Web portal for continuous Internet access Quality of Service measurement

Ricardo Jorge Francisco Nunes
Instituto Superior Técnico
Av. Professor Cavaco Silva, IST Tagus Park, 2780-990 Porto Salvo, Portugal
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 present report addresses the problem of QoS measuring and presents a solution for a comprehensive evaluation system and QoS measuring tool for private consumers on broadband connections, both wireless and fixed. 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 extracted from the 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 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
Quality of Service, measurement platform, location aware.

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. The most referenced study in Portugal that analyzes this subject, is the one presented annually by the National Authority of the Communications (ANACOM) [ANA] 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. 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.

2. RELATED WORK
In this chapter we present some related projects both national and international. For each project we shall analyze the environment conditions, project goals, system architecture and selected metrics. The studied projects include: *ANACOM's* ISP evaluation study [ANA]; *LIRNEasia* [LIR] QoS benchmarks in South and Southeast Asia regions; Italian's *Osservatorio della Banda Larga* study [OBL] that used the commercial application Isposure, developed by *Epitirio*; *United States Federal Communications Commission* (FCC) national enquiry and broadband assessment, that used *Ookla Net Metrics* [OOK] and *M-Lab Network Diagnostic Tool* [MLAB]; Portuguese *Fundação para a Computação Cientifica Nacional* (FCCN) *Speedmeter[FCCN]*.

In terms of participant selection and test environment, [ANA] differs from all others by using a closed system (where participants are individually invited) and a controlled test environment (by using the same standard set hardware in the participant's premises and by restricting participants from using their network access during the whole process). 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 [ANA] method is the correct procedure to allow 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. Although [ANA] method provides more accurate KPI, it has serious impact in user's convenience and discourages participants to enter the process.

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. 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.

For the test frequency each [ANA] test lasted for 7 days and performed 8 or 9 daily tests depending if it is a weekday or weekend. [LIR] 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 terms of estimating the RTT, 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 [ANA] and [LIR] chose to measure the packet loss. [ANA] chose to send a burst of UDP packets and [LIR] 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 [ANA] 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. [OOK] 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

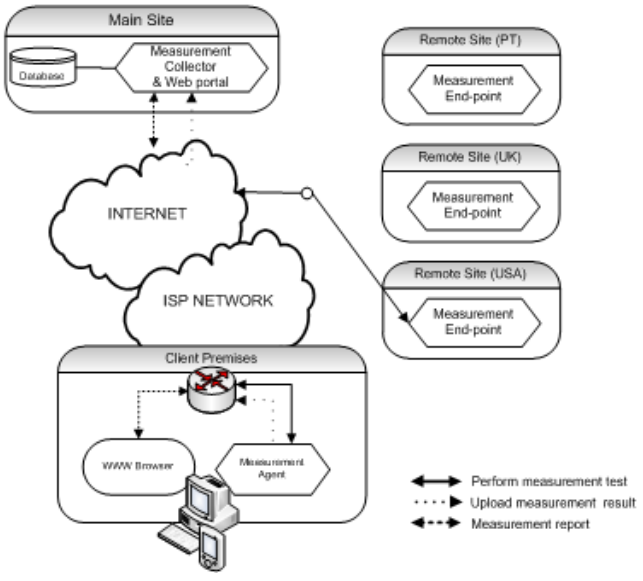

Figure 1 – Physical system architecture

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 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.

System utilization

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.

3.1 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.

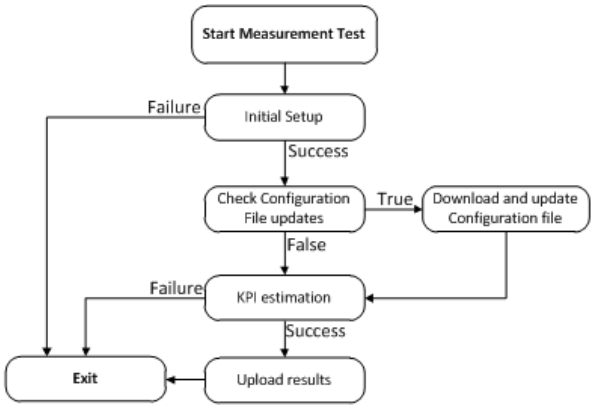

Figure 2 – Measurement process high level design

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. For the KPI estimation we will select a remote endpoint and perform the measurement test, that will then be uploaded to the Main Site.

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.

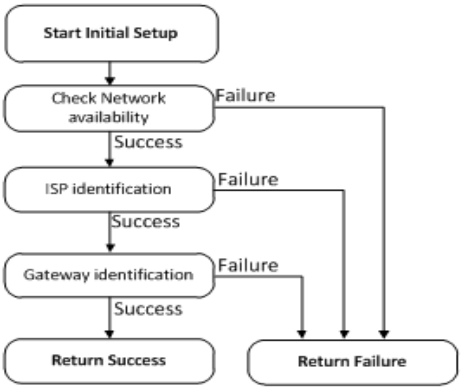

Figure 3 – Initial setup procedure high level design

The first elemental step of the measurement procedure is to assess if the network access is available. After this step we then proceed to identify the connection. Our approach to perform this task, is to have a 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 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.

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.

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 a live network capture, read the Ethernet frame of the packet that was returned by the endpoint.

In the mobile version or in case of the desktop version using a 3G USB modem we cannot get the MAC address as these connections use PPP protocol that does not have any associated Ethernet frames. Also 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. For the desktop version (3G SB modem) 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 may be changed by the user, but by default is set by the modem's software 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.

3.2 KEY PERFORMANCE INDICATORS

We tried to have a comprehensive set of KPI, having in mind however 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

Our approach to measure this KPI 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, video streaming and has a direct effect on other metrics. 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, we take a subset of 40 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. The standard average, maximum, minimum and jitter resolution time are taken from this measurement test.

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).

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. 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. The average is estimated by gathering the overall time each thread took to download its file and the file size. From the ratio of those elements we obtain the average throughput for each file. The measurement's overall average throughput speed is considered the highest reported throughput of all threads.

To obtain a maximum throughput speed, right after the file transfer process begins, we start a custom network sniffer (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.

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.

3.3 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. The main reason is that some users might not have a flat traffic plan rate and with this method we can set a higher frequency in the beginning of the month as lower as the week's progress. The scheduler is contained in the configuration file present in all measurement agents and distributed by the main site.

The first element we need to establish is the different hours in a day that tests will be performed. To prevent every test to start simultaneously we added a duration element in each hour. Every measurement test will add a random number of minutes from the established start time up to the maximum waiting time. An example of the test frequency could be:

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

Then next step is to assign these test frequencies. 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, so we have 10 different elements to which we assign the previous frequencies. 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.

3.4 SECURITY ASPECTS

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, both the measurement agent and the participants are only allowed to access the web portal through an encrypted connection through the standard X509 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.

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. We devised a custom process that takes the participant's credentials, and before registering them in our database we run a simple bash script that modifies the source code and includes these credentials (username is in clear text, but the password is firstly hashed using a SHA2 256 bit 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 section.

For the mobile version of the software 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.

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.

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 or Winpcap library. All other required libraries are fully portable and are included in the jar package. The external libraries included in the software are: the Apache HTTP Client to perform all HTTP connections; the JNetPcap, a java wrapper for the libpcap library; JSON serializabled container that holds the measurement results to be sent to the main site web portal.
The main flow of the application is composed of three main elements: the "Scheduler", the GUI and the "Controller".

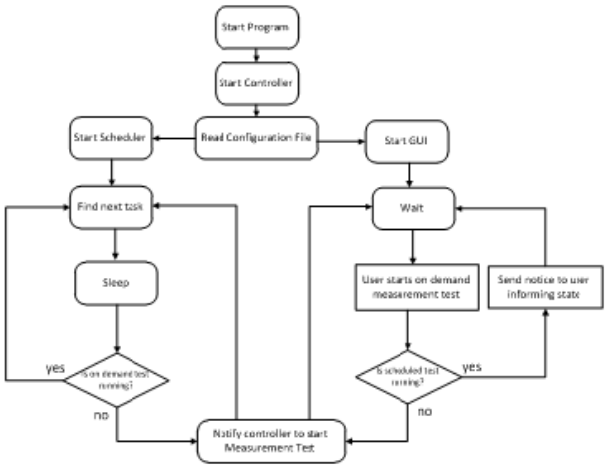

Figure 4 – Main application high level design

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.

4.2 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.

The endpoint was built from a standard LAMP (Linux + Apache + PHP + MySQL) server. The list of software used for the implementation is as follows: Linux 2.6.32-32 kernel version; Ubuntu 11.04; 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:

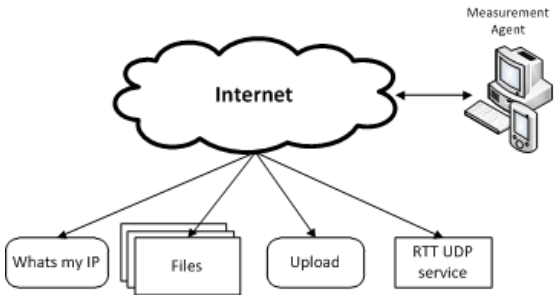

Figure 5 – Measurement endpoint conceptual design

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.

4.3 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.
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; php5-mysql (MySQL library for PHP integration); openssl 0.9.8k (for creating self-signed certificates); open flash chart2 (for graph creation).

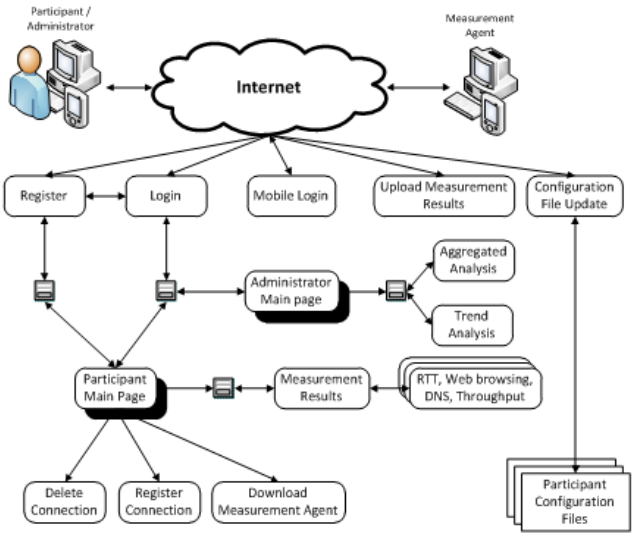

Figure 6 – web portal conceptual design

The main entry point is the Login/Register page. Here we present a form that will validate or create a new user. Upon login if the user is an administrator he will be redirected to the "Administrator Dashboard". In here he will access the project's dashboard containing information about general aspects of the platform. If the login has user level rights, he will be redirected to the "Participant's Dashboard". 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.

Participant's Report

The participant's report is comprised of two main elements: a main dashboard and a measurement analysis.

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. This management is performed through a presented list of the participant's 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 the Android version, or the dialup connection name in case of PPP).
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 is seldom used he/she may delete that connection, thereby erasing all information related to it.

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.). To accomplish that, we have performed a market search and gathered a set of network access connections from all major ISP. As new connections are developed, the administrator should add them to our database and the webpage is automatically updated.

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 thkat 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.

Measurement results

The measurement results are presented in a web page as described in Figure 7. 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.

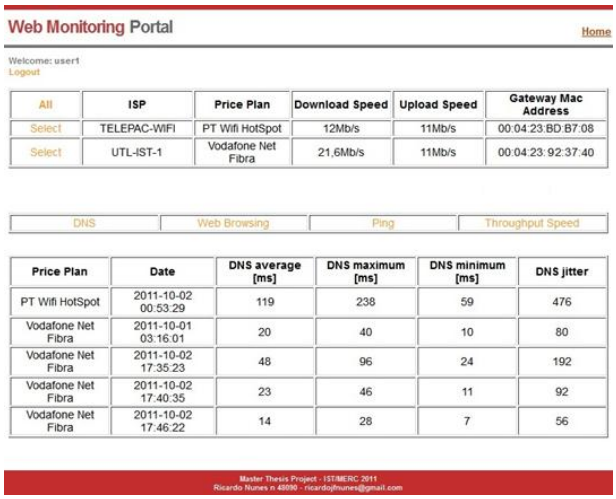

Figure 7 – Example of a participant's measurement result web page

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, the aggregated analysis and the trend analysis

Dashboard

The "Administrators report" starts in the "Dashboard" where we can obtain the overall participant adherence to our measurement platform. 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 and number of active registered connections (with events in the last month).

Following in the dashboard we have 3 graphs that allow us to have an assessment of the number of connections distributed between ISP. The first shows the distribution of connections between ISP since the beginning of the project. The next graph tells us how many of the previous connections are still considered active (at least one event in the last month). The last graph allows us to determine how many measurement results were performed in each ISP connection in the last month. If we have insufficient data from a certain ISP the system administrator may want to exclude that ISP from the report.

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.
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. 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 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.

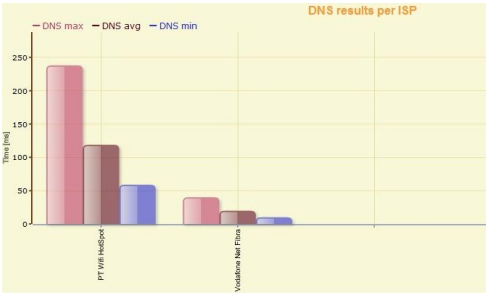

Figure 8 – Example of an aggregated analysis graph

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. 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. The available aggregations are by hour, day, month and year.

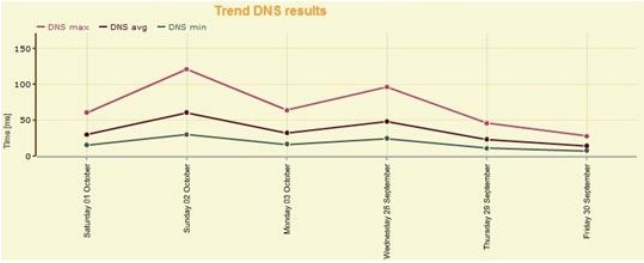

Figure 9 - Example of a trend analysis graph

5. SYSTEM TESTING

To assess the validation of our measurement platform we compared the measurement results with some established and commercial applications. As for the system setup, we installed our main site in a Laptop with a Pentium M 1,5 GHz, 512MB RAM and running the Linux Ubuntu 11.04. This equipment was behind a standard residential ISP router gateway with fiber network access type with 100Mbps bandwidth connection. In the second equipment we installed the measurement agent and the several applications that we will compare. This equipment is also a laptop with an Intel Core 2 Duo P8600 @2,4GHz with 4GB RAM and running Microsoft Windows 7. We went to several ISP clients premises and tested each one of our KPI with a several well-known applications. Furthermore we registered the network connection identification (ISP and gateway) and compared with the elements present in the web portal.

5.1 RESULTS

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 1 – Results from RTT system test

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

ISP	Average [ms]		Maximum [ms]		Minimum [ms]		Jitter	
	Java	nslookup	Java	nslookup	Java	nslookup	Java	nslookup
IST eduroam	28	106	153	470	3	72	2339	4958
PT Wifi HotSpot	257	301	1377	1024	27	20	72098	60227
ZON TvCabo	2	4	19	14	1	3	16	4
TMN Banda Larga Móvel	133	324	214	394	117	229	326	684

Table 2 - Results from DNS system test

To evaluate our method of estimating the web browsing KPI we compared with the command line application CURL. 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.

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 3 – Results from the web browsing tests

For the download speed KPI we performed a comparison with the results obtained from two applications: the GNU WGET command line application and the commercial download manager "Download Accelerator Plus" (DAP).

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 4 – Results from the download speed tests

For the upload speed this test we compared our method with the CURL application, by performing an HTTP POST with an attached file.

ISP	Average [KB]		Maximum [KB]	
	Java	CURL	Java	CURL
IST eduroam	2133	1861	2123	2220
PT Wifi HotSpot	65	79	101	116
ZON TvCabo	2170	2021	2342	2153
TMN Banda Larga Móvel	93	116	-	137

Table 5 – Results from the upload 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. 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. 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 credentials used were from a public shared account.

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.

From the QoS measurement perspective we would like 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.

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.

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 [GLAS] test server. Finally it would be interesting to emulate the behavior of a VoIP call or streaming video.

7. CONCLUSION

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

[ANA] ICP-ANACOM, Estudo de aferição do serviço de acesso à internet Banda Larga, Relatório Metodológico, July 2010

[LIR] Tim Gonsalves, "Broadband Quality of Service Experience Test Results", March 2009

[LIR2] Timothy A. Gonsalves & Anuraag 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

[OBL] Osservatorio della Banda Larga, Italian Broadband Quality Index, Rapporto Preliminare, October 2009

[FCC] Federal Communications Commission, Connecting America: The National Broadband Plan, March 2010

[GLAS] M. Dischinger, M. Marcon, S. Guha, K. Gummadi, R. Mahajan, S. Saroiu, "Glasnost: Enabling End Users to Detect Traffic Differentiation", NSDI 2010 paper.

[CN]A. Tannenbaum, "Computer Networks", 4th edition, Prentice-Hall, 2003

[ADSL] ITU-T Recommendation G.1000, Communications quality of service: A framework and definitions, November 2001.

[QoS] ITU-T Recommendation E.802, Framework and methodologies for the determination and application of QoS parameters, February 2007.