Using Wireshark verify that the HTTPS connection is being encrypted. 3. From the list of the web portal pages try to access them directly in the browser without going through the user login. Verify that access is denied. 4. With user credentials try to directly access any of the administrator's pages. Verify denied access. 5. Using Firefox browser install SQL Injection addon from "SecurityCompass" and perform all available SQL injection tests in all forms. Verify report with 0 failures.

Checks:	
HTTPS redirection	Pass: ☒ Fail: ☐
Connection encryption	Pass: ☒ Fail: ☐
URL direct access	Pass: ☒ Fail: ☐
Administrators web pages direct access using user credentials	Pass: ☒ Fail: ☐
SQL Injection	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.05
Title:	User's dashboard
Purpose:	Verify correct functionality of the user's dashboard
Procedure:	1. Login as a registered user and enter the dashboard. Verify that the webpage correctly identifies the username (upper left corner). 2. Verify that by pressing the logout button the session is destroyed and the browser is redirected to the login page. 3. Login with a registered user without connections. Verify that a message is displayed informing this situation. 4. Login with the same user and press the "Download Software". Verify that a download is performed with the filename "install.jar". 5. Login with a registered user with both unregistered and registered connections. Verify that the dashboard webpage presents both connections and distinguishes between the two states.

Checks:	
User identification	Pass: ☒ Fail: ☐
Logout procedure	Pass: ☒ Fail: ☐
Measurement agent download link	Pass: ☒ Fail: ☐
Connection presentation	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.06
Title:	User's connection management
Purpose:	Verify the correct connection management (presentation, deletion and registration)
Procedure:	1. Login as a registered user with both registered and unregistered connections. Click the delete link in a registered connection and unregistered connection. Verify that all information from those connections is deleted from the database. 2. Click in a unregistered connection's "Register" link .Verify that the browser is redirected to the register form. 3. In the register form verify that the connection is correctly identified by the label of the ISP identification. 4. Verify that the form only succeeds if all fields are entered (except "Postal Code"). 5. Complete the registration form and verify that the connection transits to the "Registered Connections" section and that price plan identification, download/upload speed are correctly mapped. 6. Click the delete in a registered connection and unregistered connection. Verify that all information is deleted from the database.

Checks:	
Delete connection	Pass: ☒ Fail: ☐
Register link	Pass: ☒ Fail: ☐
Register form ISP identification	Pass: ☒ Fail: ☐
Register form validation	Pass: ☒ Fail: ☐

Registration process	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.07
Title:	User's report
Purpose:	Verify the correct presentation of the user report
Procedure:	1. With a user with 2 or more registered connections click the "See results" link. Verify that the browser is redirected to the results web page. 2. Verify that in the top of the page all registered connections are presented. 3. Click the "All" link and cycle through all test classes. Verify that results are presented from all connections. 4. Click the "Select" link in any of the connections. Cycle through all test classes. Verify that only results from the selected application are presented.
Checks:	
Results link	Pass: ☒ Fail: ☐
Connection presentation	Pass: ☒ Fail: ☐
Results presentation (all connections)	Pass: ☒ Fail: ☐
Results presentation (single connections)	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.08
Title:	Administrator's Dashboard
Purpose:	Verify the correct presentation of the administrator report
Procedure:	1. Login as an administrator and enter the dashboard. Record summary information (number of registered users, number of connections, number of registered connections, number of connections with test performed in the last month). Perform database queries and verify the correct results. 2. Verify that the graphs are presented. 3. Using database queries verify that the registered "Registered Connections", "Number of active events", "Number of Events in the last 3 months" are correctly mapped. 4. Click the "Trend analysis" and "Aggregate analysis". Verify that browser is redirected to the respective webpages.
Checks:	
Results link	Pass: ☒ Fail: ☐
Graphic presentation	Pass: ☒ Fail: ☐
Graphic validation	Pass: ☒ Fail: ☐
Results presentation (single connections)	Pass: ☒ Fail: ☐
Report analysis link	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.09
Title:	Trend analysis report
Purpose:	Verify the correct functionality of the Trend analysis report
Procedure:	1. Login as an administrator and enter the trend analysis report. 2. Clear all data from the Connections table and create two connections from different ISP, district, access type, download/upload speed. 3. Create two events from each connection in the same day but with different hours. 4. Generate report with invalid date. Generate report with an out of range date. Verify that in both cases no graphs are presented. Select valid range date and verify that results are presented. 5. Within a valid date range select an aggregation date type "Hour". Verify that a trend analysis report is presented between two points. Select aggregation "Day", "Month" and "Year" and verify that only one point is displayed. Repeat with events from different days, months and years. 6. Generate new report only restricting one ISP. Verify that the presented values match the selected ISP. 7. Generate new report with district restriction from one of the connections and verify that only events from that connection are presented. Repeat restricting "County" parameter. 8. Generate new report restricting download speed containing one connection and verify that only events from that connection are presented. Repeat for upload speed. 9. Generate new report restricting endpoint location from one connection. Verify that only events from that connection are presented. 10. Generate new report with no restrictions with aggregation date that will generate two point trend graph. Register DNS, RTT, Throughput, Web browsing values. From the database estimate the average of all results and compare with the ones presented in the graph.

Checks:	
Date restriction	Pass: ☒ Fail: ☐
Date aggregation	Pass: ☒ Fail: ☐
ISP restriction	Pass: ☒ Fail: ☐
Regional restriction	Pass: ☒ Fail: ☐
Download / Upload speed restriction	Pass: ☒ Fail: ☐
Endpoint location restriction	Pass: ☒ Fail: ☐
Result validation	Pass: ☒ Fail: ☐
Comments:	

Test No.	WEB.10
Title:	Aggregated analysis report
Purpose:	Verify the correct functionality of the Aggregate analysis report
Procedure:	1. Login as an administrator and enter the trend analysis report. 2. Repeat the previous WEB.09 test in terms of date, ISP, District, County, Download/Upload speed, endpoint location restrictions. Verify that restriction affect the number of elements in the graph. 3. Generate new report without any restrictions, and record the presented results for each ISP. Manually estimate these values and verify that all values match.

Checks:	
Date restriction	Pass: ☒ Fail: ☐
Date aggregation	Pass: ☒ Fail: ☐
ISP restriction	Pass: ☒ Fail: ☐
Regional restriction	Pass: ☒ Fail: ☐
Download / Upload speed restriction	Pass: ☒ Fail: ☐
Endpoint location restriction	Pass: ☒ Fail: ☐
Result validation	Pass: ☒ Fail: ☐

Comments:	

Annex 4 – Measurement agent acceptance tests

Test No.	MA.01
Title:	Download process
Purpose:	Verify the download process
Procedure:	1. Using a standard browser (Firefox) login as a user and click the download link. Save the file to the local computer. Run file and start installation procedure. 2. In the first step, verify that both English and Portuguese languages are available. In the second step verify that the license is displayed and unless accepted the installation may not proceed. In the third verify that we are able to select different directories for the installation. 3. After the installation ends, verify that in the selected directory exists the following files: "webmontool.jar", "Setup.xml", "License.txt", "uninstaller.jar" (in the "Uninstaller" directory), "World.jpg" (in the "img" directory). 4. Run the application "webmontool.jar" and verify that a new icon is presented in the system tray.

Checks:	
Download link and installation file	Pass: ☒ Fail: ☐
Installation steps	Pass: ☒ Fail: ☐
Measurement agent files	Pass: ☒ Fail: ☐
Measurement agent correct startup	Pass: ☒ Fail: ☐
Comments:	

Test No.	MA.02
Title:	Measurement agent GUI
Purpose:	Verify the correct functionality of graphical user interface
Procedure:	1. Run the previously installed application. 2. Hover the mouse pointer over the system tray icon and verify that an information is presented to indicate that the user should access the application through the right mouse button click. The same information should be presented when double click action is performed but through a pop up message. 3. Access the application menu (right click) and click the "View results" option. Verify that the default system browser is opened and redirected to the application login page. 4. Access the application menu and verify that the "About" option open a popup message with the project and author identification 5. Access the application menu and click the "Run Test" option. Right after perform the same procedure. Verify that a popup message appears informing the user that a measurement test is being performed. Click OK and wait. When the test is finished a new popup message should appear informing that the test is complete. 6. Repeat the procedure but when a scheduled test is being run. 7. Repeat the procedure for hardware set 2.

Checks:	
Main menu access menu information	Pass: ☒ Fail: ☐

View results option	Pass: ☒ Fail: ☐
About option	Pass: ☒ Fail: ☐
Run test option	Pass: ☒ Fail: ☐
Test completion information message (on demand and scheduled)	Pass: ☒ Fail: ☐
Comments:	

Test No.	MA.03
Title:	Configuration file
Purpose:	Verify the correct Configuration file setup and update
Procedure:	1. Using a text editor edit the Configuration file (Setup.xml), and change version tag to 1. 2. Using the development IDE Eclipse change the source code to run a development function that prints all variables in the Setup class every time the setup procedure runs. 3. Run the measurement agent and verify that that the variables printed in the Eclipse console match the tags inside the Setup.xml (these include, version, main measurement site, remote endpoints, scheduler, list of 100 web addresses, and download/upload process file size. 4. Locally access the main site and with a system console, go to the directory of the user that was used to download the software. Inside there should be a configuration file. Edit it by changing the version tag to 2, and change some values in all of the configuration file's properties. 5. Next run a measurement test and after completion verify that the local configuration file has changed to the new version
Checks:	
Configuration file setup procedure	Pass: ☒ Fail: ☐
Configuration file update procedure	Pass: ☒ Fail: ☐
Comments:	

Test No.	MA.04
Title:	Scheduler
Purpose:	Verify the correct functionality of the scheduler
Procedure:	1. Change the configuration file for the current week and during a weekday set a scheduled test to run in a few minutes. Wait for the measurement test and login to the page to verify that results were uploaded. Repeat during a weekend. 2. Set the current day in the "Holyday" tag and schedule a test to run in a few minutes. Wait for the measurement test and login to the page to verify that results were uploaded. 3. Using the development IDE Eclipse change the source code to print the number of minutes that the application will wait until trying to run a scheduled measurement test. 4. During a weekday change the configuration file so that the next test will be in the weekend. Verify that the printed number of minutes corresponds to the time gap between the tests. 5. Repeat for the following sets: Weekday – weekday, weekday – weekend, weekday – holyday. Weekend – weekday, weekend – weekend, weekend – holyday. Holyday – weekday, holyday – weekend.

	6. Edit the configuration file and change the next scheduled test to run in a few minutes. Also change the maximum random time to wait before starting the test. Set this parameter to 2 minutes. Start the measurement agent and verify that the scheduled time to wait does not pass the time gap + the defined maximum random time. Repeat the process using several time frames.

Checks:	
Scheduled measurement test	Pass: ☒ Fail: ☐
Next test estimation function (during weekday, weekend, holyday)	Pass: ☒ Fail: ☐
Adding random time to scheduled test	Pass: ☒ Fail: ☐
Comments:	

Test No.	MA.05
Title:	Mobile version login
Purpose:	Verify the correct functionality of the Android version login procedure
Procedure:	1. Start the Android SDK emulator and the Eclipse IDE. 2. Install the project's application into the emulator. Verifying that we have network connectivity. 3. Try to perform a login with no password, then with no username, then with erroneous login credentials. Verify that the application denies access in all cases. 4. Enter valid username and password and verify that the application is redirected to the main page. Also, using the Eclipse IDE, verify that the Shared Preferences File contains the correct credentials as present in the database.

Checks:	
Incorrect login credentials	Pass: ☒ Fail: ☐
Login procedure	Pass: ☒ Fail: ☐
Comments:	

Test No.	MA.05
Title:	Mobile version login
Purpose:	Verify the correct functionality of the Android version login procedure
Procedure:	1. From the previous setup run a measurement test. After finish verify that in web portal database contains the tests results and no location information is present. 2. Using the http://itouchmap.com/latlong.html website find the GPS coordinates from a location from which the postal code is known. 3. Restart a new measurement test and during the test, and using the Eclipse Android plugin, enter the previous GPS coordinates. After the test completion, go to the web portal database and verify that the coordinates and the postal code is included.

Checks:	
Measurement test without location update	Pass: ☒ Fail: ☐
Measurement test with location update	Pass: ☒ Fail: ☐

Annex 5 – Measurement process acceptance tests

Test No.	MA.05
Title:	ISP and interface identification
Purpose:	Verify the correct ability to identify different types of connections
Procedure:	1. In Windows OS start the development Eclipse IDE. Turn off all interfaces and run a measurement test. Verify that the application returns an error indication, informing that no network is available. 2. Connect to the "MEO Fibra" residential network access using the wireless interface. Start a Wireshark network capture. Record the routers MAC address. 3. Using an internet browser go to the endpoint URL and access the "whatsmyip" page. Go to the www.whatsmyip.org webpage. Compare results from both webpages. 4. With that address perform a whois query using RIPE webpage. 5. Perform a measurement test. Login the user's dashboard and verify that a new connection was created. Compare the ISP identification and description with the obtained by the RIPE query. 6. Compare the Gateway MAC address from the one obtained in the Wireshark capture. 7. Compare the presented results in the Eclipse console with the one present in the user's result page. 8. Repeat with the following: MEO Fibra – wired interface IST eduroam – wireless interface PT WifiHotSpot – wireless interface Sapo ADSL – wireless interface ZON TvCabo – wireless interface 9. Repeat the procedure with the following dialup connections: VPN IST – PPTP connection TMN Banda Larga movel – PPP using 3G USB modem Instead of performing the Wireshark capture record the connection name as present in the OS and compare to the gateway address present in the user's dashboard.

Checks:	
Network availability check	Pass: ☒ Fail: ☐
MEO Fibra – public ip identification, RIPE query, MAC address	Pass: ☒ Fail: ☐
IST eduroam – public ip identification, RIPE query, MAC address	Pass: ☒ Fail: ☐
PT Wifi HotSpot – public ip identification, RIPE query, MAC address	Pass: ☒ Fail: ☐
SAPO ADSL– public ip identification, RIPE query, MAC address	Pass: ☒ Fail: ☐
ZON TvCabo – public ip identification, RIPE query, MAC address	Pass: ☒ Fail: ☐
ZON TvCabo – public ip identification, RIPE query, MAC address	Pass: ☒ Fail: ☐
VPN IST – public ip identification, RIPE query, connection name	Pass: ☒ Fail: ☐
TMN Banda Larga Movel– public ip identification, RIPE query, connection name	Pass: ☒ Fail: ☐
Upload test procedure	Pass: ☒ Fail: ☐