

Avaliação do Desempenho de Aplicações VoIP P2P

Rodrigo Barbosa, Arthur Callado, Carlos A. Kamienski,
Stênio Fernandes, Dênio Mariz, Judith Kelner, Djamel Sadok

Grupo de Pesquisa em Redes e Telecomunicações (GPRT)
Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Cidade Universitária, Recife/PE
{rodrigo, arthur, cak, stenio, denio, jk, jamel}@gpirt.ufpe.br

Abstract. *In this paper we evaluate the performance and behavior of P2P VoIP applications (Skype, Google Talk) under different network conditions. We adopt different values for capacities of critical links, delay and packet loss and assume the quality of received audio as the interest measurement for evaluating its performance. We use the PESQ algorithm to compare the sent and received audio in order to infer MOS and measure 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.*

Resumo. *Este trabalho avalia o comportamento e o desempenho de aplicações VoIP P2P (Skype, Google Talk), quando submetidas a condições variadas da rede. Foram considerados diferentes valores para capacidade de enlaces críticos, atraso e perda e assumimos a qualidade do áudio recebido como parâmetro de desempenho. O algoritmo PESQ foi usado para inferir o valor do MOS baseando-se na comparação entre o áudio enviado e o recebido e o grau de impacto que cada parâmetro da rede oferece para a qualidade do áudio recebido foi mensurado. Ao invés de eleger a melhor aplicação VoIP P2P, este trabalho visa analisar vários aspectos de desempenho e pontuar as qualidades e deficiências apresentadas nos cenários avaliados.*

1. Introdução

A disseminação de Voz sobre IP (VoIP) é atual e principalmente motivada pela redução de custos de telefonia para empresas e consumidores residenciais. Além disso, a convergência do serviço de voz com a rede de dados abre espaço para uma grande variedade de inovações que podem revolucionar a maneira como pessoas e empresas encaram a comunicação. Entre as várias possibilidades da utilização da tecnologia VoIP, a que mais tem tido sucesso entre os usuários finais são as aplicações que permitem ligações gratuitas na Internet, como *Skype*, *Google Talk*, *Yahoo! Messenger* e *Netmeeting*. Apesar de se basearem em uma rede que não oferece garantias, essas aplicações alcançam bons resultados, considerando o custo-benefício entre o preço e os padrões de qualidade oferecidos pelo sistema telefônico convencional, o que tem sido o principal motivo da grande difusão dos aplicativos de VoIP e da eminente revolução no acesso ao serviço público de telefonia.

VoIP não é uma tecnologia nova, mas apenas recentemente vem se apresentando como uma concorrente real às companhias telefônicas. O surgimento do *Skype*

estabeleceu um marco significativo na aceitação da tecnologia, uma vez que ele utiliza uma rede peer-to-peer (P2P) para possibilitar a comunicação direta entre os usuários e conseqüentemente gerar grandes melhorias na qualidade perceptível da ligação. A grande disseminação de sistemas intermediários (e.g. NAT, firewall) nas redes de acesso quebra a semântica original fim a fim da Internet [29] e prejudica o funcionamento de vários protocolos. Por este motivo, a maioria dos sistemas anteriores oferecia o serviço VoIP através da triangulação através de um servidor remoto. Esta característica eleva o atraso na rede e prejudica a interatividade.

Para tratar desse problema, as novas aplicações VoIP P2P possuem mecanismos de contorno de NAT e firewall que permitem a comunicação direta entre usuários. Entre elas, esse artigo avalia e compara o *Skype* com o *Google Talk*, aplicações gratuitas e amplamente difundidas que foram escolhidas devido à sua importância no atual cenário de VoIP na Internet. O objetivo deste trabalho é avaliar o comportamento das aplicações de VoIP P2P quando submetidas a diferentes condições de rede. Em 2003, a empresa *Skype*¹ disponibilizou sua aplicação VoIP gratuita, desenvolvida com parte do conhecimento e experiência adquiridos pelos seus autores com a rede KaZaA, de compartilhamento de arquivos em redes *peer-to-peer*. O *Skype* permite que pessoas com um computador com fones e microfone possam conversar via Internet, ou seja, sem usar a rede telefônica convencional, reduzindo custos principalmente com chamadas de longa distância. O desempenho do *Skype* em termos da qualidade do áudio depende das condições da rede, pelo fato de que a Internet não oferece garantia de serviço, tal como a rede telefônica convencional.

O *Google Talk*² foi lançado em 2005 e, apesar de não oferecer tantos recursos como o *Skype* (suporta atualmente apenas ligações entre computadores e troca de mensagens instantâneas), é uma aplicação VoIP. Ao contrário do *Skype*, o *Google Talk* adota um protocolo padrão para troca de mensagens instantâneas: o Jabber [13][14], de forma que seus usuários não estão presos ao programa para trocar mensagens. A sinalização das chamadas de voz também é aberta, mas ainda não há nenhum programa concorrente que a utilize. De qualquer forma, como o *Skype* utiliza sinalização proprietária para conexões de voz, não é compatível com o *Google Talk*.

Alguns fatores influenciam na qualidade da sessão de voz alcançada por esses aplicativos VoIP, como os codificadores de voz (*codecs*) e as condições dinâmicas da rede. Considerando que o *codec* e o mecanismo de sinalização usados pelas aplicações são componentes previamente selecionados e controlados, o objetivo deste trabalho é avaliar o comportamento dessas aplicações de VoIP P2P quando submetidas a diferentes condições de rede. Foram avaliados como as aplicações se adaptam dinamicamente, mudando as características do fluxo de voz nos casos em que a capacidade disponível no caminho entre dois usuários diminui. Também avaliam qual o atraso máximo suportado pelas aplicações para que uma conversa seja mantida e quão sensíveis são as aplicações à perda de pacotes na rede subjacente.

¹ Skype, <http://www.skype.com>

² GoogleTalk, <http://www.google.com/talk>

Algumas dessas perguntas poderiam ser respondidas mais facilmente com uma análise do conteúdo dos pacotes gerados pelas aplicações. Entretanto, sabe-se que o *Skype*, por exemplo, cifra os dados que trafegam na rede entre o emissor e o receptor, de maneira que não é possível, saber qual o *codec* usado em determinado momento. Assim, a metodologia adotada analisa as aplicações de um ponto de vista externo à aplicação, coletando na rede o maior número de informações possíveis em pontos de entrada e saída da aplicação.

Os resultados indicam que apesar de *Skype* e *Google Talk* afirmarem que adotam a mesma biblioteca de *codecs*, em nenhum experimento as características do fluxo de áudio gerados por eles se assemelharam. As aplicações também diferem na maneira como se adaptam dinamicamente a mudanças nas condições de rede. A política de adaptação do *Skype* altera o *codec* e/ou os seus parâmetros em suas sessões de áudio. O *Google Talk* se adapta realizando uma triangulação da comunicação. Não é objetivo deste trabalho determinar categoricamente que uma das aplicações seja melhor que a outra, mas analisar vários aspectos de desempenho e pontuar as qualidades e deficiências apresentadas nos experimentos executados.

O restante deste artigo é organizado da seguinte maneira. Na seção 2 são apresentados alguns conceitos relacionados a VoIP que são importantes para a compreensão do resto do trabalho. A seção 3 apresenta os trabalhos relacionados identificados na literatura. A seção 4 apresenta a metodologia adotada para observar o comportamento das aplicações e avaliar o seu desempenho frente às condições da rede subjacente, incluindo a descrição do ambiente de testes e os cenários adotados. Os resultados são apresentados e discutidos na seção 5, discutindo o impacto da capacidade dos enlaces críticos da rede, do atraso dos pacotes e da perda na qualidade observada do áudio que chega ao receptor. Comentários finais, conclusões e discussões sobre trabalhos futuros são discutidos na seção 6.

2. Voz sobre IP

Nos anos 80 começou-se a estudar tecnologias de tráfego de voz em redes de pacotes, (*Voice over IP*, VoIP) com o objetivo de fazer uma integração das redes de telefonia e de dados. Surgiram algumas propostas, mas somente nos anos 90 apareceram implementações voltadas para a rede IP.

Em 1998 foi publicada a primeira versão do conjunto de protocolos H.323 [8][9][10], da *International Telecommunication Union* (ITU), para padronizar a sinalização de sessões de voz e vídeo em redes IP. Tal padrão foi adotado pela indústria e ficou conhecido pela sua alta complexidade, encarecendo equipamentos e fazendo com que a maioria dos fabricantes não o implementasse completamente. Posteriormente, em 1999, foi proposto o Session Initiation Protocol (SIP) [11] pela IETF, com o objetivo de prover um protocolo leve e mais simples de implementar, para tratar sessões (e.g., áudio, vídeo e jogos). A principal consequência da criação do SIP foi o aparecimento de mais aparelhos de VoIP e o seu barateamento.

Um *codec* (contração do termo "*COder-DECoder*") é um dispositivo (software ou hardware) usado para converter sinais análogos em sinais digitalizados e vice-versa. Assim, quando usado em aplicações VoIP, o *codec* basicamente converte o áudio em uma seqüência bits para serem encaminhados pela rede IP e é um componente

importante na determinação da qualidade do áudio e no tempo de processamento envolvido na digitalização.

Existem *codecs* de diferentes características, alguns proprietários e outros padronizados por organismos como ITU, ETSI e IETF. Por exemplo, o G.711 padronizado pela ITU em 1988 [16], codifica sons graves com poucas perdas e sons agudos com perdas mais altas, sem afetar significativamente a qualidade do áudio e é usado comumente em centrais telefônicas digitais. Outros *codecs* padronizados pelo ITU apresentam diferentes taxas de amostragem, taxas de bits (a quantidade de bits usada para representar um trecho de áudio) e qualidades de áudio resultante, tais como G.723.1 [17], G.729 [18] e G.722 [19]. O codificador do sistema de telefonia celular *Global System for Mobile* (GSM), padronizado pela ETSI em 1989 [20], alcança qualidade inferior à dos demais *codecs*, mas é bastante usado devido à baixa taxa de bits e ao fato de ser livre de pagamento de licença. O *codec Internet Low Bit-rate* (iLBC) [21] foi padronizado pela IETF em 2004, é livre de licenciamento e foi projetado para lidar com taxas crescentes de perdas de pacotes de forma a ter uma degradação suave na qualidade. Existem ainda outros *codecs* proprietários recentemente desenvolvidos, por exemplo, pela empresa *IP Global Sound*³, capazes de manter boa qualidade de áudio sob diferentes condições da rede.

A adoção da biblioteca de *codecs* da *Global IP Sound* por ambos *Skype* [12] e *Google Talk* [15] levanta suposições de que usam o mesmo *codec* ou *codecs* semelhantes. Entretanto, a adoção de um *codec* ou de parâmetros específicos para um *codec* (ex: taxa de bits) não é necessariamente uma escolha estática por parte da aplicação, de maneira que é possível adotar diferentes *codecs* e/ou diferentes parâmetros dinamicamente, o que chamaremos de "adaptação de *codec*". De fato, há indícios de que as aplicações fazem adaptação de *codec* em função de certas condições da rede, embora esta investigação não seja o foco deste trabalho.

O outro aspecto de influência na qualidade do áudio nas aplicações VoIP envolve as condições da rede IP no momento da sessão de áudio. A especificação ITU G.114 recomenda que o atraso fim-a-fim em um sentido seja inferior a 150ms para um áudio de boa qualidade. Em [27], Walker *et al.* comentam que uma variação do atraso (*jitter*) superior a 50ms causa grande impacto negativo na qualidade do áudio. O *codec* G.729, por exemplo, requer uma perda não superior a 1% para evitar erros audíveis. A capacidade da rede é também importante para suportar aplicações em tempo real como VoIP. Por exemplo, uma chamada VoIP usando o *codec* G.711 operando a uma taxa de bits de 80Kbps terá péssimo desempenho sobre um enlace de 64Kbps, pelo fato de que pelo menos 16Kbps serão descartados (ou seja, 20% de perda), assumindo o uso exclusivo do enlace. Portanto, a ausência de garantias e a reconhecida sazonalidade do perfil de tráfego da Internet são, elementos de grande impacto no desempenho dessas aplicações, o que é uma das razões que motiva a análise conduzida neste trabalho.

3. Trabalhos relacionados

A importância das aplicações VoIP na Internet tem incentivado pesquisas sobre a qualidade de tais serviços de voz, tais que os provedores de serviços Internet (ISP)

³ Global IP Sound, <http://www.globalipsound.com/>

possam avaliar com certo grau de precisão a qualidade de áudio percebida pelos seus usuários. Em [1], Markopoulou *et al.* relacionam a qualidade subjetiva percebida pelo usuário com medições de atraso e perdas em diversos *backbones* da Internet. Por outro lado, o trabalho de Feiten *et al.* [2] trata de técnicas de adaptação para aplicações de áudio no contexto do padrão MPEG-21, bem como discute métricas objetivas e subjetivas de qualidade para *codecs* com alta e baixa taxa de codificação. Os autores mostram que as métricas atuais são perfeitamente adequadas a codificadores com altas taxas. Portanto, a preocupação principal de Feiten, refere-se à precisão de tais métricas em codificadores com baixas taxas. Especificamente para redes sem fio, em [7], Shen avaliou o desempenho de codificadores VoIP em redes GPRS. O trabalho mostrou que em alguns casos, a abordagem VoIP sobre GPRS pode implicar em ganho de capacidade relativo a tradicional comutação por circuitos, com garantias aceitáveis em qualidade de serviço.

Em pesquisas similares à nossa, os trabalhos de Nichols *et al.* [6] e Chung *et al.* [5], tratam da avaliação do comportamento dinâmico de aplicações populares para transmissão de vídeo, a saber, *Real Video Player* (RVP) [5] e *Windows Streaming Media* (WSM) [6]. Em [6], Nichols avalia o grau de reação do WSM às diversas condições da rede. Sua principal contribuição é a caracterização da taxa de bit do WSM às mudanças na capacidade da rede e à taxa de perdas. Uma importante conclusão dos autores é a constatação de que apesar de ser reativo às mudanças do estado da rede, o WSM pode ser não-amigável com fluxos TCP (TCP Unfriendly). Já em [5], os autores medem adaptação do RVP sobre UDP através da medição de desempenho da transmissão de diversos cliques de vídeos. Sua principal métrica é a amigabilidade com fluxos TCP. Em geral, os autores mostram que os fluxos UDP do RVP reagem bem em períodos de congestionamento na rede. Vale enfatizar que os autores usam um *testbed* com *NIST.net* [28] para controlar os experimentos, através da variação de parâmetros da rede, especificamente a capacidade do gargalo da rede. Em conclusões similares ao trabalho de Nichols [6], os autores encontraram evidência de comportamento não-responsivo do RVP às condições da rede.

O trabalho de Furuya [3] avalia a relação entre a variação de diversos parâmetros de rede (e.g., capacidade, atraso e tamanho do buffer no gargalo da rede) e a qualidade dos serviços de VoIP. Além disso, os autores avaliam métricas de rede, tais como utilização do enlace e *jitter*. Apesar de objetivos e configuração do ambiente de testes serem similares, nosso trabalho avalia o comportamento dinâmico de aplicações populares de VoIP P2P, enquanto os experimentos de Furuya foram conduzidos especificamente com o codificador G.711. Numa linha similar ao trabalho de Furuya, James *et al.* [4] avalia o efeito de perdas, atraso e técnicas de recuperação de erros, entre outros fatores, de diversos codificadores (e.g., G.711, G.728 e G.729) na qualidade percebida de voz. Finalmente, o trabalho de Hoßfeld [31] é semelhante em diversos aspectos (metodologia e métricas), mas seus experimentos são limitados, pois são restritos a sistemas 3G UMTS. Além disso, analisa apenas o Skype e não compara com o Google Talk.

Nossos experimentos com aplicações VoIP P2P exigem esforços adicionais na compreensão das políticas de controle para adaptação à mudanças do estado da rede, uma vez que seus desenvolvedores não revelam os *codecs* usados, nem os algoritmos responsáveis por estas adaptações.

4. Metodologia dos Experimentos

Como meio de avaliar o seu desempenho, submetemos o fluxo de áudio enviado pelas aplicações VoIP P2P a variadas condições da rede e assumimos como parâmetro de desempenho a qualidade do áudio recebido, a vazão de dados entre o emissor e o receptor, que oferece uma medida para a adaptabilidade da aplicação às condições da rede e o *jitter* observado no tráfego enviado (não inclui o *jitter* do *codec*). O *jitter* foi medido, pois foi um parâmetro não controlado nos experimentos e que auxilia a compreensão de alguns resultados.

Uma métrica mundialmente adotada para avaliação da qualidade em ligações telefônicas é o *Mean Opinion Score* (MOS)⁴ [22]. O MOS é uma abordagem de avaliação subjetiva computado como a média das notas individuais atribuídas por um grande número de pessoas que ouvem um áudio resultante de um processo de codificação e decodificação, onde a nota varia de 1 (ruim) a 5 (excelente). O MOS é padronizado pelo ITU-T através da recomendação P.800, a qual exige ambiente controlado e condições especiais para os testes de opinião. Embora seu resultado seja bastante significativo, a dificuldade de realizar tal avaliação em larga escala motivou o desenvolvimento de técnicas objetivas para o cálculo do MOS.

Em 1998, foi desenvolvido o Modelo-E [23] para o cálculo objetivo do MOS utilizando apenas o lado receptor da conversa, em função de métricas de rede como a taxa de perdas e do tipo de *codec* utilizado. No mesmo ano, foi proposto o *Perceptual Speech Quality Measurement* (PSQM) [24], que estima o MOS de uma comunicação com base na comparação entre o áudio transmitido e o recebido. Em paralelo a este, a *British Telecom* desenvolveu o *Perceptual Analysis Measurement System* (PAMS) [25], com os mesmos objetivos. Posteriormente, a ITU desenvolveu o *Perceptual Evaluation of Speech Quality* (PESQ) [26], combinando características do PSQM e do PAMS e melhorando os algoritmos de forma a atender uma gama maior de cenários (como cenários com ocorrência de *jitter*).

O PESQ é, portanto, um método automatizado para avaliação da qualidade do áudio recebido que faz uma predição do MOS equivalente. Para medir a qualidade, o PESQ baseia-se na comparação do áudio original X com o áudio recebido Y, possivelmente degradado ao ser transportado pelo sistema de comunicação. O resultado da comparação do sinal original com o degradado é o score de qualidade, análogo ao MOS, de acordo com a recomendação ITU-T P.800.

Nos experimentos realizados, o tráfego gerado do emissor para o receptor é capturado para computar as demais métricas: a vazão e a variação do atraso. Para calcular a vazão baseado no tráfego capturado, adotamos o *tcpstat*⁵.

O *jitter* é definido sobre dois pacotes quaisquer contidos em um fluxo, sendo calculado pela diferença entre os atrasos unidirecionais. Ao contrário do atraso, o *jitter*, quando calculado para pacotes consecutivos (o que é o mais comum), independe da sincronização de fase dos relógios (i.e., se os relógios de ambos os *hosts* marcam a

⁴ Wikipedia, http://en.wikipedia.org/wiki/Mean_Opinion_Score

⁵ *tcpstat*, <http://www.frenchfries.net/paul/tcpstat/>

mesma hora). Ainda assim, o cálculo de *jitter* depende da sincronização de frequência dos relógios (i.e., se um relógio se atrasa ou adianta em relação a outro com o passar do tempo). De qualquer forma, o erro de sincronização na frequência é várias ordens de magnitude inferior ao valor medido e pode ser ignorado. Isso permite fazer cálculos de *jitter* mesmo sem garantias de sincronização de relógios. Em resumo, o cômputo do *jitter* seguiu o método sugerido pelo WG IPPM, da IETF [30].

Para calcular o *jitter* dos experimentos, uma ferramenta foi desenvolvida em C++, usando a *libpcap*⁶. A ferramenta que tem como entrada dois arquivos gerados pelo *TCPDump*, *ethereal* ou *tethereal*⁷: o primeiro arquivo contém os pacotes capturados na máquina que originou o fluxo e o segundo possui os pacotes capturados na máquina que recebeu o fluxo de voz.

A avaliação do desempenho das aplicações é feita em um ambiente de rede controlado e envolve três cenários, onde cada cenário considera um mesmo áudio de 60 minutos dividido em 60 replicações de um minuto cada. Para cada cenário, são definidas diferentes condições de rede às quais as aplicações são submetidas. O ambiente de realização dos experimentos é definido na seção 4.1 e os cenários avaliados estão detalhados na seção 4.2.

4.1. O Ambiente de Realização dos Experimentos

A Figura 1 ilustra o ambiente montado para realização dos experimentos, onde foram utilizados 03 computadores tipo PC e um tocador de CD. O ambiente de execução foi elaborado de forma a permitir a automatização e a replicação dos experimentos.

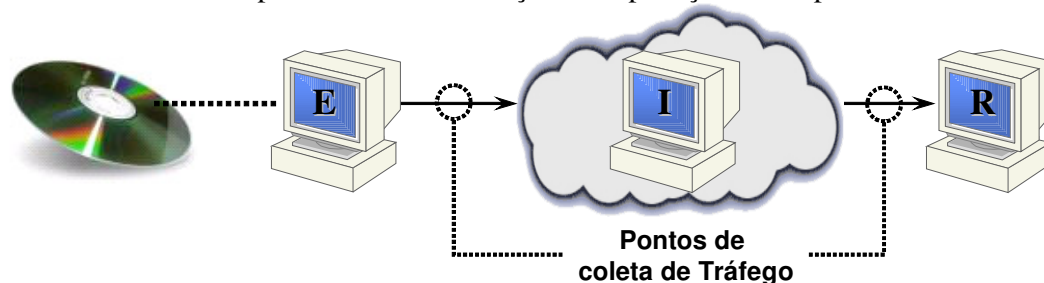


Figura 1. Ambiente de realização dos experimentos.

A máquina *E* (emissor) é responsável por executar a aplicação VoIP, estabelecer uma chamada e enviar um fluxo de áudio com destino final à máquina *R* (receptor). A máquina *R* é responsável por executar a aplicação VoIP, receber e gravar o fluxo de áudio recebido de *E*. A gravação do áudio em *R* serve para o cálculo do MOS, realizado pela implementação de referência do algoritmo PESQ [26]. O software utilizado para gravação do áudio recebido foi o *Audacity*⁸.

Um tocador de CD foi usado para reproduzir o áudio enviado de *E* para *R*. A saída de áudio do tocador de CD é conectada à entrada do microfone na máquina *E* e o

⁶ libpcap, <http://www.tcpdump.org>

⁷ Ethereal, <http://www.ethereal.com/>

⁸ Audacity, <http://audacity.sourceforge.net/>

tocador de CDs reproduz repetidamente um CD com duração de 1 hora, dividido em quatro partes de 15 minutos. Cada parte é uma gravação que imita o comportamento de uma pessoa falando ao telefone, incluindo pausas. Por recomendações do padrão PESQ, das quatro amostras de 15 min, duas vezes são masculinas e duas são femininas.

As máquinas *E* e *R* possuem tabelas de roteamento ajustadas para que todo o tráfego entre elas seja encaminhado através da máquina *I* (emulador Internet), que emula as condições da rede de acordo com parâmetros específicos para cada experimento realizado. Adotamos um emulador de rede ao invés de um simulador ou do uso de medições reais na Internet porque este permite maior controle do ambiente e garante a repetibilidade dos experimentos. A ferramenta de emulação usada foi o *NIST.Net* [28]. Embora o tráfego entre *E* e *R* seja roteado através de *I*, ambas as máquinas *E* e *R* têm acesso normal à Internet pública, para que possam ser autenticadas pela aplicação, acessar suas listas de contatos, fazer a chamada e iniciar a sessão.

Durante os experimentos o tráfego de saída de *E* e o tráfego de entrada em *R* foram capturados e usados para cálculo da vazão e *jitter*. A captura do tráfego foi feita usando o *tethereal*.

A máquina *E* é um PC com processador Athlon 2.66 GHz e 512 MB RAM com Windows XP, a máquina *R* é um PC com processador Pentium 4, 3.0 GHz e 1 GB RAM com Windows XP e a máquina *I* é um PC com processador Athlon 1.5GHz e 256MB RAM com Linux Slackware 10.1, kernel versão 2.4.

4.2. Descrição dos Cenários

Os cenários adotados para a realização dos experimentos são descritos na Tabela 1. Os cenários *A*, *B* e *C* foram estabelecidos com o objetivo de avaliar o impacto da variação das condições da rede na qualidade das sessões de áudio e no comportamento das aplicações. Cada cenário estabelece alguns parâmetros fixos e varia um dos parâmetros em cinco níveis diferentes durante a transmissão do áudio sobre a rede.

Tabela 1. Descrição dos cenários

Cenário		Capacidade	Atraso	Perda
Cenário A	A1	100Kbps	10ms	0%
	A2	50Kbps		
	A3	40Kbps		
	A4	30Kbps		
	A5	20Kbps		
Cenário B	B1	100Kbps	1ms	0%
	B2		10ms	
	B3		100ms	
	B4		500ms	
	B5		1000ms	
Cenário C	C1	100Kbps	10ms	0%
	C2			1%
	C3			5%
	C4			10%
	C5			20%

O cenário *A* busca estudar a qualidade das sessões de áudio e o comportamento das aplicações quando a rede possui enlaces críticos de diferentes capacidades. Os valores representam a capacidade residual de um caminho, também chamada de banda disponível. Como nesse cenário o interesse é estudar o impacto da variação da capacidade, adotamos um baixo atraso e perda de pacotes nula.

O cenário *B* visa estudar o impacto do atraso na qualidade da sessão de áudio e no comportamento das aplicações. Os valores de atraso indicados na tabela se referem ao atraso em um sentido (*one way delay*). No nível *B3*, por exemplo, o atraso em um sentido é de 100ms e o RTT (*round trip time*) de 200ms. A capacidade residual foi fixada em 100Kbps, pois se constatou que nenhuma das aplicações envia fluxos com vazão acima desse valor. A perda de pacotes foi configurada em 0% para todos os experimentos do cenário *B*.

O cenário *C* foi elaborado para estudar a qualidade da sessão de áudio e o comportamento das aplicações sob diferentes taxas de perda de pacotes. Nesse cenário a capacidade e o atraso são fixados em 100Kbps e 10ms, respectivamente.

Os modelos de atraso e perda são os utilizados pelo emulador citado, NIST.Net.

5. Avaliação de Desempenho

Em relação aos aspectos gerais da adaptação dinâmica das aplicações, chegaram-se aos seguintes resultados:

Antes de iniciar a conversa, no processo de estabelecimento de uma chamada entre dois usuários, o *Skype* investiga as condições de rede e determina se a comunicação será de forma direta ou através de uma terceira máquina que realizará o papel de intermediar a sessão de áudio. Uma vez que o caminho da comunicação é escolhido (direto ou indireto), ele permanece durante toda a chamada, mesmo que as condições de rede mudem. Percebeu-se também que os comportamentos dos fluxos enviados pelo *Skype* mudam à medida que as condições de rede variam. Em todos os cenários, apesar do número médio de pacotes enviados ser constante, outras características do fluxo transmitido pelo *Skype* mudam, como por exemplo, os tamanhos médio, máximo e mínimo dos pacotes.

O *Google Talk* se comporta de forma oposta ao *Skype*. No momento que a aplicação percebe que as condições de rede são adversas, mesmo que a chamada esteja em curso, a triangulação ocorre. Entretanto, as características de transmissão do fluxo nunca variam. Em todos os cenários, o *Google Talk* transmite o fluxo de áudio a uma média de 25Kbps, com muita variabilidade.

Portanto, ambas as aplicações se adaptam dinamicamente a variações nas condições de rede, onde o *Skype* se adapta mudando as características de transmissão do fluxo de voz (provavelmente mudando o *codec*, ou alterando parâmetros do *codec*), e o *Google Talk* se adapta realizando uma triangulação na comunicação, procurando caminhos melhores para comunicação.

Em condições ideais de rede, com atraso mínimo, taxa de perda de pacotes nula e banda disponível suficiente, o *Skype* apresentou uma qualidade de áudio melhor que o *Google Talk*, pois atingiu valores maiores de PESQ MOS (3.9 contra 3.4).

Nesse artigo, os valores relacionados à taxa de transmissão incluem o cabeçalho IP e todos os resultados mostrados incluem intervalos com nível de confiança de 95%, na forma de barras verticais sobre as médias obtidas, visíveis onde o intervalo é significativo.

5.1. Impacto da capacidade residual

Nesta subseção, discute-se detalhadamente o impacto causado pela redução da banda disponível no comportamento das aplicações e na qualidade de áudio.

Nos cenários com capacidade de 100, 50 e 40Kbps, respectivamente, apesar de a capacidade residual diminuir gradativamente, o MOS obtido com o *Google Talk* ficou estável, em aproximadamente 3.4, pois a aplicação utilizou menos banda que a disponível e não enfrentou condições adversas de rede.

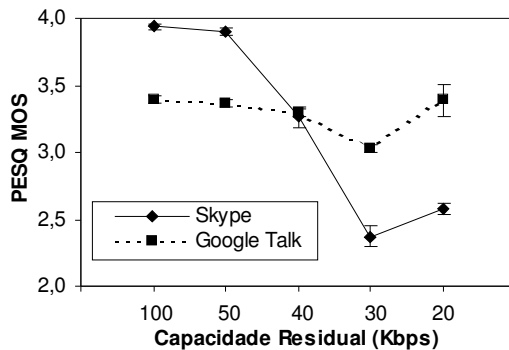


Figura 2. MOS para cenários variando a capacidade residual

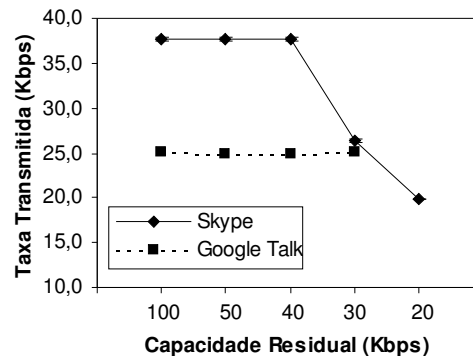


Figura 3. Taxa transmitida para cenários variando a capacidade residual

Observando isoladamente o *Skype*, os mesmos cenários de 100, 50 e 40Kbps tiveram valores semelhantes relacionados à vazão, com média aproximada de 37.7Kbps. O tamanho médio dos pacotes foi de 140Bytes, com desvio padrão do tamanho dos pacotes de 21Bytes. Como a taxa transmitida média nesses três cenários foi abaixo das capacidades configuradas, esperavam-se valores de MOS próximos. Entretanto a Figura 2 ilustra uma diminuição significativa do MOS no cenário de 40Kbps. Observando a Figura 4, percebe-se que houve picos acima de 40Kbps na taxa transmitida. Esses picos causaram enfileiramento de pacotes na máquina emulada, seguido de um conseqüente acréscimo dos *jitters* médio e máximo, que no cenário de 50Kbps eram de 0.04ms e 18.74ms e foram para 2ms e 33.29ms (Figura 5 e Figura 6). Apesar de ainda estar em uma faixa aceitável para voz, acredita-se que o alto valor do *jitter* influenciou o cálculo do MOS, pois de acordo com [26], o algoritmo do PESQ é bastante sensível ao *jitter*. Apesar de a qualidade da conversa ter diminuído no cenário de 40Kbps, o *Skype* não se adaptou dinamicamente às novas condições de rede.

Comparando as aplicações nos dois primeiros cenários, pelos gráficos de MOS e taxa transmitida, conclui-se que o *Skype* obteve um desempenho superior ao *Google Talk*, pois utilizou melhor a banda disponível para transmitir o áudio com um *codec* de maior qualidade que o do *Google Talk*, não subutilizando os recursos da rede. No terceiro cenário, com 40Kbps, as aplicações tiveram uma degradação de sinal semelhante.

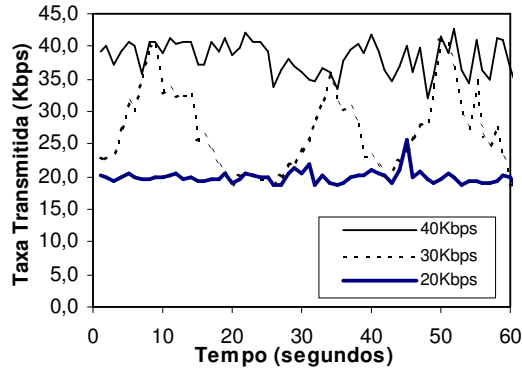


Figura 4. Taxa transmitida pelo Skype em 1 minuto de conversa para cenários variando a capacidade residual

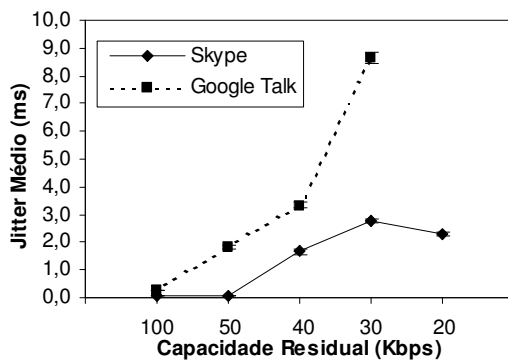


Figura 5. Jitter médio para cenários variando a capacidade residual

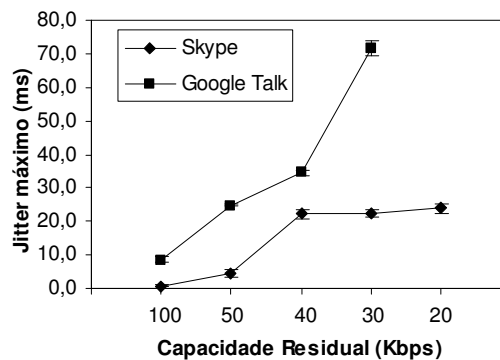


Figura 6. Jitter máximo para cenários variando a capacidade residual

Diferente dos três primeiros, o cenário de 30Kbps foi o primeiro que despertou no Skype uma adaptação dinâmica nas suas taxas de transmissão devido às novas condições de capacidade impostas pela rede emulada. Nesse cenário, a vazão média enviada pelo Skype mudou para 26Kbps e o tamanho médio dos pacotes para 100Bytes. À primeira vista, ele mudou o *codec*, ou mudou alguns parâmetros do *codec*, ou diminuiu a redundância dos pacotes. Supõe-se que houve uma mudança no *codec* ou em seus parâmetros devido ao fato do PESQ calcular um MOS menor, apesar da taxa de perdas ser zero. Essa hipótese é reforçada pela Figura 4, onde a curva de vazão para uma capacidade de 30Kbps se diferencia bastante das outras.

Analisando o Google Talk no quarto cenário, percebe-se um aumento brusco no jitter, causado pela alta variabilidade da taxa de transmissão. Entretanto, houve apenas uma ligeira queda no MOS, ao contrário do Skype, onde o aumento no jitter foi seguido de uma brusca queda no MOS. Há suspeita que o Google Talk é mais tolerante a jitter que o Skype. Outra conclusão é que nesse cenário, apesar de o Google Talk (MOS 3.03) não realizar adaptação, foi superior ao Skype, que embora tenha utilizado uma política de adaptação, teve uma pontuação de 2.37 para o MOS.

Ao se reduzir a capacidade de 30 para 20Kbps, acredita-se, pela Figura 4, que o *Skype* alterou novamente o *codec* ou os parâmetros do *codec* utilizado. Nesse cenário, o tamanho médio dos pacotes gerados pelo *Skype* foi de aproximadamente 74Bytes. Um fato curioso é que o MOS do *Skype* foi superior ao MOS no cenário de capacidade maior, de 30Kbps. Novamente, poderia ser explicado pelo alto valor do *jitter* médio apresentado no cenário de capacidade de 30Kbps, que é consequência dos picos de vazão que excedem a capacidade da rede, associado à alta variação do fluxo gerado pelo *Skype*. Portanto, nos cenários de 30 e de 20Kbps se percebeu uma forte deficiência do *Skype*, que não soube se adaptar eficientemente às mudanças nas condições de rede.

Dois minutos após o início do cenário de 20Kbps, o *Google Talk* manifestou uma adaptação dinâmica pela primeira vez, realizando a triangulação da comunicação através de uma máquina com IP 216.239.37.194, pertencente ao Google. Como ilustrado pela Figura 2, houve uma melhora significativa dos MOS e o *Google Talk* nesse cenário foi novamente superior ao *Skype*. A Figura 3, a Figura 5 e a Figura 6 não possuem valores para o *Google Talk* no cenário de 20Kbps devido à triangulação.

Embora não enumerado na Tabela 1, para o *Skype* foi realizado um experimento onde a capacidade foi de 20 para 15Kbps. Nesse experimento, a frequência do som no receptor diminuiu gradativamente, se assemelhando a um vinil em baixa rotação. Após 11 minutos a chamada foi desligada aparentemente através de pacotes TCP Reset enviadas pelo receptor. As 11 amostras de MOS tiveram como valor médio 1.42. Conclui-se que o *Skype* consegue se adaptar dinamicamente até que a banda disponível seja de aproximadamente 20Kbps.

5.2. Impacto do atraso

Nesta subseção, discute-se detalhadamente o impacto causado pelo atraso no comportamento das aplicações e na qualidade de áudio.

Para o *Skype*, através de cálculo de valores de tamanho médio, máximo e mínimo dos pacotes, quantidade de bytes e de pacotes enviados por unidade de tempo, verificou-se que os fluxos nos experimentos com atrasos de 1ms e 10ms têm o mesmo comportamento dos fluxos nos experimentos com capacidades de 100Kbps, 50Kbps e 40Kbps, onde provavelmente o mesmo *codec* é utilizado.

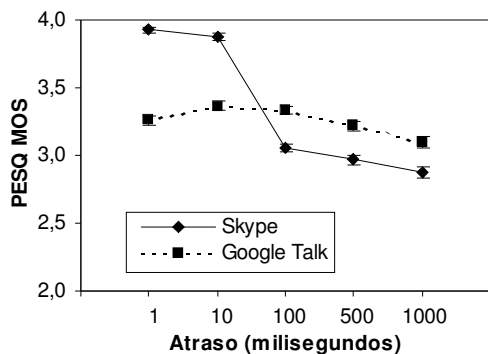


Figura 7. MOS para cenários variando o atraso

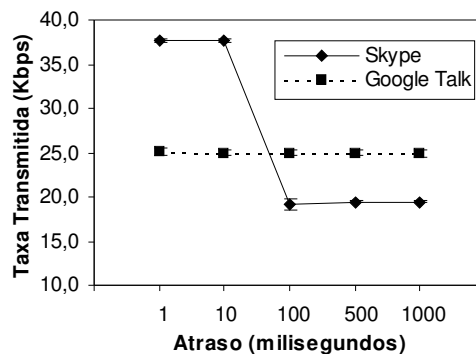


Figura 8. Taxa Transmitida para cenários variando o atraso

O *Skype* alterou as características de envio do fluxo quando o atraso foi configurado para 100ms, apesar de 100ms de atraso (em um sentido) ser um valor razoável para voz [27]. Acredita-se que o *Skype* restringiu o *codec* utilizado na série de experimentos para um *codec* com uma maior degradação do sinal, onde o MOS foi de 3.05. Nesse mesmo cenário, a banda utilizada caiu consideravelmente. Os cenários de 500ms e 1s apresentaram resultados semelhantes ao cenário de 100ms. Supõe-se que ao observar tal nível de atraso, o *Skype* tenha estimado que a tecnologia utilizada em pelo menos uma das pontas seja de linha discada e conseqüentemente resolveu utilizar um *codec* com menor taxa.

O *Google Talk* em todos os cenários apresentou resultados semelhantes. A aplicação desenvolvida pelo Google foi superior ao *Skype* no cenário com 100ms de atraso, pois apresentou uma pontuação maior para o MOS que o *Skype*. Nos cenários com 500ms e 1000ms de atraso, esperava-se que o *Google Talk* realizasse a triangulação, são valores inaceitáveis para uma sessão de áudio. Ainda sobre esses cenários, apesar de o *Google Talk* alcançar valores maiores de MOS que o *Skype*, eles não refletem a qualidade da conversa, pois o PESQ desconsidera o atraso no cálculo do MOS.

5.3. Impacto da perda de pacotes

Nesta subseção, discute-se detalhadamente o impacto causado pela perda de pacotes no comportamento das aplicações e na qualidade de áudio.

A Figura 10 mostra que no cenário com a perda configurada de 1%, houve uma mudança na taxa transmitida pelo *Skype*, que saltou de um valor médio de 37.70 Kbps para 41.62 Kbps. Sobre a política de adaptação do *Skype*, levantam-se duas hipóteses. A primeira é que o *Skype* detectou perda de alguns pacotes de seu fluxo de voz transmitido e passou a incluir informações redundantes para diminuir o impacto da perda de pacotes. Nesse caso, essa mudança não foi suficiente para manter o valor do MOS, que decresceu de 3.92 para 3.70 (Figura 9). A segunda hipótese é que o *Skype* tenha mudado o *codec* para um menos sensível à taxa de perdas.

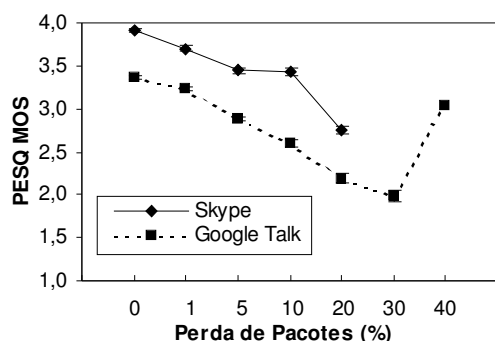


Figura 9. MOS para cenários variando a perda de pacotes

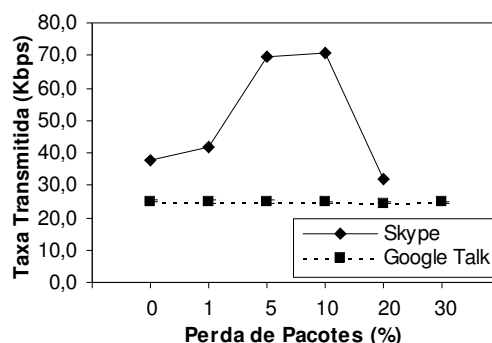


Figura 10. Taxa transmitida para cenários variando a perda de pacotes

A adaptação do *Skype* às novas condições de rede impostas pelo cenário configurado com 5% de perda, apresentou um comportamento semelhante à adaptação realizada na mudança do cenário de 0% para o de 1% de perda, onde se percebeu que a

taxa transmitida aumentou. Desconsiderando-se os cabeçalhos, IP, UDP e supondo que o *Skype* também usa alguns bytes para cabeçalho, pode-se afirmar que a carga útil praticamente dobrou na adaptação do segundo cenário para o terceiro cenário. Além de piorar as condições de rede em situações onde a perda é causada por congestionamento, a adaptação não foi suficiente para manter o valor do MOS.

Analisando o *Skype*, na transição do terceiro cenário (5% de perda) para o quarto cenário (10% de perda), aparentemente não houve adaptação, pois a taxa de transmissão nos dois cenários apresentaram as mesmas características. Um aspecto curioso é que o MOS se manteve constante. Acredita-se que o *codec* utilizado não provoca degradação do sinal quando é submetido a um aumento na perda de pacotes. O mesmo valor do MOS nos cenários de 5% e 10% também pode ser um indicativo que, no cenário de 5%, o *Skype* tenha aplicado um esquema de redundância mais forte que o suficiente, consumindo largura de banda desnecessária, evocando assim, uma deficiência no *Skype*.

O quinto cenário (20% de perda) mudou abruptamente as características do fluxo de voz do *Skype*. A aplicação alterou a taxa de transmissão de dados, reduzindo-a para mais da metade e como esperado, o MOS diminuiu. No último cenário, a ligação caiu após 10 minutos.

Nos cinco primeiros cenários, o *Google Talk* foi inferior ao *Skype* e apresentou, a cada cenário, uma diminuição do valor do MOS devido às perdas de pacotes. Entretanto, ao contrário do *Skype*, no cenário com 30% de perda, o *Google Talk* sustentou a ligação. Onze minutos após o início do cenário onde a perda foi configurada para 40%, o *Google Talk* realizou a triangulação e com isso o MOS sofreu um aumento.

6. Conclusões

Esse artigo avaliou e comparou o desempenho de duas aplicações de voz sobre IP P2P: *Skype* e *Google Talk*. Foram discutidas as políticas de adaptação dinâmica das aplicações através da observação do PESQ MOS, da taxa transmitida e do *jitter*.

Percebeu-se que apesar de ambas as aplicações afirmarem que adotam a biblioteca de *codecs* da Global IP Sound, em nenhum experimento as características do fluxo de áudio do *Skype* e do *Google Talk* se assemelharam. As aplicações também diferem na maneira como se adaptam dinamicamente a mudanças nas condições de rede. A política de adaptação do *Skype* altera o *codec* e/ou os seus parâmetros em suas sessões de áudio. O *Google Talk* se adapta realizando uma triangulação da comunicação.

Em condições ideais de rede, verificou-se que o áudio transmitido pelo *Skype* sofre menos degradação que o áudio transmitido pelo *Google Talk*. Entretanto, quando submetido a condições de rede muito abaixo do aceitável (e.g. perdas acima de 40% e capacidade residual inferior a 20Kbps), o *Google Talk* tem a vantagem de mudar o caminho da comunicação durante a chamada (triangulação), ao contrário do *Skype*, que desliga a chamada. Em cenários variando a capacidade residual, deduziu-se que o *Google Talk* é menos sensível a *jitter* do que o *Skype*. Quando submetido à perda de pacotes, foi observado que o mecanismo de adaptação do *Google Talk* demora a reagir.

Os resultados revelam indícios de que o *jitter* da rede tem impacto direto sobre a qualidade das sessões de áudio e sobre a política de adaptação das aplicações. Embora não tenha sido considerado neste trabalho, o controle explícito do *jitter* da rede poderá

mensurar o grau de impacto deste parâmetro sobre o comportamento das aplicações, e poderá ser importante tópico de pesquisa. Cenários onde a variação da capacidade, atraso, perda e *jitter* variam de forma decrescente e depois crescente poderão avaliar como as aplicações se adaptam e se recuperam de condições desfavoráveis. A inclusão de outras aplicações na análise e o estudo da amigabilidade dessas aplicações com o TCP (*TCP friendliness*) em um cenário controlado também é um interessante tópico a se investigar.

7. Referências

- [1] Athina P. Markopoulou, Fouad A. Tobagi, Mansour J. Karam, “Assessment of VoIP Quality over Internet Backbones”, IEEE INFOCOM 2002
- [2] Bernhard Feiten, Ingo Wolf, Eunmi Oh, Jeongil Seo, and Hae-Kwang Kim, “Audio Adaptation According to Usage Environment and Perceptual Quality Metrics”, IEEE Transactions on Multimedia, Vol. 7, No. 3, June 2005
- [3] 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 Volume Telecommunications, 2003. ICT 2003. 1, 23 Feb.-1 March 2003 Page(s):543 - 552 vol.1
- [4] J. H. James, Bing Chen, and Laurie Garrison, Implementing VoIP: A Voice Transmission Performance Progress Report, AT&T, IEEE Communications Magazine, July 2004.
- [5] Jae Chung, Mark Claypool, Yali Zhu, “Measurement of the Congestion Responsiveness of RealPlayer Streaming Video Over UDP”, In Proceedings of the International Packet Video Workshop (PV), Nantes, France, 2003
- [6] James Nichols, Mark Claypool, Robert Kinicki, and Mingzhe Li, “Measurements of the Congestion Responsiveness of Windows Streaming Media”, NOSSDAV’04, June 16–18, 2004, Kinsale, County Cork, Ireland.
- [7] Shen, Q., “Performance of VoIP over GPRS”, Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA’03)
- [8] International Telecommunications Union. “Packet-based multimedia communications systems”, Recommendation H.323, Fevereiro de 1998.
- [9] International Telecommunications Union, “Call Signalling protocols and media stream packetization for packet-based multimedia communication systems”, Recommendation H.225, Fevereiro de 1998.
- [10] International Telecommunications Union, “Control Protocol for multimedia communication”, Recommendation H.245, Setembro de 1998.
- [11] Rosenberg, J., Schulzrinne, H. *et al.* “SIP: Session Initiation Protocol”, RFC 3261, Junho de 2002.
- [12] Baset, Salman A. Et Schulzrinne, Henning, “An Analysis of the Peer-to-Peer Internet Telephony Protocol”, Setembro de 2004.
- [13] Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Core”, RFC 3920, Outubro de 2004.
- [14] Saint-Andre, P., “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence”, RFC 3921, Outubro de 2004.

- [15] Website do Global IP Sound, <http://www.globalipsound.com/newsroom/newsroom.php?newsID=106&tot=83>, Último acesso em dezembro de 2005.
- [16] International Telecommunications Union. "Pulse code modulation (PCM) of voice frequencies", Recommendation G.711, Novembro de 1988.
- [17] International Telecommunications Union. "Speech coders: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s", Recommendation G.723.1, Março de 1996.
- [18] International Telecommunications Union, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)", Recommendation G.729, Março de 1996.
- [19] International Telecommunications Union, "7 kHz audio-coding within 64 kbit/s", Recommendation G.722, Novembro de 1988.
- [20] European Telecommunications Standards Institute, "GSM full rate speech transcoding", GSM Recommendation 6.10, version 3.2.0, Janeiro de 1991.
- [21] Andersen, S., "Internet Low Bit Rate *codec* (iLBC)", RFC 3951, Dezembro de 2004.
- [22] International Telecommunications Union, "Subjective performance assessment of telephone-band and wideband digital *codecs*", Recommendation P.830, Fevereiro de 1996.
- [23] International Telecommunications Union, "The E-model, a computational model for use in transmission planning", Recommendation G.107, Dezembro de 1998.
- [24] International Telecommunications Union, "Objective quality measurement of telephone-band (300-3400 Hz) speech *codecs*", Recommendation P.861, Fevereiro de 1998.
- [25] A. W. Rix and M. P. Hollier. "The perceptual analysis measurement system for robust end-to-end speech quality assessment". IEEE ICASSP, Junho de 2000.
- [26] 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, Fevereiro de 2001.
- [27] Walker, J. Q., Hicks, J. T., "The essential guide to VoIP implementation and Management". White paper, NetIQ Corporation, 2002. http://www2.cs.uh.edu/~sujeev/projects/Ad-Hoc/NetIQ_VoIP_Introduction.pdf
- [28] Carson, M. & Santay, D., "NIST Net - A Linux-based Network Emulation Tool", ACM SIGCOMM Computer Communication Review, Julho de 2003.
- [29] Kamienski, C., Mariz, D., Sadok, D. & Fernandes, S., "Arquiteturas de Rede para a Próxima Geração da Internet", SBRC 2005, Maio 2005.
- [30] Demichelis, C., Chimento, P., "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, Novembro de 2002.
- [31] Tobias Hoßfeld, "Measurement and Analysis of Skype VoIP Traffic in 3G UMTS Systems", Internet Performance, Simulation, Monitoring and Measurements 2006 (IEEE/ACM IPS-MoME 06), Fevereiro de 2006.