Voice to the People

Gustavo Felisberto
<gustavo@student.dei.uc.pt>

February 28, 2007

Abstract

The Asterisk[1] framework promises to be a low cost with high performance solution for VoIP infrastructures that need interoperability with legacy systems. In this paper we analyze the performance in a small scale system and try to extrapolate the behavior in a larger scale.

Contents

State of the Art

1 Introduction

On large scale organizations the need to make voice calls is continuing to grow despite the emergence of alternative communication system's based on the IP protocol such as e-mail, the web and Instant Messaging systems. Nevertheless IP based networks are becoming the standard for the internal and external networks. This paves the way for the integration of these two worlds.

Also the need for greater voice quality, the standard PSTN only has a bandwidth of 9600bps, is pushing for changes in the way that we carry voice between two points.

In this paper we study the Free Software version of the Asterisk PBX system in terms of scalability of the system.

[1]Asterisk is a registered trademark of Digium, Inc.

2 Standards

2.1 Signaling

Signaling is the process by which two or more parties will create, modify and terminate a session. The leading standards today for signaling are *SIP* [1] and *H.323* [6] with the recent addition of *Jingle* [7].

The level of complexity and the lack for extended features[9] makes H.323 an option very hard to implement, specially when compared to SIP. Also H.323 lacks the capability to traverse simple NAT[2] firewalls, having to depend of special software, gatekeepers, to be installed at the firewalls. This and other issues have contributed for the progressive abandon of H.323 in favor of SIP.

Jingle is still in it's early stages, but is already used by many Jabber/XMPP clients such as Google Talk, Psi, Kopete and others.

In SIP when using Asterisk the signaling between User Agents and the server, as well as between servers, uses UDP for the transportation of messages. The fact that UDP does not guarantee delivery is not an important issue, because UA's and server's maintain a list of messages waiting for response, and will retransmit these messages if needed. Also with the advance in networking technology packet loss is diminishing, as we will see in the profilling that was done.

2.2 Transport

The need for audio and video synchronization, was well as timestamps of the presentation data, make the usage of RTP[2] a natural choice.

3 Available system's

3.1 Free software system's

With the advent over the last decade of the *GNU/Linux* operating system, Free Software has become increasingly important. In this section an overview is given of IP Telephony systems that conform to the FSF[3] definition of *Free Software*[8].

Asterisk

Asterisk is a full telephone private branch exchange (PBX). It supports all the major standards for VoIP, as well as some not so standard, and the hability to work with dedicated hardware to route calls to and from the standard PSTN. Asterisk is mainly developed by Digium with contributions from programmers around the world.

OpenPBX

OpenPBX is a fork of the Asterisk software that removes some parts that where not considered to be under the GPL. Over time work has continued and the list of diferences to the Asterisk original code include for example integrated STUN[4] support, SpanDSP which allows better codecs and T.38 FAX over IP. The development model of OpenPBX trys to mimic the one of the Linux kernel, with fast release cycles and distribution of tasks to developers. It is very likely that this project will gaing momento over time.

OpenSER

OpenSER originated after FhG FOKUS research institute in Berlin released the source code of their SIP Express Router. The software is a SIP only Registrar server, Location server, Proxy server and Redirect server that targets SIP only instalations. The development team claims that a single server with 4GB of ram can handle as many as 300.000 registered users[4].

---

[2] Network Address Translation

[3] Free Software Foundation `http://www.fsf.org/`

[4]

STUN Simple Traversal of UDP (User Datagram Protocol) through NATs (Network Address Translators) is a protocol that allows a client behind NAT to find it's public IP, type of NAT used and associated NAT port [3].

Trixbox

Trixbox is a project sponsored by Fonality that consists of a Linux distribution based on CentOS that comes with Asterisk pre-installed. It also includes some extra software to enable web based configuration of the underlying operating system as well as Asterisk itself. At this time this seems a very good solution to those interested in learning about Asterisk as it makes the learning curve much less steeper.

3.2 Non-free system's

Asterisk Business Edition

This system is based on the free version of Asterisk, but comes with one year support. Also it allows for the integration of external, non GPL, packages for some types of audio transcoding.

Skype

Skype was started by Niklas Zennström and Janus Friis from what they learned developing the P2P file sharing system Kazaa[5]. When the project started it was necessary to use the provided Skype-only SoftPhone. This product attracted many users due to easy installation, it's ability to connect to the network in almost any circumstances, and low prices on calls to land lines.

It the last months some company's started providing standalone Skype phones, and even Skype gateways to connect to normal PBX systems[21, 19, 20].

VoIPBuster

This is one of many services by Betamax GmbH & Co KG that provides low cost calls to and from the *PSTN*. This service is extremely popular right now because calls to land lines in many country's are for free. The interface infrastructure of VoIPBuster uses *SIP* which makes it possible to integrate it with an existing *SIP* system like Asterisk[10].

4 Business models and Future perspectives

Several business models are emerging in the VoIP world. Companies are either selling solutions to the corporate world, or targeting the *SOHO*[5] network. The first target's medium to large "entities" that wish to cut down costs on internal communications while keeping usability as untouched as possible, while the second seduce clients with low rates in calls to the PSTN.

The future will see many changes and many business models will appear. Even standard telco companies are starting the migration, with they're internal infrastructures being VoIP based, and also starting to offer VoIP services to they're clients, including the ability to buy large packages of pre-paid minutes to the PSTN that can be route through the Internet before exiting.

[5] Small Office or Home network

Validation

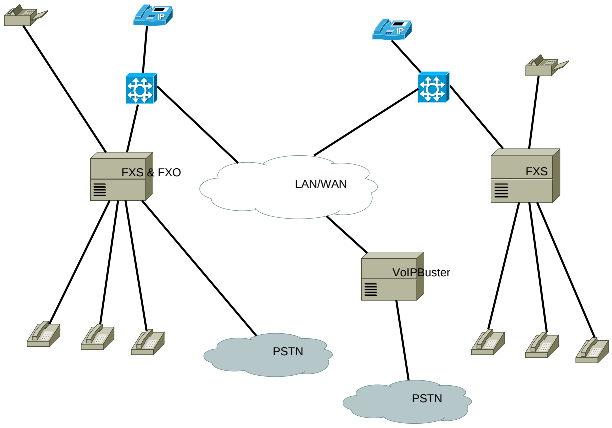

Figure 1: Diagram of the testbed

The diagram in Figure 1 shows a typical Asterisk-VoIP infrastructure. Two or more PC's running Asterisk are equipped with *FXS*[6] cards. These systems are interconnected through a LAN/WAN so that calls can be routed from one system to the other. One of the servers can also have a *FXO*[7] to handle connections from and to the PSTN. In some cases the VoIP infrastructure may be open to external entity's, making the process of placing a call to destination as simple as typing the persons e-mail address.

Experimental Evaluation

In our setup a simplified version of the layout in Figure 1 was used. The testbed consisted of the server described in section 5, where multiple user agents where connected. The main server was located in the data center for the Informatics Department of the University of Coimbra.

The goal to our tests was to know how would the Asterisk server handle many incoming calls and to have some idea on the Hardware requisites for larger installations.

To better simulate real world situations the UA's[8] where connecting from inside the Campus or from external networks.

5 Server Setup

The server used in our tests consisted of a AMD Duron(TM) running at 800Mhz with 380Megabytes of ram, a Realtek 8139C+ network card and a Digium TDM22B[18] analog card. The server was running Gentoo GNU/Linux 2006.1 and used the Gentoo overlay from the VoIP herd to be able to use the latest versions of packages.

The kernel version used was the Linux 2.6.17 with Gentoo's patcheset r8. Glibc version was 2.4-r4 with GCC 4.1.1 . The asterisk version used was 1.2.14 with Zaptel 1.2.12-r1. Sipp version used for the final measurements was the 2007-01-24 snapshot.

6

FXS Foreign Exchange Station is a telephone interface which provides battery power, sends dial tone, and generates ringing voltage. A standard telephone plugs into such an interface to receive telephone service.[12]

7

FXO Foreign Exchange Office is a telephone interface that receives POTS, or Plain old telephone service. It generates the on-hook and off-hook indicators used to signal a loop closure at the FXO's end of the circuit. Analog telephone handsets, fax machines and (analogue) modems are FXO devices.[13]

[8] User Agents - Software or hardware based telephones

Instalation of the base Linux system was trivial, but the same cannot be said about Asterisk. Asterisk is a large software and documentation is very sparse in some areas, namely the way extensions and dialplans work. That was not a big issue with our tests where simple dialplans were used.

6 Evaluation Tools

UDP Ping by Huahui Wu[22][9] and Iptraf[23] were used to do profiling of the underlying network, latency times as well as available bandwidth where the more important parameters that needed evaluation.

Sipp[15] is a traffic generator for the SIP protocol. It was the tool used to test the testbed because it was the Free Software tool that seamed more developed.

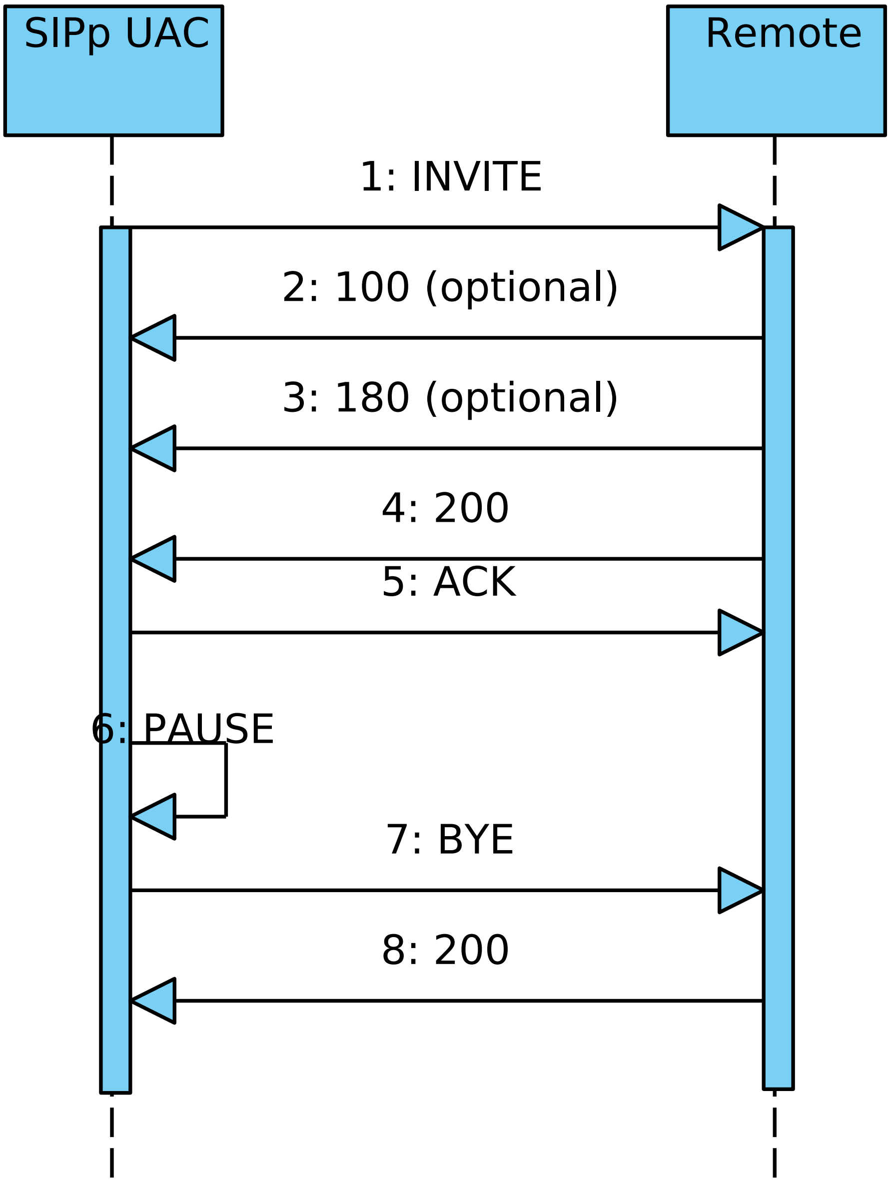

Figure 2: Call Graph

The main test consists of a *client* UA that will generate batch calls to a *server* UA. The *client* will try to generate 2000 calls at different rates. The sequence diagram in Figure 2 shows the flow of each one of such calls.

[9] Some changes were made to the original code to make it possible to stop the test after a pre-determined number of packets were received, count the time in usecs and not in msecs and some minor bug fixes. The patch is available at: `http://student.dei.uc.pt/~gustavo/cUDPingLnx.patch`

ten consecutive runs where made with increasing call rate. First 10 calls per second where placed, then 15, 20 etc untill 55 calls where placed every second.

6.1 Local Lan

The first batch of tests where conducted with the *client* and *server* UA running on the Asterisk server LAN, where network latencies and congestion is very low.

6.2 External Wan

The second batch of tests where conducted with the *server* UA running on the Asterisk server LAN and the *client* UA running a different network[10]. This second set of tests try's to mimic real world conditions where many of the UA's will be moving in external networks.

6.3 Internal Wireless LAN

The campus wireless LAN is part of e-U[16] initiative. It was impossible to generate tests in this network due to the fact that latencies and packet loss to the Asterisk server where:

- Ten to twenty times larger than in other networks
- Several times larger than what is admissible for proper VoIP to work

This will probably change in the near future was more resources are transfered to this infrastructure.

6.4 Results

Figure 3, figure 4, figure 6 and figure 7 show the results of the 10 consecutive batches of calls.

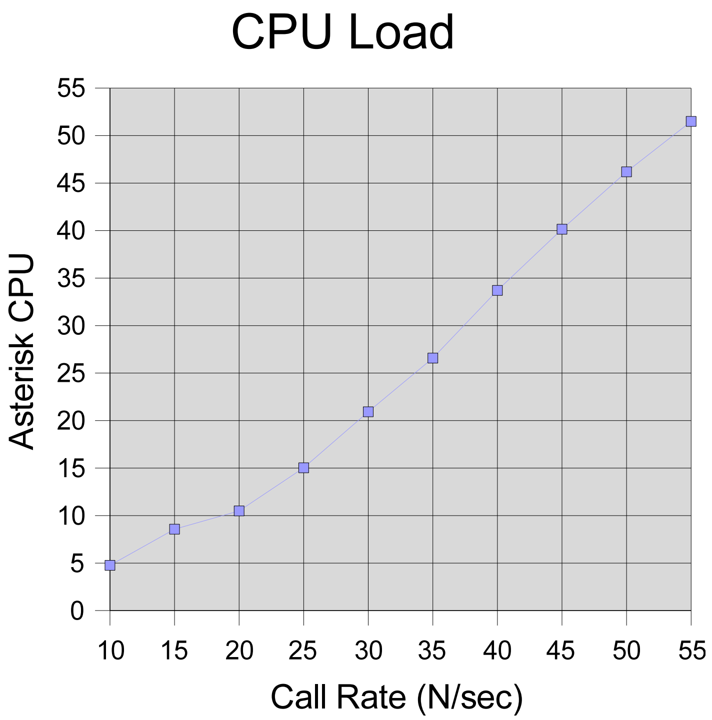

Figure 3: Lan Asterisk CPU Chart

[10]The network used was a asymmetrical 16Mb/1Mb cable modem network

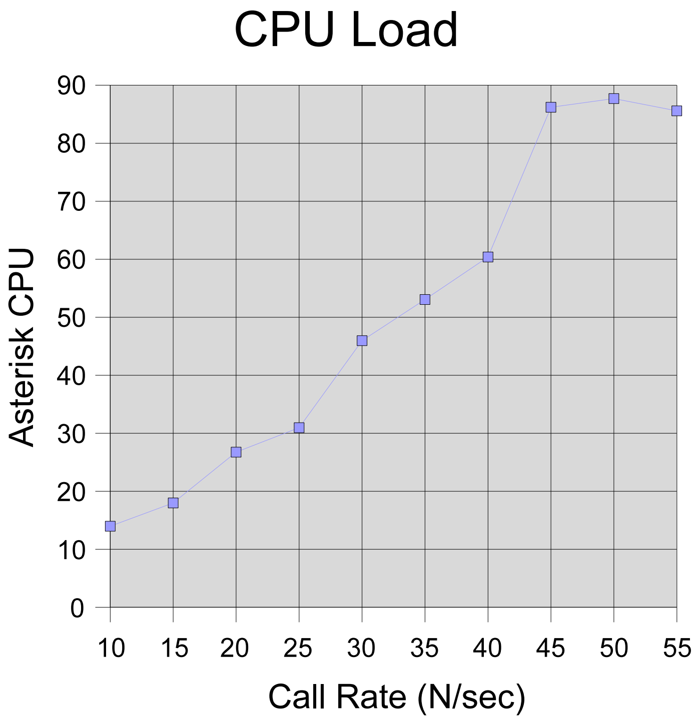

Figure 4: Wan Asterisk CPU Chart

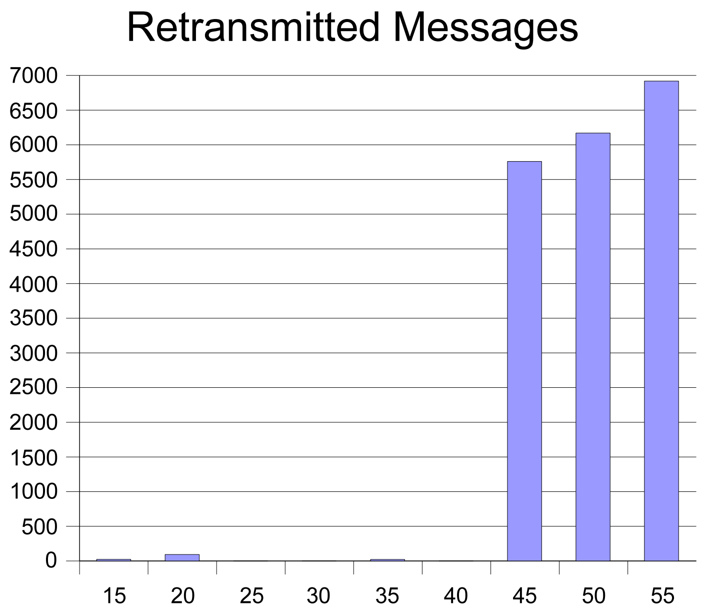

Figure 5: Retransmitted Messages in WAN UA

As can be seen the CPU time taken by Asterisk to process the calls grows in a linear way when using the LAN UA, but the same is not true when the calls originate from the WAN UA. In figure 5[11] we see that when the call rate reached 45 calls per second the number of messages that were lost in transit and had to be resent spiked. The lost messages are easy to explain due to the connection of the UA being saturated.

We can also see that for the same call rate the load on the Asterisk server doubles for the WAN UA when

[11]No messages were lost with the LAN UA.

compared to the LAN UA. This is probably due to the increased time that Asterisk must handle the messages in queue.

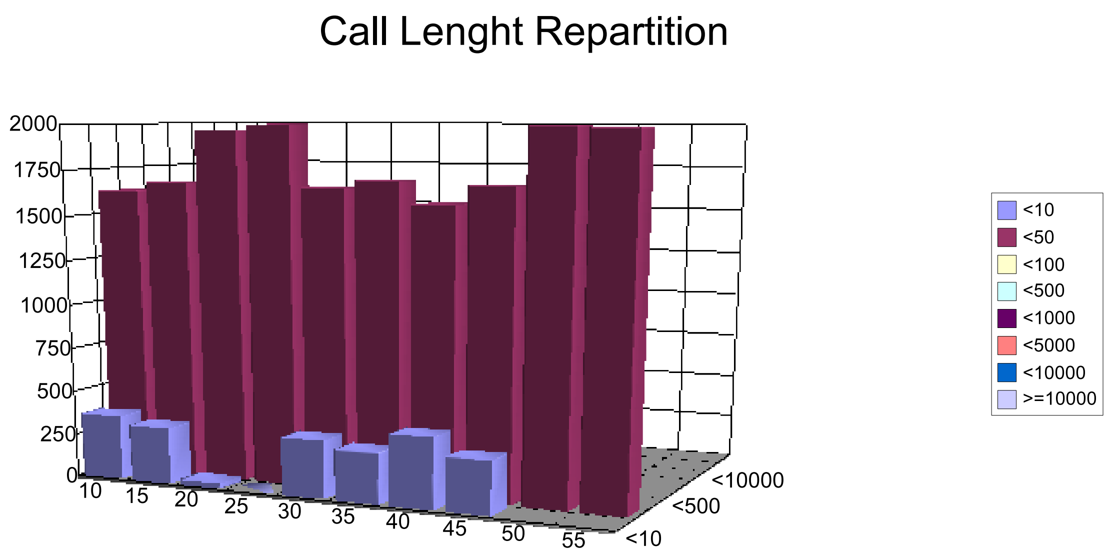

Figure 6: Lan Call Length Repartition (in ms)

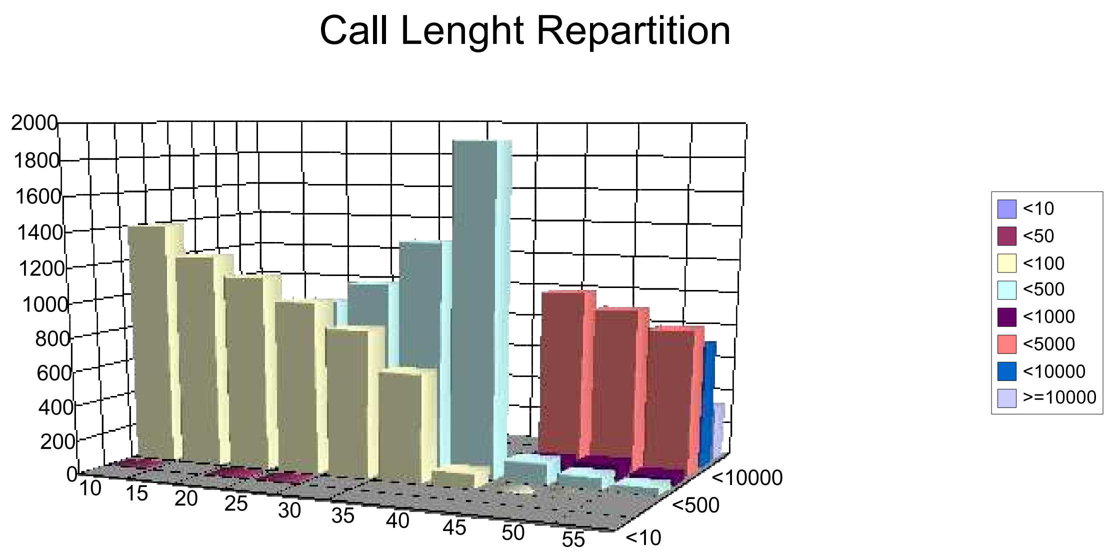

Figure 7: Wan Call Length Repartition (in ms)

Part I
Conclusions and Future Work

This study does not test when real RTP traffic is routed via Asterisk. To do advanced CPU load testing with RTP traffic would be nice to see how asterisk would handle that.

In any case it seems clear that even with low end hardware it is possible to have a full Asterisk based VoIP PBX that is capable of handling a substantial amount of calls, if no transcoding[12] is needed. But careful monitoring is also needed as with any network service.

Part II

Acknowledgments

The author would like to acknowledge the motivation given by Professor Edmundo Monteiro for the realization of this work. Also the Laboratory of Communications and Telematics of the University of Coimbra has given support in the form of Hardware as well has the HelpDesk for the department.

References

[1] RFC 3261: Session Initiation Protocol (SIP) <http://tools.ietf.org/html/rfc3261>.

[2] RFC 3550: RTP: A Transport Protocol for Real-Time Applications <http://tools.ietf.org/html/rfc3550>.

[3] RFC 3489: STUN <http://tools.ietf.org/html/rfc3489>.

[4] OpenSer Features <http://www.openser.org/index.php?option=com_content&task=view&id=33&Itemid=50>.

[5] Skype at Wikipedia <http://en.wikipedia.org/wiki/Skype>.

[6] ITU Recommendation H.323: Packet-based Multimedia Communications Systems (September 1999).

[7] JEP 166: Jingle <http://www.jabber.org/jeps/jep-0166.html>.

[8] Free Software Definition <http://www.fsf.org/licensing/essays/free-sw.html>.

[9] A Comparison of SIP and H.323 for Internet Telephony, Henning Schulzrinne and Jonathan Rosenberg

[10] VoIPBuster and Asterisk integration <http://www.voip-info.org/wiki/view/Asterisk+VoIPBuster>.

[11] OpenPBX Overview<http://wiki.openpbx.org/tiki-index.php>.

[12] FXS Definition <http://en.wikipedia.org/wiki/Foreign_exchange_station>.

[13] FXO Definition <http://en.wikipedia.org/wiki/Foreign_exchange_office>.

[14] RTP Overview<http://geocities.com/intro_to_multimedia/RTP/>.

[15] Sipp <http://sipp.sourceforge.net/>.

[16] e-U <http://www.e-u.pt/>.

[17] Gentoo VoIP Herd <http://overlays.gentoo.org/proj/voip/wiki>.

[18] Digium TDM22B Specifications <http://www.digium.com/en/wheretobuy/digiumdirect/productview.php?product_code=RTDM22B>.

[19] Philips VOIP841 <http://www.consumer.philips.com/consumer/catalog/tree/en/gb/consumer/PHONES_GR_GB_CONSUMER/INTERNET_TELEPHONY_DEVICES_CA_GB_CONSUMER/ce/_productId_VOIP8411B_05_GB_CONSUMER/Internet__DECT_phone+VOIP8411B_05>.

[20] Linksys CIT400 <http://www.amazon.com/o/ASIN/B000JI75GU/ref=s9_asin_title_1/104-3622623-8972706>.

[12] If two generic UAs which to engage in communication, but no set of codecs are found that both support, it is possible for that the PBX will act as a proxy and convert the audio/video formats.

[21] Callfree Gateway E1 PRI < http://eshop.spin3d.com.tw/product_info.php/cPath/46/products_id/94 >.

[22] UDP Ping < http://svn.smallwhitecube.com/pcap-tools/trunk/src >.

[23] Iptraf < http://iptraf.seul.org/ >.

[24] Trixbox < http://www.trixbox.org >.