

Ambiente para Simulação e Monitoração de Ligações Telefônicas IP

Cesar A. C. Marcondes, Paulo H. de Aguiar Rodrigues, Fabio David, João Carlos Peixoto da Costa

Laboratório de Voz sobre IP – Núcleo de Computação Eletrônica –
Universidade Federal do Rio de Janeiro (NCE/UFRJ)¹

{cesarmarcondes, aguiar, fabio, peixoto}@nce.ufrj.br

Abstract. *We describe an infrastructure to collect and monitor perceived quality of service from simulated voice over IP calls over the Brazilian Research Network (RNP2) backbone. We focus on details of development of this experimental environment based on the analysis of multimedia packets behavior under RTP (Real-Time Protocol). An integrated Javascript based Web interface presents the statistics with different granularity and a search mechanism was developed to facilitate the location of troubled VOIP connections. The received VOIP message is recorded and saved along measured statistics to allow future studies towards VOIP models and voice quality correlation with jitter, loss and delay.*

Resumo. *O presente artigo descreve uma infra-estrutura para coleta e monitoramento da qualidade percebida de ligações simuladas de voz sobre IP na Rede Nacional de Pesquisa (RNP2). São apresentados os detalhes do desenvolvimento deste ambiente experimental, que é baseado no comportamento de fluxos multimídia enviados via protocolo RTP (Real Time Protocol). Uma interface Web escrita em Javascript permite a visualização das estatísticas com diferente granularidade e um método de busca permite a localização eficiente de chamadas com qualidade degradada. A mensagem de voz recebida é gravada e salva junto com as estatísticas da ligação, permitindo estudos futuros de modelos de VOIP e de correlação entre qualidade da chamada e os parâmetros de jitter, atraso e perda.*

¹ Os integrantes do Laboratório VoIP são parcialmente suportados com recursos do GT-VOIP, grupo tarefa mantido pela RNP. Paulo H. de Aguiar Rodrigues é professor adjunto do Departamento de Computação do IM/UFRJ e analista do NCE/UFRJ. Cesar Marcondes é pesquisador contratado e recebe também apoio direto do NCE/UFRJ. Fabio David e João Carlos Peixoto da Costa são analistas do NCE/UFRJ e mestrando pelo Programa de Pós-Graduação NCE/IM.

1. Introdução

Atualmente, temos vivenciado o surgimento de serviços avançados na Internet, em contraposição ao fluxo de dados sobre TCP, que ainda é o tráfego predominante na rede. A telefonia IP é um exemplo destes novos serviços avançados, tráfego interativo de tempo real com sérias restrições de atraso, jitter e perda. O surgimento destas novas aplicações é muito importante porque revoluciona a maneira de interagir das pessoas e enriquece o processo colaborativo. A convivência de tráfegos com restrições variadas de qualidade de serviço na rede urge a implantação de mecanismos de suporte a QoS no backbone e nas redes locais das instituições, para garantir a qualidade fim a fim. A Rede Nacional de Pesquisa (RNP) está desenvolvendo um projeto de implantação de telefonia IP, que, além de facilitar a interação remota de pesquisadores e permitir acesso a voz em localidade servida por rede, mas não por telefonia convencional, irá possibilitar o domínio tecnológico das técnicas de VOIP e QoS no cenário de rede acadêmica, livre e heterogênea, onde questões de segurança, autenticação e garantia de QoS se tornam verdadeiros desafios.

Para um efetivo provisionamento de novos serviços no backbone é necessário realizar medidas tanto na forma passiva (com o monitoramento do tráfego real que atravessa a rede) ou de forma ativa (através da geração de tráfego e verificação de variáveis de interesse fim-a-fim).

Apresentamos uma infra-estrutura de geração, coleta e monitoramento da qualidade de ligações de voz sobre IP, baseada no desenvolvimento de um conjunto próprio de ferramentas que são utilizadas para a realização de medidas ativas na RNP, especialmente no contexto do serviço de telefonia IP. Tais ferramentas armazenam os dados sobre o comportamento de fluxos multimídia enviados via protocolo RTP (Real-Time Protocol) [1], criando uma base para comparação da qualidade atual da telefonia IP na RNP com a qualidade obtida com a implantação de novas políticas de QoS no backbone. Futuramente, os próprios fluxos reais de voz serão a base para a monitoração constante da qualidade percebida por este serviço.

Medidas ativas de geração de telefonia IP e obtenção de uma avaliação da qualidade das ligações é um assunto explorado em algumas ferramentas da Internet, como a *Chariot* da *NetIQ* [2] e o J4618A Telegra RQM software da HP [17]. Estas ferramentas permitem a geração de vários fluxos simultâneos e a criação de um relatório final sobre a qualidade das ligações. Entretanto apresentam deficiências quanto à granularidade das métricas e na identificação de pontos problemáticos.

Uma das premissas deste trabalho foi basear o desenvolvimento em software livre, de modo a ser amplamente utilizado pela comunidade acadêmica sem qualquer ônus. Também consideramos outros aspectos, que permitiriam subsidiar a engenharia de tráfego no backbone, como encontrar uma forma de detectar automaticamente a origem do problema da degradação da voz na rede, pesquisando o possível gargalo, ou degradação devida à mudança de rota. Para efeitos de validação e/ou aprimoramento de modelos de qualidade para VoIP baseados no MOS (Mean Opinion Score), a disponibilização do arquivo de áudio na recepção é fundamental para permitir a correlação entre as métricas de atraso, jitter e perda com diferentes codificadores de áudio e a qualidade da voz percebida pelo usuário através de métodos subjetivos, ou objetivos como o e-model [15].

O artigo está dividido em 6 seções. Na seção 2 é feita uma breve revisão do mecanismo de *feedback* da qualidade de serviço pelo protocolo RTCP. Na seção 3 é apresentada a metodologia de captura em testes reais, usando programas de código fonte aberto de telefonia IP, destacando a estrutura interna dos mesmos, as modificações realizadas para melhorar a precisão das medidas bem como a descrição dos logs e integração com outras medidas, como traceroute. Na seção 4 é descrita a consolidação dos logs das gerações de telefonia IP realizadas, de modo a termos a capacidade de visualização da qualidade como um todo, e ao mesmo tempo sem perder as informações mais detalhadas para avaliações mais precisas. Isso é exibido de forma gráfica, baseada na Web, contendo os resultados estatísticos. Na seção 5 temos as conclusões do trabalho, bem como sugestões de trabalhos futuros, e, na seção 6, a bibliografia.

2. Mecanismo de Feedback da Qualidade de Serviço pelo RTCP

O Real-Time Protocol (RTP) [1] é um protocolo que provê serviços ao transporte fim-a-fim dos dados para aplicações com restrições de tempo, como o áudio interativo de ligações telefônicas. Entre os serviços providos estão: identificação do tipo de carga de mídia transportada, numeração em seqüência dos pacotes, marcação de tempo e monitoramento da qualidade da sessão multimídia, usado pela aplicação. O RTP não provê sozinho quaisquer mecanismos que garantam a entrega dos dados no tempo desejado, ou com uma qualidade de serviço diferenciada. Estes requisitos devem ser providos pela camada de rede.

Na telefonia IP, cada participante da ligação envia continuamente o áudio em pequenos trechos (20 ms, por exemplo) encapsulados num cabeçalho RTP, por sua vez encapsulado em um pacote UDP. Esses pacotes de áudio são enfileirados no destino para que possam ser tocados, com a devida correspondência de tempo, de modo a se manterem sincronizados. Isto é feito através do buffer de playout, cujo propósito é absorver a variação de atraso, segurando os pacotes que chegam por tempo suficiente para garantir que eles sejam tocados continuamente. Quando um pacote chegar após o seu tempo de playout, ele deverá ser descartado, portanto existe uma relação direta entre o atraso e a perda [12].

Em conjunto com o RTP, fazendo o monitoramento da qualidade de serviço está o RTCP (Real-Time Control Protocol) [1], também transportado sobre UDP. Nele são definidos vários tipos de mensagens RTCP, que transportam uma variedade de informações de controle, como o SR ou *Sender Report*, usado para a transmissão e recepção de estatísticas de participantes que são fontes emissoras ativas de áudio; e o RR ou *Receiver Report*, usado na descrição da qualidade da recepção de participantes que não são fontes emissoras. Estes dois tipos de mensagens são as mais importantes para nosso experimento.

Nos pacotes SR ou RR estão definidos os campos, que contém algumas das principais variáveis de interesse, usadas para avaliar a qualidade, como:

- NTP *timestamp*: (64 bits) – captura o horário local de sistema da máquina fonte, no momento do envio da mensagem SR, usando a formatação do NTP (Network Time Protocol) [3].

- *RTP timestamp* (32 bits) – corresponde ao instante de tempo (normalmente múltiplo da unidade de amostragem de áudio) em que o áudio, pertencente ao último pacote RTP, foi enviado por determinada fonte.
- *Sender's packet count* (32 bits) – Contador cumulativo do total de pacotes de dados RTP transmitidos pelo emissor desde o início da transmissão até o momento em que um pacote SR foi gerado.
- *Fraction lost* (8 bits) – Esta fração é definida pelo número de pacotes perdidos dividido pelo número de pacotes esperado. Tratando-se de informação do RR.
- *Cumulative number of packets lost* (24 bits) – O número total de pacotes de dados RTP que foram perdidos desde o começo da recepção (valor cumulativo). Este número é definido pelo número de pacotes esperado menos o número de pacotes atualmente recebidos. O número de pacotes esperado é definido pelo campo do RR chamado “*extended highest sequence number received*”.
- *Interarrival Jitter* (32 bits) – esta é uma estimativa da variância estatística do tempo entre chegadas dos pacotes de dados RTP, medido em unidade de amostragem e expresso sempre como um inteiro positivo, ou seja, se o *jitter* for negativo, ele é apresentado com o sinal invertido.

2.1. Características dos Relatórios de Qualidade

Os relatórios SR e RR provêm subsídios para que o emissor possa, na indicação de perda de QoS, modificar seus buffers na recepção e transmissão, ou trocar o codificador, baseando-se neste *feedback*.

Os relatórios fazem largo uso de uma contabilidade cumulativa dos parâmetros. Isto é explicado na RFC do RTP[1], como sendo uma forma de realizar medidas sobre período de tempo, curtos onde a diferença entre os últimos dois relatórios recebidos pode ser usada para estimar os parâmetros de qualidade de serviço mais recentes, e longos, para prover confiabilidade frente a perda de relatórios intermediários em possíveis situações de congestionamento da rede.

Outra característica importante é a detecção de congestionamentos, onde podemos usar o campo *interarrival jitter* como um indicativo. Assim, enquanto a perda de pacotes rastreia congestionamentos persistentes, a medida de *jitter* vai rastrear congestionamentos transientes. Isto acontece porque o *jitter* aumenta indicando a iminência do congestionamento antes que este gere qualquer perda de pacotes. Como o campo de *jitter* entre chegadas é somente um retrato do *jitter* no momento do relatório, é necessário analisar vários relatórios de um receptor ao longo do tempo dentro de uma única rede, para verificar o comportamento dos congestionamentos, e o impacto deles na qualidade da ligação.

Recentemente, novos relatórios RTCP têm sido propostos para melhor representar os parâmetros de qualidade de serviço para voz sobre IP [5], mas veremos que nosso ambiente permite capturar todos os pacotes necessários para também obter estas novas estimativas sugeridas devido à característica de granularidade.

2.2. Cálculo do jitter pelo RTCP

O *jitter* entre chegadas J é definido como: a diferença D correspondente ao espaço de tempo entre a chegada de um par de pacotes, comparado com os tempos estampados neste par de pacotes no momento do envio. Como mostrado na equação abaixo.

Logo, seja S_i o *timestamp* RTP do pacote i , e R_i o tempo de chegada em unidades de *timestamp* RTP do pacote i , então para dois pacotes i e j , D pode ser expresso como:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) \quad J = \frac{J + (|D(i-1, i)| - J)}{16}$$

ou $= (R_j - S_j) - (R_i - S_i)$

O *jitter* (J) entre chegadas é calculado continuamente por uma equação filtrada desta diferença. Quando cada pacote i for recebido da fonte, calcula-se a diferença D em relação ao pacote prévio ($i-1$) em ordem de chegada (não necessariamente em ordem de seqüência), de acordo com a fórmula acima.

Toda vez que um relatório de recepção RR for montado, o valor corrente deste J é amostrado. O cálculo do *jitter* é prescrito na RFC 1889 para permitir o monitoramento independente do perfil (áudio, vídeo, usando um ou outro codificador) e tornar válida a interpretação de relatórios vindos de diferentes implementações. O algoritmo usado em J usa um estimador empírico com parâmetro de ganho $1/16$, o que segundo a RFC dá uma boa redução da taxa de ruído (ocasionado por picos de jitter), enquanto mantém uma taxa razoável de convergência.

Cabe ressaltar, que apesar do cálculo do jitter filtrado estar previsto na RFC do RTP, este valor não é apropriado como entrada em modelos de avaliação objetiva da qualidade de voz sobre IP como o *e-model* [15]. Porque este acaba escondendo os valores instantâneos de jitter em detrimento das médias. Por isso, veremos na seção a seguir que também obtemos os valores instantâneos de D em nossas medidas.

3. Metodologia de Geração e Captura de Telefonia IP

Utilizamos uma infra-estrutura baseada no projeto OpenH323 [5] como plataforma operacional do ambiente, onde as ligações de telefonia IP são baseadas em H.323 [6]. Esta infra-estrutura é formada por dois programas: *openam* (ou *Open Answer Machine*), um tipo de secretária eletrônica de telefonia IP desenvolvida no âmbito do projeto OpenH323, e o programa *caller* (ou *call generator*), uma modificação do *openam* desenvolvida inicialmente por [7]. Nossa modificação, ao invés de receber chamadas, as realiza, enviando um arquivo pré-gravado de áudio durante a ligação.

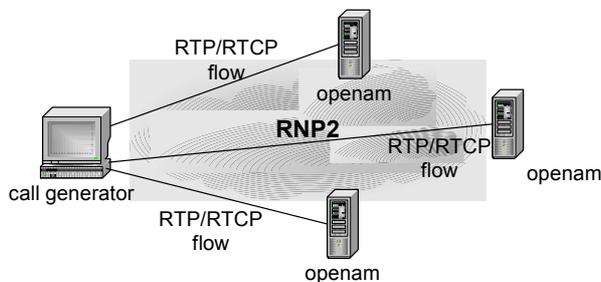


Fig. 1 – Arquitetura de Geração de chamadas e Coleta de Estatísticas

Na Fig.1, temos a ilustração da arquitetura desenvolvida. De um lado, a ferramenta de geração de chamadas *caller* (NCE/UFRJ), que ativa threads concorrentes e pode realizar

até 254 chamadas IP simultâneas, sem precisar ter nenhum recurso de áudio (seja placa de som, ou microfone) envolvido. Do outro lado, espalhado em diversos pontos do *backbone* da RNP2, em vários estados (AM, DF, CE, MG, PR, SC, RJ), está o programa receptor das chamadas, o *openam*.

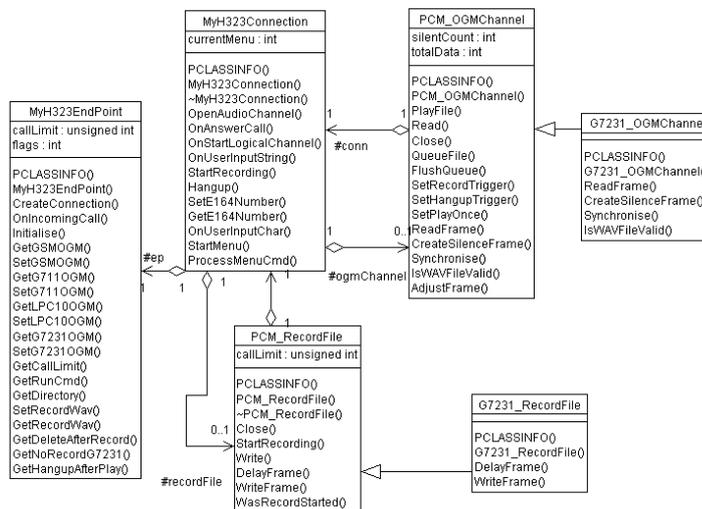
Ambos os programas realizam o armazenamento dos *logs* das chamadas em arquivos de texto contendo entre outras informações: toda a sinalização H.323 envolvida (útil na detecção de problemas de sinalização entre PoPs), todos os pacotes RTCP enviados e recebidos, bem como outras informações sobre o funcionamento interno do programa, e uma análise final do buffer de playout e de outros parâmetros estatísticos.

3.1. Estrutura Interna do Software Open Answer Machine

O *openam* é uma secretária eletrônica IP desenvolvida usando a biblioteca *openh323*. Nós utilizamos a versão 1.1.12 neste trabalho. Ela trabalha com base em objetos C++ que gerenciam as funções de sinalização H.323 (SETUP, CONNECT, H.245), a pilha de protocolo RTP/RTCP, além dos codificadores de áudio em software como o G.723.1, G.711ulaw, G.711alaw, GSM, MS-GSM, e LPC-10. A figura abaixo (Fig. 2) mostra o diagrama de classes usado pelo *openam*.

Fig. 2 – Diagrama de Classes do Openam

Toda a execução desta secretária está vinculada ao recebimento de sinalização H.225 [8] (como SETUP para estabelecimento de chamada) por parte de um usuário remoto, para que então o programa crie os objetos relacionados e responda àquela ligação. Portanto, em um primeiro momento, o *openam* fica bloqueado esperando possíveis conexões H.323.



Quando outro programa baseado em H.323 se conecta ao *openam*, ele inicia o objeto #ep (Fig. 2, seta ao lado de *MyH323EndPoint*), instância de *MyH323EndPoint*, acionado pela captura do evento “*MyH.323Endpoint::OnIncomingCall*”, e que por sua vez, criará o objeto *MyH323Connection*, relacionado especificamente àquela ligação. Esta capacidade permite ao *openam* receber ligações simultâneas, separando-as, respectivamente, como threads e objetos concorrentes para cada ligação. Esta é uma característica desejável para testes mais críticos de telefonia IP, que envolvam múltiplas conexões VOIP.

O objeto #conn (Fig. 2), instância de *MyH323Connection*, cria dois objetos internos importantes, cujas responsabilidades são respectivamente de tocar e gravar mensagens de áudio, são eles: o objeto (para tocar) #ogmchannel, instância de *PCM* ou *G7231OGMChannel*. O OGM da sigla dos objetos indica *outgoing message*, ou mensagem de boas vindas, que estará associado a um arquivo de áudio pré-gravado e começara a ser tocado assim que a ligação for atendida. O outro objeto (para gravar)

#recordFile, instância de *PCM* ou *G7231RecordFile*, criará um novo arquivo (cujo nome será formado pelo horário de sistema do instante de início da gravação) e gravará o conteúdo da mensagem sendo enviada nos pacotes RTP pelo emissor.

Para entender o procedimento usado por estes importantes objetos, relacionados tanto com o envio da mídia quanto com a gravação, é preciso dizer que existe um mecanismo de gerenciamento de tempo envolvido através do escalonamento dos frames de áudio a serem tocados e gravados. Ambos os objetos precisam de métodos de sincronização de tempo, como o método *PCM_OGMChannel::Synchronise*, que vai inserindo atrasos caso seja necessário em cada leitura de um frame de áudio, obtido do arquivo pré-gravado e armazenado temporariamente no buffer de envio até o seu encapsulamento RTP. A operação contrária é bastante similar. Na escrita do arquivo contendo a gravação, o método *PCM_RecordFile::DelayFrame* vai ditando o ritmo de gravação dos frames de áudio que estão armazenados no buffer de recepção ou layout.

Portanto, devido a esta bufferização e sincronismo, não é possível “ouvir” a degradação obtida pelos atrasos fim-a-fim, ao tocar posteriormente arquivo de áudio gerado pela secretária. Uma vez que esta degradação somente poderia ser percebida em uma conversação interativa, o que não é o caso da secretária.

3.2 - Modificações para criar o *Call Generator* e Melhorar a Precisão das Ferramentas

Foram realizadas algumas modificações no *openam*, para que ele pudesse suportar o acionamento automático de uma ligação H.323. Inicialmente, nos baseamos no código-fonte do *call generator* [6], que usava a versão 1.1.9 do *openam*. As modificações realizadas foram as seguintes:

- Desativar o mecanismo de *listenerH323*, do *openam*, que fica bloqueado esperando receber ligações. Trocando este procedimento por uma chamada a um método *MakeOutgoingCall*.
- Estender alguns métodos de captura de eventos do *H323Endpoint* que são específicos para um elemento (*endpoint H.323*) que realize chamadas ao invés de recebê-las, como *OnAlerting*, *OnConnectionEstablished*, *OnConnectionCleared*, *AwaitTermination* e o próprio *MakeOutgoingCall*.
- No caso do método *MakeOutgoingCall*, é obtido como parâmetro do vetor de argumentos de entrada do programa, o endereço relacionado ao ponto que deve ser chamado, e dentro deste método é feita a chamada propriamente dita para este IP.
- Foi modificado o código do método *OnConnectionEstablished* do *caller* e do *openam*, para fazer acesso a sessão RTP iniciada e ajustar a frequência de envio de pacotes RTCP para 1 pacote a cada segundo. Anteriormente, esta taxa de envio era de 1 a cada 12 segundos (no caso do Netmeeting esta taxa é de 1 a cada 5 seg.), valores inapropriados para medidas mais precisas dos parâmetros de qualidade.
- No código do *caller/openam* foi criado mais um parâmetro de entrada (*--dejitter*), onde pode ser inserido o valor do buffer de compensação de jitter em ms. Tendo como faixa de valores válidos de 20 ms até 10 s. Lembrando que para conversas interativas o atraso máximo one-way não pode ultrapassar 150 ms [18].

- Também foram feitas modificações na biblioteca OpenH323, mais especificamente, nos objetos que controlam o funcionamento do stack RTP/RTCP, onde foi criada uma variável chamada *JitterInstant* que representa o valor D da fórmula do *jitter*. Complementando, esta mudança, foi alterado o evento de chegada de pacotes RTP para poder imprimir no log, o valor de D a cada intervalo de dois pacotes consecutivos (ver seção 2.2).
- Outra mudança na pilha RTP/RTCP foi à alteração do código para imprimir no *log* o evento de perda de pacotes, portanto relacionando cada perda com seu respectivo tempo mais próximo.

Todas estas modificações tiveram como objetivo aumentar a capacidade da ferramenta (*openam*) em realizar medidas mais precisas, no contexto da avaliação de performance do backbone.

3.3 - Característica das Fases de Play e Record

É importante descrever uma característica não modificada nos programas que formam a infra-estrutura de coleta devido a sua complexidade. Tanto *openam* quanto *caller*, internamente, após o estabelecimento da chamada, têm o mesmo comportamento em suas máquinas de estado, tendo duas fases bem definidas, nesta ordem: a fase de tocar a mensagem OGM, e depois a fase de gravar a mensagem vinda da parte remota. Não se podem realizar as duas ao mesmo tempo.

Apesar da comunicação ser, durante toda a ligação, bidirecional (os pacotes RTP são enviados nos dois sentidos enquanto houver arquivos para serem tocados ou gravados), caso uma das partes caia na fase de gravação, não será mais enviado o arquivo de OGM (Ongoing Message), e o programa passará então a transmitir silêncio, até que o término da chamada seja acionado por uma das partes.

Na Fig. 3, temos a ilustração de como isso funciona. De um lado temos o programa *caller* com um *arquivo wav + DTMF 4* como sua mensagem de OGM, enquanto que no lado *openam* temos somente o *arquivo wav*. Após ter sido feito o estabelecimento de chamada (fases de Setup/Connect no H.225 e OpenLogicalChannel no H.245), os programas automaticamente entram no estado de “*play OGM*”. Assim nos primeiros segundos o que temos é os dois programas enviando pacotes RTP com o arquivo de áudio, mas nenhum deles gravando.

No passo seguinte, na recepção do arquivo de mídia pelo *openam*, existe um objeto que captura eventos relacionados à detecção de DTMF (dual-tone multi-frequency), os dígitos discados dentro do próprio fluxo de áudio. Esta funcionalidade de detecção de DTMF faz parte do esforço de flexibilização do *openam* pelos programadores do *openh323*, para que este possa ser estendido futuramente para um sistema de Call Center IP. No caso, somente o **DTMF 4**, é o que aciona a mudança de estado de “*play OGM*” para “*record file*”, os outros dígitos (DTMFs) não tem sentido para o programa.

Uma vez no estado de “*record file*”, o *arquivo wav* que está sendo tocado remotamente, fica sendo gravado em um *arquivo wav* local, correspondendo ao arquivo original mais as degradações sofridas por perda de pacotes nesta comunicação.

Outra condição para a mudança de estado acontece, quando o *arquivo wav* que está partindo do *caller* acaba (Fig.3). Então o *caller* passa para o estágio de “*record file*”,

mas já é tarde demais e o procedimento de término da chamada (Release Complete) é ativado no 150^o segundo, no *openam*.

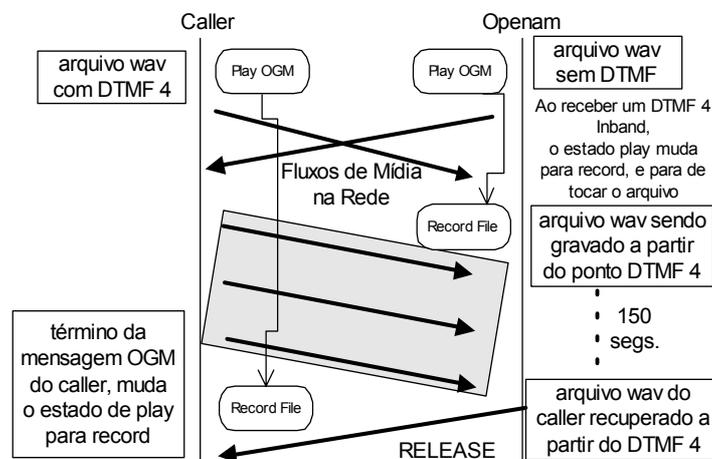


Fig. 3 – Gravando arquivos degradados no cliente remoto

O arquivo wav do *caller* é chamado de *arquivo wav* de referência e com ele podemos avaliar a qualidade perceptual dos usuários frente a rede, comparando subjetivamente este com o arquivo degradado sendo gravado remotamente.

Para o propósito deste arquivo de referência, construímos um arquivo contendo um locutor contando números pausadamente, isto porque queríamos ter períodos de fala e períodos de silêncio bem definidos.

Outra razão para usar a contagem de números é que fica mais fácil buscar no arquivo a posição em segundos de uma degradação. Este arquivo pode ser visualizado com a ferramenta Audacity [8], ferramenta que foi inicialmente concebida para realizar a análise de algoritmos de áudio (Fig.4), e que para nós ajuda a comparar “visualmente” os arquivos de referência e degradado. Bastando para isso, colocar lado a lado, os respectivos “gráficos” dos dois arquivos de áudio, e verificar a falta de certo picos de frequência ao longo da ligação (*gaps* no gráfico).

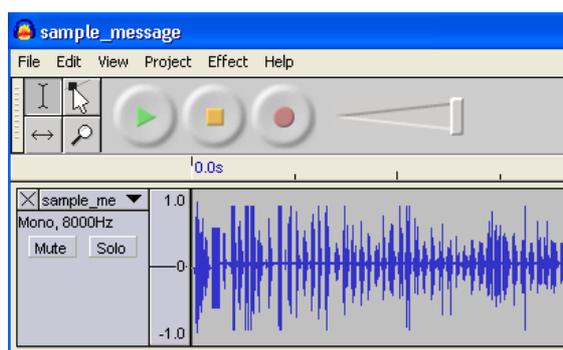


Fig. 4 – Visualização do arquivo de referência

3.4 – Arquivos de Logs

Os programas, tanto o *caller* ou *openam*, envolvidos nos testes geram a cada ligação um extenso arquivo de texto, contendo os logs com todas as medidas realizadas (perdas instantâneas, perdas acumuladas, RTT, diferença entre chegadas, jitter suavizado RTCP, entre outras), com cerca de 1 MB de tamanho.

A este arquivo é dado um nome formado pela par de nomes dos locais envolvidos na ligação, um L de local, dia, hora, minuto e segundo (ex. nceufrij-brasilia-2002-L-10-24-15-12-34.txt). Esta marcação de tempo é extraída do primeiro pacote contendo o NTP do emissor para o receptor. Desta forma, quando este pacote é recebido no receptor, temos um nome semelhante, na localidade e sentido, apenas diferenciado pela letra –R– de remoto, ao invés do –L– de local.

Uma preocupação constante nos testes é com relação a sincronização de tempo das medidas, dado que as máquinas podem estar em tempos diferentes, tomamos sempre como base das medidas a mesma máquina emissora, sincronizada constantemente com o servidor NTP da UFRJ, que é sincronizado pela RNP2.

Cada arquivo de log tem as seguintes linhas de texto, dos eventos relacionados a recepção de cada pacote RTP, importantes para as medidas (Tab.1):

```
0:12.284 RTP Jitter:864de88 RTP Receive statistics: packets=396 octets=95040
lost=0 jitterInstant=0 jitter=10
0:12.311 RTP Jitter:864de88 RTP Receive statistics: packets=397 octets=95280
lost=0 jitterInstant=24 jitter=9
0:12.353 RTP Jitter:864de88 RTP Receive statistics: packets=398 octets=95520
lost=0 jitterInstant=120 jitter=10
```

Tab.1 – Arquivo de Log com medidas instantâneas

Em negrito, temos os tempos de ativação dos eventos relacionado à recepção de pacotes RTP, com as estatísticas instantâneas para cada pacote recebido. No caso, três medidas são particularmente úteis: **lost** que representará quando um pacote for perdido, **jitterInstant** ou o valor D da equação do jitter, e o jitter propriamente dito segundo a RFC RTP, sendo suavizado pela fórmula do RTCP, que mostra tendências de congestionamento.

Outras linhas de logs importantes são as que mostram os pacotes RTCP (no caso do caller e openam, eles são pacotes compostos por 1 SR, 1 RR e 1 SDES; observe que o instante de recepção do pacote é o mesmo para os três relatórios):

```
0:12.374 RTP Jitter:864de88 RTP OnRxSenderReport: ssrc=1803467860 ntp=2002/10/24-10:12:59.610288 rtp=95520 psent=398 osent=95520
0:12.374 RTP Jitter:864de88 RTP OnRxSenderReport RR: ssrc=2232852623 fraction=0 lost=0
last_seq=0 jitter=39 lsr=0.000 dlsr=0.000
0:12.374 RTPJitter:864de88 OnSourceDescription: ssrc=1803467860 item[0]: type=CNAME
data="voip@server3.pop-df.rnp.br" item[1]: type=TOOL data="OpenAM"
```

Tab.2 – Arquivo de Log com os pacotes RTCP

No pacote RTCP acima (Tab. 2), temos os campos com parâmetros envolvidos nos relatórios de emissor (SR) como **ssrc** (identificação da fonte), **ntp** (tempo absoluto em formato NTP da máquina emissora), o **timestamp RTP** daquele pacote SR, entre outras (**psent** = packet sent). Também no mesmo pacote, podemos verificar um relatório de receptor (RR) da outra fonte (lembrar que o fluxo é bidirecional) identificado por outro número de **ssrc**, além dos valores da fração de perda (**fraction**), e valor acumulado do número de pacotes perdidos (**lost**). Um dado importante, é que os valores de **LSR** (Last Sender Report) e **DLSR** (Delay Since Last Report) usados para calcular o *Round-Trip-Time* (RTT) não são preenchidos pelos programas.

Para calcular o RTT, usamos um algoritmo simples baseado nos eventos de chegada e partida dos pacotes RTCP, ordenados por tempo e procurando uma correspondência temporal entre eles.

3.5 - Integração com Outras Medidas Complementares (traceroute e MRTG)

Além de realizar a série de medidas RTP/RTCP, e armazenar um arquivo de áudio degradado simulando perfeitamente uma ligação de telefonia IP baseada em H.323, também foi projetado para este ambiente de coleta, uma forma de integrar informações complementares sobre o estado da rede nos nós intermediários para facilitar a exploração da origem de certa falha.

Com este propósito, decidimos juntar à ferramenta de geração de chamada, um *script* desenvolvido em *perl*, para realizar automaticamente um traceroute do caminho origem da ligação IP até o ponto final. Caso neste caminho, exista algum roteador do núcleo da rede RNP2, então o *script* busca no site de estatísticas da RNP [9], os valores dos relatórios de vazão em bits, em pacotes e atrasos, tudo baseado no sistema MRTG [10] sobre o estado atual daquela interface de rede.

Com isso, não temos somente as informações fim a fim, como também alguma informação do meio do caminho que pode ser considerada útil. Assim sendo, outro arquivo de pouco mais de 500 bytes é gerado a cada ligação tanto na máquina do *caller* quanto na máquina do *openam* remoto. Eis um exemplo de um arquivo de traceroute concatenado com as estatísticas obtidas das páginas web do MRTG dos roteadores do núcleo da RNP2, como o 200.143.254.137.

```
146.164.247.193---- 1 146.164.247.193 0.325 ms
146.164.8.193---- 2 146.164.8.193 1.632 ms
200.20.94.58---- 3 200.20.94.58 2.343 ms
200.143.254.22---- 4 200.143.254.22 1.942 ms
200.143.254.137----(Vazão em bits - Entrada) 10382171 6697466 7117723 (Vazão em bits - Saída) 23904966 15202608 22127953 | (Vazão em pacotes - Entrada) 3749 2331 3096 (Vazão em Pacotes - Saída) 4979 3415 4441 | (Delay - Entrada) 66 37 37 (Delay - Saída) 100 0 0 | - 5 200.143.254.137 39.598 ms
200.19.119.123---- 6 200.19.119.123 53.111 ms
```

Tab. 3 – Log do traceroute com integração MRTG RNP2 (RJ-DF)

No exemplo acima (Tab. 3), temos uma ligação partindo da máquina 146.164.247.193 (Lab. VOIP NCE/UFRJ) até uma máquina em 200.19.119.123 (Brasília no POP-DF).

4 - Consolidação e Visualização das Estatísticas

Como vimos na seção anterior, cada máquina ou espalhada pelos pops RNP ou máquina interna do lab. VOIP vai gerando três arquivos a cada ligação: o arquivo com o áudio degradado com cerca de 2 MB (2 minutos e meio), o arquivo com os logs com cerca de 1.0 MB, e o arquivo contendo o traceroute + MRTG daquele instante com cerca de 1 KB. Ou seja, no caso de medidas sendo realizadas a cada hora, temos ao final de 24 horas, cerca de 80 MB em arquivos na máquina remota. Portanto, faz-se necessário um mecanismo de limpeza dos arquivos e de consolidação dos logs em base de dados centralizada.

Assim, foi desenvolvido um *script perl* que é acionado todo dia em horários de pouca utilização da internet, como de madrugada, para fazer este serviço. Nestes horários quando o *script* entra em ação ele abre uma conexão TCP com uma máquina com alta capacidade do lab.VOIP, e envia todos os arquivos.

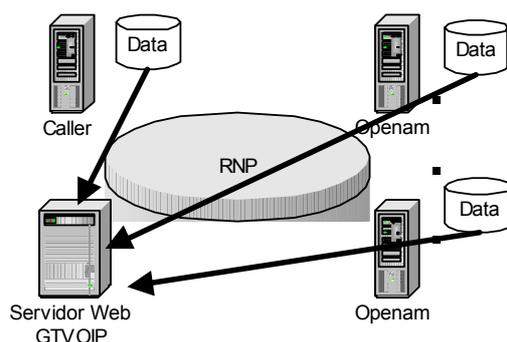


Fig.5 – Consolidando os arquivos logados

Na máquina centralizadora dos logs entra em ação um script de pós-processamento dos dados cujo objetivo é extrair medidas estatísticas (como média, desvio padrão, valor máximo e valor mínimo) que possam resumir todo o log de uma única ligação. Isso é importante para termos um panorama mais amplo sobre as ligações ao longo do dia ou da semana. Embora as medidas mais precisas não sejam perdidas, pois elas permanecem armazenadas neste servidor, no caso de ser necessário visualizá-las.

A Fig. 5 ilustra o sistema de consolidação de arquivos dos logs estatísticos. Dentro do servidor de consolidação, além da funcionalidade de pós-processamento e armazenamento dos logs, também foram desenvolvidas interfaces Web para visualização gráfica dos dados. Estas interfaces estão divididas em três níveis:

- Visualização condensada de informações;
- Visualização das estatísticas de uma ligação em particular;
- Pesquisa de ligações pelos parâmetros de qualidade de serviço como jitter / RTT / perda

4.1 – Visualização Condensada de Informações

A interface Web (Fig. 6, ao lado) tem como objetivo condensar a informação de um período (em dias) de testes de VOIP em um grupo resumido de gráficos (RTT, perda e jitter). Tem, como entrada principal, os dois pontos envolvidos na ligação e o sentido da ligação (de emissor para receptor ou vice-versa). A página web com o mapa do Brasil e os ícones do lab.VOIP ilustram esta interface, totalmente baseada em menus Javascript.

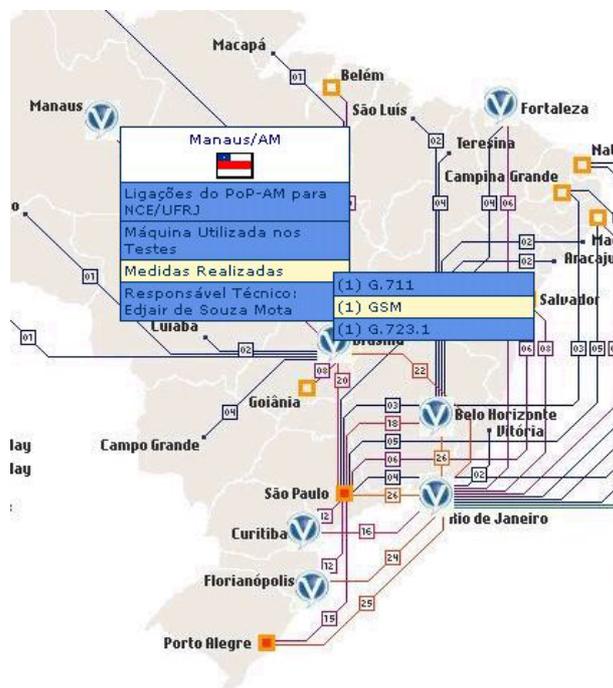


Fig. 6 – Interface Web baseada em Javascript para visualizar resultados

Escolhida a ligação, é gerado automaticamente, através de bibliotecas *perl* (como GD.pm e GDGraph.pm [11]), uma página web contendo um conjunto de 3 gráficos gerados: gráfico do RTT, perda e jitter.

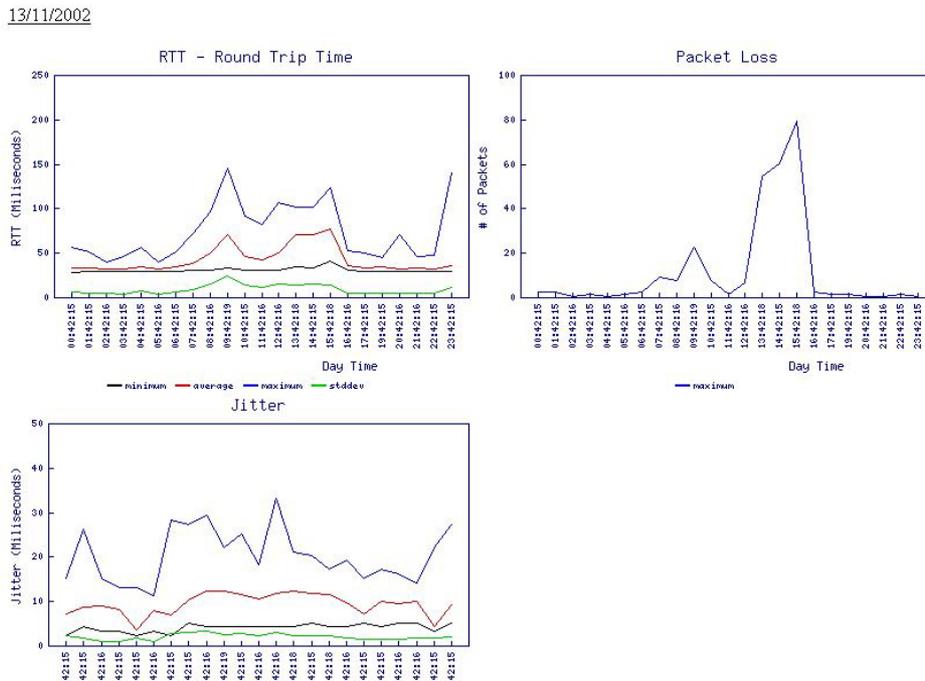


Fig. 7 – Exemplo de Resumo RTT (Round Trip Time), Packet Loss e Jitter ao longo do dia 13/11/2002 entre Brasília e Rio de Janeiro. O eixo X representa a hora da ligação.

Cada ponto do eixo X deste gráfico de RTT (Fig. 7) representa uma ligação que foi feita contendo o dia e hora da mesma. Sendo que as linhas coloridas, respectivamente descritas na legenda, informam quais os pontos mínimos, médios, máximos, e o desvio padrão destas ligações. Ao final desta página de visualização condensada é possível escolher realizar um zoom de uma destas ligações em particular para explorar mais detalhadamente os parâmetros da ligação (Fig. 8).

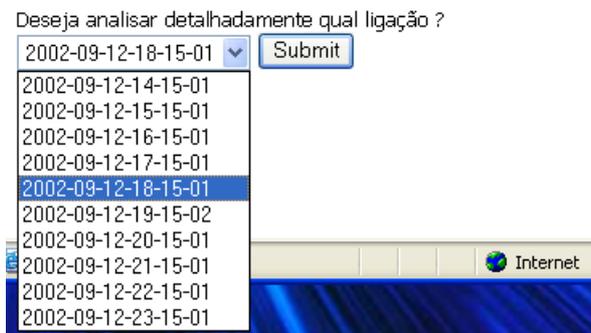


Fig. 8 – Detalhe da página Web, onde é feita a escolha de uma ligação a ser detalhada quanto aos seus parâmetros ao longo dos 3 minutos.

4.2 – Visualização das Estatísticas de Uma Ligação em Particular

Depois de escolher uma ligação, e clicar em *Submit*, uma nova página web é gerada dinamicamente, contendo os detalhes daquela ligação. Estes detalhes são: o round-trip time, calculado pela integração dos valores dos relatórios RTCP SR/RR de dois pacotes

consecutivos nos dois logs (ou seja, cerca de 150 RTCP SR/RR), a perda instantânea e o jitter entre chegadas de pacotes RTP. A Fig. 9 ilustra um gráfico de RTT detalhado e a Fig. 10 ilustra um gráfico da perda instantânea, ambos foram medidas realizadas entre Amazonas e Rio:

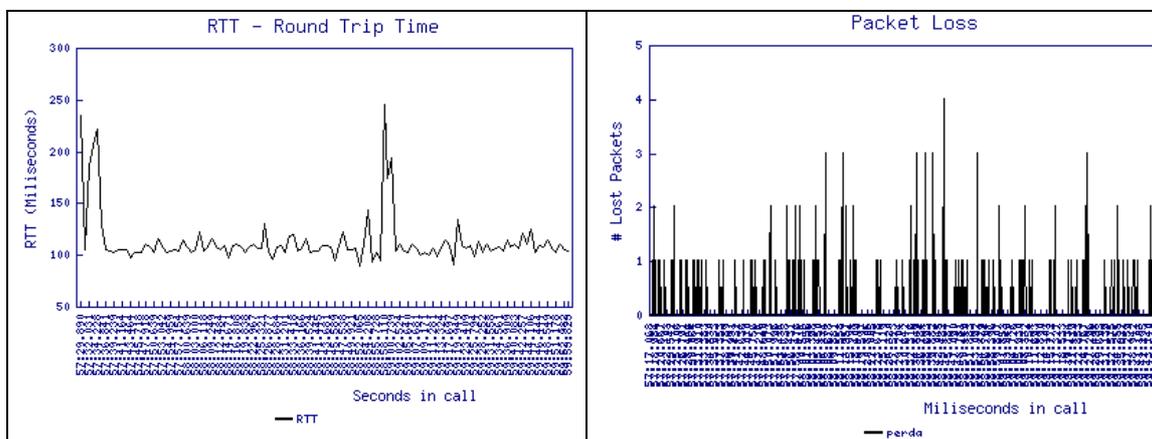


Fig. 9 – Detalhamento de uma ligação, valores de RTT ao longo de 150 segundos de ligação

Fig. 10 – Detalhamento de uma ligação, perdas instantâneas ao longo de 150 segundos de ligação

Obs.: O eixo Y da Fig.9 representa o RTT em milissegundos, enquanto o eixo X representa o tempo em segundos dentro da chamada onde foi obtido tal valor de RTT (150 pontos). No caso da Fig. 10, está representada a perda de pacotes, onde no eixo Y temos quantidade de pacotes perdidos em um determinado intervalo entre pacotes expresso em milissegundos (cerca de 50.000 pontos) no eixo X.

Estas informações detalhadas são todas aquelas armazenadas nos arquivos de logs gerados ao longo da ligação, portanto, é possível usar, a qualquer momento, estes dados para gerar automaticamente estes gráficos.

Com a sobreposição dos gráficos dos principais parâmetros de qualidade de serviço é possível verificar como problemas de RTT, perda e jitter se correlacionam naquela ligação com uma maior clareza.

Também a informação do caminho (*traceroute*) utilizado, bem como as estatísticas obtidas no MRTG naquele instante de ligação, são exibidos na página de zoom, e finalmente são criados links web para os arquivos de áudio de referência e o degradado relacionado com aquela ligação.

Detalhamento da chamada
 De : amazonas
 Para : nceuffj
 Hora : 11:35:21
 Data : 25/11/2002
 Codificador : ITU-T G.723.1
[Ligação original=> Ouvir esta ligação](#)

Traceroute

IP	Tempo	MRTG					
200.129.156.100	1 ms						
200.129.156.200	2 ms						
200.143.254.242	28 ms	In			Out		
	%bps	Maximum	Average	Current	Maximum	Average	Current
		19.49%	2.94%	6.69%	91.54%	6.34%	34.98%
	pcts/seg	Maximum	Average	Current	Maximum	Average	Current
		153	14	55	121	12	69

Fig. 11 – Detalhe dos parâmetros de uma ligação, incluindo os arquivos de áudio para serem ouvidos, e o traceroute gerado naquele momento.

4.3 – Pesquisa de Ligações pelos Parâmetros de Qualidade de Serviço como jitter / RTT / perda

Muitas vezes, a quantidade de dados de ligações é tão grande, que pesquisar página por página, ligação a ligação a fim de analisar as estatísticas pode se tornar uma atividade massacrante. Neste sentido desenvolvemos uma página de busca por ligações que atendam certos requisitos. A partir desta página é possível acessar o resumo de certas ligações e então seguir diretamente para a ligação problemática.

As figuras 12 e 13 apresentam, respectivamente, a interface Web de pesquisa e o resultado de uma pesquisa por perda de pacotes alta acima de 150 pacotes.

Localidade : <input type="text" value="Do PoP-AM para NCE/UFRJ"/>	
Codec : <input type="text" value="G.723.1"/>	
RTT : <input type="text" value="100"/>	
Perda : <input type="text" value="150"/>	
Jitter : <input type="text"/>	
<input type="button" value="Enviar"/> <input type="button" value="Limpar"/>	

Ligação	RTT	Perdas	Jitter
nceufri-amazonas-R-2002-11-25-11-35-21	190	191	47
nceufri-amazonas-R-2002-11-26-08-35-21	244	175	37

Fig. 12 – Interface Web de Pesquisa por ligações com certos parâmetros de qualidade
Fig. 13 – Resultado da pesquisa da Fig. 11 sobre medidas realizadas entre PoP-AM e NCE/Rio

5 - Conclusões e Trabalhos Futuros

Neste trabalho, apresentamos um conjunto de ferramentas de geração, monitoramento e coleta de estatísticas a serem utilizados para verificar o nível de qualidade que ligações VOIP sofrem ao serem feitas através da RNP2.

Entre as contribuições alcançadas temos um estudo dos parâmetros de feedback de qualidade de serviço utilizado pelo protocolo RTP/RTCP. A modificação do programa de secretária eletrônica (openam) do projeto OpenH323 aumentando a precisão das medidas, bem como viabilizando uma flexibilização da geração de relatórios RTCP em um intervalo pequeno, e modificando o buffer de compensação de jitter.

Também foi estudado, um método de gravação de arquivo de áudio, que capture a característica de degradação por perda obtidas na transmissão de multimídia na RNP2, e criado um arquivo de referência para posterior avaliação subjetiva por parte de usuários.

O sistema de geração de ligações de telefonia IP simuladas e coleta de estatísticas é pequeno (com cerca de 10 MB entre os scripts e programa executável), leve (não demanda muitos recursos computacionais e gera 80MB no máximo em disco por dia). Ele foi portado para vários ambientes como Linux e FreeBSD, podendo também ser portado para Windows facilmente.

Alguns parâmetros usados nos testes preliminares foram obtidos da literatura, como o tamanho considerado bom para o buffer fixo de compensação de jitter para a Internet pública não interativa (por meio de secretárias eletrônicas), com cerca de 150 ms [12] de modo a diminuir as perdas de pacotes por jitter. Também o tamanho da ligação foi fixado em cerca de 2 minutos e meio, valor este que representa a grande maioria de ligações VOIP, na rede MCI, segundo estudo feito em [13]. No futuro, esperamos

trabalhar com outros parâmetros, como pelo desenvolvimento de buffer de playout adaptativo que diminui o atraso fim-a-fim, e mecanismos de FEC (Forward Error Correction) que embutem redundância no fluxo de áudio para protegê-lo de perdas [14]. Este estudo pode comprovar a eficácia destes métodos na RNP2.

Além disso, foi desenvolvido um sistema de recolhimento de logs, e de *arquivos wav* dos pontos remotos; e a centralização dos mesmos em uma máquina com excelente poder de processamento e armazenamento, para facilitar a consolidação e visualização online dos resultados.

Entre os passos futuros, vislumbramos o aprofundamento nas questões da avaliação automática, dita objetiva, da qualidade da mídia através de algum algoritmo padronizado, como o e-model da ITU-T [15]. Recentemente, já foram feitas experiências neste sentido no âmbito do projeto TIPHON, inclusive testando ligações de VOIP [16]. Logo, essa será a continuação natural deste trabalho.

6 – Bibliografia Consultada

- [1] IETF – RFC 1889 **RTP: A Transport Protocol for Real-Time Applications**.
- [2] **NetIQ Chariot VoIP** Test Module. <http://www.netiq.com/products/chr/default.asp>.
- [3] IETF – RFC 1305 **Network Time Protocol (Version 3) Specification, Implementation and Analysis**.
- [4] IETF – Draft **RTCP Reporting Extensions**. draft-ietf-avt-rtcp-report-extns-01.txt. Trabalho em andamento.
- [5] <http://www.openh323.org>
- [6] ITU-T Recommendation H.323, **Packet-Based Multimedia Communications Systems**, Setembro 1999.
- [7] Milen Stoychev. <http://cns.cnsys.bg/voice/> (page visited 05/2002)
- [8] <http://audacity.sourceforge.net>
- [9] Estatísticas dos roteadores do backbone RNP2. <http://www.rnp.br/ceo/ceo-estatisticas.html> (page visited 01/2003)
- [10] **Multi Router Traffic Grapher**. <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>
- [11] <http://www.cpan.org/>
- [12] Athina P. Markopoulou, Fouad A. Tobagi, Mansour J. Karam. **Assessment of VoIP Quality over Internet Backbones**. IEEE Infocom 2002.
- [13] J. van der Merwe, R. Cáceres, Y-H. Chu, C. Sreenan. **Mmdump: A Tool for Monitoring Internet Multimedia Traffic**. ACM Computer Communication Review, 30(4), October 2000.
- [14] J. Rosenberg, **Distributed Algorithms and Protocols for Scalable Internet Telephony**, Tese de Doutorado, Univ. de Columbia, NY. maio 2001.
- [15] ITU-T Recommendation G.107, **The Emodel, a computational model for use in transmission planning**, December 1998.
- [16] ETSI TS 101 329-5. **Quality of Service (QoS) measurement methodologies**. TIPHON, 2000.
- [17] Hewlett Packard Products. **HP J4618A Telegra RQM software Manual**. Setembro 1999.
- [18] ITU-T Recommendation P.800 **Methods for subjective determination of transmission quality**. Agosto, 1996.