Uma Análise Comparativa da QoS do Skype, Yahoo! Messenger e Google Talk

Douglas C. P. Barbosa, Marcos A. A. Gondim, Rafael D. Lins e Rafael S. de Souza

***Resumo*— Este artigo apresenta uma análise comparativa do desempenho dos aplicativos Skype, Yahoo! Messenger e Google Talk na comunicação de voz sobre IP entre computadores através da Internet. Será avaliada a correlação entre os sinais de voz transmitidos e recebidos a partir de suas amostras no tempo e periodogramas, com o intuito de observar a fidelidade com que cada um dos aplicativos reproduz a fala e o consumo de banda decorrente do uso desses softwares para comunicação de voz sobre IP.**

***Palavras-Chave*— VoIP, Skype, Yahoo! Messenger, Google Talk, QoS.**

Abstract— **This paper presents a comparative analysis of the performance of Skype, Yahoo! Messenger and Google Talk for VoIP communication between computers over the Internet. The correlation between the transmitted and received speech signals both in the time and frequency domains are analyzed. The bandwidth usage will be evaluated for each of those VoIP system.**

***Keywords*— VoIP, Skype, Yahoo! Messenger, Google Talk, QoS.**

I. INTRODUÇÃO

Desde o lançamento do Skype em 2003 [1], os aplicativos de VoIP têm chamado a atenção por sua rápida difusão e utilização por diversos tipos de clientes. Tal sucesso pode ser explicado porque VoIP é uma solução de fácil implementação e de baixo custo, o que agrada tanto aos usuários domésticos quanto aos empresariais. A América Latina é o destino da maior parte do tráfego de VoIP no mundo desde 2001 e o Brasil é o país onde o uso de VoIP cresce mais rapidamente, registrando, apenas no ano de 2004, um crescimento de 112% [2]. Estudos realizados em 45 países com mais de 1,5 mil profissionais de TI indicam que o mercado mundial de VoIP cresceu 152% entre os anos de 2002 e 2006. No Brasil, somente no ano de 2007, uma em cada dez empresas optará por utilizar voz sobre IP como sistema de telefonia para longas distâncias [3].

A qualidade da voz em tais aplicativos depende não somente dos CODECs utilizados, mas também de como os pacotes são roteados e encaminhados pela rede, sobretudo em momentos de congestionamento [4]. Este é um tema que tem sido alvo de inúmeros trabalhos científicos que buscam compreender o funcionamento de aplicativos para comunicação VoIP e analisar seu desempenho [5,6,7,8,9,10]. Neste artigo será apresentado um estudo comparativo da qualidade da conversação através dos aplicativos Skype, Yahoo! Messenger e Google Talk, com enfoque na fidelidade da voz recebida em relação à que foi transmitida, além de uma breve análise das características desses aplicativos no que diz respeito ao roteamento de pacotes de dados durante os processos de autenticação e estabelecimento de chamadas.

Douglas Contente Pimentel Barbosa, Marcos Antonio Alves Gondim, Rafael Dueire Lins e Rafael Santos de Souza, Departamento de Eletrônica e Sistemas, Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco, Recife, Brasil, E-mails: douglas.contente@gmail.com, marcos_aag@yahoo.com.br, rdl@ufpe.br e rafael.souza@globo.com.

II. METODOLOGIA

Os experimentos que possibilitaram a análise de desempenho de VoIP foram realizados com três dos aplicativos mais difundidos na Internet para comunicação de Voz sobre IP: Skype versão 3.1.0.152 [1], Yahoo! Messenger versão 8.1 [11] e Google Talk versão Beta [12].

Os testes consistiram na realização de chamadas VoIP entre dois computadores onde um dos interlocutores executava um arquivo de áudio padrão, cujo som era capturado pelo microfone e transmitido via aplicativo VoIP para o outro interlocutor. O conteúdo da chamada VoIP era gravado em arquivos de áudio no formato *wave* (.wav) a uma taxa de amostragem de 48kHz através do *software* Mx Skype Recorder versão 3.2.1 [13] nas duas extremidades da comunicação. Paralelamente, todos os pacotes enviados e recebidos pelos computadores durante as chamadas eram capturados pelo *software* Wireshark versão 0.99.5 anunciado em [14] como substituto do Ethereal.

Para garantir uniformidade nos testes, foi utilizado um único texto em língua inglesa para gerar dois arquivos de voz sintetizada, sendo um de voz masculina e outro de voz feminina. O sintetizador utilizado foi o TTS (*Text-to-Speech*) da AT&T Labs que se encontra disponível em [15]. Os arquivos gerados possuem formato *wave* com taxa de amostragem de 16 kHz e duração aproximada de 20 segundos. A densidade espectral de potência estimada através dos periodogramas dos arquivos gerados é mostrada nas figuras 01 e 02. Decidiu-se utilizar arquivos de curta duração para simular trechos unidirecionais de uma conversação telefônica, viabilizando a análise através da correlação cruzada dos sinais transmitido e recebido.

Foram realizados um total de 60 chamadas VoIP, sendo 20 para cada um dos sistemas VoIP em estudo. Destes 20, 10 foram executados com uso da voz masculina e outros 10 com uso da voz feminina. Visto que se tratava de um experimento longo, e a rede poderia sofrer mudanças significativas nesse intervalo de tempo, os dados foram coletados alternando-se o aplicativo em teste. Desta forma, buscou-se minimizar a influência das oscilações nas

condições da rede do provedor de acesso a Internet no resultado final do experimento. Por fim, cada chamada gerava 02 arquivos de áudio (transmitido e recebido) e 02 arquivos com a captura dos pacotes realizada nos computadores envolvidos no processo.

Os dois computadores estavam situados em pontos dentro da região metropolitana da cidade de Recife e ambos encontravam-se conectados a Internet através de linhas ADSL configuradas com taxas de 512 kB/s no sentido *down* e 128 kB/s no sentido *up*. Um dos computadores utilizados era um *laptop* Intel Centrino Duo 1.8 GHz com 1 GB de RAM e o outro um PC Pentium IV 1.4 GHz com 512 MB de RAM.

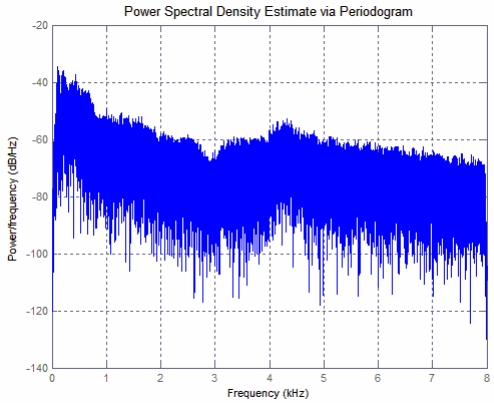

Figura 01. Densidade espectral de potência estimada via periodograma do sinal de voz masculina sintetizado.

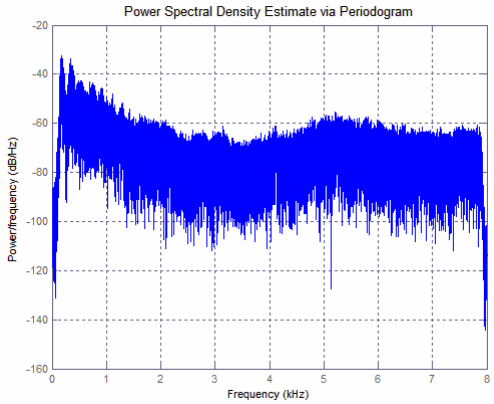

Figura 02. Densidade espectral de potência estimada via periodograma do sinal de voz feminina sintetizado.

III. CARACTERÍSTICAS DE ROTEAMENTO

Nesta seção serão descritos alguns aspectos importantes dos processos de autenticação e estabelecimento de chamada por cada um dos aplicativos testados bem como a topologia da rede utilizada por eles.

A. *Funcionamento do Skype*

De acordo com as referências [5,6,7] o Skype utiliza um esquema *peer-to-peer* de comunicação baseado em três entidades principais:

1) *Cliente Skype*

É uma aplicação Skype que pode ser utilizada para a realização de chamadas de voz, envio de mensagens de texto ou troca de arquivos.

2) *Super-Nó*

O Super-Nó provê à rede funcionalidades como encaminhamento de pedidos aos destinos apropriados e responder às solicitações de Clientes Skype ou de outro Super-Nó. Os Super-Nós podem ainda prover serviços de *proxy* para mídia de Clientes Skype que possuam acesso restrito à Internet através de NATs (*Network Adress Translators*) ou *firewalls*. Tipicamente os Super-Nós mantêm uma rede de *overlay* entre si enquanto um Cliente Skype liga-se com um único ou um pequeno número de Super-Nós. Uma análise de [7] mostra que um Cliente Skype pode ser promovido a Super-Nó se ele possui ampla largura de banda e é facilmente alcançável na rede.

3) *Servidor de Login Skype*

É o único servidor central da rede Skype. Ele é o servidor que guarda os registros de nomes dos usuários e seus *passwords*, realizando a autenticação dos mesmos garantindo que eles sejam únicos no espaço de nomes do Skype.

Um Cliente Skype deve se conectar a pelo menos um Super-Nó e ao Servidor de Login para ser registrado na rede. O Cliente Skype possui uma lista que é atualizada periodicamente de pares endereço IP e número de porta dos Super-Nós acessíveis da rede. Essa lista, chamada segundo [5] de *Host Cache*, pode conter até 200 entradas. Pelo menos uma dessas entradas deve ser válida, ou seja, deve ser uma máquina *on-line* rodando o Skype. Após o primeiro *login* depois da instalação, a referência [5] observou ainda que a *Host Cache* é sempre inicializada com 07 entradas chamadas de *Bootstrap Super Nodes*. Essas entradas correspondem a Super-Nós conectados à Internet através de quatro servidores: Superb, Suscon e ev1.net nos Estados Unidos e um quarto de extensão dinamarquesa.

Na primeira vez que se conecta à rede, o Cliente Skype envia pacotes UDP para algum dos *Bootstrap Super Nodes* que o responde com outros pacotes UDP. Em seguida, o Cliente Skype estabelece com ele uma conexão TCP e provavelmente recebe o IP do Servidor de Login (80.160.91.11), com o qual estabelece uma nova conexão TCP, realiza a autenticação e finaliza a conexão TCP. A conexão TCP com o Super-Nó dura até que este se torne indisponível e quando isto ocorre o Cliente Skype inicia uma nova conexão TCP com outro Super-Nó. Nos acessos subseqüentes o procedimento é o mesmo, com exceção que ao invés de conectar um *Bootstrap Super Node,* o usuário pode conectar-se a qualquer Super-Nó presente na *Host Cache*.

Após serem ambos registrados na rede, para estabelecer uma chamada, o Cliente Skype chamador inicia uma conexão TCP com o chamado. Troca de sinalização é realizada através dessa conexão, o que indica a existência de um mecanismo de *challenge-response* entre os usuários. Além

disso, pacotes UDP são enviados pelo chamador para alguns nós que estão *on-line* e foram descobertos durante o *login*. Ao ser finalizada a sinalização, inicia-se a troca de pacotes UDP de voz entre os dois usuários.

O processo descrito aqui se refere ao estabelecimento de chamadas entre dois Clientes Skype que estejam on-line com endereços IP válidos e presentes na lista de contatos um do outro, sem nenhuma restrição de NAT ou *firewall*. Variantes deste esquema são possíveis, mas fogem do escopo deste texto e são descritas em [5].

B. Funcionamento do Yahoo! Messenger

Através do monitoramento do tráfego de pacotes entre os usuários observa-se que o Yahoo! Messenger troca pacotes no momento da autenticação utilizando um esquema cliente-servidor através do protocolo SSLv3 que provê a privacidade e a integridade de dados entre duas aplicações que se comunicam pela Internet. Isto ocorre por meio da autenticação das partes envolvidas e da criptografia dos dados transmitidos entre as mesmas. Esse protocolo ajuda a prevenir que intermediários entre as duas pontas da comunicação tenham acesso indevido ou falsifiquem os dados que estejam sendo transmitidos. A figura 03 ilustra um usuário com acesso ADSL e IP válido 201.50.209.91 efetuando a autenticação no servidor Yahoo! Messenger com IP 209.73.168.74. Utilizando-se a ferramenta My Trace Route do Fedora 6 [16], identificaram-se todos os saltos até o servidor de autenticação do Yahoo! Messenger, com estatísticas de perda e atraso dos pacotes como mostra a tabela I.

TABELA I
ESTATÍSTICA DO ROTEAMENTO DOS PACOTES DURANTE O LOGIN NO YAHOO! MESSENGER.

My traceroute [v0.71]					
Thu May 3 09:45:28 2007					
Trace to 209.73.168.74					
Host	Loss%	Avg	Best	Wrst	StDev
1. 192.168.253.254	0.0%	0.6	0.3	1.6	0.3
2. 192.168.0.254	0.0%	1.3	0.4	2.7	0.7
3. 200.165.149.177	0.0%	4.3	1.8	26.0	5.2
4. Gi0-0.NBV-PE-ROTD-03.telemar.net.br	0.0%	4.0	1.9	8.3	1.8
5. PO6-0.NBV-PE-ROTN-01.telemar.net.br	0.0%	4.2	2.0	10.6	2.8
6. 200.223.131.205	0.0%	81.5	34.5	276.2	70.7
7. PO4-0.BOT-RJ-ROTN-01.telemar.net.br	0.0%	42.3	38.7	59.1	4.8
8. PO0-3.ARC-RJ-ROTN-01.telemar.net.br	0.0%	153.9	130.5	296.5	37.0
9. 198.32.124.115	0.0%	149.9	129.4	216.3	21.3
10. ge-3-3-4-p447.pat2.pao.yahoo.com	0.0%	207.8	187.1	245.3	15.7
11. ge-4-0-0-p451.msr2.scd.yahoo.com	0.0%	213.7	191.6	264.8	22.4
ge-3-0-0-p251.msr2.scd.yahoo.com					
ge-3-0-0-p241.msr1.scd.yahoo.com					
ge-4-0-0-p441.msr1.scd.yahoo.com					
12. ten-2-3-bas1.scd.yahoo.com	0.0%	211.8	186.2	257.8	23.4
ten-2-3-bas2.scd.yahoo.com					
ten-1-3-bas2.scd.yahoo.com					
ten-1-3-bas1.scd.yahoo.com					
13. l2.login.vip.scd.yahoo.com	0.0%	211.2	190.4	241.5	15.0

Os pacotes que foram capturados no decorrer das chamadas indicam que a partir do momento em que o usuário disca para seu destino, basicamente quatro endereços IP da rede Yahoo! foram conectados:

1. **68.142.233.145 – sip48.voice.re2.yahoo.com:** servidor de sinalização SIP. Observou-se que os computadores que rodavam os aplicativos VoIP trocaram pacotes com este servidor do início ao fim da chamada;
2. **68.142.233.72 – relay1.voice.vip.re2.yahoo.com:** foram observados pacotes relacionados a solicitação para ligar (*Binding Request*) a esse servidor;
3. **68.142.233.76 – stun2a.voice.re2.yahoo.com:** o protocolo STUN (*Simple Transversal of UDP Through NAT*), é usado para transposição de NAT;
4. **68.142.233.79 – relay6.voice.re2.yahoo.com:** servidor que é contatado no momento que antecede a inicialização da troca de pacotes de voz (UDP) entre os usuários.

A figura 04 mostra a troca de pacotes entre o usuário e os IPs dos servidores citados correspondente a somente uma das capturas realizadas, num total de 60. Embora tenha sido observada a troca de pacotes com outros endereços IP no decorrer das outras capturas, realizadas em momentos diferentes, o destino sempre correspondia a um servidor com as mesmas funcionalidades que as citadas acima.

C. Funcionamento do Google Talk

Analisando-se o Google Talk no momento do login, observa-se que o mesmo trabalha na forma de cliente-servidor, como ilustrado na figura 05. O IP com o qual o aplicativo troca pacotes no momento do login, é o 64.233.161.147.

TABELA II
ESTATÍSTICA DO ROTEAMENTO DOS PACOTES DURANTE O LOGIN NO GOOGLE TALK.

My traceroute [v0.71]					
Thu May 3 13:29:50 2007					
Host	Loss%	Avg	Best	Wrst	StDev
1. 192.168.253.254	0.0%	0.5	0.4	0.9	0.1
2. 192.168.0.254	0.0%	1.2	0.8	2.7	0.5
3. 200.165.149.177	0.0%	4.5	2.5	13.3	2.9
4. Gi0-0.NBV-PE-ROTD-03.telemar.net.br	0.0%	3.3	1.6	5.0	1.1
5. PO6-0.NBV-PE-ROTN-01.telemar.net.br	0.0%	3.0	2.1	4.5	0.8
6. 200.223.131.205	0.0%	78.4	51.1	191.6	40.6
7. PO4-0.BOT-RJ-ROTN-01.telemar.net.br	0.0%	58.4	45.3	89.2	9.9
8. 200.187.128.66	0.0%	191.7	165.8	228.3	24.1
9. GigabitEthernet5-0.GW12.NYC1.ALTER.NET	0.0%	302.6	281.8	340.7	21.4
1-0.GigabitEthernet.GW12.NYC1.ALTER.NET					
10. 0.so-1-3-0.XL1.NYC1.ALTER.NET	7.1%	305.8	286.5	348.3	20.1
11. 0.so-6-1-1.XL3.NYC4.ALTER.NET	0.0%	301.2	279.2	340.9	20.7
12. 0.ge-5-0-0.BR2.NYC4.ALTER.NET	0.0%	297.4	278.5	337.6	20.1
13. Te-3-4.car1.NewYork1.Level3.net	0.0%	376.3	261.9	688.9	129.6
14. ae-1-53.bbr1.NewYork1.Level3.net	0.0%	194.7	162.3	263.4	30.2
15. as-3-0.bbr1.Washington1.Level3.net	0.0%	286.6	257.7	344.5	24.0
ae-0-0.bbr2.Washington1.Level3.net					
16. ae-21-54.car1.Washington1.Level3.net	0.0%	287.9	258.9	338.0	23.1
ae-21-52.car1.Washington1.Level3.net					
ae-11-51.car1.Washington1.Level3.net					
ae-11-53.car1.Washington1.Level3.net					
ae-21-56.car1.Washington1.Level3.net					
17. GOOGLE-INC.car1.Level3.net	0.0%	277.0	248.7	326.5	25.1
18. 72.14.238.136	0.0%	292.2	262.0	332.4	23.8
66.249.95.149					
19. 72.14.236.173	0.0%	292.5	257.0	333.5	20.5
216.239.47.1					
216.239.49.45					
72.14.236.15					
20. 216.239.43.1 el-in-f125.google.com	0.0%	292.4	267.6	337.6	26.0

Capturando-se somente os pacotes TCP através do Wireshark [14], foi observado que o Google Talk envia periodicamente pacotes para o IP 209.85.163.125 que tem como DNS el-in-f125.google.com. Através do My Trace

Route do Fedora 6 [16] foram obtidos os saltos desde a máquina que continha o aplicativo até o servidor da Google para autenticação de usuário. O resultado obtido encontra-se na tabela II.

O caminho feito desde a rede onde se encontra a máquina do usuário até o servidor Google pode ser descrita de forma resumida na seguinte seqüência de saltos:

1. 192.168.253.254, LAN onde se encontra a máquina do usuário;
2. 200.165.149.177 IP válido atrelado a LAN do usuário;
3. Estação da Telemar localizada no bairro da Boa Vista, Recife – PE;
4. Estação da Telemar localizada no bairro de Botafogo, Rio de Janeiro – RJ;
5. Rede Gigabit em Nova York DC;
6. Washington;
7. Servidor Google.

A figura 06 mostra um trecho de uma captura, de uma conversação em andamento no Google Talk. Em destaque, Na linha 41 da captura, observa-se uma mensagem "*Jabber Request"* (Requisição para falar), enviada para o servidor do Google Talk. Esses pacotes foram trocados antes do início da conversação. A partir da linha 49 as duas máquinas começam a trocar pacotes de voz entre si somente trocando pacotes com o servidor 209.85.163.125 em casos de retransmissão.

IV. ANÁLISES E RESULTADOS

A análise dos dados foi realizada em duas etapas: uma para tratamento dos arquivos de voz gravados e identificação da correlação entre os pares de sinais (transmitido e recebido) nos domínios do tempo e da freqüência; outra para análise e determinação das características dos pacotes capturados.

Basicamente, o tratamento dado aos arquivos de áudio consistiu na normalização do tamanho de cada sinal através do corte de trechos no início e no final de cada seqüência, de forma a preservar apenas o conteúdo da fala. Tal operação foi realizada através do software Nero Wave Editor versão 3.0.0.0 [17] e tinha o intuito de viabilizar a análise comparativa entre os sinais através da correlação cruzada entre eles, além de eliminar ruídos provenientes de tons de chamada ou encerramento das ligações produzidos pelos próprios aplicativos VoIP. Os arquivos finais possuíam duração de 22 segundos e tamanho de 2.063 kB.

Foi utilizada correlação cruzada como um dos métodos para comparação entre os sinais. A função de correlação cruzada entre *f[n]* e *g[n]*, duas funções de tempo discreto, é definida por [18]:

$$c[m] = E\{f[n]g[n+m]\}, \quad (1)$$

em que *E{X}* é o valor esperado da variável *X*.

Sejam *f[n]* as amostras de voz transmitidas e *g[n]* as amostras de voz recuperadas no receptor. A função de correlação cruzada, portanto, nos informa o quão *f[n]* é semelhante a cópias deslocadas de *g[n]* em função do deslocamento *m*. No caso em questão, como se espera que idealmente as amostras transmitidas e recebidas sejam iguais, se ambas as amostras possuem comprimento *N*, a função de correlação cruzada possui comprimento *2N-1*, simetria em relação a *N* e um máximo nas redondezas deste ponto (em função do atraso inerente a transmissão dos pacotes [19]).

Assim, através do Matlab7®, o valor máximo da função de correlação cruzada pôde então ser determinado e utilizado como um indicador de fidelidade entre o sinal de voz transmitido e o sinal recebido.

O segundo parâmetro utilizado na comparação entre os aplicativos baseou-se no cálculo da correlação entre a densidade espectral de potência dos sinais de voz transmitido e recebido, densidade esta estimada através do método de periodograma [18] e com suporte do Matlab7® .

Nosso objetivo, com esses procedimentos, foi avaliar a fidelidade da reprodução do sinal de voz não apenas no domínio do tempo, mas também no domínio da freqüência.

Na segunda etapa da análise, foram extraídas, a partir das capturas realizadas pelo *software* Wireshark [15], informações como *jitter*, perda de pacotes, tamanho médio dos pacotes transmitidos entre os computadores, taxa de envio dos pacotes e consumo de banda para cada um dos aplicativos testados. O tempo relativo de envio e chegada de cada pacote foi utilizado para estimar o valor do *jitter* imposto pela rede durante as chamadas, a fim de nos certificar de que os aplicativos enfrentaram em média as mesmas condições durante a bateria de testes.

O fato de os relógios das máquinas não estarem sincronizados não gera problemas para o cálculo do *jitter*, dado que o mesmo é definido como a variância dos tempos de chegada dos pacotes enviados da máquina A para a máquina B, e não é influenciado pelo *offset* inicial entre os relógios. O descompasso entre os *clock*s das máquinas, foi desprezado em virtude da duração das chamadas não ser suficientemente longa para que este efeito se tornasse considerável.

Não foi observada nenhuma perda de pacotes durante o experimento, o que pode ser explicado pela curta duração das chamadas – cerca de 20 segundos. Observou-se ainda que o *jitter* sofreu poucas variações no decorrer do ensaio como mostrado na tabela III.

TABELA III
JITTER ESTIMADO DURANTE A COMUNICAÇÃO (ms).

Aplicativo	**Média**	**Desvio Padrão**
Google Talk	9,20	2,40
Skype	7,92	3,03
Yahoo! Messenger	8,50	3,17

As tabelas IV e V mostram que o Skype transmitiu quase o dobro dos pacotes em cada chamada se comparado ao Google Talk e ao Yahoo! Messenger, que ficaram praticamente empatados neste item.

TABELA IV
NÚMERO DE PACOTES TRANSMITIDOS PELOS APLICATIVOS.

Aplicativo	**Média**	**Desvio Padrão**
Google Talk	469,00	23,66
Skype	830,25	38,07
Yahoo! Messenger	453,85	69,85

TABELA V
PACOTES TRANSMITIDOS POR SEGUNDO PELOS APLICATIVOS.

Aplicativo	Média	Desvio Padrão
Google Talk	18,01	0,35
Skype	23,59	1,05
Yahoo! Messenger	17,87	2,76

Não há diferenças significativas no tamanho dos pacotes gerados pelos três programas, no entanto foi observada no Yahoo! Messenger uma maior variação no tamanho desses pacotes no decorrer da transmissão.

TABELA VI
TAMANHO MÉDIO DOS PACOTES (BYTES).

Aplicativo	Média	Desvio Padrão
Google Talk	172,95	2,19
Skype	160,91	1,97
Yahoo! Messenger	188,83	22,27

Quanto à taxa de transmissão, é mostrado na tabela VII que o Skype apresentou em média um maior consumo de banda que os outros dois aplicativos, custo esse que pode estar relacionado ao o uso de *piggybacking*.

TABELA VII
TAXA DE TRANSMISSÃO (kB/s).

Aplicativo	Média	Desvio Padrão
Google Talk	3,11	0,08
Skype	3,78	0,18
Yahoo! Messenger	3,36	0,48

As tabelas VII, IX, X e XI exibem os resultados obtidos no cálculo da correlação entre os sinais de voz transmitido e recebido nos domínios do tempo e da freqüência, de acordo com os métodos expostos na seção IV.

TABELA VIII
CORRELAÇÃO MÁXIMA ENTRE OS PERIODOGRAMAS DOS ARQUIVOS DE VOZ TRANSMITIDOS E RECEBIDOS.

Aplicativo	Média	Desvio Padrão
Google Talk	0,243	0,027
Skype	0,215	0,036
Yahoo! Messenger	0,210	0,033

TABELA IX
CORRELAÇÃO MÁXIMA ENTRE OS SINAIS DE VOZ.

Aplicativo	Média	Desvio Padrão
Google Talk	0.164	0.054
Skype	0.188	0.095
Yahoo! Messenger	0.158	0.038

TABELA X
CORRELAÇÃO MÁXIMA ENTRE OS SINAIS DE VOZ MASCULINA.

Aplicativo	Média	Desvio Padrão
Google Talk	0,188	0,064
Skype	0,217	0.113
Yahoo! Messenger	0,146	0,030

TABELA XI
CORRELAÇÃO MÁXIMA ENTRE OS SINAIS DE VOZ FEMININA.

Aplicativo	Média	Desvio Padrão
Google Talk	0,140	0,031
Skype	0,159	0,066
Yahoo! Messenger	0,170	0,043

V. CONCLUSÕES E TRABALHOS FUTUROS

Os sistemas de comunicação de voz sobre IP tornaram-se muito populares nos últimos anos. Poucos estudos comparativos como o realizado em [9] são encontrados na literatura, apesar da enorme quantidade de usuários que utilizam esses sistemas. A tabela VIII nos mostra que a codificação PCM de taxa de bit variável utilizada pelo Google Talk consegue manter uma maior fidelidade entre os espectros dos sinais transmitidos e recebidos em relação à atingida pelo codificador iLBC [20] do Skype. Os valores observados para o Yahoo! Messenger indicam que esse aplicativo deve utilizar uma estratégia de codificação similar à utilizada pelo Skype. As tabelas IX, X e XI mostram, no entanto, que em média a fidelidade dos sinais no Skype é ligeiramente superior às dos outros dois aplicativos sofrendo, porém, oscilações maiores, provavelmente devido à estratégia de *play-out* FIFO (*Firt-in-first-out*) tornando-o mais dependente das condições da rede, já que os pacotes que chegam desordenados são reproduzidos nesta mesma ordem [9].

O Skype apresentou um maior consumo de banda durante a realização das chamadas enquanto o Yahoo! Messenger e Google Talk mostraram características bastante semelhantes no que diz respeito a desempenho, número e tamanho dos pacotes transmitidos entre os computadores e largura de banda utilizada. Em trabalhos futuros pretende-se buscar uma maior compreensão das diferenças entre os aplicativos e seus métodos de codificação e transmissão de voz, acrescentando outras formas de comparação dos arquivos de áudio, tanto quantitativas quanto subjetivas, correlacionando os resultados aqui apresentados com os obtidos pelos métodos já existentes para a qualificação de voz, como o MOS (*Mean Opinion Score*). É intensão deste estudo, repetir os experimentos com enlaces intercontinentais e com chamadas de maior duração também pode trazer à tona novas características a serem analisadas. Pretende-se também comparar os aspectos da transmissão conjunta de voz e vídeo na Internet, a degradação dos sinais de voz trafegando em meios sem-fio nas extremidades do circuito de comunicação (rede 802.11 + rede cabeada + rede 802.11), e a qualidade da voz em ligações entre VoIP e telefones TDM.

REFERÊNCIAS

[1] Skype: http://www.skype.com/

[2] Telegeography: http://www.telegeography.com/press/releases/2005-12-15.php. Acessado em 30/04/2007.

[3] Estadão:.http://www.estadao.com.br/tecnologia/internet/noticias/2006/dez/14/228.htm. Acessado em 30/04/2007.

[4] William C. Hardy, "VoIP Service Quality – Measuring and Evaluating Packet-Switched Voice", McGraw-Hill Book Co, 2003.

[5] S. A. Basset., H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", *Proceedings of IPTPS*, Dec. 5th, 2004.

[6] S. Ehlert, S. Petgang, "Analysis and Signature of Skype VoIP Session Traffic", *Proceedings of Communications*, Internet and Information Technology, July 25th, 2006.

[7] S. Guha, N. Daswani, R. Javi, "An Experimental Study of the Skype Peer-to-Peer VoIP System", *Proceedings of IPTPS*, 2006.

[8] T. Hoβfeld, A. Binzenhöfer, M. Fiedler, K. Tutschku, "Measurement and Analysis of Skype VoIP Traffic in 3G UMTS Systems", *Research Report Series*, University of Würzburg, December 2005.

[9] S. Batu, B. W. Wah, "Analysis and Evaluation of the Skype and Google-Talk Voip Systems", *IEEE International Conference on Multimedia and Expo*, July 2006.

[10] H.Nishida and T.Suzuki, "Performance Analysis of a P2P-Based VoIP Software", *International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications*, February 2006.

[11] Yahoo Messenger: http://messenger.yahoo.com/

[12] Google Talk: http://www.google.com/talk/

[13] Mx Skype Recorder: http://www.skyperec.com/

[14] Wireshark: http://www.wireshark.org/

[15] TTS: http://www.research.att.com/~ttsweb/tts/demo.php. Acessado em 30/04/2007.

[16] My Trace Route: http://fedoraproject.org/wiki/

[17] Nero: www.nero.com

[18] A. V. Oppenheim, R. W. Schaffer, "Discrete-Time Signal Processing", Prentice-Hall, 1989.

[19] O. Hersent, J.P. Petit, D. Gurle, "Beyond VoIP Protocols – Understanding Voice Technology and Networking Techniques for IP Telephony", John Wiley & Sons, 2005.

[20] S. Andersen et. al, "*Internet Low Bit Rate Codec (iLBC)*", http://www.ietf.org/rfc/rfc3951.txt, December, 2004.

21	14.109223	201.50.209.91	209.73.168.74	TCP	1529 > https [SYN] Seq=0 Len=0 MSS=1360	70 14.109223
22	14.234835	201.50.209.91	216.155.193.150	TCP	1528 > 5050 [ACK] Seq=56 Ack=140 Win=65396 Len	62 14.234835
23	14.569939	209.73.168.74	201.50.209.91	TCP	https > 1529 [SYN, ACK] Seq=0 Ack=1 Win=65535 L	68 14.569939
24	14.571738	201.50.209.91	209.73.168.74	TCP	1529 > https [ACK] Seq=1 Ack=1 Win=65535 Len=0	62 14.571738
25	14.572514	201.50.209.91	209.73.168.74	SSLv3	Client Hello	164 14.572514
28	15.102690	209.73.168.74	201.50.209.91	SSLv3	Server Hello, Certificate, Server Hello Done	918 15.102690
29	15.104872	201.50.209.91	209.73.168.74	SSLv3	Client Key Exchange, Change Cipher Spec, Encry	266 15.104872
30	15.595401	209.73.168.74	201.50.209.91	SSLv3	Change Cipher Spec, Server Hello[Malformed Pacl	129 15.595401
31	15.598592	201.50.209.91	209.73.168.74	SSLv3	Application Data	639 15.598592
32	16.186567	209.73.168.74	201.50.209.91	SSLv3	Application Data	1198 16.186567
33	16.188486	201.50.209.91	209.73.168.74	TCP	1529 > https [FIN, ACK] Seq=884 Ack=2060 Win=6	62 16.188486
34	16.189709	201.50.209.91	216.155.193.150	TCP	[TCP segment of a reassembled PDU]	889 16.189709
35	16.194574	209.73.168.74	201.50.209.91	SSLv3	Encrypted Alert	85 16.194574

Figura 03. Captura dos pacotes durante a autenticação do Yahoo! Messenger.

4	0.438477	201.32.165.30	68.142.233.145	TCP	1768 > https [ACK] Seq=497 Ack=1157 Win=16123	54 0.438477
5	0.438477	201.32.165.30	68.142.233.145	SSL	Continuation Data	570 0.438477
6	0.447266	201.32.165.30	68.142.233.72	TCP	1782 > https [SYN] Seq=0 Len=0 MSS=1440	62 0.447266
7	0.861328	68.142.233.72	201.32.165.30	TCP	https > 1782 [SYN, ACK] Seq=0 Ack=1 Win=65535	60 0.861328
8	0.861328	201.32.165.30	68.142.233.72	TCP	1782 > https [ACK] Seq=1 Ack=1 Win=17280 Len=0	54 0.861328
9	0.861328	201.32.165.30	68.142.233.72	SSL	Continuation Data	174 0.861328
10	0.945313	68.142.233.145	201.32.165.30	TCP	https > 1768 [ACK] Seq=1157 Ack=1013 Win=65535	60 0.945313
11	1.269532	68.142.233.72	201.32.165.30	SSL	Continuation Data	86 1.269532
12	1.269532	201.32.165.30	68.142.233.72	TCP	1782 > https [FIN, ACK] Seq=121 Ack=33 Win=172	54 1.269532
13	1.276368	201.32.165.30	200.165.132.155	DNS	Standard query A stun.voice.bro.yahoo.com	84 1.276368
14	1.325196	200.165.132.155	201.32.165.30	DNS	Standard query response CNAME stun.voice.yahoo	158 1.325196
15	1.325196	201.32.165.30	68.142.233.76	STUN	Message: Binding Request	70 1.325196
16	1.328125	201.32.165.30	68.142.233.79	TCP	1783 > https [SYN] Seq=0 Len=0 MSS=1440	62 1.328125
17	1.333008	201.32.165.30	68.142.233.76	STUN	Message: Binding Request	70 1.333008
18	1.334961	201.32.165.30	68.142.233.79	TCP	1784 > https [SYN] Seq=0 Len=0 MSS=1440	62 1.334961
19	1.335938	201.32.165.30	68.142.233.79	UDP	Source port: 8009 Destination port: 13844	154 1.335938
20	1.336914	201.32.165.30	68.142.233.79	UDP	Source port: 8010 Destination port: 13845	154 1.336914
21	1.426758	201.32.165.30	68.142.233.76	STUN	Message: Binding Request	70 1.426758
22	1.447266	201.32.165.30	68.142.233.76	STUN	Message: Binding Request	70 1.447266
23	1.634766	201.32.165.30	68.142.233.76	STUN	Message: Binding Request	70 1.634766
24	1.647461	201.32.165.30	68.142.233.76	STUN	Message: Binding Request	70 1.647461

Figura 04. Troca de pacotes entre o usuário e o servidor no Yahoo! Messenger.

65	3.333805	192.168.15.236	64.233.161.147	TCP	4732 > https [ACK] Seq=1 Ack=1 Win=65535 [TCP	54 3.333805
66	3.334117	192.168.15.236	64.233.161.147	SSLv2	Client Hello	132 3.334117
67	3.550928	64.233.161.147	192.168.15.236	TCP	https > 4732 [ACK] Seq=1 Ack=79 Win=8190 Len=0	60 3.550928
68	3.575549	64.233.161.147	192.168.15.236	TLS	Server Hello,	1484 3.575549
70	3.882691	64.233.161.147	192.168.15.236	TLS	[TCP Retransmission] Server Hello,	1484 3.882691
71	3.882737	192.168.15.236	64.233.161.147	TCP	4732 > https [ACK] Seq=79 Ack=1431 Win=64105 [	54 3.882737
72	4.087389	64.233.161.147	192.168.15.236	TLS	Certificate, Server Hello Done	342 4.087389
73	4.088918	192.168.15.236	64.233.161.147	TLS	Client Key Exchange, Change Cipher Spec, Encry	236 4.088918
74	4.289972	64.233.161.147	192.168.15.236	TLS	Change Cipher Spec, Encrypted Handshake Messag	97 4.289972
75	4.290405	192.168.15.236	64.233.161.147	TLS	Application Data	258 4.290405
76	4.529547	64.233.161.147	192.168.15.236	TCP	https > 4732 [ACK] Seq=1762 Ack=465 Win=5720 L	60 4.529547
77	4.529581	192.168.15.236	64.233.161.147	TLS	Application Data	235 4.529581

Figura 05. Captura de pacotes durante a autenticação do Google Talk.

41	5.341797	201.32.165.30	209.85.163.125	Jabber	Request: \027\003\001\000]\207\354\276t\275\26	152 5.341797
42	5.343750	209.85.163.125	201.32.165.30	Jabber	Response: \027\003\001\000\210P\353A\204\037\0	195 5.343750
43	5.347656	209.85.163.125	201.32.165.30	Jabber	Response: \027\003\001\000\210h\215.z\213\211D	195 5.347656
44	5.347656	201.32.165.30	209.85.163.125	TCP	1585 > 5222 [ACK] Seq=3646 Ack=1364 Win=16170	54 5.347656
45	5.578125	209.85.163.125	201.32.165.30	Jabber	Response: \027\003\001\000\210\250]B\026?v#\33	195 5.578125
46	5.614257	209.85.163.125	201.32.165.30	TCP	5222 > 1585 [ACK] Seq=1505 Ack=3646 Win=18720	60 5.614257
47	5.684570	201.32.165.30	209.85.163.125	TCP	1585 > 5222 [ACK] Seq=3646 Ack=1505 Win=16029	54 5.684570
48	6.051757	200.255.242.189	201.32.165.30	UDP	Source port: 64904 Destination port: 53379	124 6.051757
49	6.967773	189.13.172.233	201.32.165.30	STUN	Message: Binding Request	98 6.967773
50	6.967773	201.32.165.30	189.13.172.233	STUN	Message: Binding Error Response	123 6.967773

Figura 06. Captura de pacotes ligeiramente antes do início da conversação no Google Talk.