Performance evaluation of P2P VoIP applications

Rodrigo Barbosa
Universidade Federal de Pernambuco
Recife, PE, Brazil
rodrigo@gprt.ufpe.br

Carlos Kamienski
Universidade Federal do ABC
Santo André, SP, Brazil
cak@gprt.ufpe.br

Dênio Mariz
Centro Federal de Educação Tecnológica da Paraíba
João Pessoa, PB, Brazil
denio@gprt.ufpe.br

Arthur Callado
Universidade Federal de Pernambuco
Recife, PE, Brazil
arthur@gprt.ufpe.br

Stênio Fernandes
Centro Federal de Educação Tecnológica de Alagoas
Maceió, AL, Brazil
stenio@gprt.ufpe.br

Djamel Sadok
Universidade Federal de Pernambuco
Recife, PE, Brazil
jamel@gprt.ufpe.br

ABSTRACT

In this work we evaluate the performance and behavior of two widely spread VoIP applications, namely Skype and Google Talk under different network conditions. Using a controlled environment we adopt different values for the capacities of critical links, delay, packet loss and jitter and assume the quality of received audio as the measurement of interest for evaluating its performance. We use the PESQ – an ITU algorithm that compares the original and degradated audio – in order to infer voice quality and evaluate the impact of each network parameter over the quality of received audio. Instead of ranking VoIP P2P applications, this work aims at analyzing various performance aspects and pointing out the observed weaknesses and strengths.

Categories and Subject Descriptors

C.2.5 [**Local and Wide-Area Networks**]: Internet; C.4 [**Performance of Systems**]: *Performance attributes, Measurement technique;* H.4.3 [**Information Systems Applications**]: Communications Applications

General Terms

Measurement, Performance, Experimentation.

Keywords

Peer-to-Peer Systems, Voice over IP

1. INTRODUCTION

The dissemination of Voice over IP (VoIP) technologies is considered the main enabler of telephony cost reduction nowadays. The convergence of voice and data networks makes room for a number of innovations that may change the manner people see communications. The most successful VoIP applications among end-users are those that allow free calls directly among Internet users, in a peer-to-peer (P2P) fashion, such as Skype and GTalk. Although based on the best effort Internet, a network without quality of service (QoS) assurances, such applications often achieve adequate results considering the cost-benefit and the quality standards of traditional telephony networks. These good results are the main reason for the great dissemination of VoIP applications.

One of the main factors influencing the quality achieved in a VoIP call (a session) is the voice coder/decoder (codec) utilized and its configuration parameters. However, the adoption of a codec or the selection of its specific parameters (e.g., bit rate) is not a static choice for an application. The application needs to dynamically adapt to network conditions by choosing different codecs and/or their adequate parameterization. Voice-carrying packets should not excessively be dropped, delayed or subject to high variation in delay to ensure an intelligible audio reception. Therefore, the absence of guarantees and the known variability at several time scales in the Internet traffic profile are elements that greatly impact such applications' performance. These impairments will eventually induce its developers to deploy robust approaches to deal with a variety of traffic profiles.

This work investigates, evaluates and compares Skype and GTalk, both being free VoIP applications broadly utilized in the Internet. Considering that the codecs and their parameters are components defined exclusively by the applications and that cannot be directly manipulated, this work evaluates the behavior of the P2P VoIP applications when submitted to a number of varying network conditions. Among other contributions, this paper evaluates how applications dynamically adapt to network conditions, changing the voice flow characteristics when available capacity increases or decreases. It also evaluates the maximum delay and jitter levels accepted by voice applications for providing a satisfactory voice session and how sensitive are these applications to packet losses in the underlying network. To achieve this purpose, this work proposes a methodology involving a controlled environment for emulating Internet behavior and performing several measurements.

In the rest of the paper, section 2 presents some VoIP fundamentals; section 3 discusses important related work. Section 4 proposes the methodology and also the metrics of interest for evaluating Skype and GTalk. Section 5 presents the performance evaluation regarding the voice quality metrics and discusses their adaptation mechanisms when submitted to different network conditions. Section 6 summarizes some lessons learned and draws some recommendations for application developers. Finally, section 7 presents the conclusions.

NOSSDAV'07, Urbana, Illinois USA.

2. P2P VoIP APPLICATIONS

The essential idea of VoIP technology is to transport voice using an IP network, conceived for transporting data packets. One of the challenges for this convergence is that the best effort Internet does not offer the quality of service guarantees that a conventional telephony network does. Therefore, a high delay in transmission, a high delay variation, or a high packet loss rate has a major impact on the quality of a voice session transported over IP [10].

Skype and GTalk allow programmers to create applications that work together with their code through a closed-source API. They use either TCP or UDP as transport protocol and both applications use the Global IP Sound (GIPS) [1] codec suite. They also inherit features from the peer-to-peer architecture, which is characterized by cooperating and sharing resources among network participants, even if some machines are hidden by NATs and firewalls. This characteristic makes them extremely robust and fault-tolerant, diminishing the possibility of service interruption. Skype relies on an overlay network with only two types of nodes: ordinary nodes and supernodes. The ordinary node is a Skype application that only performs trivial tasks, such as making/receiving voice calls. The supernodes are special nodes spread all over the Internet, besides performing the same tasks an ordinary node does, they also help the Skype network, by managing contact lists and relaying data flows when necessary. Any host with an active Skype client that is capable of receiving connections from the Internet, has a fast network access speed, has enough memory and has a fast enough processor is a candidate for being a supernode. Supernode activity is transparent to the user. There is no user choice on whether to become a supernode or not. Another P2P characteristic in Skype is the use of the *hole punching* technique for traversing NAT boxes.

GTalk adopts an IETF standard as a protocol, freeing its users to use other applications to communicate with. GTalk network provides interoperability with other VoIP networks and other instant messaging networks (e.g., the Gizmo Project). The service is hosted in the google.com site and can be accessed in the port 5222. We did not find any record of a GTalk overlay network.

3. RELATED WORK

VoIP applications in the Internet have attracted research on QoS for IP voice services. In [12], Shen evaluated the performance of VoIP codecs on GPRS networks and showed that the VoIP approach may create some capacity gain over traditional circuit switching, with acceptable guarantees in quality of service. Furuya [4] evaluates the relationship between network parameters (e.g., capacity and delay) and the quality of VoIP services. Although the objectives and the test environments are similar to ours, this work evaluates the dynamic behavior of popular P2P applications, while Furuya's experiments were conducted specifically with the G.711 codec. James et al. [8] evaluate the effect of loss, delay and error recovery, among other factors, in the perceived voice quality using many codecs (e.g., G.711, G.728 and G.729).

Due to its success, a number of research studies have already been developed around Skype. Baset and Schulzrinne [1] were the pioneers in analyzing Skype (version 1.4). Their paper contains an incipient discussion of Skype's network behavior (e.g., quantity of messages exchanged, supernodes location) in the login process, NAT and firewall traversal, call establishment and media transfer. In [11], Guha et al. carried out an experimental study between Sep/2005 and Jan/2006, focusing on the behavior of supernodes and ordinary nodes, by taking into account their exchanged traffic, life cycle and supernodes geographic location. His work serves as a base for P2P VoIP traffic modeling projects. The work of Chen et al. [9] correlates the duration of Skype calls with QoS factors: transmitted rate, delay, jitter and packet loss. Assuming that call duration may affect quality, the work defines and validates the User Satisfaction Index (USI), an index to measure user satisfaction based on QoS factors.

In [13], Suh et al. characterize Skype sessions passing through relays and propose a method to identify this type of traffic. Our work is similar to the work of Hoßfeld [14] in many aspects (methodology and metrics), but his experiments are restricted to 3G UMTS systems and both papers only analyze Skype.

The experiments with P2P VoIP applications conducted in this paper demand additional efforts in understanding the control policies used for application adaptation to changes in the state of the network, since the developers of the applications do not publish the algorithms responsible for such adaptation. To the best of our knowledge, this is the first research study that provides a reasonable comparison between Skype and GTalk audio quality under several network conditions.

4. EVALUATION AND METHODOLOGY

This work has the intention to analyze and to compare Skype and GTalk when submitted to adverse and favorable network conditions, under the aspects of voice quality and adaptability. We understand adaptability as the applications' capacity and efficiency for reacting to changes in network behavior.

One way to analyze these two criteria would be through an analysis of the packet payload generated by the applications, and by doing so, discover the codec used and its parameters. Based on this information, one could infer the voice quality and adaptability through techniques previously discussed, such as Furuya [4] and James [8], or through a model to measure the quality of voice, such as the E-Model [7], a method that obtains voice quality objectively and provides the results based don factors that has influence on the áudio quality (e.g., transmission delay, echo and distortions introduced by the codecs)

However, besides using a proprietary protocol, Skype communication sessions are encrypted, which prevents the analysis of packet content. Although it is possible to get essential information for the analysis of voice quality (codec and its parameters) through Skype's and GTalk's programming API, codecs of both applications studied are proprietary, there are no established models that allow relating these codecs to voice quality levels.

Due to these factors, the proposed methodology considers Skype and GTalk as black boxes and performs measurements at the entry and exit points of the applications (the network interface of the sender and the soundcard of the receiver) to infer performance parameters.

4.1 Experiment Environment

A testbed was built to allow the automation of the experiments (see Figure 1). Machine S (Sender) is responsible for executing the VoIP application, establishing a call and sending the audio flow to machine R (Receiver). R is responsible for executing the VoIP application, receiving and recording the audio flow from S. The software utilized for audio recording was Audacity[1]. The traffic from S to R was captured at both network interfaces of S and R.

[1] http://audacity.sourceforge.net

NAT-S and NAT-R are NAT boxes used to reproduce the same conditions the applications face on the Internet. The network emulator, namely NIST.Net [2], emulates network conditions according to specific parameters for each experiment. Only the traffic among S and R is routed through the network emulator and the traffic from/to S or R to/from the Internet does not suffer interference, thus not interfering in the communication between the application and any other support peer (e.g. supernodes).

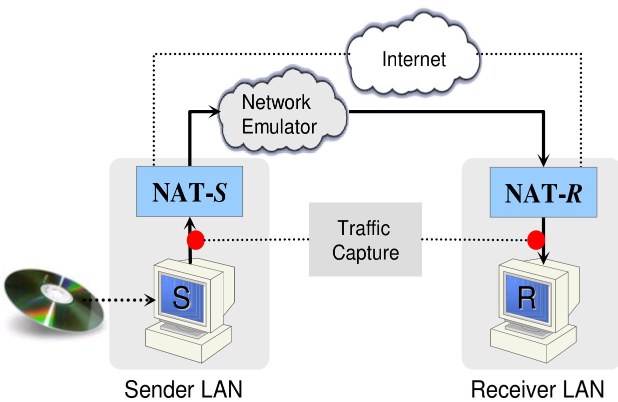

Figure 1 – The testbed for the experiments

We choose a network emulator instead of a simulator or real measurements in the Internet because it allows greater control over the environment and allows the replication of the experiments with the same environmental conditions. NIST.net adjusts the traffic that passes through its interfaces, while being able to modify several network parameters (unidirectional delay, packet loss rate, capacity, router queue size, etc) for different flows and representing the behavior of an entire network. Although not allowing the direct configuration of jitter, it can be approximated in the experiments through a parameter called delaySigma, which represents the standard deviation of delay.

The audio output of a CD player was connected to the microphone input in machine S while the CD player repeatedly reproduced a one-hour long audio sample, which was sent to R through the network emulator. Following recommendations by the standard used for voice quality measure, the audio consisted of a normal conversation between two people and was divided in four 15-minute parts, being two blocks of male voices and two of female voices.

It was necessary to separate the machines S and R in distinct networks because it was detected that, when both peers are in the same LAN, GTalk uses TCP for the voice sessions, while uses UDP when users are located in different LANs. Since this research aims at understanding the behavior of the applications in the Internet, it was necessary to elaborate a special network topology, as shown in Figure 1.

Both applications operate differently when the call is initiated already under adverse network conditions. On such occasions, the traffic relaying occurs frequently, and even after a readjustment for favorable network conditions, the traffic does not flow directly between the two hosts again. Since traffic relaying eliminates the controlled characteristic of the experiments, we prevent it, establishing the desired network conditions in the emulator only after the call is established.

4.2 Metrics of Interest

The performance parameters of the applications evaluated in this work are: a) the quality of the audio received; and b) the transmission rate from the sender to the receiver, which serves directly as a measurement for adaptability. The calculation of the transmitted rate is based on the traffic captured at the network interface of the sender. For that, the tcpstat[2] tool was used.

Although the experiments were conducted in a controlled environment, in this case it is not always possible to control all network variables. For example, when configuring a network path with a lower capacity than the required by applications, some packets are buffered and, consequently, some jitter occurs. Besides, some packets may be lost, and the received rate may be different from the transmitted rate. Knowing that packet loss and jitter do affect the received voice quality, we also measured these parameters to better understand the PES MOS results. The jitter calculation followed the method proposed in [3].

A widely adopted metric for quality evaluation of phone calls is the Mean Opinion Score (MOS) [5] standardized by ITU-T. MOS is a subjective evaluation, calculated by averaging the grades given by a large sum of people that listen to an audio sample that went through a coding/decoding process. The grade is in the range from 1 (bad) to 5 (excellent). Although its result is significant, the difficulties in performing such a large scale evaluation motivated the development of objective techniques for MOS calculation.

The E-Model [7] is a method that calculated the voice quality objectively and provides the results base don factors that influence the áudio quality (e.g., transmission delay, echo and distortions introduced by the codecs). To be able to utilize the E-Model, some information relating to the functioning of the codec are necessary. Since the details of the codecs used in the GIPS (used by GTalk and Skype) are not publicly available, this work could not utilize the E-Model for voice quality evaluation. The Perceptual Evaluation of Speech Quality (PESQ) [6] estimates the MOS of a communication based on the comparison of the audio sent with the audio received. This work utilizes the PESQ MOS as a metric of voice quality.

4.3 Experiments Description

To evaluate applications when submitted to diverse network conditions, we created four scenarios where the following network parameters were tuned in a controlled fashion: a) network capacity in the path from sender to receiver; b) network delay; c) packet loss rate; and d) network jitter. The values of the parameters were defined based on considerations about acceptable configurations for VoIP services available in the literature [10].

The parameters were modified from a favorable to an adverse situation (aggravation) or from an adverse to a favorable situation (progression), hence generating two evaluations. Due to lack of space, only the aggravation evaluation is shown in this work.

Each value assumed for a network parameter is called a level. Levels are adjusted dynamically during a call and, for each level, the same 1-hour long audio is transmitted and 60 1-minute long samples are collected at the receiver. The experiments are repeated with the same input for both Skype and Gtalk.

A preliminary evaluation was performed to discover which codecs can be used by both applications and whether they were changed

[2] http://www.frenchfries.net/paul/tcpstat/

during adaptation. Skype allows the monitoring of a voice session's technical information, and among this, the codec utilized, through a simple choice selection in the application settings. In GTalk, based on the knowledge of its protocol behavior, this investigation was possible through the inspection of the packets' payload during call signaling and during the call itself.

We carried out our experimental study between Aug/2006 and Oct/2006 using Skype version 2.0.0.81 and GTalk version 1.0.0.92.

5. PERFORMANCE EVALUATION

In this work, transmission rates are calculated including IP headers. The results shown in the graphics are the average of the samples. Vertical bars (visible only when significant) represent the confidence interval at a confidence level of 95%.

5.1 Capacity Impact

This scenario investigates the behavior of the applications when the network has critical links of varying capacities: 50, 40, 30, 20 and 15Kbps. Other parameters took the following values: 25ms delay, no explicit packet loss (only loss caused by full router queues) and no induced jitter.

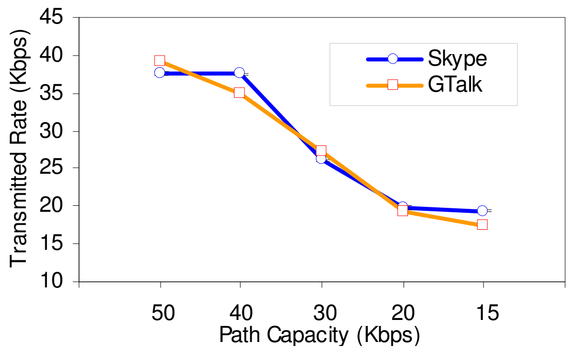

Figure 2 – Throughput for capacity variation

Observing the transmitted rate (Figure 2) and PESQ MOS results (Figure 3), it is possible to observe that when the capacity of the network path is 50Kbps GTalk utilizes more bandwidth, transmitting at a rate higher than Skype, but obtains the same audio quality (the confidence intervals overlap).

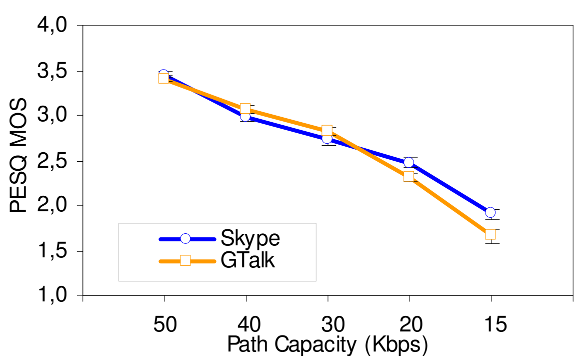

Figure 3 – PESQ MOS for capacity variation

Adjusting the bottleneck link capacity to 40Kbps, GTalk clearly adapted, reducing the transmitted rate to 35Kbps. Such behavior improved its PESQ MOS score when compared to Skype, which in turn did not adapt its sending rate and continued to transmit near to the capacity limit. Both applications achieve similar results when the capacity was adjusted to 30Kbps.

With the capacity in 20Kbps, for the first time, Skype performed better than GTalk and with the capacity in 15Kbps, GTalk adapted its rate to 17.29Kbps, which was not enough to overcome Skype, which achieved a PESQ MOS 5.5% higher.

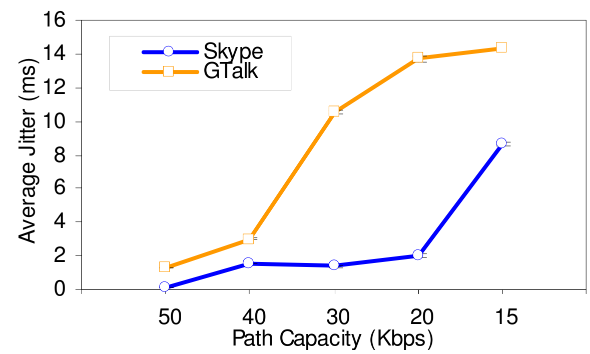

Figure 4 – Average jitter for capacity variation

An interesting observation is that despite both applications transmitted at the same rate when the capacity of the network path was configured to 20Kbps (see Figure 2), they achieved different PESQ MOS scores, as is shown in Figure 3. An explanation for this phenomenon is the ocurrence of the high jitter for GTalk, as can be observed in Figure 4, which depicts the average jitter measured for both applications. Also, analysing the time series of the transmitted rate for GTalk and Skype (Figure 5), one can notice higher variability for the GTalk traffic, which causes queueing at transit network and a subsequent jitter increase.

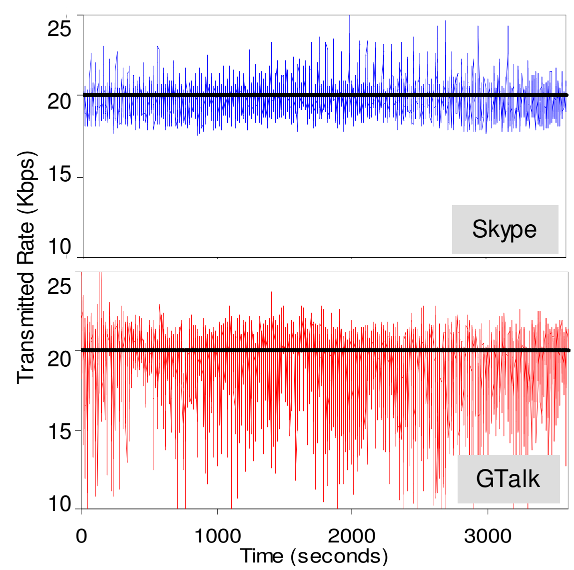

Figure 5 –Transmission rate for a bottleneck link of 20Kbps

5.2 Delay Impact

We are now interested in the impact of the network delay. The capacity was fixed at 50Kbps (superior to the maximum transmitted rate of both applications), the packet loss was configured to be 0% and jitter for 0ms. Following Miras [10], we set the delay level to take acceptable (1ms, 10ms, 100ms) and unacceptable values (500ms and 1000ms) for VoIP applications.

Skype's adaptation policies are clearly more sensitive to delay than GTalk's. When changing from 100ms to 500ms delay, as shown in Figure 6, the transmitted rate changed from 37.5Kbs to 19.36Kbps. However, we did not find a reasonable explanation for such behavior. GTalk did not show any sign of adaptation, since its transmitting rate did not change in this scenario whatsoever.

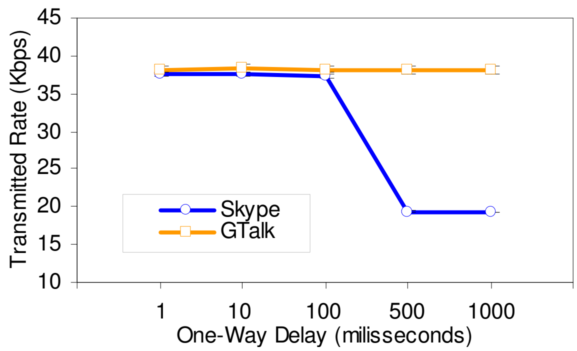

Figure 6 –Throughput for different delays

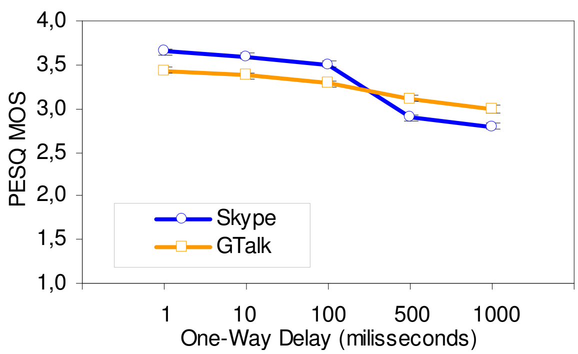

Figure 7 –PESQ MOS achieved under different delays

Observing Figure 7, Skype had a superior performance in terms of ideal network conditions, which means delays of 1ms and 10ms, while GTalk was superior with 500ms and 1000ms for delay.

5.3 Packet Loss Impact

In order to study application behavior under different packet loss rate, the bottlenecked link capacity was at 50Kbps, delay at 25ms and jitter at 0ms, while loss rates were 0%, 1%, 5%, 10%, 20%, 30% and 40%.

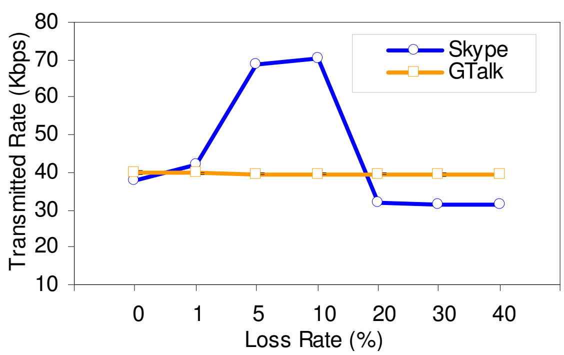

Figure 8 – Throughput for different loss rates

Figure 8 shows that Skype adapted at various levels of loss, while GTalk behavior indicates it has no mechanism of adaptation to packet loss. We raise the hypothesis that Skype added redundant information to the audio flow to reduce the impact of loss on the audio quality, which explains the increase in transmitted rate when the loss rates were 1%, 5% and 10% (disregarding IP headers, the payload was doubled in some cases).

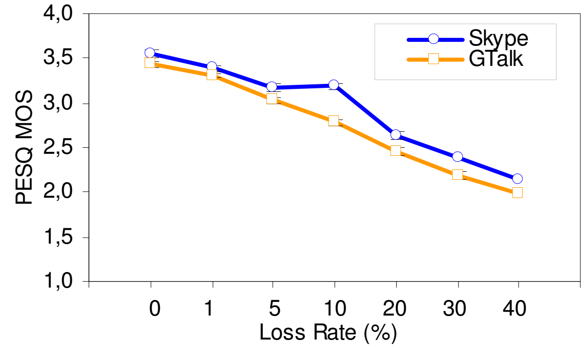

Figure 9 – PESQ MOS for different loss rates

However, these adaptations were not enough to keep the MOS value high (Figure 9). Notably, with the loss rate at 5%, Skype used more redundancy than was necessary, consuming network resources in excess without benefiting significantly its MOS score. We concluded it by observing that with the loss rate at 10%, the MOS value didn`t change.

5.4 Jitter Impact

To study the impact of jitter, the capacity was set to 50Kbps, the loss to 0% and the delay to 100ms. Jitter took the levels 0ms, 20ms, 40ms, 60ms and 80ms, but measurements did not show any indication of adaptation within the applications. It is possible that in case of jitter the applications only alter the size of their buffer, without any visible impact in the transmitted traffic. As expected, the PESQ MOS metric showed a negative correlation with the jitter. In all levels both applications had a very similar behavior (not shown due to lack of space).

6. LESSONS LEARNED

Although we compare two VoIP applications, the goal of our work was not to point out which one is better application. Our aim was to find out open research questions and to come up with recommendations for VoIP application developers.

The choice of the codec has a major influence in the perceived quality and this may be verified by the fact that both Skype and GTalk use proprietary GIPS codecs, even though open codecs such as iLBC exist. However, this is not enough, since applications using the same codec may present different qualities under the same network conditions. Therefore, as far as design decisions are concerned, the use of effective adaptation mechanisms under varying network conditions primarily defines the difference in the quality obtained by VoIP applications. Since varying network conditions is currently the rule for wireless users, adaptation will increasingly play an important role, as convergence goes on. A summarization of the main findings regarding adaptation is presented below.

While adaptation usually yields better results than no adaptation at all, applications should control the impetus to adapt too fast. During the adaptation period the quality becomes unstable as we observed in some preliminary experiments with GTalk. In situations where bandwidth is seriously restricted, GTalk consistently tries to transmit at a higher rate than the permitted by the network, which induces higher jitter levels and drop the PESQ measure. Figure 9 and Figure 3 for 20 and 15 Kbps are examples of the harmful effects of the GTalk aggressiveness. On the other hand, Skype adapts to higher delays by sharply decreasing the transmission rate. While we could not find any reason for that design decision in the literature, it clearly results in worse quality levels.

Adaptations that send redundant information may yield positive results, as depicted by Figure 8 and Figure 9 in the loss scenario. Skype was able to keep PESQ MOS values above 3 up to 10% packet loss with such adaptation, while GTalk did not adapt at all and the quality decreased linearly with the increase in loss rate. However, there is a trade-off in increasing the redundancy level (and bitrate) and the improvement obtained, since transmitting at higher rates is not desirable in most voice systems. More research is needed to come up with smarter redundancy schemes targeted to specific network variation conditions (especially in wireless environments). In general, we think that codecs should be developed having in mind different adaptations strategies, i.e. they should not only be designed to allow different choices to be made, but to assume that these choices would be changed in runtime according to different network conditions.

Finally, adaptations that make triangulations with the supernodes might be performed as a last resort when the quality keeps unacceptable for longer periods. As a matter of fact, GTalk used to rely on this mechanism in previous versions, although it has not been observed in this version. However, considering that triangulation is a workaround for congested routes and frequently congestion occurs at access networks that have single paths, the real benefits of such mechanism would be an interesting research topic.

7. CONCLUSIONS AND FUTURE WORK

This work compared the performance of two P2P VoIP applications: Skype and GTalk. We discussed the dynamic adaptation policies and evaluated the audio quality of applications through the observation mainly of the PESQ MOS and the transmitted rate.

This work showed that Skype and GTalk rely on estimators to analyze the quality of the service offered by the network and control the use of their codecs. Despite both companies affirm their applications use the GIPS codec library, sometimes both the characteristics of the audio stream and the measured audio quality were different under equal scenarios.

Under ideal network conditions, it was observed that, although the difference is very small, the audio transmitted by Skype suffers less degradation than that by GTalk but, under high delay Skype performed unnecessary adaptation. We conclude that GTalk does not implement any mechanism for adaptation when submitted to packet loss, whereas there are strong indications that Skype uses a data redundancy mechanism against loss. Under high levels of jitter, none of the applications adapted their sending rate.

When considering the voice quality aspect for both applications and also including the experiments that vary from an adverse to a favorable situation (progression), one will see that Skype slightly overcame GTalk in 24 occasions, while GTalk performed better only in 4 scenarios and in 14 occasions there was a draw. However, in most occasions where Skype performed better, the PESQ MOS difference was below 0.1, showing that even with Skype's apparent advantage, both applications are very close in terms of voice quality.

It is known that PESQ MOS is not the most adequate algorithm to evaluate voice quality for high values of delay. Therefore, despite the results involving delay variation being interesting, conclusions must be taken carefully. The study of the friendliness of both applications towards TCP flows and also between both applications when competing together for network resources is also an interesting future work.

8. REFERENCES

[1] Baset, Salman A., Schulzrinne, Henning, "An Analysis of the Peer-to-Peer Internet Telephony Protocol", IEEE INFOCOM 2006, April 2006.

[2] Carson, M. & Santay, D., "NIST Net - A Linux-based Network Emulation Tool", ACM SIGCOMM Computer Communication Review, July 2003.

[3] Demichelis, C., Chimento, P., "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.

[4] Furuya, H.; Nomoto, S.; Yamada, H.; Fukumoto, N.; Sugaya, F., "Experimental investigation of the relationship between IP network performances and speech quality of VoIP", 10th International Conference on Telecommunications (ICT 2003), March 2003 .

[5] International Telecommunications Union, "Methods for subjective determination of transmission quality", Recommendation P.800, August 1996.

[6] International Telecommunications Union, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs", Recommendation P.862, February 2001.

[7] International Telecommunications Union, "The E-model, a computational model for use in transmission planning", Recommendation G.107, Dezembro de 1998.

[8] J. H. James, Bing Chen, and Laurie Garrison, "Implementing VoIP: A Voice Transmission Performance Progress Report", IEEE Communications Magazine, July 2004.

[9] Kuan-Ta Chen, Chun-Ying Huang, Polly Huang, Chin-Laung Lei, "Quantifying Skype User Satisfaction," ACM SIGCOMM 2006, Pisa, Italy, September 2006.

[10] Miras, D. "A survey on network QoS needs of advanced internet applications", Working document, Internet2 – QoS Working Group, 2002.

[11] Saikat Guha, Neil Daswani, Ravi Jain, "An Experimental Study of the Skype Peer-to-Peer VoIP System", 5th Workshop on Peer-to-Peer Systems, (IPTPS), February 2006.

[12] Shen, Q., "Performance of VoIP over GPRS", 17th International Conference on Advanced Information Networking and Applications (AINA'03), 2003.

[13] Suh, K.; Figueiredo, D.R.; Kurose, J. and Towsley, D. "Characterizing and Detecting Relayed Traffic: A case study using Skype". IEEE INFOCOM`06, April 2006.

[14] Hoßfeld, T. "Measurement and Analysis of Skype VoIP Traffic in 3G UMTS Systems", IEEE/ACM IPS-MoME'2006, February 2006.