Monitoring based on statistical analysis for evaluating quality of calls in VoIP environment

Ana Flávia M. de Lima[1], Francisco R. Ribeiro[2], Maiquel S. de Melo[2], Leandro S. G. de Carvalho[3], Regeane Aguiar[3], José Neuman de Souza[1,2] e Edjair de Souza Mota[3].

[1]Department of Engineering of Teleinformatic (DETI), Federal University of Ceará (UFC)
Post Office Box 6007 - 60.755-640 - Fortaleza - CE - Brazil.
anaflavia@deti.ufc.br
[2]Departament of Computer Science (DC), Federal University of Ceará (UFC)
60.455-760 - Fortaleza - CE - Brazil.
raphael@pop-ce.rnp.br, maiquel@lia.ufc.br, neuman@ufc.br
[3]Departament of Computer Science, Federal University of Amazonas (UFAM)
69077-000 Manaus - AM - Brazil.
leandrogalvao@ufam.edu.br,{rba,edjair}@dcc.fua.br

Abstract. Monitoring call quality in a Voice over IP (VoIP) environment has great importance to collect information about those calls on the network. Packet loss on the network and de-jitter buffer, delay and subjective speech evaluation using the E-Model will enable the manager to verify the quality of calls. This article proposes the development of an agent to implement objects with statistical data about quality of call evaluation in a VoIP environment

Key-words: VoIP, Quality of call, Monitoring, Statistical data.

1. Introduction

Voice over IP (VoIP) technology performs communication among people through out the world transmitting human voice on IP networks. The great contribution attained by the users is a significant cost reduction on long distance calls. As a user's IP address is transparent, the cost of a call using this technology is not charged by the distance between the sourcer and the receiver. Using Voice over IP, a local or long distance call is always considered local by the telephone company.

There are some protocols responsible for signaling VoIP calls, as reported in the following recommendations: H.323 from ITU-T (International Telecommunication Union – Telecom Standardization Sector) [1], SIP (Session Initiation Protocol) from IETF (Internet Engineering Task Force) [2] and H.248.1 from ITU-T [3]. Each one of them contains its own architecture and entity group that are responsible for signaling VoIP calls on the network. This article does not aim to describe these protocols nor their entities.

Management is also necessary on VoIP, mainly because it's a real time transmission. This characteristic must be taken into account when the service is available because users praise having it always ready for use. The ITU-T standardized the MIB (Management Information Base) H.341 [4], that contains a group of MIBs to manage multimedia systems based on H.323. Each one of these MIBs contain a group of objects responsible for returning relevant information to the implemented module. The various IETF work groups also defined a series of specific MIBs to managing equipment, protocols, services and applications [5]. Equipment manufacturers may develop MIBs employed in VoIP that meet requirements not anticipated in IETF MIBs.

This article presents an agent to monitor and analyze the quality of calls in VoIP networks. In order to attain this goal, a tool was employed to collect data on quality of real calls, and stores them in a database. A demon presents reports based on statistical samples, such as mean, mode, maximum value, minimum value, and standard deviation. The manager can access these data by means of the agent.

The greatest advantage of this agent is being adaptable, that means, new statistical data can be added without the need to altering the agent nor the MIB structure.

The next section remarks some studies related to this research. Section three describes some factors that influence the quality of speech on voice over IP. Section four presents a monitoring management environment based on statistical analyses to evaluate voice quality. Section five presents the conclusions and suggestions for future studies.

2. Related Works

Some researches on voice over IP management and monitoring have been performed. In [5] a survey on a management environment was made by analyzing some MIBs that are likely to be implemented. It also describes a way of managing tariffs and proposes a tool for managing voice over IP networks. The need to develop a MIB to monitoring the quality of calls in voice over IP networks was perceived based on this study, for the ones presented by [5] did not meet the requirements of VoIP network monitoring.

In [6], were introduced some tools to generating, monitoring and collecting statistical data to evaluate the degree of quality of calls in VoIP environment. The management approach by means of a protocol was not exploited. Instead, a Web tool that delivers statistical data was implemented, applying quality data (delay, loss, and jitter) of simulated environment employing voice from a recorded-voice file. This solution does not supply real call statistical information because of the simulation environment employed. The work presented hereby proposes an architecture agent/manager by means of SNMP (Simple Network Management Protocol), in which the agent will supply the manager with statistical information regarding real calls from a certain period of time.

A tool based on the extended E Model for appraising the quality of calls was developed by [7], supplied by the RTCP (Real Time Control Protocol), defined in RFC 3550[8], generating a report containing information regarding quality of calls. That information is useful to be mapped to the subjective quality evaluation MOS (Mean Opinion Score) [9] method. This work is based on that result by means of which the statistical monitoring takes place.

An architecture to monitoring quality of IP telephone calls was developed by [11], based on the development of a library named Vquality, that implements the objective appraisal supplied by the E Model [10], and its extensions. That allows the whole

VoIP environment: IP telephones (hardware and software) and voice gateway, to supply information regarding quality of calls by means of CDRs (Call Detail Record). The date supplied by those reports made possible statistical monitoring of VoIP calls and point some reasons for degrading quality of calls.

This research differentiates from the related works by the development of a monitoring environment employing the SNMP protocol and its entities to monitor VoIP calls.

3. Factors that influence speech quality in VoIP

Various challenges must be surpassed in an IP (Internet Protocol) network so as to make call quality gain credibility among users. The Internet itself does not guarantee reliable packet delivery and, regarding real time voice application, some factors have to be taken into account to provide quality to the service.

The choice of the codec to be employed, the end-to-end delay, the lost packets on the network and voice packets late discarded by jitter buffer, can affect the final quality of speech. Therefore these factors should be evaluated so as to guarantee quality to a VoIP network.

Codecs – are voice signal encoder algorithms that can perform speech compression. They can be divided into three categories: waveform encoders, source encoders and hybrid encoders [12] [13].

In order to perform a good compression ratio, the encoder algorithms have to save a certain amount of samples to be able to construct a frame. This way, the encoder inserts a delay in the ongoing communication that corresponds to the amount of time necessary to build the voice frame [14].

The factors that must be taken into account when comparing different techniques of vocoding or vocoders are: bit ratio, delay, algorithm complexity and quality [13].

Delay – the way the packets travel on the network causes communication delay. The voice packets may contain one or more voice frames. After being built they are set in a transmission queue where they'll wait their turn to be sent to the network physical link. The end-to-end delay limit for a quality voice communication is 150ms. A large packet delivery creates a great delay that produces a significant decline in the quality of speech. To soften this problem, the large packets can be divided so as to be sent mixed with smaller ones. After waiting for some time in the transmission queue, the packet is sent to next destination node. At the physical layer, this packet will be serialized bit to bit, thus introducing a serialization delay. After the serialization the bits will take a constant amount of time to be transmitted to the next data network node. Therefore the total time spent by the network is the addition of all amounts of time spent in the queues, in serialization and in transmission. [7].

According to [13], delays between 150 ms and 400 ms can be acceptable, but are not ideal, and delays over 400 ms can seriously harm real time conversation interactivity. The VoIP software receiver usually discards any packets whose delay exceed a certain standard, e.g. 400 ms. Hence such a delay implies definitive loss of these packets.

An important end-to-end delay component is the router queue delay. Because of these variable network delays, the time elapsed between the moment the packet is generated at the source until it is received at the destination may vary. This delay variation is called jitter [15].

The problem involving delay variation can be solved by the de-jitter buffer, that stores voice packets for a certain period of time, delaying their delivery enough time so as to enable the majority of the packets to sent to encoder within schedule. The packets that arrive behind schedule are considered lost by the buffer. The ones that are received in advance wait for their turn to be sent to the decoder.

Packet losses – voice packet losses are important speech quality transmission factors on a VoIP network.

Basically, three reasons may account for voice packets losses: transmission errors, packets discarded at the network routers and at the de-jitter buffer [7].

Lost voice packets should never exceed 5% for the entire conversation. In case of loss rate excelling this limit, interactivity problems will arise damaging speech quality.

To reduce lost packet problems on the network, some packet repair techniques may be implemented.

In case of losses arisen at the de-jitter buffer, buffer size has to be taken into account, once it influences the most important parameters associated to speech quality: end-to-end delay and lost packets [16].

De-jitter buffer – jitter problems arise because of the delays occurred in the operational system and at the packet queue in the network nodes. If the packets are played immediately after their arrival at the receiver's end, communication quality will become unintelligible.

The necessity to supply a synchronous transmission of voice packets is met by storing them temporarily at a buffer, delaying their delivery for a span of time so as to enable the majority of packets to be received within the scheduled time. This buffer is known as de-jitter buffer [7].

The delay may be constant throughout the duration of a call or it may vary adaptively [15].

Regardless the employed algorithm (constant or adaptive), the de-jitter buffer completely eliminates the jitter problem, assuming a longer but constant delay without increasing the amount of lost packets [7].

Subjective Method of Speech Quality

The MOS method is derived from the ACR (Absolute Category Rating) method for assessing voice transmission systems.

This method defined in the ITU-T P.800 [9] recommendation requires that some people appraise the general quality of voice samples submitted to vocoders for the purpose of telephone communication. The appraisers estimate grades from 5 (excellent) to 1 (poor) to the quality of voice performed in the communication system being tested [14]. The goal of these tests is to present a figure of personal assessment of the jury members regarding the signal transmitted by the communication system compared with transmitted by the algorithm under test [17].

Although the method seems very clear, the MOS grading is a hard procedure to be performed. Objective methods have been developed and implemented to estimate this punctuation for communication system employing voice over IP technology. Among these objective methods stand out the E Model, as defined in the recommendations G.107 and G.108 by ITU-T [10] [18].

The E Model is based on the concept that "psychological factors on the psychological scale are additive", that means, each contribution caused by a loss factor degrading a voice communication system may be calculated separately, although this does not imply that these factors are not related. The final calculation of these loss factors is a scalar factor R, that varies from 0 (worst case) to 100 (excellent).

The R factor is obtained by the following formula:

$$R = Ro - Is - Id - Ie + A$$

Where **Ro** represents the effects of the signal-noise ratio (SNR); **Is** represents the combination of simultaneous voice signal impairments; **Id** represents impairments associated with end-to-end delay; **Ie** represents impairments associated to the equipment employed; and **A** corresponds to the advantage factor, or expectation factor.

The R factor may be converted into the MOS grading through the following third power formula [10]:

For $R < 0$: MOS = 1
For $0 \leq R \leq 100$: $MOS = 1 + 0{,}0035\,R + 7.10^{-6}\,R\,(R-60)\,(100-R)$
For $R > 100$: MOS = 4,5

The R factor is usually described in value categories, as shown in Table 1.

Table 1: Speech transmission categories. Source: [10].

R Factor	MOS	User's satisfaction
90 ≤ R < 100	4,34 – 4,50	Very satisfied
80 ≤ R < 90	4,03 – 4,34	Satisfied
70 ≤ R < 80	3,60 – 4,03	Some users are dissatisfied
60 ≤ R < 70	3,10 – 3,60	Many users are dissatisfied
0 ≤ R < 60	1,00 – 3,10	Nearly all users are dissatisfied

4. VoIP Management

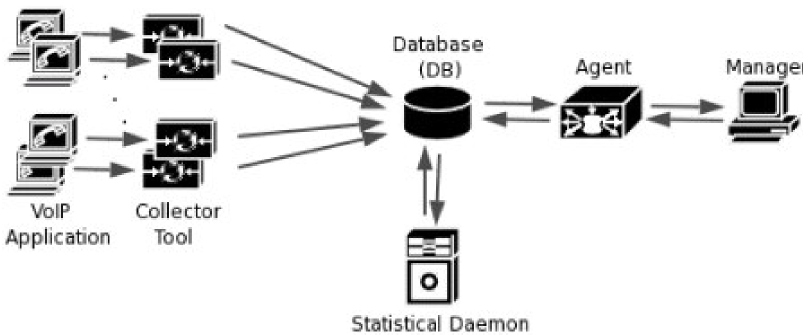
Due to growth and expansion of VoIP networks, arises the need to manage service quality so as to justify its use.

In order to manage devices on IP networks, the SNMP [19] is usually employed. This protocol operates based on two mechanisms: a pooling that employs instructions requisition/response; and on an event reporting that indicates that some important events took place, started by the agents [20].

A management environment based on statistical data analysis for voice quality evaluation on VoIP networks will be shown in subsection 4.1, which will detail each of the illustrated entities.

4.1 Management Environment for Monitoring Appraisal of Quality of Calls in Voice over IP

The management proposed for VoIP networks on article's scope comprises six interrelated functional entities. Picture 1 shows the management environment architecture.



Picture 1. Management environment architecture.

VoIP Application - the software employed, OhPhone [21], is a command line version of OpenPhone [22]. The two of them are the softphones from the OpenH323 software family project. Both of them allow initiating and receiving direct calls, or registering on a Gatekeeper (pre-configured or automatically configured on the network), in order to have its calls directed. It is configured to generate a trace of all events since the initialization, including the calls made and their statistical information (such as total of packets, amount of lost packets , delay and jitter of voice packets) that are filtered afterwards by the Collector Tool. It was chosen because it is a free software and because it has a version for Linux.

Collector Tool – this module analyses the trace file generated by the VoIP application OhPhone and updates the database with real time information to be employed in the statistical calculations available to the manager about ongoing calls. The trace contains some information about ongoing calls as well as: timestamp, lost packets, delay, call identifier, among others.

This tool comprises two parts: the filter and the parser. The first part only separates the relevant information from the trace about the calls to be evaluated concerning quality of speech. This part was adapted based on the work [7] in order to filter real time trace generated by OhPhone. The second part processes the filter outputs and updates the database with quality of speech parameter on calls.

The output of the trace filter shown in Picture 2 contains event signalization of VoIP application regarding the calls. The beginning of the trace file is filtered, since it shows the **Trace start timestamp**, which is necessary, because the subsequent ones are chronometered from the initial one. **AdmissionRequest** and **AdmissionConfirm** signalize the register in the gatekeeper of a tentative call between the **src** and the **dest**, which are the numbers associated with the caller and the receiver respectively. **CallAnswerd** signalizes a call established. **Pt** shows the type of codec employed, **psz** is the packet size in bytes, **ts** indicate voice duration in each packet (milliseconds). The next lines refer to statistical information received form each voice packet: sequence number, timestamp referring to initial trace file timestamp, **lost** (lost packets on the network), **toolate** discarded packets by the de-jitter buffer and delay in milliseconds respectively. **CallEnd** signalizes the end of a call.

Parser is a two status machine: 'awating' and 'on'. It is started simultaneously with the VoIP application and the filter. It starts on the 'awaiting mode'. The filter's output signalizes consecutive request and confirm of ongoing calls, altering parser's status to 'on'. Thereby the R factor is processed as shown in [23], mapped to MOS [24] and updated in the database. At the end of a call the status changes from 'on' to 'awaiting'.

```
Trace start timestamp: 19:54:32.070 2005/3/15
AdmissionRequest(2e b2 60 b6 79 f2 18 10  9a 48 00 10 b5 61 a7 e5)
AdmissionConfirm(2e b2 60 b6 79 f2 18 10  9a 48 00 10 b5 61 a7 e5)
 src:Ana Flavia-UFC (050002) dest:"050007"
CallAnswered
pt=PCMU psz=160 ts=320
1   1:46.207  lost=0 tooLate=0 (116 ms)
2   1:46.223  lost=0 tooLate=0 (116 ms)
3   1:46.243  lost=0 tooLate=0 (116 ms)
4   1:46.263  lost=0 tooLate=0 (116 ms)
5   1:46.283  lost=0 tooLate=0 (116 ms)
CallEnd   1:46.303
```

Picture 2. Filter Output

Real time filter output collection was performed employing pipes [25], between OhPhone and filter, between filter and parser. VoIP application output is processed during the calls, while being written.

The filter is being developed in Perl profiting from work [7], while the parser is being developed in PHP (PHP: Hyper-text Pre-processor) [26] because of its easy interaction with the MySQL database [27] by means of PHPs native communication interface, which is well documented and is also free software.

Database (DB) – its task is to store data received from the Collector Tool and make them available for daemon statistical calculations. After being processed the information are stored so as to become available for the agent whenever necessary.

According to Picture 1, the management environment needs a data server so as to concentrate the information and make them accessible to the agent.

Statistical Daemon – this module is responsible for calculating statistical figures: mean, mode, maximum value, minimum value and standard deviation regarding the calls; as well as store the output in a database. The agent, who has access to the database, makes these statistical information available to manager.

The goal of this work is to present a monitoring environment for VoIP calls, not to choose the best statistical information to evaluate quality of calls. The statistical models employed are meant to validate the monitoring proposal hereby introduced.

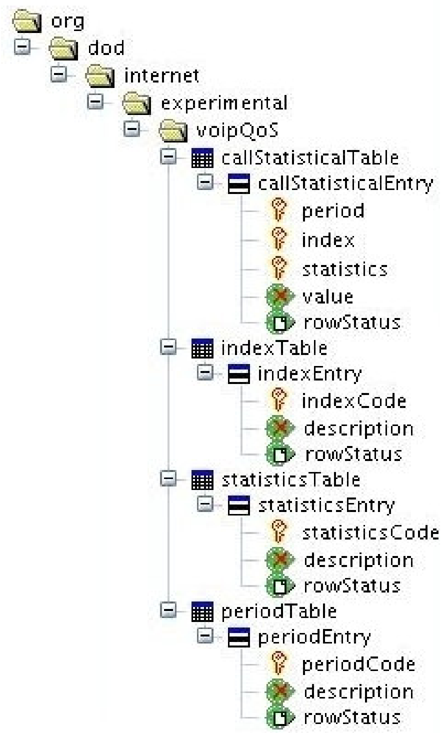
The statistical calculations are employed by the following indicators regarding the calls of a certain period of time: MOS, delay, loss on the network, and discarded packets by the de-jitter buffer. The output of those statistical calculations allow the appraisal monitoring of quality of calls by the manager.

O daemon is being developed in PHP since it interacts easily with the MySQL database applying PHPs native communication interface, it is well documented as well as a free software.

Agent – the agent module was developed in Java [28] because it is multiplatform. The management protocol employed was SNMPv3, since it implements security improvements compared to previous versions. AdventNet [29] agent development environment was applied, hence it is a tool that enables quick development and supplies a set of integrated applications: MIBs SMI (Structure of Management Information) designer (MIB Editor), agent code generator (MIB Compiler) and a SNMP graphical reference application (MIB Browser).

Basically, the agent makes statistical information stored in the database accessible to any SNMP manager application. When a reference is performed on any MIB identifier, the agent accesses the MySQL database employing JDBC [30] technology capturing the results and transforming them into SNMP answers to be sent to the manager.

The set of information made available by the agent was designed employing SMIv2. The details in this model can be seen in Picture 3.



Picture 3. MIB VoIP QoS Tree

The information is divided into three tables: **callStatisticalTable, indexTable, statisticsTable** and **periodTable.** Each table consists of a set of corresponding entries.

Under **callStatisticalEntry** are the information that allow the manager to monitor the VoIP network. **Period** informs the intervals in which the measurement was taken; **index** represents the indicators under analysis; **statistics** indicates the statistical sample employed; **value** shows the result of each statistical calculation; **rowStatus** is a textual convention that was employed in the four tables to control line generation, modification and exclusion by the manager, as defined in RCF2579 [31].

The other tables were created to list the group of important data to the manager application:

IndexTable stores a set of quality indicators under analysis (MOS, delay, loss on the network and de-jitter buffer loss);

StatisticsTable contains a set of statistical samples (mean, median, mode and standard deviation) employed in these calculations;

PeriodTable gathers a set of intervals (minute, half an hour, hour, six hours, day, week, month) in which the information is observed.

The objects: **indexCode**, **statisticsCode**, and **periodCode** are employed as indexes in accessing information on the **callStatisticalTabel.**

The data can be rescued by the GetBulk primitive or by repeated calls employing the GetNext primitive.

New statistical models and time intervals can be added, according to the necessity, without altering the MIB structure. Therefore it will be necessary to add a new routine to the daemon to deal with these new indicators. Moreover, other parameters can be included by adding filters that are able to rescue new information. Easy maintenance is what characterizes the agent as adaptive.

Manager – this module can access the agent's statistical data employing the SNMP get operations. These data can be visualized depending on the manager employed, e.g. they can be shown graphically applying programs as MRTG [32] or Whatsup [33].

5. Conclusions and Future Works

This article presented a monitoring environment based on statistical analysis for evaluating quality of speech on voice over IP networks. Therefore a collection tool was developed to search quality of speech parameter by means of traces.

This article did not aim to describe the protocols that signal VoIP (H.323, SIP, among others) calls. Another study is being prepared to implement the management environment calls signalized by SIP, since the data collection was based on signalization H.323. Therefore this research was not limited to a single type of signalization.

A database was employed to store data from a parser and some information processed by the statistical daemon. A MIB was specified for the development of the agent, which was developed in Java applying AdventNet environment. A manager that shows monitored information graphically communicates the agent the by means of a SNMP protocol.

As a continuation to this work, the development of a Web manager application that allows customized and flexible monitoring. This tool would be able to be able to be accessed by any computer connected to the Internet.

The creation of a Web Service [34] is being planned. It is known that there are researches nowadays that indicate this architecture might substitute the SNMP [35] management model.

It would be advisable to replace the AdventNet API by a free software as Mibble [36] and SNMP4J [37], just like the database and the statistical daemon. Another goal is to create a baseline for the VoIP network information, in order to develop a proactive management tool to inform the manager by means of trap commands and/or by alarms when something goes wrong menacing the performance of the VoIP network.

6. Acknowledgement

The authors thank RNP Point of Presence in Ceará Brazil, for the facilities made available and for the technical help. Special gratitude to Wellington Albano and Marcos Frota.

7. Referências Bibliográficas

[1] ITU-T Recommendation. H.323. Packet-based multimedia communications systems. Genève, May 1999.
[2] SIP - Session Initiation Protocol. Available: http://www.ietf.org/rfc/rfc2543. Accessed in August 2004.
[3] ITU-T Recommendation. H.248.1 Gateway Control Protocol. Genève, March 2002.
[4] ITU-T Recommendation. H.341. Multimedia management information base. Genève, May 1999.
[5] Costa, J.C.P.A. "Implementação e Gerência de uma Arquitetura de Voz sobre IP". Masters Dissertation. UFRJ, September 2003.
[6] Marcondes, C.A.C.; Rodrigues, P.H.A.; David, F. & Costa, J.C.P.A. "Ambiente para Simulação e Monitoração de Ligações Telefônicas IP". In proc. SBRC'03 – Natal – RN, May 2003. pp.615-630.
[7] Carvalho, L.S.G. "Implementação do Modelo E para Avaliação Objetiva da Qualidade da fala em Redes de Comunicação VoIP". Masters Dissertation. UFAM, September 2004.
[8] RFC 3550 - RTP: A Transport Protocol for Real-Time Applications. Available: http://www.ietf.org/rfc/rfc3550.txt?number=3550. Accessed in February 2005.
[9] ITU-T Recommendation P.800. Methods for subjective determination of transmission quality. Genève, November 1996.
[10] ITU-T Recommendation G.107. The E-Model, a computational model for use in transmission planning. Genève, May 2003.
[11] Lustosa, L.C.G.; Rodrigues, P.H.A.; David, F & Quinellato, D.G. "Arquitetura de Monitoração de Qualidade de Chamadas Telefônicas IP". Proceedings – SBRC'05 – Fortaleza – CE, May 2005. pp.1073-1086.
[12] Balbinot, R.; Silveira G. J.; Castello C. F.; Santos M. P. & Quadra, S. A. "Voz sobre IP - Tecnologia e Tendências", Minicourse – SBRC'03 – Natal – RN, May 2003. pp.321-363.
[13] ITU-T Recommendation G.114. One-way transmission time. Genève, May 2003.
[14] Hersent, O.; Guide, D. & Pierre P.J. "Telefonia IP – Comunicação Multimídia Baseada em Pacotes". Addison-Wesley, 2002.
[15] Kurose, J.F. & Ross, K.W. "Computer Networking: A top-down approach featuring the Internet". Addison-Wesley, 2001.
[16] Choudhury, G.L. & Cole, R.G. "Voice over IP performance monitoring". Computer Communications 27, February 2004. pp.529-573.
[17] Oliveira, S.; Júnior, C. J. N. C. & Loureiro, A.A.F. "Telefonia IP para ambientes móveis usáveis". In proc. SBRC'01– Florianópolis – SC, May 2001. pp.542-548.
[18] ITU-T Recommendation G.108. Application of the E-model: A planning guide. Genève, September 1999.
[19] A Simple Network Management Protocol (SNMP) Available: http://www.ietf.org/rfc/rfc1157.txt?number=1157. Accessed in April 2004.
[20] Stallings, W. "SNMPv1, SNMPv2, SNMPv3 and RMON1 and 2". Addison-Wesley, 1999.
[21] Openh323 Project - Ohphone - Command line H.323 client. Available: http://www.openh323.org/. Accessed in March 2005.

[22] Openh323 Project - Openphone - Command line H.323 client. Available: http://www.openh323.org/. Accessed in November 2004.
[23] Markopoulou, A.; Tobagi, F. & Karam, M. "Assessment of VoIP Quality over Internet Backbones", in: Proceedings of Infocom 2002. New York, June 2002. 10p.
[24] Lustosa, L.C.G.; Carvalho, L.S.G.; Rodrigues, P.H.A. & Mota, E. "Utilização do Modelo E para avaliação da qualidade da fala em sistema de comunicação baseados em voz sobre IP". SBRC'04 – Gramado – RS, May 2004. pp. 603-616.
[25] Pipe (computing). Available: http://en.wikipedia.org/wiki/Pipe. Accessed in March 2005.
[26] PHP: Hyper-text Pre-processor. Available: http://www.php.net/. Accessed in March 2005.
[27] MySQL Network. Available: http://www.mysql.com. Accessed in February 2005.
[28] Java Techonology. Available: http://java.sun.com. Accessed in: February 2005.
[29]AdventNet SNMP Agent Toolkit Java Edition 6, Available: http://www.adventnet.com/products/javaagent/. Accessed in February 2005.
[30] JDBC Technology. Available: http://java.sun.com/products/jdbc. Accessed in February 2005.
[31] Textual Conventions for SMIv2. Available: http://www.ietf.org/rfc/rfc2579.txt. Accessed in April 2004.
[32] Multi Router Traffic Grapher. Available: http://people.ee.ethz.ch/~oetiker/webtools/mrtg/. Accessed in March 2005.
[33] Whatsup. Available: http://www.ipswitch.com. Accessed in March 2005.
[34] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. "Web Services Description Language (WSDL) 1.1". Available: <http://www.w3c.org/TR/wsdl>, 2001.
[35] Pavlou, G., Flegkas, P., Gouveris, S., Liotta, A. "On Management Technologies and the Potential of Web Services". IEEE Communications Magazine, julho de 2004.
[36] Mibble. Available: http://www.mibble.org. Accessed in March 2005.
[37] The SNMP API Java. Available: http://www.snmp4j.org. Accessed in March 2005.