MASTER THESIS

TITLE: Proxy performance analysis in a community wireless network

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Pablo Pitarch Miguel

DIRECTOR: Roc Meseguer

DATE: October, 21st 2013

Títol: Anàlisi de rendiment dels proxis en una xarxa comunitaria sense fils

Autor: Pablo Pitarch Miguel

Director: Roc Meseguer

Data: 21 d'octubre de 2013

Resum

Atès que la connexió a la Internet s'ha convertint en un aspecte fonamental per al desenvolupament de la societat cada vegada més millores i noves tecnologies d'accés han aparegut. Guifi.net proposa una xarxa sense fils comunitària lliure, neutral i oberta amb l'objectiu principal d'arribar als usuaris que no tenen moltes altres opcions de connexió.

L'optimització dels recursos de la xarxa es torna més important per a aquest tipus de connexions. Aquest treball té com a objectiu proposar un mètode per millorar la figura principal d'accés a Internet d'aquesta xarxa, els servidors proxy. Aquest element de xarxa permet a l'usuari final, que estigui degudament registrat, accedir a Internet amb autenticació. Les portes a Internet a Guifi.net són administrades per organitzacions i institucions públiques i són comunament federades el que significa que tots els usuaris registrats en un proxy poden accedir a un altre proxy federat amb la mateixa autenticació. Així, els usuaris finals tenen diverses opcions per connectar-se.

Per procedir a fer l'anàlisi de rendiment d'aquests elements de xarxa, dos trams diferenciats han estat definits, el pas entre el client final i el proxy i el pas entre el proxy i el servidor web on hi ha el contingut desitjat. Per analitzar el primer tram s'ha utilitzat l'eina ping mentre que per analitzar el segon tram s'ha utilitzat l'eina wget. Ping es basa en la capa IP mentre wget treballa a la capa TCP, per tant es necessari fer un reajustament per poder comparar els resultats obtinguts. En aquest sentit, s'ha aplicat un calibratge del mètode d'adquisició de ping amb una eina basada en TCP com iperf fent la prova en alguns servidors iperf disponibles a la xarxa.

Les figures necessàries per poder avaluar el rendiment són la latència, el throughput, la pèrdua de paquets, els nombre de nodes intermedis entre el client i el proxy i la distància. Les proves revelen que no tots els servidors intermediaris federades estan disponibles per a un sol usuari a causa de la singular topologia de la xarxa. També revelen que hi ha una connexió directa entre el nombre de nodes intermedis i la latència, el throughput i el rendiment de la pèrdua de paquets, per la seva banda la distància no segueix cap relació amb la resta de figures. Per tant, és possible concloure que per a aquestes proves el criteri adequat per a l'elecció proxy és el nombre de nodes intermedis en comptes de la distància que és el criteri comunament utilitzat.

Title: Proxy performance analysis in a community wireless network

Author: Pablo Pitarch Miguel

Director: Roc Meseguer

Date: October, 21st 2013

Overview

Since the connection to the Internet has becoming a fundamental issue to the development of the society more and more improvements and new access technologies have appeared. Guifi.net proposes a free, neutral and open community wireless network with the principal aim to arrive to users that doesn't have many other options.

The optimization of the resources of the network becomes more important for this kind of connections. This study pretends to analyze and learn the behavior of the principal Internet access figure of this network, the proxies. This network element allows to the final user that is properly registered to access Internet with an authentication. Proxies in Guifi.net are administrated by organizations and public institutions and they are commonly federated what means that all the users registered in a proxy can access to another federated proxy with the same authentication. So, final users have several options to connect.

To proceed with the performance analysis of these network elements two differentiated stretches have been defined, the step between the final client and the proxy and the step between the proxy and the web server where the content is. To analyze the first stretch, ping tool has been used while to analyze the second step wget tool has been selected. Ping is based in the IP layer meanwhile wget works in the TCP layer so some readjustment have to be done to compare the results obtained. In this sense, a calibration of the ping method with a TCP based tool as iperf has been applied, doing the test to some iperf servers available in the network.

The required figures to evaluate the performance are latency, throughput, packet loss, intermediate nodes between client and proxy and distance. Tests reveal that not all the federated proxies are available to a single user due to the singular topology of the network. Also reveal that there is a direct connection between the number of intermediate nodes and the latency, throughput and packet loss performance, meanwhile distance doesn't follows any relation. So, it is possible to conclude that for these tests the proper criterion for the proxy election is the number of intermediate nodes instead the distance which is the commonly used.

CONTENTS

INTRODUCTION

Currently the Internet world is evolving; the amount of devices that demand a connection to the network to adapt to the constantly changing needs of users is increasing. A great engine for this phenomenon to occur is that the new technologies of connection and network access are becoming a reality at high speed. So, for the objective of the whole system can meet these needs, requires that all players respond in a compensated way for such advancement.

To provide an efficient connection to users is important to make the right decisions in the network management. This project emphasizes the latter in order to make decisions that allow the user to enjoy an optimized perception in the navigation. Here comes into play one of the key pieces in the client-server communication within the network, the proxies.

Proxies are intermediate components between the user client and the web content server and they are responsible for providing a range of services to improve the Internet connection such as the cache for provisioning of recursive content or search the content server faster. In a network usually there is more than one proxy available for a client and it must be able to discern what the option that gives a better service is. This is the main point on which this study focuses, which discusses various factors that may affect this choice such as the distance, intermediate nodes has to go through the information to get to the proxy or the transfer rate allowing each case.

The principal goal of this study is to establish a method to analyze the performance of elements in a Wireless Community Network (WCN). To do this, the first step is to set a test scenario to design and create the proper probes to evaluate the system. Once these probes are designed the next step is to apply these tests in a real network. For this particular case the tests will be done at Guifi.net network, a neutral, free and open ad hoc WCN that makes the network special.

The desired elements of the network to analyze are the federated proxies available from a single client and his behavior. These elements are in charge of providing Internet to the network and are very susceptible to the response time. In this order, the study will be centered in evaluate the throughput and the latency of the proxies.

Once the test results have been obtained, the bases to be able to choose the best option for a certain client node that wants to be connected to the network will be established.

The first chapter outlines the main features of the network in which the tests are conducted. The network in question is Guifi.net, a network classified as ad-hoc and administered entirely free and open by the users themselves.

Chapter two focuses on presenting the software used for testing, as well as the primary operating system that the various devices in the network use. We also evaluate its characteristics and role in interpreting and evaluating system for choosing the best solution.

In the third chapter discusses the various figures that purport to measure within the network and that directly influence the system behavior. Figures are evaluated which can produce some indicators to better manage the acquired data.

The fourth chapter presents the steps, prior to the tests performed, within the Guifi.net network. This will detail the steps to simulate a scenario similar to that will stand the various methods for measuring the required figures. Once constituted this test environment, the different probes are designed to collect the desired data. Also describes preparations to perform the tests mentioned in previous chapters.

In the fifth chapter the results once implemented different methods of data have been collected. Thus it is possible to make an analysis of proxy servers available on the network and argue which ones offer a better service. With the results obtained it is possible to determine the best criterion to choose a proxy.

CHAPTER 1. NETWORK CHARACTERISTICS

1.1. Wireless Community Network (WCN)

Wireless Community Networks has his origin in some particulars organized to develop a network that provides an alternative connection to the final clients. They have become more popular as wireless technology has evolve in the last decade and they are able to offer a properly connection in disposition to compete with the products offered by the common Internet providers.

The administrators of this kind of network are the own people that propagates the network and adds the services to the system making it open and free. This propagation of the network is coordinated by a system of co-operation backed by a webpage, where the users have to registered and announce their updates to be known by the community and also receive assistance for any issue or necessity. Wireless Community Networks are spread all over the world; they are normally organized by reduced regions that can be easily administrated.

1.2. Guifi.net network

1.2.1. Introduction

Guifi.net is an open, free and neutral telecommunications WCN with an interconnection agreement between users that extends the network to connect and get connectivity. Guifi.net network is mostly composed by wireless connections but there are also some fiber sections. This kind of structures is named Ad-hoc networks. Guifi.net foundation is registered at the CMT as a telecommunications operator.

The principal focus of Guifi.net network is Catalonia and Valencian Community, but there are nodes located all over Spain making it one of the largest in the world with these characteristics. If a new user wants to be added to Guifi.net network via wireless connection has to check for the coverage in his area in the Guifi.net website and place a node with the characteristics and situation recommended in the Guifi.net bases. There is any barrier for users to access to this network, this fact makes it open. Every user is in charge of his own node and has the responsibility of his maintenance and management.

Guifi.net was born with the purpose of provide shared resources connection to rural locations in Catalonia in a cheap and easy way. With these resources, users could have some basic network utilities as file transferring, mail, videoconferencing or access to the Internet, although any service is warrantied.

1.2.2. Network specifications

Guifi.net is composed of more than 28.000 nodes where around 18.000 are active nodes. This numbers make an estimation of the magnitude of the network in Spain. Every node can have an access point associated. Each AP provides connectivity to the user's devices to share their resources.

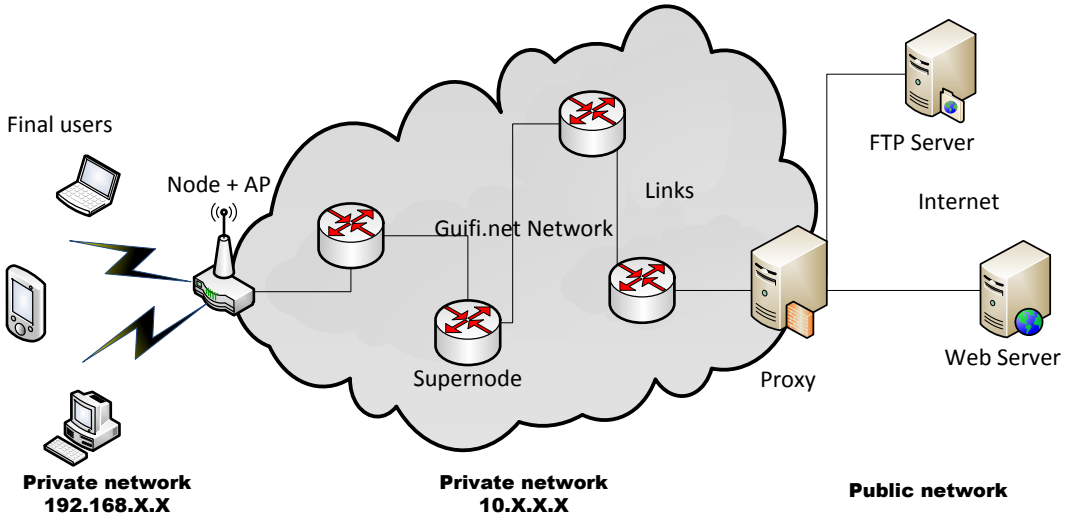

Figure 1.1 Basic scheme of Guifi.net architecture

The Guifi.net network is composed by several important parts to provide the demanded services to the final users. The basic main actors are the supernodes, simple nodes, access points, links and proxies. Supernodes are in charge of extending the network geographically to big distances and provide a connection to different areas. Simple nodes are placed by users to capture the signal and accede to the network. Access points receive the signal from the nodes and share it to the different final user's devices (laptops, computers, mobiles, tablets…). Links are the mediums where the signal transmits, in this particular case can be, normally, fiber or wireless. Proxies will be analyzed in sections 1.2.3. and 1.2.4.

1.2.3. Guifi.net proxy definition

There are lots of servers all along the network that provides services to the users. The most demanded one is the Internet proxy. The vast majority are federated proxy based on squid. Squid is GNU GPL software that consists on a proxy server for web with cache. It also allows other protocols traffic instead HTTP as FTP and offer a serial of services like users admissions, improvement of the connection with the cache or security filter for some contents. There is an open and free packet to federate a proxy provided by Guifi.net admins. Once the federation packet is installed, the signing on of the proxy in the Guifi.net web is needed to can be reached by users. There are three different forms of

federating a proxy, federated IN, federated OUT or both at the same time. The IN proxy accept validated Guifi.net users belonging to another federated proxy, meanwhile OUT accepts that users that belongs to this proxy can accede other federated proxys. Users also have to be registered in the system with name and password to accede to the proxy.

In figure 1.2 it is possible to see the basic scheme of a Guifi proxy cache. Proxies are administrated by the owners of the server where they are installed and they can apply their own policies. These policies can be consulted at the Guifi.net web for each proxy. Most of them limit the upstream and downstream bandwidth available to the users due to it is a shared medium, for instance, it can share the bandwidth with the connection of a town hall. This limitation can affect to the user perception when visiting some Internet pages but, on the other hand, the cache minimizes this effect due to it stores the information in the server to proportionate it faster. So, the performance of acceding to the Internet will be different for each user depending on what proxy they are connected to.

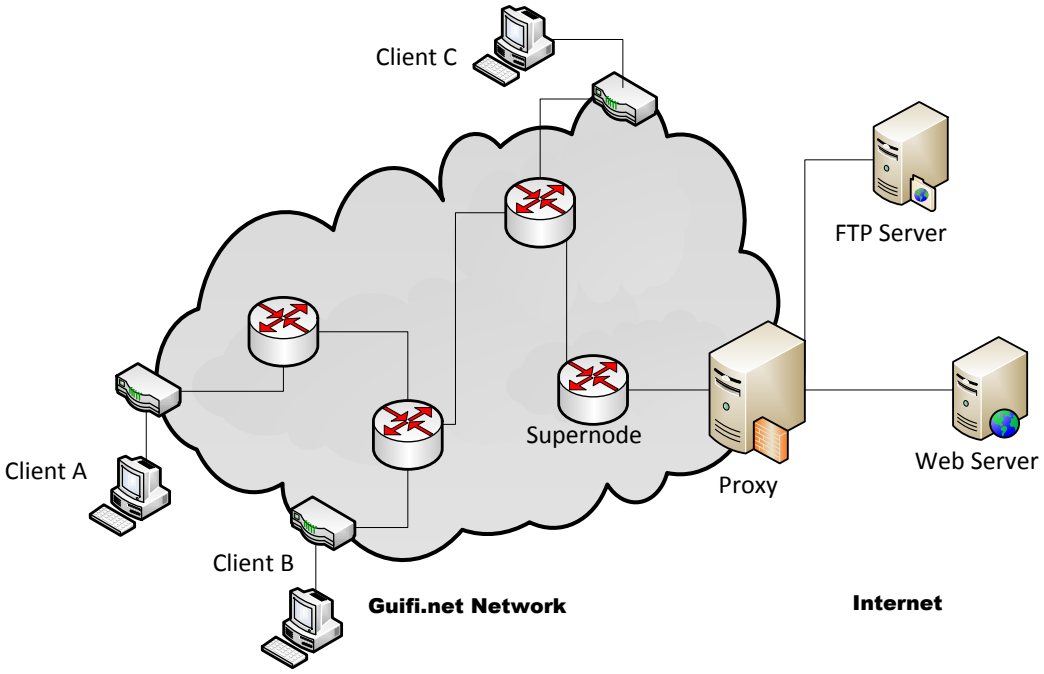

Figure 1.2 Tipical proxy structure

1.2.4. Guifi.net proxies performance

All the proxies are administrated by their owners, which provide an Internet access to the users without any benefit. Normally, these owners are institutions and they have to ensure a proper behavior of the system controlling the resources. A large number of proxies not only limit the contents that can be transferred but also the upstream and downstream bandwidth from the web servers to the proxy.

In figure 1.3 it is possible to see how the 67 operative and federated proxies perform in terms of bandwidth. This data is published in Guifi.net by every administrator and it goes from 64Kbps to 40Mbps for the downstream and from 64Kbps to 20Mbps for the upstream. At the graphic it is possible to see that near the 80% of the proxies has a bandwidth below 8Mbps and near a half below 4Mbps. For the upstream this data is reduced to the half, the 80% of the servers provide less than 4Mbps and the 60% of the proxies allows rates below 2Mbps.

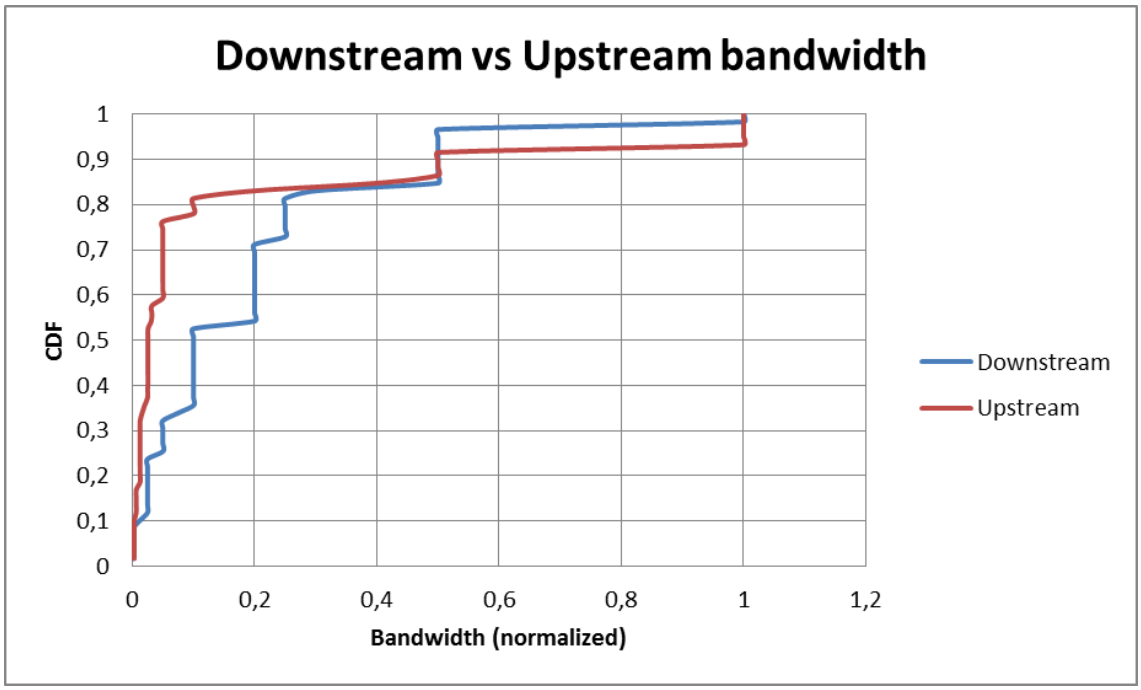

Figure 1.3 Normalized downstream and upstream bandwidth limitation by federated proxies performance

1.3. Confine project

The confine project is a complement of Guifi.net network that has an important paper in controlling the system. Confine is specialized in large community networks that include a wide variety of commodity wireless and optical links, heterogeneous nodes, different routing protocols, applications and a large number of end-users, following an innovative model of self-provisioning using unlicensed and public spectrum.

The project develops a unified access to an open testbed with tools that allow researchers to deploy, run, monitor and experiment with services, protocols and applications on real-world community IP networks. This integrated platform will provide user-friendly access to these emerging networks supporting any stakeholder interested in developing and testing experimental technologies for open and interoperable network infrastructures, strengthening open community networks. Figure 1.4 shows the typical installation of the Confine nodes in the

Guifi.net antennas to do external tests with the condition that it has at least three nodes with direct visibility.

Confine project is in charge of various networks in Europe. To the already named Guifi.net, join networks as Athens Wireless Network (AWNM), Funkfeuer in Vienna and Freifunk in Germany.

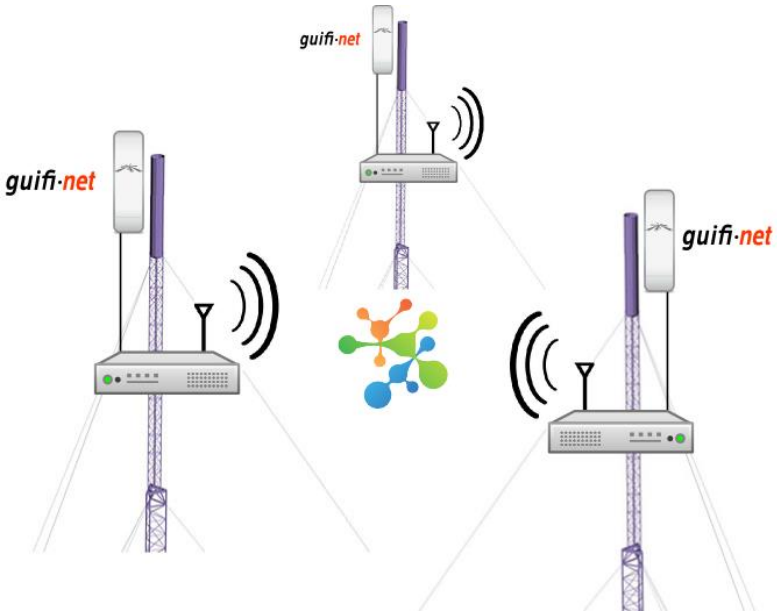

Figure 1.4 Confine nodes' in a Guifi.net network

CHAPTER 2. SOFTWARE SPECIFICATIONS

Choosing software to do the test is one of the most critical decisions. There is the need of selecting software that fits better taking into account the scenario and the objective to achieve. First of all, the OS have been chose. Afterwards, the basic tools to do the tests have been selected.

2.1. OpenWrt

The first decision that has to be taken is the OS used to do the tests, not only in the real scenario but in the preproduction scenario. In this case, the best option is OpenWrt due to it is the distribution used in Guifi.net nodes.

OpenWrt is a free Linux distribution which principal characteristic is that is thought to wireless ad-hoc networks. Instead of trying to create a single, static firmware, OpenWrt provides a fully writable filesystem with package management. This frees you from the application selection and configuration provided by the vendor and allows you to customize the device through the use of packages to suit any application.

In order to prepare the tests there is a list of packages available in the OpenWrt nodes. This is an important limitation to take into account in order to select the proper tools to do the tests because there is the possibility of finding some useful procedures that cannot be used. So the solution selected is to use Iperf and ping that fits perfectly as for the objectives and for the list of packages available in the OpenWrt nodes.

2.2. Iperf

Iperf is one of the most popular tools for throughput and bandwidth measurement. It although measures other network figures as jitter or the number of packets lost. In order to have more versatility at the tests, iperf offers the possibility of doing the probes with TCP and UDP depending on the configuration used.

This particular tool creates TCP or UDP flows of a certain size and taking into account the time invested it calculates the real throughput of the system. It is possible also to determine the port used (5001 by default) or even the size of the windowing in the TCP transfers.

2.3. Ping

Ping is a traffic control tool based on ICMP protocol. It is intended to test the reachability of a certain host on a network. It also measures the Round Trip Time (RTT). With this tool it is possible to define the size of the ping packets in bytes and preset the quantity of pings sent in each command.

Once a ping test is done, it is possible to take some conclusions. If the result is that host is unreachable, it means that the destiny host or the origin host is unavailable or the connection between them is failing, so these three parts have to be checked. If the RTT is too large means that there is not a good connection in this link and will cause a problem of user perception.

There are many reasons to have a RTT large. The principal one is that the link has a small bandwidth in relation with the traffic, and taking into account that ping traffic often has least priority inside the network in relation with other protocols, the RTT will be affected. So, it is possible to conclude that in these cases ping gives the worst case RTT of the connection. Other possible reason for a large RTT is a bad configured policy at the destiny host.

When a network is symmetric as Guifi.net, it is possible to calculate the worst case throughput of any reachable link. This method consists on taking the relation of size of packets sent and the RTT inverted. There are many factors that can affect to this measure, so an improvement of this method has been proposed in chapter 4. In an asymmetric link this method cannot work properly because RTT doesn't discriminate upstream and downstream and, for instance, only the average throughput of the two parts can be determine.

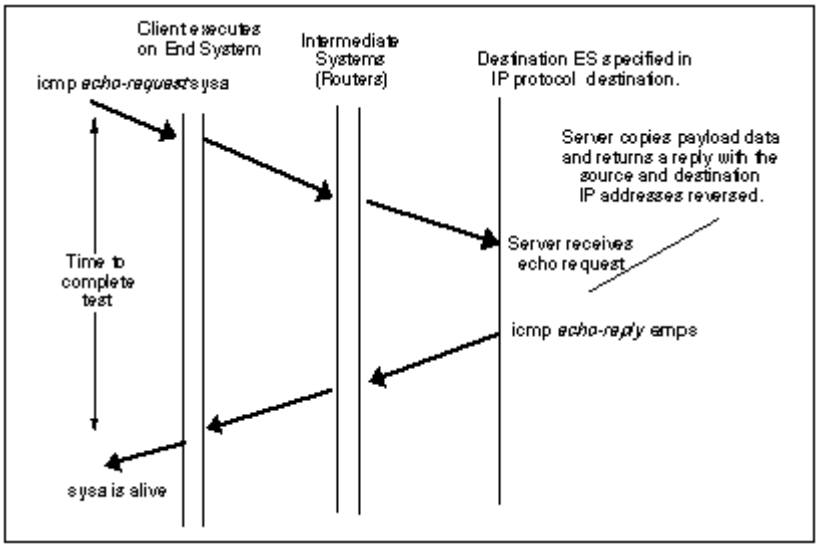

Figure 2.1 Ping life cycle

Ping also has a field for the time to live TTL. This figure means the number of routers hops for a packet before being discarded to avoid these packets to remain in a network without reaching a destination and wasting bandwidth. TTL is set to 255 hops by default but can be set from 1 to 255 by the command 'ping –i x' where x is the maximum desired number of hops. This figure can also give useful information due to it is possible to know how many intermediate routers are between the transmitter and the receiver.

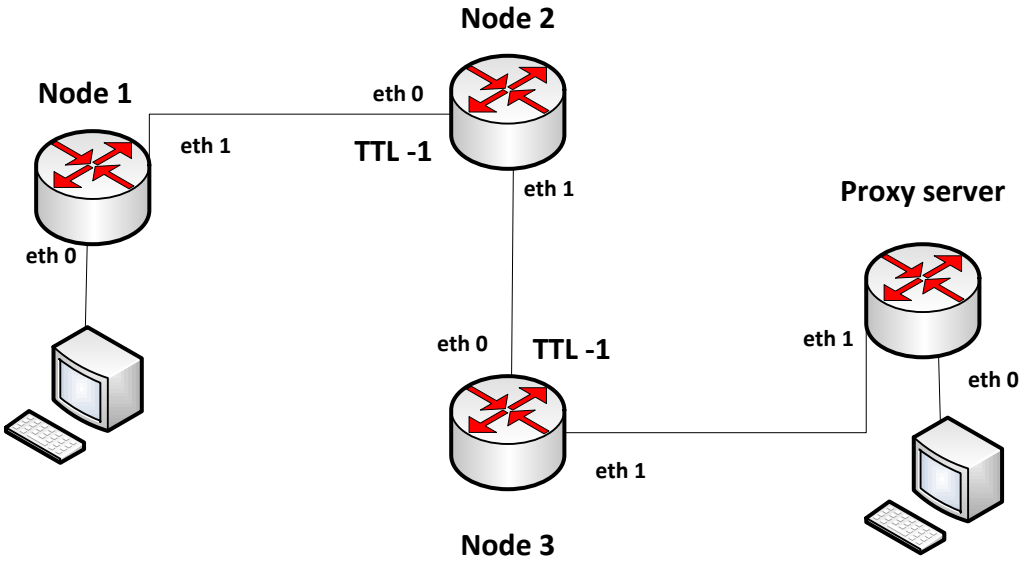

Figure 2.2 TTL decrement in a network of 2 intermediate nodes in a communication between a final user and a proxy

In figure 2.2 it is possible to see that nodes 2 and 3 decrease TTL in 1 each. For instance, if a ping to the proxy is done and the ping response has a TTL settled to 8 the field TTL in node 1 will be 6 and this two intermediate nodes can be identified.

Knowing the number of intermediate nodes is important due to this fact can influence in the routing. Sometimes is preferable to choose a route with a low number of hops to minimize the possibility of going through routers that can have congestion or a poor performance.

2.4. Wget

GNU Wget is a free software package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive command line tool, so it may easily be called from scripts, cron jobs, terminals without X-Windows support, etc.

Wget fits to the porpoise of this project due to it calculates the throughput and time spent when it downloads the content demanded. Throughput is calculated dividing the size of the file transferred with the time spent. Also, it is possible to change the proxy which will demand the information requested to evaluate the performance of all the proxies available in the network. The way to change the proxy is as follows:

export http_proxy="http://username:password@foo.bar:8080"

Where foo.bar is the domain of the proxy. Most of Guifi.net proxies demand a username and password for authentication. For this test only federated proxies are going to be evaluated, so only one user will be need. The next command will be executed to download and image from a page located in a Google's server:

wget http://www.google.es/images/srpr/logo4w.png

This image has a size of 19KB.

CHAPTER 3. MEASURES

One important issue is to establish the different measures that are wanted to be acquired. Once they are determined, studying them can give an extra value to the research. To achieve the goals proposed there are four interesting main figures as RTT, throughput, packet loss and the number of hops. These four figures can be determined with the tools iperf and ping analyzed in chapter 2.

3.1. Round Trip Time (RTT)

Round trip time is defined as the time that spends a packet since it is transmitted from the source until the response from the receiver returns back to the source. There are many factors to take into account that affects to the final RTT. The principal one is the propagation delay in the link of the communication; this factor will have a relative impact in the round trip time and is determined by the speed of the transmission in the medium. Another factor that affects to the RTT is the process of the packet in the nodes where it passes through; these nodes can behave in a different way each one depending on their characteristics and add time to the total RTT of a communication. Buffers management has a big influence in this last point and the configuration of the queueing. Lastly, the protocol used in the communication also affects to the final RTT, for instance, an efficient election of the TCP window of transmission will reduce the RTT.

Table 3.1. Figures that affect to the total RTT

Figure	
Propagation delay	$RTT_{Propagation}$
Queuing	$RTT_{Queuing}$
Node routing	$RTT_{Routing}$
Protocol	$RTT_{Protocol}$

In the equation 3.1 is possible to see the influence of each factor that affects to the final RTT discussed in [8]. Each one has to be treated to have an effective system to reduce it.

$$RTT_{Total} = RTT_{Propagation} + RTT_{Queuing} + RTT_{Routing} + RTT_{Protocol} \quad \textbf{(3.1)}$$

The RTT is a significant figure due to it is related with the user's perception, so it is important to improve the system of communication to have a good value in this figure. The user perception has more importance when web browsing.

Latency can be defined as the one way trip time and the indicative acceptable values of this figure for the different purposes are as follows:

Table 3.2. Acceptable values of latency for different types of traffic

Purpose	Acceptable value of latency
Real time games (TCP)	100ms
VoIP (RTP)	150ms
Remote administration (SSH)	300ms
Web browsing (HTTP)	200ms

G.114 ITU recommendations says that a latency beyond 150ms is acceptable meanwhile above 400ms is unacceptable for VoIP. In [7] the effect of latency in games is analyzed arriving to the conclusions that depending on the kind of game there is a different threshold between 100ms and 1s. For web browsing a latency of less than 100ms is undetectable while a value of 200ms is still an acceptable delay.

3.2. Throughput

Throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot.

There are many tools and ways to measure the throughput of a network. For this study, iperf is the selected tool to do these measurements. In this case, iperf, sends a quantity of bytes of data to a server and it does the relation between the amount of data sent and the time that this data took in arrive to the destiny. This test can be done with TCP and UDP protocols and it allows setup a serial of option as the TCP window or the amount of data sent.

Nevertheless, not always there is an iperf server running at the nodes, so, a new method to the estimation of the throughput has to be designed. After the evaluation of some methods to calculate this figure such as passive throughput measurement with traffic sniffers, ICMP ping has been selected due to it adjust to the requirements for the measurements desired. In chapter 4 there is an explanation of how the throughput can be estimated in this way. To probe the reliability of this method, the test will be done at the same iperf servers to compare the results. Finally, the throughput between the web servers and the client node will be calculated with wget program.

Guifi.net network is generally symmetric, so the values of downstream and upstream bandwidth are supposed to be the same. However, there are some

factors that affect the throughput such as server's configuration that provoke traffic shaping.

3.3. Packet loss

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. A system with a high rate of packet loss will have different problems in UDP and TCP. In UDP the lost packets will not reach the destiny, normally it will be video and voice traffic so if the rate is very high the communication will be deficient. In a TCP link the effect will be different. In this case, the protocol demands a retransmission for every lost packet which leads to a reset of the TCP window, making the transference of data much more slow due to the sender will wait to receive the ack that assure the correct reception to transfer the next packet, instead of sending numerous packets without waiting for an ack.

Packet loss can be caused by a number of factors including signal degradation over the network medium due to multi-path fading, packet drop because of channel congestion, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines

The packet loss can be detected with the tools iperf and ping. Due to the ping test can be done in all the proxies and nodes proposed it will be the method selected to collect this data.

3.4. Number of hops

Another interesting measure is the number of hops that separates the two ends of a network communication. Each hop is considered a node that routes the packet to the next node of the routing table or the destiny. This figure can be calculated with multiple applications as tcpdump with the option –vv which gives additional information of the traffic capture or, as in this case, with the tool ping which have a count of the TTL, knowing that each decrement of this field means an intermediate hop [see section 2.3].

```
root@pablo-desktop:/home/Scripts# ping 10.138.27.98
PING 10.138.27.98 (10.138.27.98) 56(84) bytes of data.
64 bytes from 10.138.27.98: icmp_req=1 ttl=54 time=46.8 ms
64 bytes from 10.138.27.98: icmp_req=2 ttl=54 time=54.2 ms
64 bytes from 10.138.27.98: icmp_req=3 ttl=54 time=21.2 ms
64 bytes from 10.138.27.98: icmp_req=4 ttl=54 time=70.5 ms
64 bytes from 10.138.27.98: icmp_req=5 ttl=54 time=19.8 ms
64 bytes from 10.138.27.98: icmp_req=6 ttl=54 time=95.9 ms
^C
--- 10.138.27.98 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 19.867/51.443/95.933/26.713 ms
```

Figure 3.2 Measure of the number of hops with ping

CHAPTER 4. PREVIOUS SETUPS

To know how the network will behave, it is important to design a setup similar to the environment where the tests will take part. In this way, it is possible to do all the necessary probes for tuning the tests that will be done at the real network. For doing this, two nodes have been configured with the necessary characteristics to do the tests properly.

4.1. Scenario

To configure the preproduction scenario some decisions have to be taken. The first decision is to choose the virtualization software that will be used. For this case, Virtualbox fits perfectly for the needs required. Virtualbox is an x86 virtualization software package developed by Sun Microsystems. It is distributed under either the GNU GPL or a proprietary license with additional features. Once the software is installed and running in the PC two virtual machines are configured with an internal network that allows seeing each other and being able to communicate.

The second decision is to choose the operative system. Due to Guifi.net network uses Openwrt [see section 2.1] in its nodes, the Linux distribution Openwrt Backfire 10.03.1 have been installed and configured to perform as a node. This OS is specially developed to work in ad hoc wireless networks and is very typical find it in routers working as AP.

Once the virtualization software and OS is selected is the moment of set the network configuration of each node editing the file “network” in the directory “etc/config/” with the tool of text edition *vi*. The configuration will be as follows:

config interface lan

option ifname	*eth1*
option proto	*static*
option ipaddr	*192.168.56.235*
option netmask	*255.255.255.0*

Any address between 192.168.56.2 and 192.168.56.254 are allowed. The second node is configured with the IP address 192.168.56.13.

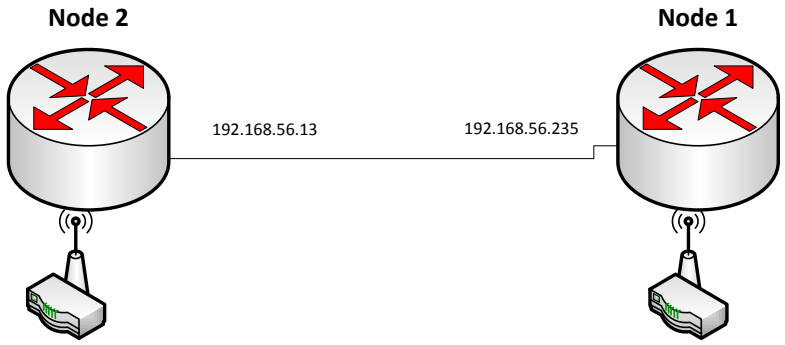

Figure 4.1 Scheme of the configured Openwrt nodes

The specifications of the host PC are the followings:

Table 4.1. Host PC specifications.

Component	
Processor	Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz (8 CPUs)
Memory	4096MB RAM
OS	Windows 7 Home Premium 64-bit
Wireless	Intel(R) WiFi Link 1000 BGN

4.2. Software set up

Once the scenario is prepared is time to set up the software part. In this term there are two main tools that have to run properly, iperf and ping. First of all the availability of the tools for the distribution of Openwrt has to be consulted in http://downloads.openwrt.org/backfire/10.03/brcm47xx/packages/. Then they have to be tested in the settled scenario.

4.2.1. Iperf test

For the iperf test a server have to be configured in one extreme with the option iperf –s to receive TCP traffic or iperf –s –u to receive UDP traffic. When the server is up it is possible to connect to it with the client indicating the servers IP. In the next capture it is possible to see the communication between node 1 (client) and node 2 (server) with iperf.

Server (receiver):

```
$ iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 107 KByte (default)
------------------------------------------------------------
[ 3] local 192.168.56.13 port 5001 connected with 192.168.56.235 port
65299
[ 3] 0.0-10.0 sec 1.25 MBytes 1.05 Mbits/sec 0.008 ms 0/ 893 (0%)
```

Client (transmitter):

```
$ iperf -u -c 192.168.56.13 -b 1M
------------------------------------------------------------
Client connecting to
192.168.56.13, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 9.00 KByte (default)
------------------------------------------------------------
[ 3] local 192.168.56.235 port 65300
connected with 192.168.56.13 port 5001
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-10.0 sec 1.25 MBytes 1.05 Mbits/sec
[ 3] Server Report:
[ 3] 0.0-10.0 sec 1.25 MBytes 1.05 Mbits/sec 0.003 ms 0/ 893 (0%)
[ 3] Sent 893 datagrams
```

After checking the correct behavior of the tool in the Openwrt environ, the specific tests have to be determined. To analyze the TCP flows of the network it is interesting to see how the system behaves with different sizes of TCP windows. With this method the efficiency in each case can be determined. Taking into account that the maximum window size is 65.535 bytes two different sizes has been selected, 64KB and 8KB, to acquire data from one large window and a short window. Windowing allows TCP protocol to send a determined amount of data without receiving the ack whenever there weren't retransmissions before. In case of retransmission the window decrease and will be increasing until the maximum value initially settled.

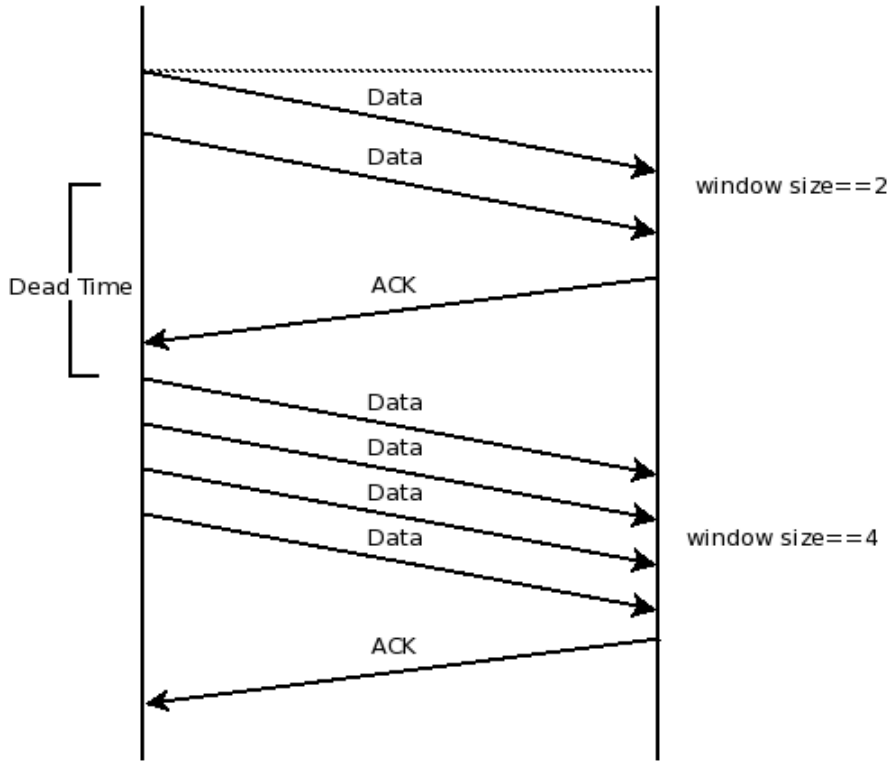

Figure 4.2 Segment of increasing of the windowing when an ack arrives

As it can be seen in figure 4.2 waiting for the reception of the ack adds a dead time to the transfer that increments the throughput. In that sense, it is important to see how the system handles this problem with the test suggested.

The two tests for the TCP protocol with iperf are the followings:

Iperf with short window:

```
for line in $(cat IPLIST.txt);
do

IPERF2=`iperf -c $line -w 8KB –p 80 | grep 'bits'`

THR2=`echo $IPERF2 | awk '{ print $7}'`

echo $line >> TCPsthr.txt

echo $THR2 >> TCPsthr.txt

done
```

Iperf with large window:

```
for line in $(cat IPLIST.txt);
do

IPERF3=`iperf -c $line -w 64KB –p 80 | grep 'bits'`

THR3=`echo $IPERF3 | awk '{ print $7}'`
```

```
echo $line >> TCPlthr.txt

echo $THR3 >> TCPlthr.txt

done
```

Where IPLIST are the IPs of the different iperf servers. With these two tests it is possible to acquire the TCP throughput of the communication with the servers.

4.2.2. Ping test

Not all the nodes in Guifi.net network have an active iperf server so it is not possible to calculate the throughput in this way. So, the method selected to calculate this rate in the proxy servers is with the tool ping. As seen in section 2.3, ping allows sending data of different sizes and it calculates the total RTT, the TTL and the rate of packet loss.

There are some methods to calculate an estimate throughput taking into account the amount of data sent and the RTT resultant of various probes. For this project the test of three different sizes of data transferred has been selected.

In these test, three different sets of 10 pings has been done. With the first set, pings of 4000B are sent. In the second one, the size of the ping is of 10000B. The size of the pings of the third set is of 14000B. The average RTT of the three sets are saved in the variables RTT1, RTT2 and RTT3. Once the different RTT are acquired the throughput is evaluated with the next formula:

$$\text{Throughput} = (\Delta B)/ (RTT_y - RTT_x) \quad \textbf{(4.1)}$$

Where ΔB is the difference of bytes transmitted between two of the tests done. This throughput is calculated three times with the three probes acquired and then an average of the three results is done. In figure 4.3 there is a graphic illustration of this test, where the pendent of the line that joins the three points is the throughput calculated.

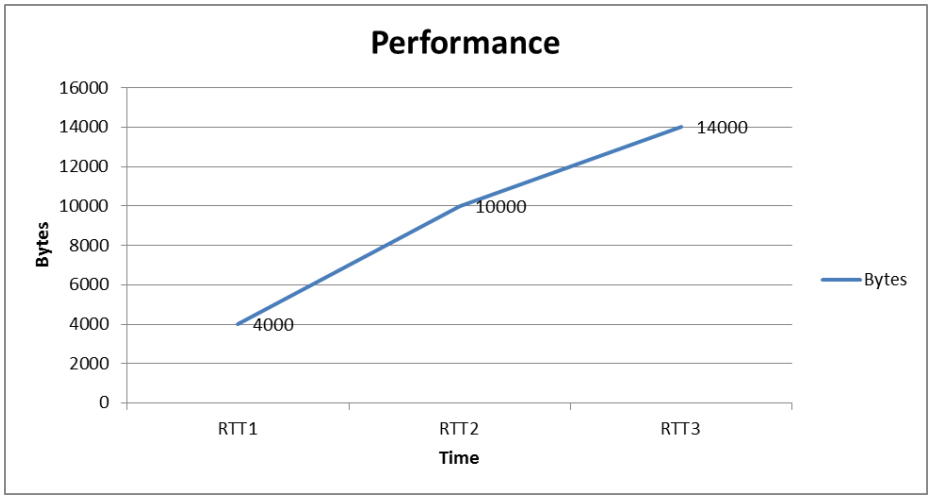

Figure 4.3 Result of the RTT calculated by the three points method

The next algorithm does the three sets of ten pings with a different payload and applies the formula 4.1 to return the estimate throughput and it stores the result in a file. It also stores the packet loss.

```
for line in $(cat IPLIST.txt)
do

PING1=`ping -c 10 -s 4000 $line | grep 'received\|rtt'`

LOST1=`echo $PING1 | awk -F',' '{ print $3}' | awk '{ print $1}'`

RTT1=`echo $PING1 | awk -F'/' '{ print $5}' | awk -F'.' '{ print $1}'`


PING2=`ping -c 10 -s 10000 $line | grep 'received\|rtt'`

LOST2=`echo $PING2 | awk -F',' '{ print $3}' | awk '{ print $1}'`

RTT2=`echo $PING2 | awk -F'/' '{ print $5}' | awk -F'.' '{ print $1}'`


PING3=`ping -c 10 -s 14000 $line | grep 'received\|rtt'`

LOST3=`echo $PING3 | awk -F',' '{ print $3}' | awk '{ print $1}'`

RTT3=`echo $PING3 | awk -F'/' '{ print $5}' | awk -F'.' '{ print $1}'`

PING4=`ping -c 10 -s 4000 $line | grep 'ttl'`

HOP=`echo $PING4 | awk '{ print $6}'`

TTM1=`expr $RTT2 - $RTT1`

TTM12=`expr 96000 / $TTM1`

TTM2=`expr $RTT3 - $RTT1`

TTM13=`expr 160000 / $TTM2`
```

```
TTM3=`expr $RTT3 - $RTT2`

TTM23=`expr 64000 / $TTM3`


TTM=`expr $TTM12 + $TTM13`

TTM=`expr $TTM + $TTM23`

TTM=`expr $TTM / 3`


echo $line >> RTTpingthr.txt

echo $TTM >> RTTpingthr.txt

echo $line >> RTTpinghop.txt

echo $HOP >> RTTpinghop.txt

echo $line >> RTTpinglost.txt

echo $LOST1 >> RTTpinglost.txt

echo $LOST2 >> RTTpinglost.txt

echo $LOST3 >> RTTpinglost.txt

done
```

The RTT to calculate the one way latency will be stored separately to fit with the size required for each test.

Once the different tests have been probed in a preproduction environment and the results have been the initially spectated the tests are ready to be executed at the real network. To implement this experiment there are some previous configuration steps to follow.

4.3. Steps before the tests

First of all the client PC where the test will take part have to be settled. This PC will be connected to a node administrated by Confine located at the UPC University. This client PC will be accessible via SSH to be administrated. The OS installed is Ubuntu Desktop with the graphic interface disabled. Once the PC has been installed in the node all the elements in Guifi.net network are accessible so the probes to the other nodes can be done.

4.3.1. Calculus of the distances between nodes

One important aspect to take into account is the distances between the node where the probes are done and the different destinies. The node is located at the following geographic decimal degrees coordinates:

Latitude: 41,389833
Longitude: 2,113004

Figure 4.4 Place where the test node is located

To measure the distance between the test node and the different nodes where the iperf servers and proxies are located the coordinates of each one are needed. This information is available in the Guifi.net web, where every node is registered and the coordinates are listed. To calculate the distance in kilometers the next three formulas are applied:

$$P = \sin(\text{latitude 1}) * \sin(\text{latitude 2}) + \cos(\text{latitude 1}) * \cos(\text{latitude 2}) * \cos(\text{longitude 1} - \text{longitude 2}) \quad \textbf{(4.2)}$$

$$D = \text{ACOS}(P) \quad \textbf{(4.3)}$$

$$Km = D * 111{,}194 \quad \textbf{(4.4)}$$

Latitude 1 and longitude 1 corresponds to the decimal coordinates of the node and latitude 2 and longitude 2 corresponds also the decimal coordinates from the different nodes.

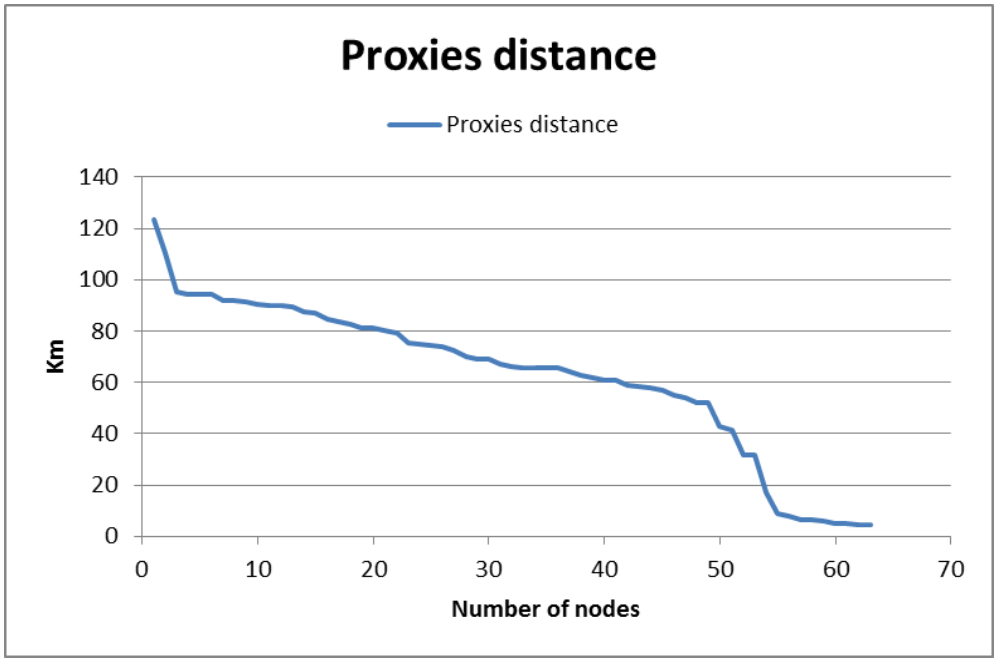

Figure 4.5 Distribution of the distances of the different proxies respect the test node

In figure 4.5 it is possible to see the order of distances of the proxies published in Guifi.net. They are placed between 123km and 4km of distance. The physical topology is an important aspect to analyze due to in some of the cases can have an impact in the results of the tests done.

4.3.2. Calibration of the system with iperf

As far as the ping test is not a truly reliable method to calculate throughput, due to this protocol is not intended to do this duty, a calibration of the method have to be done. This adjust consists in doing the ping test to a node with iperf installed and running as server. Then, the iperf test designed in section 4.2.1 will be done to compare the results and estimate the possible error that the network can add.

The reality of the network reveals that 13 of the 24 iperf servers listed in Guifi.net are reachable with ping and 7 of these 13 nodes have iperf running in server mode to do the test. Nevertheless, it is enough to measure the error that the ping based method add to the results.

The probe has been done to the nodes listed in table 4.2 with the next command that evaluates the TCP throughput:

```
iperf -c “Iperf server IP” –p 80
```

Four sets of tests have been acquired. And the results have been as follows.

Table 4.2. Results of the throughput of the ping and iperf tests and the correspondent percentage of error based on the maximum value

Iperf Servers	Iperf (kbps)			Ping (kbps)			Percentage of error
	Max	Min	Standard deviation	Max	Min	Standard deviation	
10.138.96.3	2330	2140	21,21	1816	1718	48,49	22%
10.138.33.130	2280	1980	149,75	3252	2821	199,69	43%
10.138.20.101	7880	7400	198,58	5809	5654	64,23	26%
10.138.160.98	3850	3780	31,62	3891	3689	84,29	1%
10.138.10.2	4000	3770	102,43	4668	4536	60,60	17%
10.140.150.18	1240	1040	85,24	1189	1009	73,76	4%
10.138.97.2	2130	1920	90,55	1934	1823	46,63	9%

The probes have been done one after the other four times to minimize the difference in the conditions for each test. That leads to an average difference of 645 kbps and an error of the 17% for the maximum values acquired. This error appears due to the different way of processing the traffic for each protocol by the intermediate nodes. The average distance between the client and the iperf server is of 55.800 meters and 10 intermediate nodes.

Table 4.3. Intermediate nodes and distance from the client node and the different iperf servers

Iperf Servers	Hops	Distance (km)
10.138.96.3	9	57,111
10.138.33.130	11	67,081
10.138.20.101	7	70,080
10.138.160.98	11	60,922
10.138.10.2	8	58,827
10.140.150.18	13	33,532
10.138.97.2	11	43,050

This test provides a magnitude of the error. As a rule, ping traffic is used to be the less priority one, so, in normal conditions, ping throughput will be the most restrictive one comparing with other protocols, but there are many other factors that influence the final results and provoke these trending swaps. These factors are the congestion of the network in punctual moments and the rules and capacities of intermediate and final nodes to handle the arriving data, for instance, some nodes gives more priority to control traffic as ICMP when they are congested.

With these results an error of 17% in the measures of the throughput to the proxies will be assumed, taking always the most optimistic case.

4.3.3. Evaluation of the TCP window size with iperf

Another interesting aspect of the network is to evaluate the behavior of the system against the changes of the TCP window. By default, iperf has a window size of 21Kbytes. As seen in section 4.2.1 a window size of 8KB and another of 64KB are going to be analyzed to see the difference between the two scenarios.

Table 4.4. Throughput results of changing the TCP window size with iperf.

Iperf server	Window size of 64KB (kbps)	Window size of 8KB (kbps)	Percentage of difference
10.138.96.3	3000	748	75%
10.138.33.130	1224	915	25%
10.138.20.101	5300	1940	63%
10.138.160.98	3270	695	79%
10.138.10.2	4160	1040	75%
10.138.97.2	2250	1420	37%

The average of change is a 59%, so, the performance of the biggest TCP window size is more than the double of the shortest TCP window. So in this system it is possible to see that the efficiency will be higher with a large TCP window. Nevertheless, there is not a substantial difference between the default TCP window size used in calibration tests and the large size window, so it has been decided to keep the default window for these tests due to the penalization for retransmission with a large window increases.

4.4. Tests to be done

The selection of the proxy that provides a better performance for the user can be evaluated with the six next indicators:

- Geography (in km)
- Number of hops (intermediate nodes)
- Client to Proxy latency (Ping)
- Client to Web Server latency (Wget)
- Client to Proxy throughput (Ping)
- Client to Web Server throughput (Wget)

Geography is the geographical distance in km from the node where the user is connected to the different proxy servers available in the network. These distances are explained in section 4.3.1.

Figure 4.6 Example of the geographical distance calculated between a proxy and a client

Three different groups of proxies have been detected depending on the grade of availability of each one. The first group of 67 is the federated up proxies listed in Guifi.net web. The second one is the 29 proxies of these 67 that are reachable with ping from the client. The last group is compound by the 17 proxies of the second group that allows a connection with authentication from the client node. Since now, these groups will be named as federated proxies, reachable proxies and available proxies.

Number of hops represents the number of intermediate nodes that the data flow has to cross to arrive from the client to the proxy server. Each intermediate node decrements the TTL field from ping response in one, so it is possible to determinate the total number of hops in the client.

Figure 4.7 Example of the intermediate node hops between a proxy and a client

In the example of figure 4.7 it is possible to detect three intermediate nodes between the client and the proxy server. This will be the real path that the data flow follows to reach the destiny; most of the times this path has a large difference with the geographical distance. Each intermediate node adds time to the transmission due to they have to process the routing information and they could be congested. Also, when the amount of nodes where information has to turn over is a large number, the probability of having a poor rate of packet loss increases.

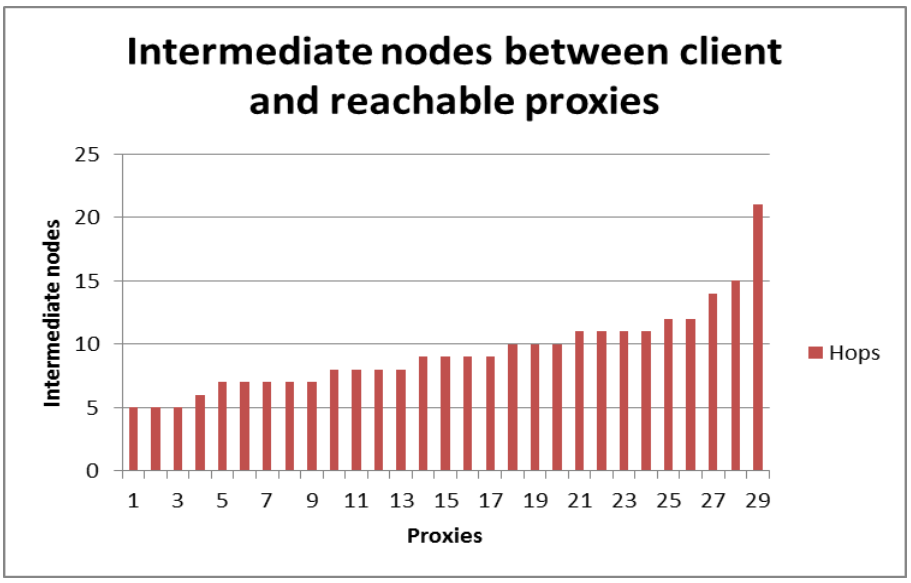

Figure 4.8 Graph of the intermediate nodes in the reachable proxies

Client to Proxy latency reveals the time that a packet spends to arrive from the proxy server to the client. This time acquired with ping will determinate the impact in time of the sector between the client and the proxy and it is one of the parameters to decide which server provides a better connection.

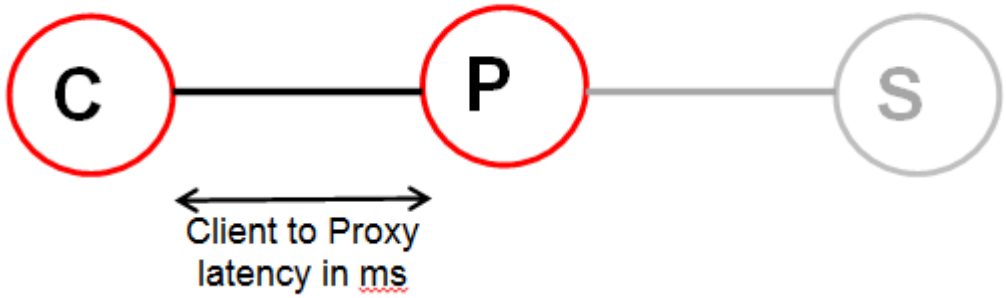

Figure 4.9 Latency between the client and the proxy calculated with ping

Proxy to Web Server latency consists in the time that the system spends in download a certain component of a webpage calculated with wget. For the porpoise of this project, an image of Google of 19kB is going to be downloaded from the servers where this content is stored going throw the different accessible proxies to compare the results.

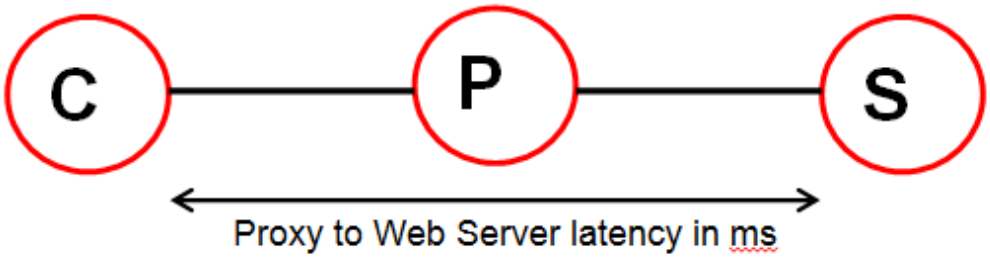

Figure 4.10 Latency between the client and the content server calculated with wget

The next two commands determine the intermediate proxy by parameter with wget to download Google's logo:

"export http_proxy=http://USER:PASSWORD@PROXYIP:3128/"

"wget http://www.google.es/images/srpr/logo4w.png"

wget reports the time spent to download the images, the size and the throughput.

The times to the connections to the servers going throw the different proxies can vary, so, it is important to quantify this variation to evaluate the impact. Also the ping and wget latencies will be compared to determine which stretch is more restrictive, this comparison can be critical to choose one proxy or another or if one of the latencies is simply irrelevant.

Client to Proxy throughput will be calculated with the three-points method with ping explained in section 4.2.2. This method calculates the average throughput reached between the client and the proxy server selected.

Proxy to Web Server throughput will be acquired with wget software explained in section 2.4. This throughput includes all the communication between the client and the web server where the content is sited passing through the proxy. This measure gives an idea of the total throughput of the entire process of downloading data from the Internet and comparing with ping throughput can determine where the bottleneck is in the system.

It has to be clear that when the “Client to Proxy” stretch is evaluated the calculus are done in the IP layer due to ping is an ICMP based protocol meanwhile the “Proxy to Web Server” stretch uses wget to acquire the measures through HTTP, so it depends on the TCP layer. These differences can produce some disarrangement in the measures due to the treatment in the nodes of the network is not the same for each layer. That is the principal reason to do the calibration process of the tool ping with iperf which also is on the TCP layer.

CHAPTER 5. NETWORK RESULTS

5.1. Test results

Once it is clear the tests that have to be done and the figures that have to be identified in the system is time to implement the tests in a real environment. As it has been explained in chapter 1, these tests will take part in Guifi.net network which principal characteristic is that it is a free ad hoc network. That means that it can change its topology. Also the owners of each node can apply different policies to their devices that affects to the behavior of the system and the result of the tests.

To do the tests 67 federated proxies have been detected at the official Guifi.net page located all around Catalonia. 29 of these 67 proxies are accessible from the node where the tests are being done. There are two main reasons for proxies to be inaccessible, the first one is that the node where the proxy is located is down and it is not communicated in the web. The second main reason is that the node where the proxy service is located is in an unreachable part of the network. This last case happens due to not all the entire network is interconnected and can appear some sets of isolated nodes.

From this group of 29 reachable proxy servers 17 allows to the node to be connected with authentication. So, the analysis will be done to these 17 nodes available.

5.1.1. Scenario test results

The first aspects that will be evaluated are the physical properties as geographical distance and the number of intermediate nodes between the client node and the different proxy servers. The distance will always remain equal in time due to the nodes are located in the same place. The number of intermediate nodes can change due to the ad hoc nature of the network. Nevertheless, these changes are not often and for this study will be assumed as a static figure like distance is.

The acquisition of the distance values is explained in 4.3.1. In this case only the available proxies that provide the service to the client node have been taken into account. In figure 5.1 it is possible to see the distance between the client node located in Campus Nord and the different proxy servers evaluated.

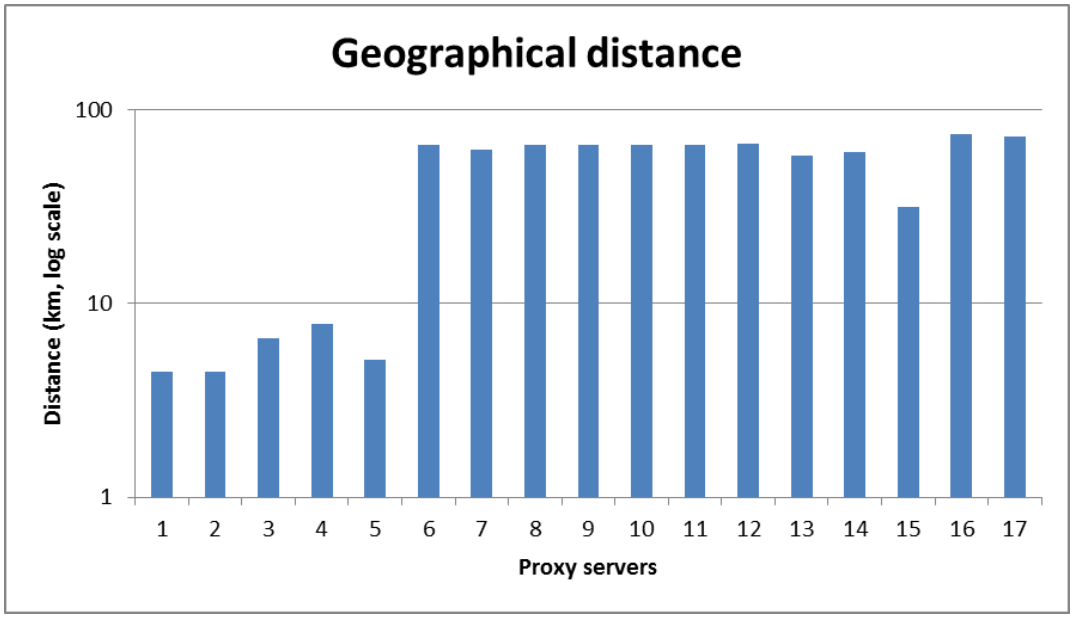

Figure 5.1 Distance between client node and the different available proxy servers

Five of the nodes where the proxy servers are hosted are located at less than 10 km. That means that almost the 25% of these nodes are installed in the metropolitan area of Barcelona while the 75% of the nodes are located between 30 and 75 km what makes that all the nodes are in the Barcelona province. Therefore, the closers proxies are the more accessible for the clients because of the characteristics of the network.

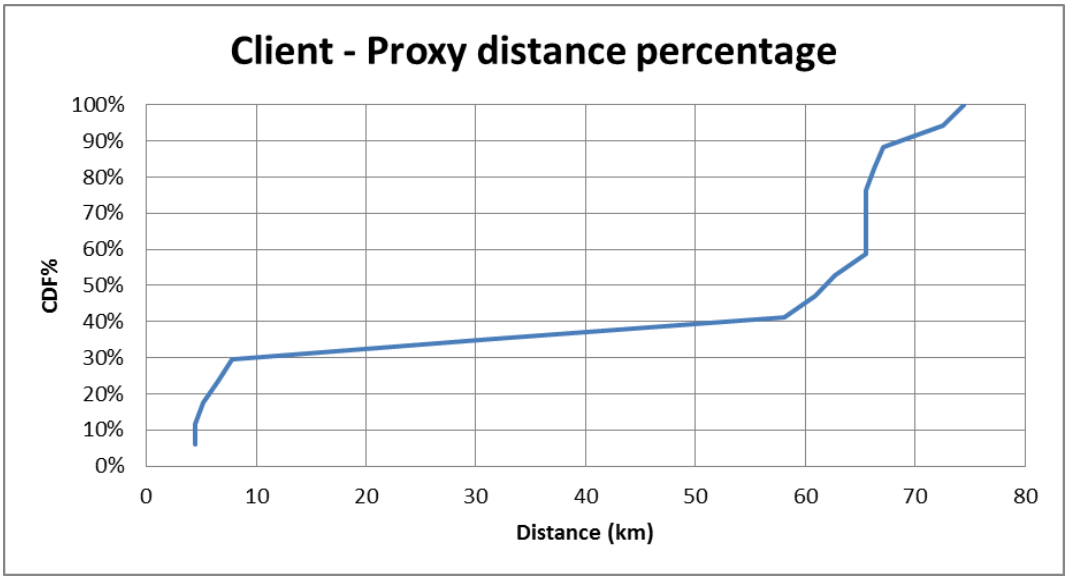

Figure 5.2 Percentage distribution of the distance between the client and the proxy

Once the geographical distance has been acquired is the moment to analyze the number of nodes that separates the client node and the proxy servers. The acquisition method of this figure is explained in section 2.3.

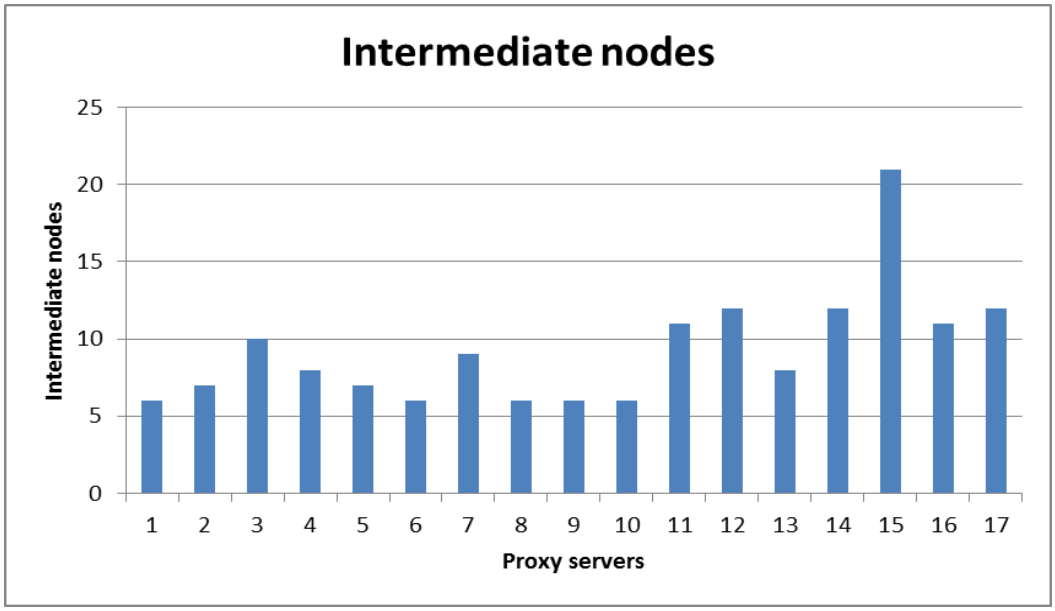

Figure 5.3 Intermediate nodes between the client and the proxy server

As it can be seen in figure 5.3 all the nodes located between 6 and 12 hops of distance with the exception of node 15 that has 21 intermediate nodes.

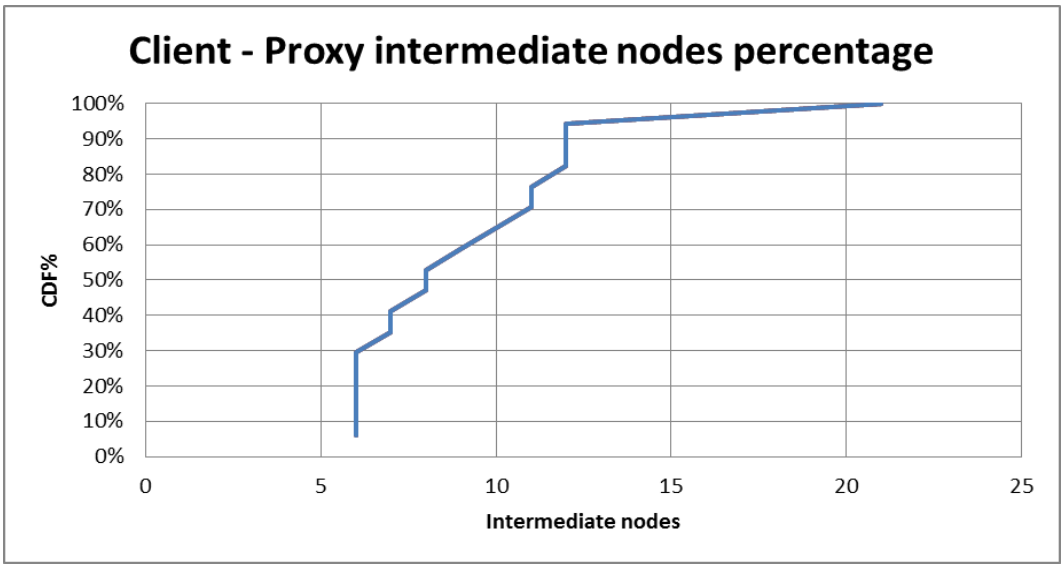

Figure 5.4 Percentage cumulative distribution function of the intermediate nodes between the client and the available proxy servers

5.1.2. Latency test results

Another interesting issue evaluated is the response time that the final user experiments. As explained in section 4.4 this measure has been taken in two segments of the communications to see what is the most restrictive one and detect bottlenecks. The first segment has been measured with the tool ping and it includes the link between the node where the proxy server is located and the client node. The second segment comprises the web server where the file that is going to be downloaded is and the client node link.

These values have been taken with the control tool ping. Ping uses ICMP protocol that gives lower value results than TCP in terms of latency. Therefore, an adjustment of the results acquired, explained in section 4.3.2, has been applied. This makes an uncertainty of the 17% taking the most optimistic case for the measures. This calibration have to be done due the most common protocol used by proxies is TCP.

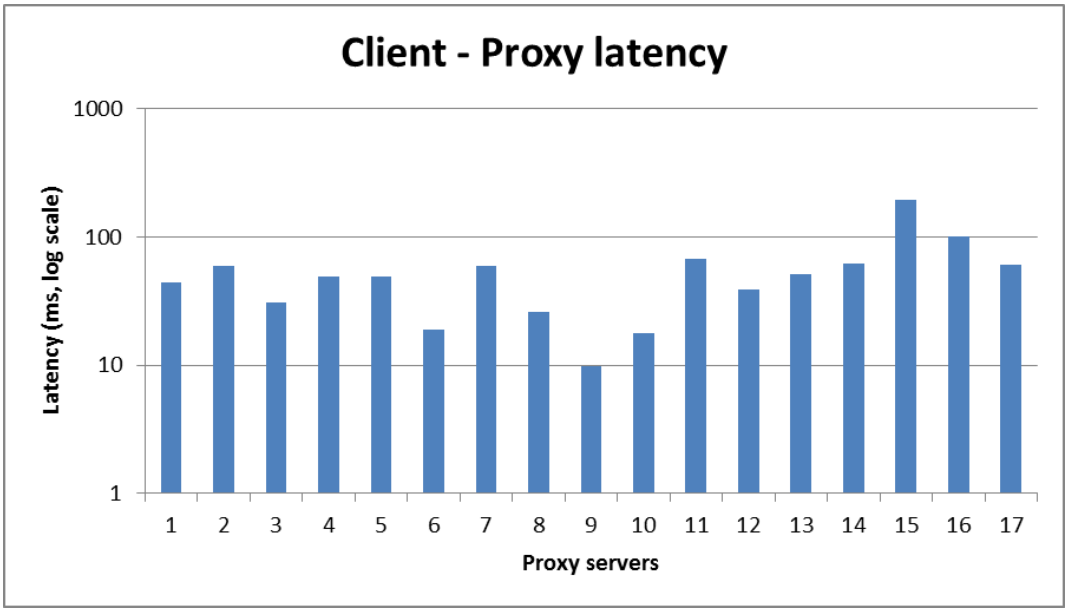

Figure 5.5 Latency between the client and the available proxy servers

Figure 5.5 shows the latency calculated for each proxy. All the values are between 10ms and 100ms excepting node 15 that has a value close to the 200ms. The 50% of the proxy servers offers latency below 50ms and near the 85% of the nodes are below the 70ms that are acceptable times for user's perception.

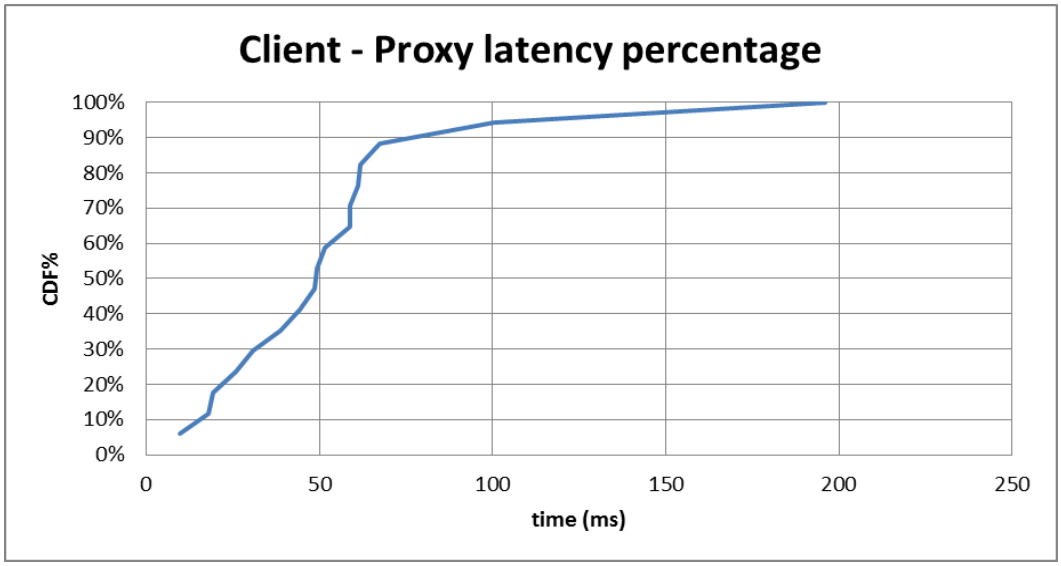

Figure 5.6 Percentage of latency between the client and the available proxy servers

The next latency values have been calculated with wget tool. In this manner it is possible to evaluate the time spent for a small file to be downloaded from the web server.

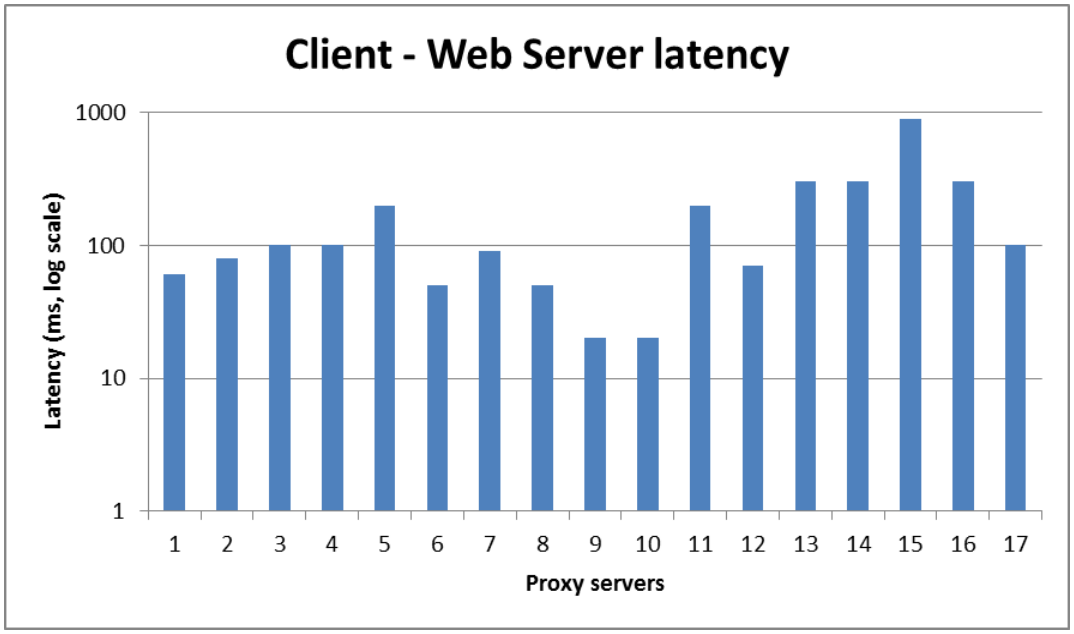

Figure 5.7 Latency between the client and the web server

In figure 5.7 it is possible to determine which proxies provides better latency values to the client. 40% of the nodes give a value lower than 100ms which is an acceptable value.

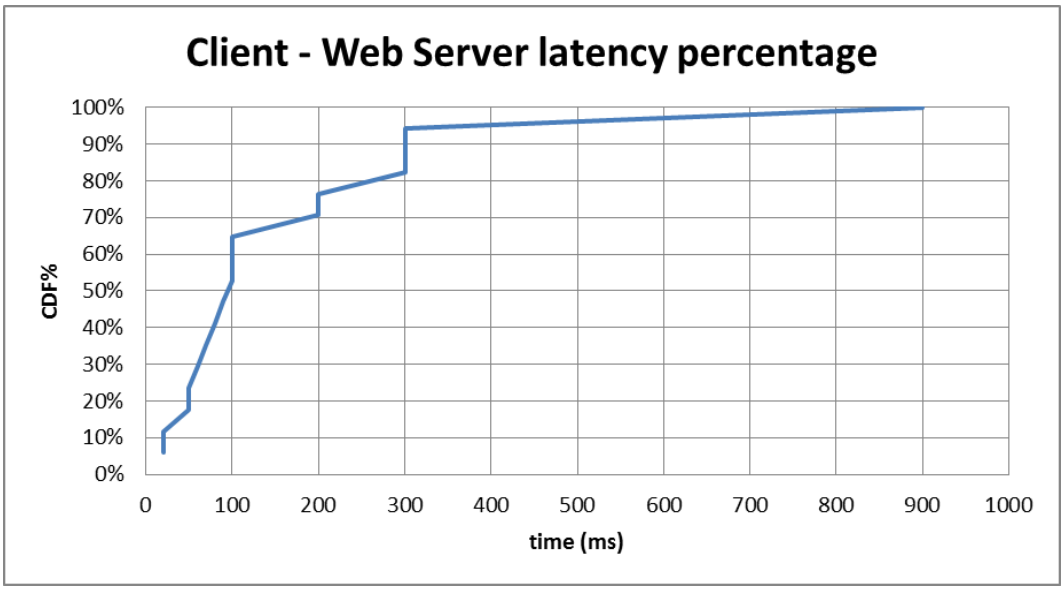

Figure 5.8 Percentage cumulative distribution of latency between the client and the web server

For a better point of view of what is happening in the entire communication it is advisable to determine which stretch is consuming more time. In figure 5.9 the response time or latency have been plotted separating it into two parts, the time spent between the client and the proxy and the latency added by the web server to the proxy communication for each proxy available.

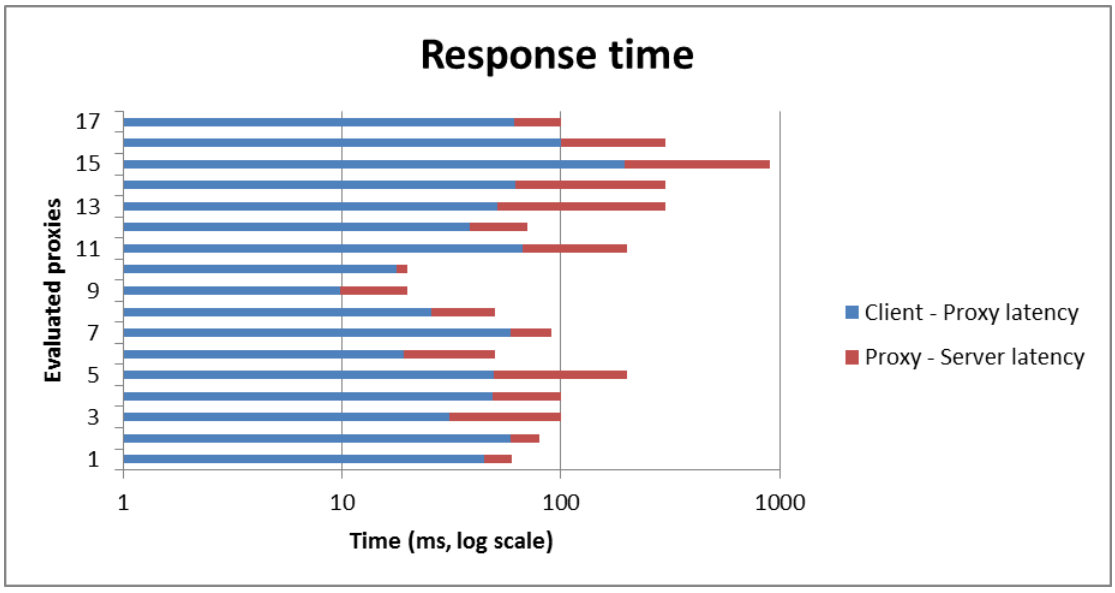

Figure 5.9 Overall latency of the communication between client and web server discriminated by steps

In figure 5.10 the percentage distribution of latency is listed for each selected proxy. In seven cases the time spent between the client and the proxy is higher than the time spent by the proxy to web server communication. Nevertheless, the average of the seventeen probes tells that the 54% of the time is spent by the communication between the steps from the web server to the proxy against the 46% spent in the rest of the link. So, in terms of latency, it is more probably to find bottlenecks in the proxy to web server link. That will minimize the impact of the election of the proxy with a best time response in the first step, although the percentages are very similar.

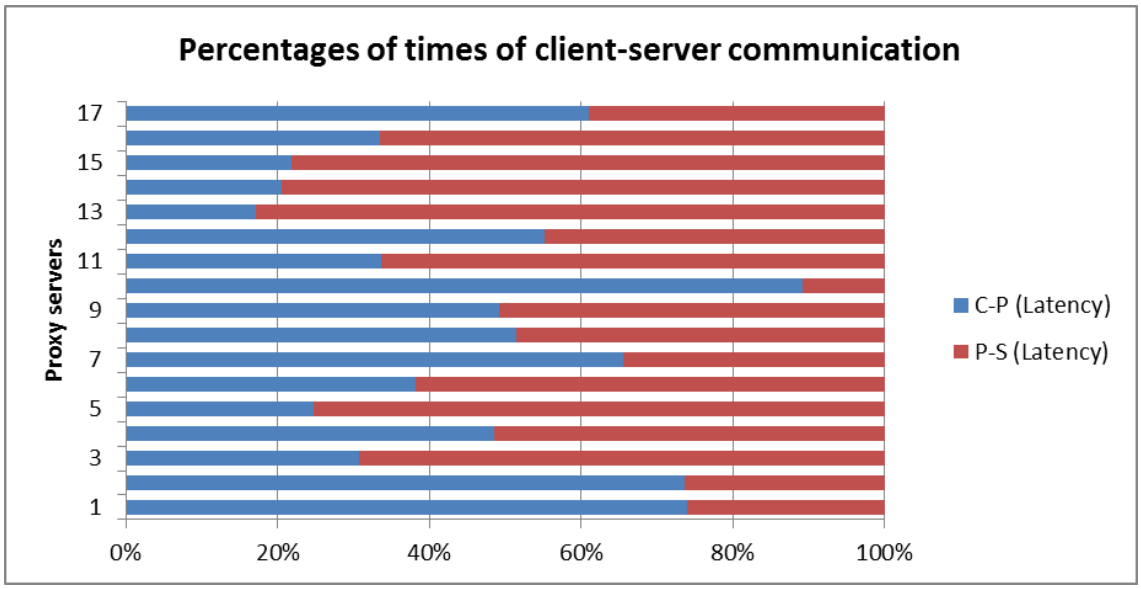

Figure 5.10 Percentage of latency of the steps in the communication between the client and the web server

5.1.3. Throughput test results

Throughput is another essential figure to be acquired. This data allows determining in which rate the packets will be sent in the medium independently of the size. For the acquisition of the data in these tests the tools ping and wget have been used.

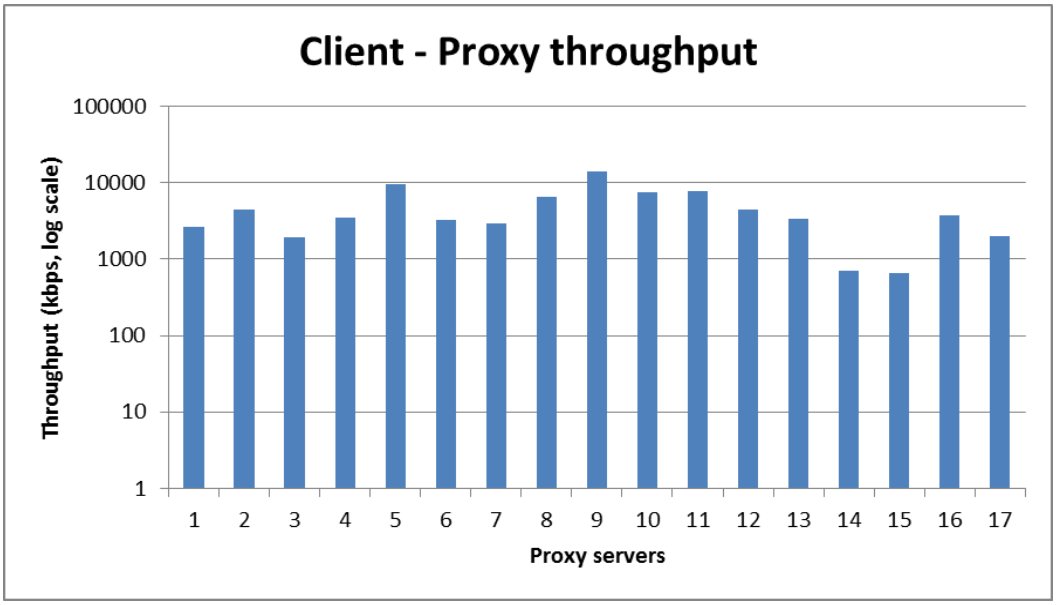

Figure 5.11 Throughput in the client to available proxy servers stretch

As seen in figure 5.11 the real throughput between the client and the proxy servers are from the 600kbps to the 14Mbps. The performance of the 40% of the proxies above the 4Mbps is quite good and it allows the client to have a great rate to be connected to the Internet via these proxies.

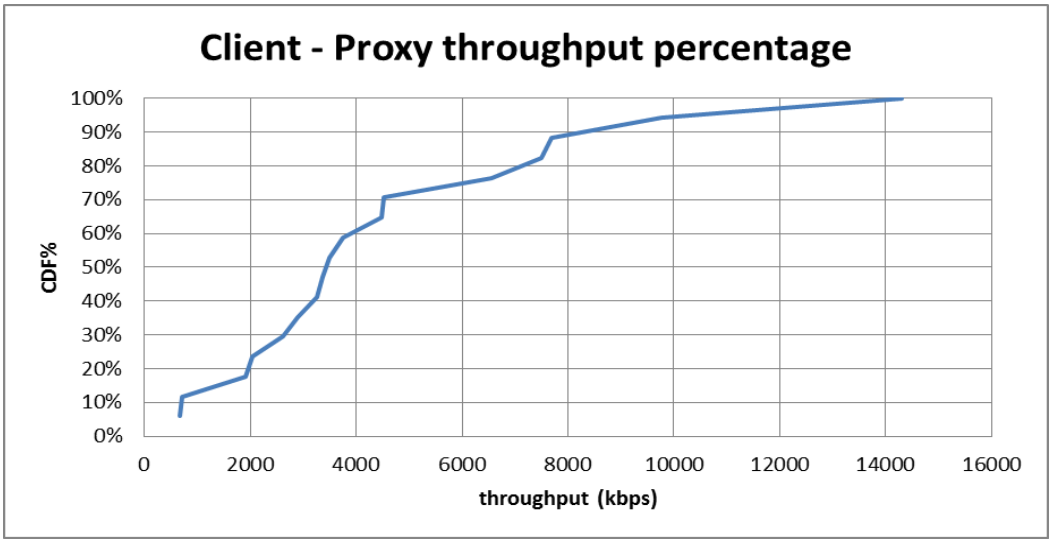

Figure 5.12 Percentage cumulative distribution of throughput between the client and proxies

These throughput values disagree in some of the cases from the bandwidth published in Guifi.net for each proxy server. Although it is a normal situation that throughput has a lower value than the bandwidth produced by several factors, it is not possible to have a throughput higher than the bandwidth. This variation

can be produced by improvements in the network that have not been updated in the web.

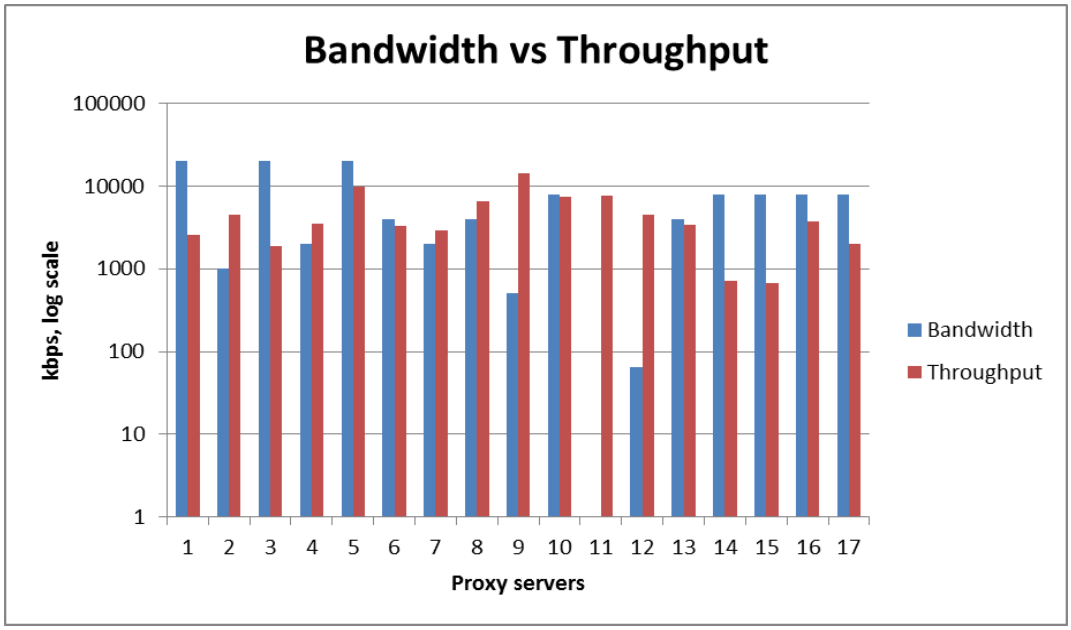

Figure 5.13 Comparison of the bandwidth published in Guifi.net and the throughput measured for each available proxy

In figure 5.14 it is possible to see the overall throughput of the communication measured with wget. These values vary between the 700kbps to the 8Mbps with an average of 2Mbps what is still a good rate for a normal connection to the Internet.

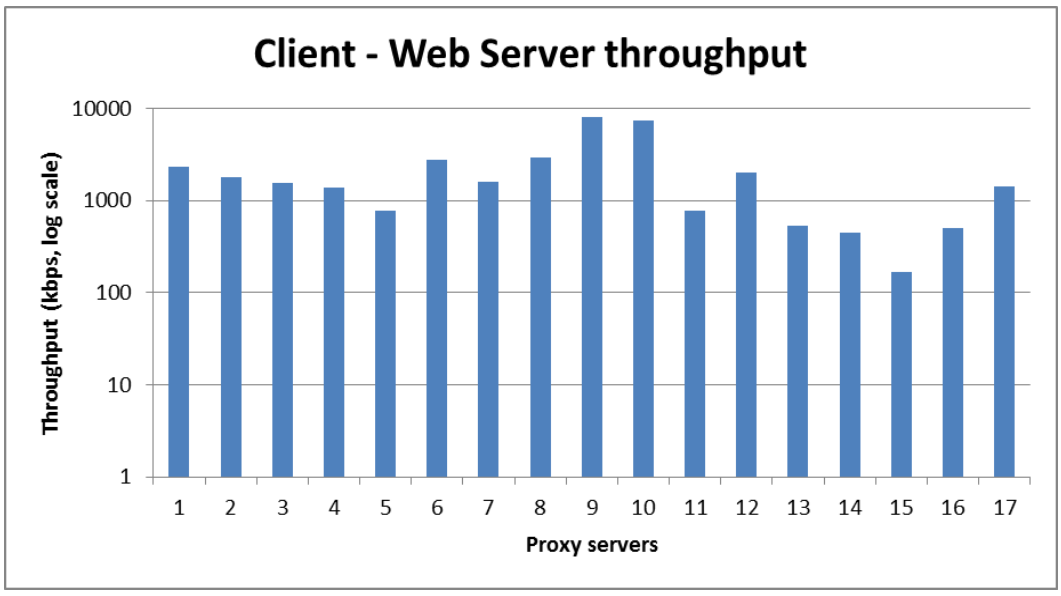

Figure 5.14 Throughput measured between the client node and the web server

Once the client to proxy and the client to web server throughput have been calculated it is interesting to compare this first step with the proxy to web server step. To calculate the throughput of this stretch the next formula has been used:

$$Thr_{P\text{-}S} = 1/ ((2/Thr_{C\text{-}S}) - (1/Thr_{C\text{-}P})) \quad \textbf{(5.1)}$$

As it can be seen in figure 5.15 the proxy to web server step is the most restrictive stretch. The average value of the first part is 4,6Mbps while the second part is about 1,6Mbps, so it is clear that bottlenecks appear in the proxy to web server link, being only the third part.

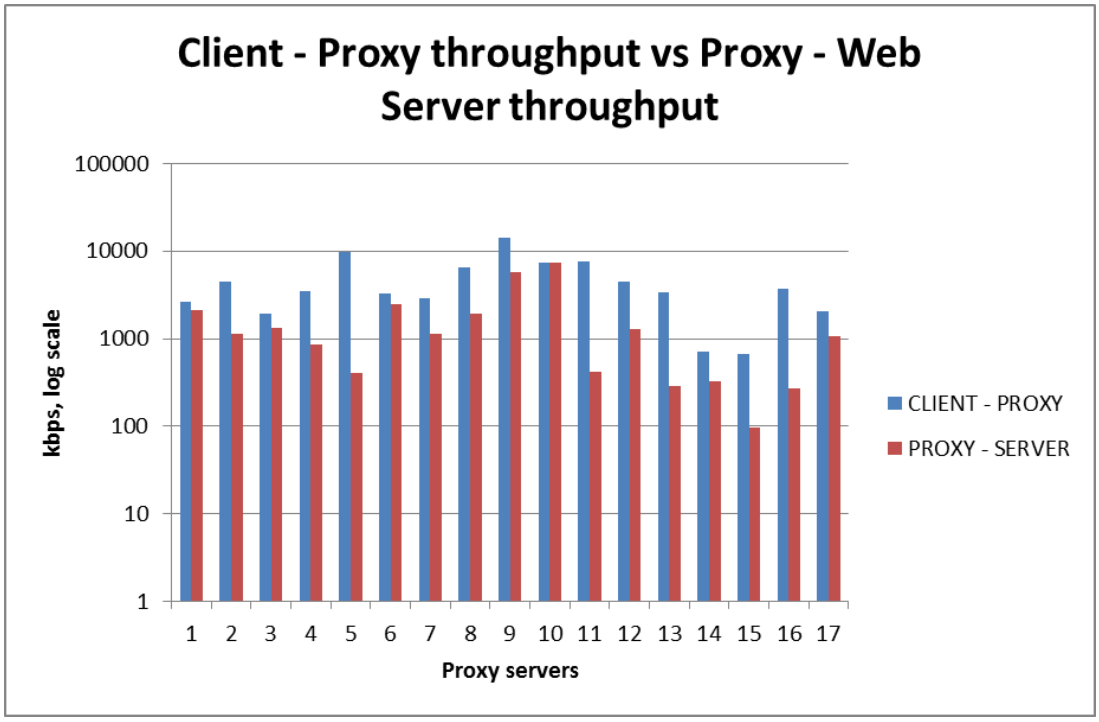

Figure 5.15 Throughput comparisons between the client to proxy stretch and the proxy to web server

5.2. Analysis of the test results

All the figures measured in chapter 5 can describe some rules that influence the proxy selection for a certain client. In this section, the performance of the connections to the different proxy servers will be discussed.

5.2.1. Figures relationship

The first aspect that will be evaluated is the relation of the distance and the number of intermediate nodes with the packet loss rate. As it can be seen in figure 5.16, packet loss increases when the number of intermediate nodes is higher. This phenomenon is produced due to each node adds a probability of failure and when the number of hops is high the probability of packet loss increases. However, the effect of distance in figure 5.17 is not so pronounced although packet loss also increases with the distance. Packet loss directly affects to the throughput due to it provokes retransmissions and it increase the transference time. So, in terms of packet loss it is advisable to choose a proxy server connection with a short number of intermediate nodes in spite of distance.

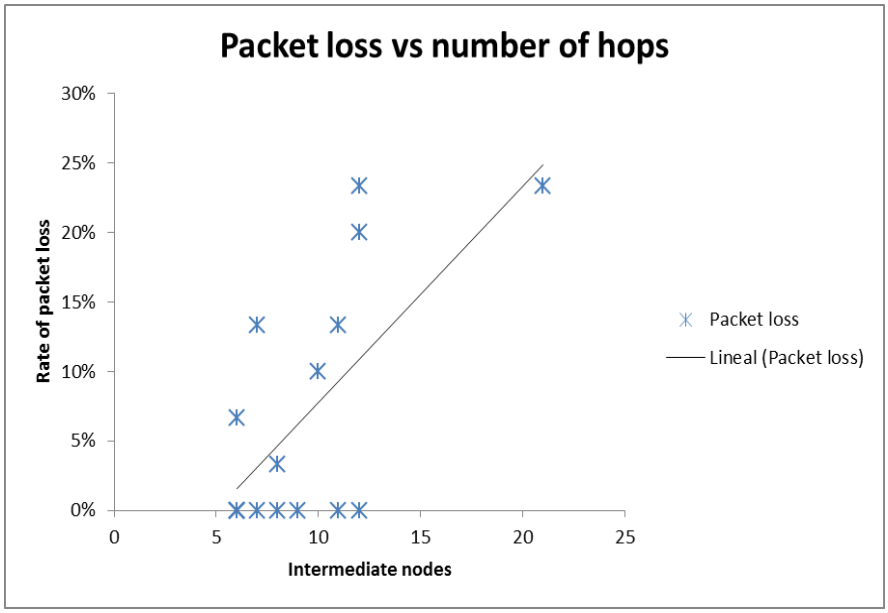

Figure 5.16 Packet loss vs. intermediate nodes relation

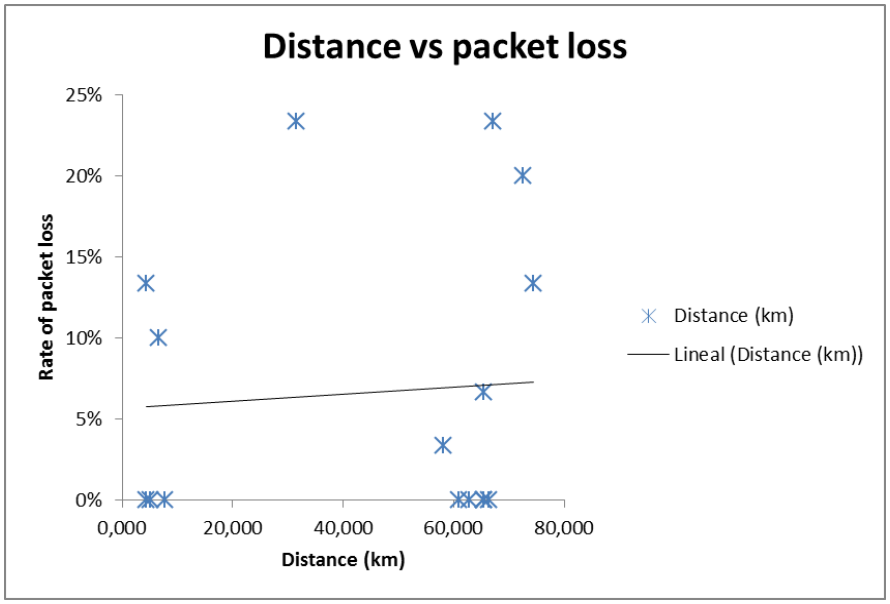

Figure 5.17 Packet loss vs. distance between client and the available proxy relation

Once the impact of the intermediate nodes on the packet loss has been discussed, the impact on the latency will be analyzed. For this experiment it is possible to determine that latency will be lower when the number of intermediate nodes is also low. An increasing of nodes will add time in the communication due to they have to process the routing information and also they can be congested or applying some traffic policies as it can be seen in formula 3.1.

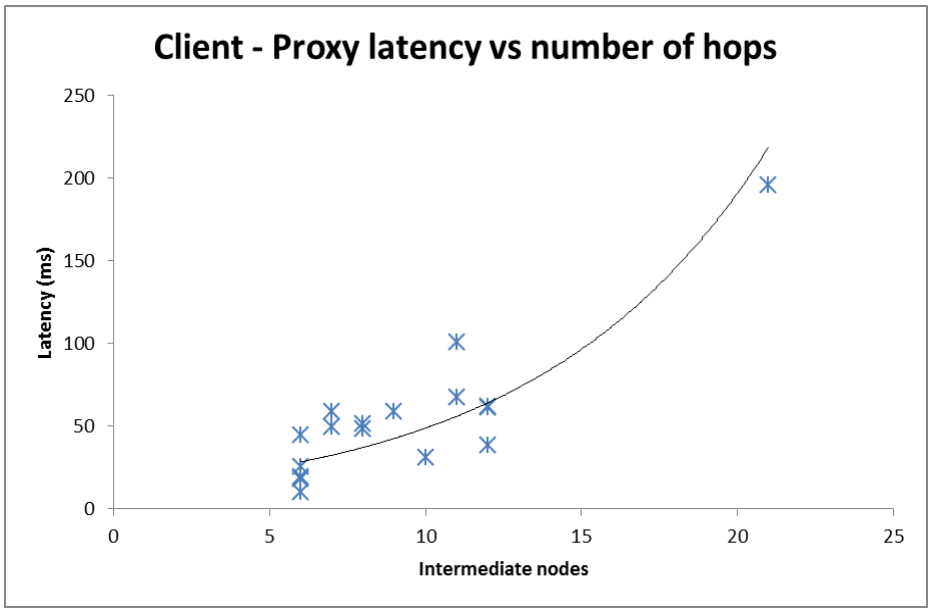

Figure 5.18 Intermediate nodes vs. latency between client and proxy relation

In figure 5.19 it is possible to see that there is a common trend between latency and throughput calculated in the client to proxy link. When latency becomes lower throughput increases because latency has an important influence in TCP throughput. So, it is possible to affirm for this experiment that the overall figures follow a trend i.e. if there is a link with 5 intermediate nodes and another one with 6 intermediate nodes it is more probable that the first one has less latency, more throughput and a lower value of packet loss.

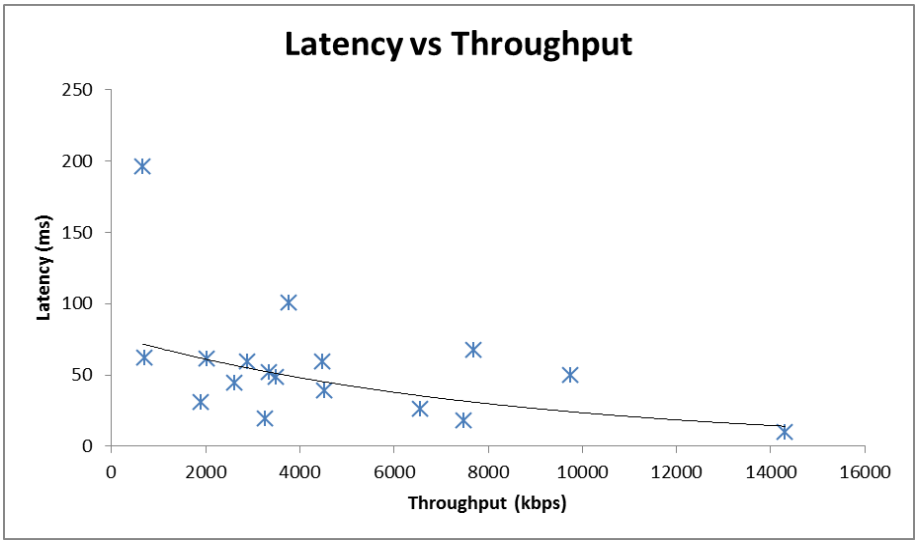

Figure 5.19 Latency vs. throughput between client and proxy relation

Finally, in figure 5.20 it is possible to discern which paths have the intermediate nodes that spend more time handling data. It is possible to see that node 9 spends an average of 1,5ms for each node while the nodes with the poor performance lasts around 9ms.

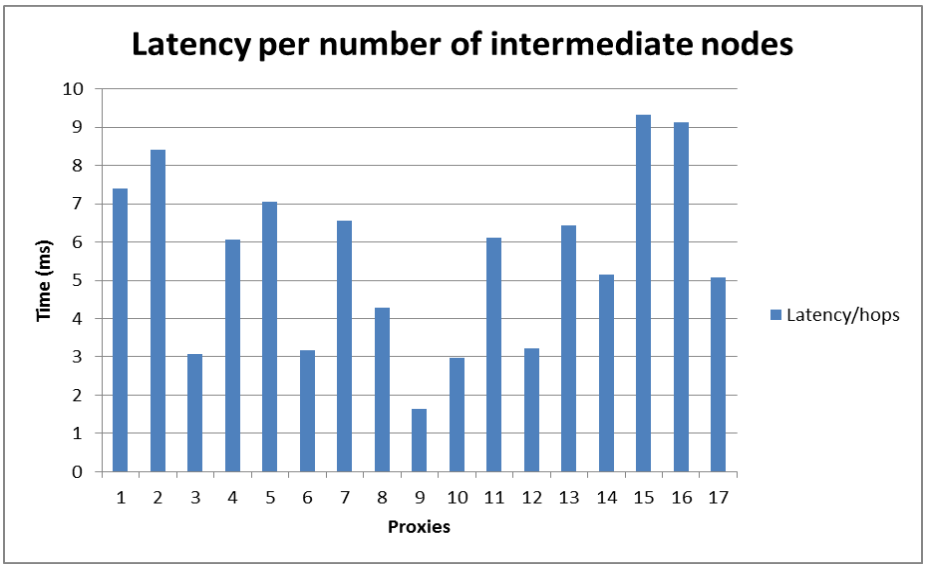

Figure 5.20 Latency between client and proxy divided by the number of intermediate nodes

5.2.2. Proxy selection criteria

To determine an order of selection of the available proxy servers some criteria has to be defined. To do this the next six measured figures will be evaluated for each proxy server:

- Number of hops (intermediate nodes between Client and Proxy)
- Client-Proxy throughput (with ping)
- Client-Proxy latency (with ping)
- Client-Proxy packet loss (with ping)
- Proxy-Server throughput (with wget)
- Proxy-Server latency (with wget)

In table 5.1 the performance of the six figures listed above and analyzed in chapter 5 can be seen. With this global view it is easier to determine which proxy offers better characteristics for each property. In general terms there is a consonance of the five figures due to all are closely related with the intermediate nodes that the communication have to go through.

Table 5.1. Value of the six most influencing figures for each federated proxy selected

Federated Proxy Server	Client to Proxy				Proxy to Web Server	
	Hops	Throughput (kbps)	Latency (ms)	Packet loss %	Throughput (kbps)	Latency (ms)
Node 1	6	2613	44,4	0%	2139	15,6
Node 2	7	4476	58,9	13%	1133	21,1
Node 3	10	1905	30,8	10%	1321	69,2
Node 4	8	3498	48,6	0%	850	51,4
Node 5	7	9761	49,4	0%	401	150,6
Node 6	6	3268	19,1	0%	2449	30,9
Node 7	9	2895	58,9	0%	1129	31,1
Node 8	6	6563	25,7	7%	1898	24,3
Node 9	6	14308	9,8	0%	5715	10,2
Node 10	6	7488	17,8	0%	7267	2,2
Node 11	11	7693	67,2	0%	415	132,8
Node 12	12	4517	38,6	23%	1284	31,4
Node 13	8	3359	51,5	3%	290	248,5
Node 14	12	717	61,8	0%	322	238,2
Node 15	21	665	195,9	23%	95	704,1
Node 16	11	3762	100,4	13%	273	199,6
Node 17	12	2031	61,0	20%	1077	39,0
Average	9	4678	55,3	7%	1651	117,7

However, for the point of view of the final user, the most important is to achieve good values in latency and throughput in whole communication from the client to the web server due to this is the final values that he will experiment. Hereby, table 5.2 represents the cumulative percentage of the throughput and the complementary of the latency, so it will be easier to identify which nodes offer a better performance. As it can be seen, node 9 has the best values and it has to be the first selection option for the user evaluated. Also, node 9, has the best performance in table 5.1 and is remarkable that is the one of the proxies with less intermediate nodes even though there is one of the further proxies from the client node.

Table 5.2. Rate of evaluation of throughput and latency between the client and the web server

Federated Proxy Server	Client to Web Server	
	Throughput %	Latency %
Node 1	29%	93%
Node 2	22%	91%
Node 3	19%	89%
Node 4	17%	89%
Node 5	9%	78%
Node 6	34%	94%
Node 7	20%	90%
Node 8	36%	94%
Node 9	100%	98%
Node 10	90%	98%
Node 11	10%	78%
Node 12	24%	92%
Node 13	7%	67%
Node 14	5%	67%
Node 15	2%	0%
Node 16	6%	67%
Node 17	17%	89%

Node 15 is a case of study due to his poor performance. This node is located in Granollers at 31,5km of distance from the client node in Campus Nord. It has the lower values of throughput and unacceptable values of latency. It is also surprising that the communication with this proxy has to go through 21 intermediate nodes, more than the double of the average, and the process of the packets in these nodes are also the poorest.

Table 5.3. Traceroute of proxy 15

1	UPCpangea-1193.lan	0.526ms
2	10.228.205.1	2.794ms
3	10.228.207.1	3.783ms
4	172.25.35.126	5.425ms
5	172.25.56.253	11.837ms
6	10.139.95.225	12.565ms
7	10.139.37.193	15.267ms
8	172.25.46.33	18.673ms
9	172.25.50.186	22.203ms
10	172.25.49.241	24.163ms
11	172.25.57.33	29.101ms
12	172.25.43.54	34.390ms
13	10.139.8.161	295.976ms
14	10.90.35.225	301.410ms
15	10.91.26.65	302.459ms
16	*	*
17	10.91.26.33	48.608ms
18	10.90.212.33	49.144ms
19	172.25.41.117	54.684ms
20	172.25.34.250	62.632ms
21	172.25.47.177	64.057ms
22	10.139.1.67	68.334ms

In table 5.3 inefficient hops 13, 14 and 15 can be detected with the ICMP based tool traceroute. A possible effect can be that these three nodes are congested or they have a deficient configuration that would be advisable to be checked.

Counterpart, node 9 is sited in Gurb at 65,5km of distance from the client node and it only has 6 intermediate nodes. This node counts with several fiber links that improves the performance respect the wireless links.

CONCLUSIONS

One of the more demanded services in a network is the Internet access. Most of the private companies that offer networks connections provide this service naturalness, but in a free, open and neutral wireless community network as Guifi.net it has to be a particular who shares the connectivity to the Internet, as many other services do. Normally they are shared by institutions as hall towns or associations and they are in charge of the maintenance. This gate to the Internet is done with an element called proxy. A properly configuration of the proxy by the owner and a good choice of the proxies available by the client are the keys for the correct function of the system.

The principal goal of this study is to do a valid procedure to be able to identify which proxy offers a better performance. There are several parameters to analyze in a network with multiple choices of proxies. This study has centered in the transport layer evaluating typical TCP figures as throughput, latency or rate of packet loss. Also other factors as intermediate nodes or physical distance have been analyzed. For this porpoise, some tools as ping, iperf and wget have been used.

Iperf is the optimum tool to measure the figures required but as far as proxies doesn't have this service running; these figures have to be measured with ping. For this study the three points method has been used to measure throughput. Before that, the possible error of the method has to be discerned comparing it with iperf results in an iperf server of the network. The error acquired is a 17% that will be applied to the correspondent measures. This error appears because ping uses ICMP protocol which has a different treatment in the different nodes respect the other tests done with TCP based tools as wget.

After that, the probes with ping and wget to the proxy can be done. Tests determined that the stretch between the proxy and the web server is slightly poor in latency and throughput than the client to proxy communication. So, the bottleneck of the system will be in this part of the network.

Test results reveal that select a proxy taking into account the distance, as it is mostly done, is not the best choice. However, the number of intermediate nodes is in general a more effective way to determine which proxy gives a better performance in terms of latency and packet loss.

This thesis has been useful to analyze deeply some elements studied during the degree as proxy servers' behavior or the real influence on the performance of a communication of the routing nodes that compounds a network. It has been also interesting to analyze the different protocols and the figures that affect them. The final remarkable useful aspect has been the study of a singular community network as Guifi.net with a formidable organization and an always encourage for the common society knowledge.

Future line works

One interesting line of work to improve this study is to do the TCP measurement passively. There are some discussed tools as tcpdump that allows measuring for instance latency and throughput in this way. This is possible due to these kind of software adds a timestamp to the packets. The principal benefit of this method is that the network is not overload with control traffic and also can provide more protocol measures.

Another way to improve the tests done is to achieve some nodes in the network with the UDP iperf server activated to do the tests proposed in this study. With this method a reference for real-time traffic will be acquired and analyzed inside the network and other figures as jitter can be studied.

BIBLIOGRAPHY

Papers

[1] Li, J., Blake, C., De Couto, D., Imm Lee, H. and Morris, R. “Capacity of Ad Hoc Wireless Networks”

[2] Afanasyev, A., Tilley, N., Reiher, P. and Kleinrock, L. “Host-to-Host Congestion Control for TCP”

[3] Jiang, H. and Dovrolis, C. “Passive estimation of TCP Round-Trip Times”

[4] Wang, Z., Zeitoun, A. and Jamin, S. “Challenges and Lessons Learned in Measuring Path RTT for Proximity-based Applications”

[5] Dischinger, M., Gummadi, K. P., Haeberlen, A. and Saroiu, S. “Characterizing Residential Broadband Networks”

[6] Krishnan, R., Madhyastha, H. V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T. and Gao, J. “Moving Beyond End-to-End Path Information to Optimize CDN Performance”

[7] Claypool, M. and Claypool, K. “Latency and Player Actions in Online Games” Communication of the ACM

[8] “What is Network Latency and Why Does It Matters” O3b Networks

Links

Guifi.net

https://guifi.net/
http://sfeldc.guifi.net
http://guifi.net/es/guifi/device/51580
http://guifi.net/catalunya
http://guifi.net/ca/elserrat
http://confine-project.eu/
http://blogs.guifi.net/fundacio/2011/10/20/projecte-europeu-de-recerca-cofine/

Software

https://openwrt.org/
http://downloads.openwrt.org/backfire/10.03/brcm47xx/packages/
https://www.virtualbox.org/
http://www.ubuntu.com/
http://linuxcommand.org/writing_shell_scripts.php

Proxy

http://es.wiki.guifi.net/wiki/Servidor_proxy
http://guifi.net/es/node/2413/view/services

Figures

http://en.wikipedia.org/wiki/Latency_%28engineering%29
http://en.wikipedia.org/wiki/G.114
http://en.wikipedia.org/wiki/Network_performance
http://en.wikipedia.org/wiki/Throughput
http://en.wikipedia.org/wiki/Packet_loss

Tools

http://www.gnu.org/software/wget/
http://iperf.sourceforge.net/
http://es.wikipedia.org/wiki/Ping
RFC 792: INTERNET CONTROL MESSAGE PROTOCOL
RFC 6349: Framework for TCP Throughput Testing


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

ANNEXS

ANNEX I: UDP TEST

I.I. UDP test with Iperf

The second protocol that can be analyzed with iperf is UDP. In this case there are not a connection established between the client and the server and there are not retransmissions in case that the packet doesn't arrives to the destiny. This protocol is orientated to real time traffic which has a tolerance to packet lost.

In UDP there is another important figure as jitter [see section I.II]. Jitter is an annoying effect when a real time communication has been established because the stream can suffer cuts. With iperf it is possible to determine the jitter and it can be stored to be analyzed in the file UDPjitter.txt.

The test to acquire the data from UDP is as follows:

```
for line in $(cat IPLIST.txt);
do

IPERF1=`iperf -u -c $line -b 1M | grep 'bits'`


THR1=`echo $IPERF1 | awk '{ print $7}'`

echo $line >> UDPthr.txt

echo $THR1 >> UDPthr.txt

JIT=`echo $IPERF1 | awk '{print $17}'`

echo $line >> UDPjitter.txt

echo $JIT >> UDPjitter.txt

done
```

To do this test it is necessary that nodes with Iperf installed have the Iperf server in UDP mode. With this, it is possible to acquire the desired figures. However, any node analyzed had this function activated, so, this test has only done in the preproduction environment. To develop these tests would be interesting to install and run this functionality in each node where the proxy servers are installed to study the behavior of this protocol.

I.II. Jitter

Jitter is often used as a measure of the variability over time of the packet latency across a network. In simpler words, jitter is the variation in latency. A network with constant latency has no variation so it has no jitter. This figure is an important indicator for the quality of service of the communication system. It specially affects in real time communications (RTP) and other protocols based on UDP. A high value of jitter makes impossible the correct performance of live video and voice applications.

The most common method to avoid jitter is to place a buffer at the receiver. The length of the buffer has to be bigger than the jitter effect and if a packet exceed the maximum jitter allowed by the buffer it will be discarded. In figure I.I it is possible to see the effect in the system of a buffer. Jitter is generally caused by congestion in the IP network. The congestion can occur either at the router interfaces or in a provider or carrier network if the circuit has not been provisioned correctly.

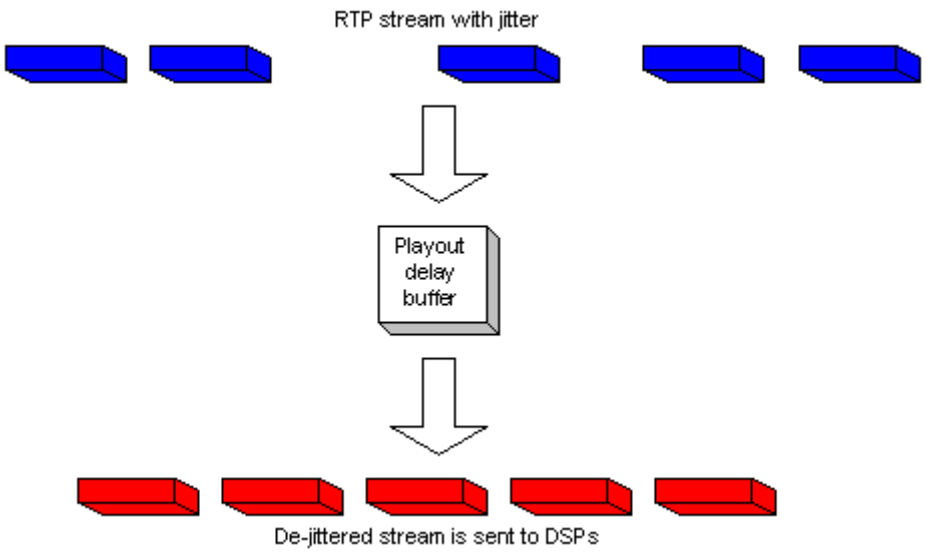

Figure I.I Effect of the buffer in a RTP stream with jitter

Jitter can be detected with iperf tool in UDP mode as seen in figure I.II. Due to it only can be detected with a server at the receiver; jitter will be only calculated at the nodes with iperf server settled to determine a possible correlation with the properties of the system. Actually, there is any node with the UDP iperf server function running in Guifi.net network so jitter has not been measured in this study.

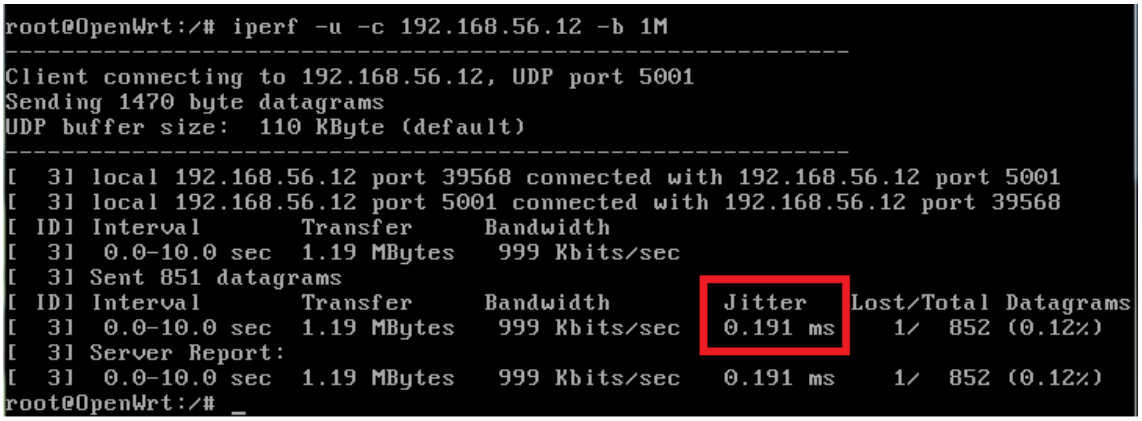

Figure I.II Measure of jitter with iperf