

Arquitetura de Monitoração de Qualidade de Chamadas Telefônicas IP

Leandro C. G. Lustosa, Paulo H. de A. Rodrigues, Fabio David, Douglas G. Quinellato

Laboratório de Voz Sobre IP – Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro (NCE/UFRJ)*
Caixa Postal 2324 – 20001-970 – Rio de Janeiro – RJ – Brasil

{leandro,aguilar,fabio,douglasq}@nce.ufrj.br

Abstract. *A voice quality monitoring architecture based on the development of the VQuality library is described. VQuality implements the E-model and its extensions for objective voice quality measurement. The library also supports the generation of a voice quality CDR with extensions for the transfer of quality parameters collected over different instants of time along the call, besides interacting with RADIUS for data collection. The framework is exemplified in the context of the fone@RNP VoIP service and the use of a modified Openphone client incorporating the new functionality is shown.*

Resumo. *Uma arquitetura para a monitoração da qualidade de chamadas VoIP é apresentada, baseada no desenvolvimento da biblioteca VQuality, que implementa a medição objetiva calculada pelo Modelo E e por suas extensões. Esta biblioteca incorpora ainda a geração de CDR próprio com suporte ao envio de dados referentes à variação da qualidade ao longo da chamada, além de um mecanismo para a coleta do CDR via RADIUS. A descrição do ambiente é apresentada no contexto do serviço fone@RNP e é exemplificado o uso em um cliente Openphone incorporando a funcionalidade descrita.*

1. Introdução

Os sistemas de comunicação baseados em voz sobre IP (VoIP) se difundiram e representam hoje uma solução tecnológica viável para provisão de serviços de voz. Neste contexto, a Rede Nacional de Ensino e Pesquisa (RNP) oferece um serviço de telefonia IP que utiliza sua infra-estrutura de redes para prover comunicação VoIP à comunidade de ensino e pesquisa usuária da rede.

O serviço, denominado fone@RNP, utiliza uma arquitetura baseada no padrão H.323 [18] que, além de interconectar os sistemas privados de telefonia (PBX) de uma série de instituições acadêmicas nacionais e fornecer uma plataforma para utilização de

*Pesquisadores parcialmente suportados com recursos do GT-VoIP Avançado, grupo de trabalho mantido pela RNP. Paulo H. de A. Rodrigues é também professor do Departamento de Computação/IM da UFRJ.

telefones IP (hardwares dedicados ou softwares rodando em PCs com capacidade multimídia), permite a interconexão com a rede de telefonia pública comutada (PSTN) e com redes de telefonia IP acadêmicas de outros países, via Internet2.

Cada instituição usuária do serviço possui um ambiente composto por um *gatekeeper* (GnuGK [12]) para registro e localização de terminais, um servidor RADIUS [11] (com banco de dados SQL associado) para autenticação de usuários e contabilização das chamadas, e de um *gateway* de voz opcional, para a interconexão com o PBX/PSTN, conforme mostrado na Figura 1.

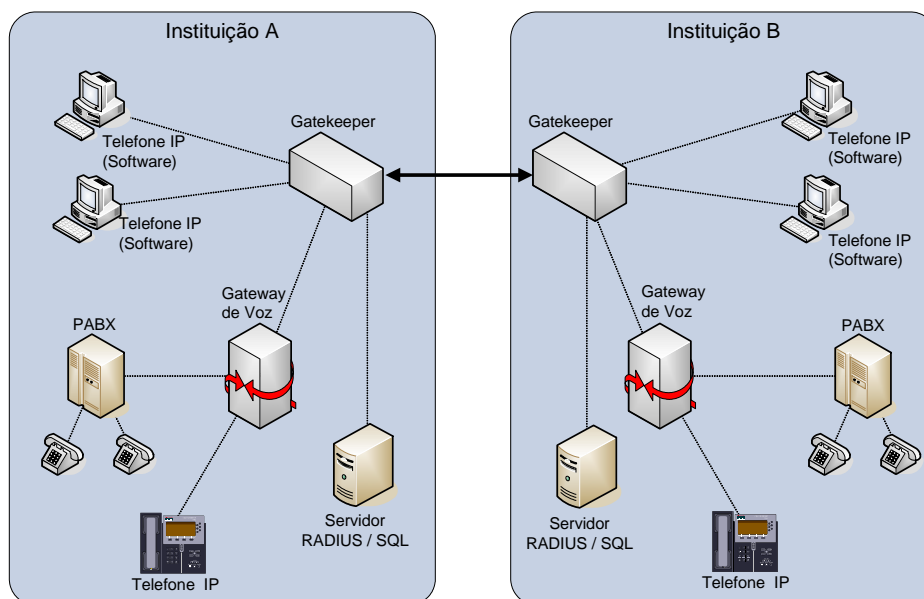


Figura 1: Arquitetura do serviço fone@RNP

As estatísticas de qualidade das chamadas representam uma informação muito valiosa para o monitoramento do sistema, auxiliando na detecção de problemas, na validação de novas configurações da rede e no acompanhamento do grau de satisfação dos usuários. A contabilidade e acompanhamento da qualidade das chamadas são realizados coletando-se os registros de detalhamento da chamada (CDR) emitidos pelo *gatekeeper* [12] e pelo *gateway* de voz [16]. Entretanto, somente os CDRs gerados pelo *gateway* de voz fornecem indicativos de qualidade que, embora sejam simplistas comparando-se aos fornecidos por modelos mais sofisticados como o Modelo E estendido [2,4,5], oferecem alguma estimativa da qualidade da chamada. Dessa forma, somente as chamadas que envolvem o *gateway* são qualificadas, tornando as estatísticas pouco confiáveis para caracterizar o sistema como um todo, já que a qualidade de chamada realizada entre telefones IP não é obtida.

A CESNET, rede de ensino e pesquisa da República Tcheca, implementou uma infra-estrutura similar de monitoração e contabilização de chamadas por meio de utilização de servidores RADIUS e coleta de CDRs emitidos por *gateways* de voz [19]. Em [20], é apresentada uma arquitetura denominada *Enterprise Call Analysis System* (ECAS), também baseada na coleta e análise de CDRs gerados por *gateways* de voz. Estas e outras soluções apresentam a mesma limitação do serviço fone@RNP, pois não permitem contabilizar a qualidade das chamadas envolvendo apenas telefones IP.

Em [21], é apresentada uma solução pouco confiável para acompanhamento da qualidade oferecida pelo sistema. Agentes distribuídos pela rede simulam chamadas periodicamente e disponibilizam, via SNMP, informações de perdas de pacotes, atraso e variação de atraso (*jitter*). Esta solução não levanta estatísticas das chamadas reais do sistema e utiliza uma amostragem que pode não ser fiel às condições reais da rede.

Em [22], assim como em diversos produtos comerciais [23,24,25], uma solução baseada em monitoramento passivo é utilizada. Elementos (*probes*) com capacidade de capturar e analisar fluxos de voz são situados em pontos estratégicos da rede com o intuito de monitorar todos os fluxos de voz do sistema. Os resultados dessas análises são transformados em relatórios de acompanhamento do nível de qualidade oferecida. Esta solução nem sempre é aplicável, seja pelo uso de criptografia, que impede a análise das chamadas, ou pela natureza de tráfego de voz, que pode ser ponto-a-ponto (*peer-to-peer*), dificultando assim a captura das chamadas.

Uma solução alternativa está sendo operacionalizada no ambiente do serviço fone@RNP. Uma biblioteca com capacidade de avaliação de qualidade de voz foi desenvolvida e está sendo integrada aos telefones IP. Esta biblioteca, denominada Voice Quality Library (*VQuality Lib*), implementa o Modelo E [3] e suas extensões [2,4,5] para avaliar a qualidade de voz recebida, e um mecanismo de envio de CDR, chamado Voice Quality CDR (*VQCDR*).

Objetivando tornar o levantamento de estatísticas do serviço fone@RNP mais confiável e abrangente, são utilizados os recursos fornecidos pela biblioteca *VQuality* para criar uma arquitetura de coleta de CDRs capaz de obter indicadores de qualidade diretamente dos telefones IP. Estes indicadores são extremamente detalhados e mostram-se mais fiéis à percepção humana que os indicadores fornecidos pelo CDR dos *gateways* de voz [16,17], devido ao uso do Modelo E estendido [2,4,5]. A arquitetura resultante permite que, ao final de cada chamada, dois CDRs com informações de qualidade sejam coletados, seja a chamada estabelecida entre dois telefones IP, entre um telefone IP e um *gateway* de voz, ou entre dois *gateways* de voz.

O restante do artigo está estruturado como segue. Na seção 2, é descrita a biblioteca *VQuality*. Na seção 3, a arquitetura para a coleta dos CDRs é apresentada. Finalmente, as conclusões são apresentadas na seção 4.

2. Voice Quality Library

Um dos esforços para a operacionalização do serviço fone@RNP tem sido a caracterização e a avaliação do transporte de voz em tempo real no *backbone* da RNP2. A primeira iniciativa neste sentido foi a criação de uma infra-estrutura [1] de geração, coleta e monitoramento da qualidade de ligações VoIP baseada na biblioteca OpenH323 [7] e capaz de extrair, por meio de monitoração ativa, dados estatísticos referentes ao nível de transporte das chamadas, além de apresentar graficamente informações de perdas de pacotes, variação de atraso (*jitter*) e RTT (*Round Trip Time*) ao longo do tempo.

A continuação deste trabalho foi o estudo de modelos analíticos de avaliação objetiva de qualidade de voz, como o Modelo E [3], da ITU (*International Telecommunication Union*), e de sua extensão [4,5], proposta pela ETSI (*European Telecommunications Standards Institute*). Como fruto deste estudo, foram

desenvolvidas uma proposta de revisão a estes modelos [2] e uma implementação conhecida como MOBDEM (*Modified OpenH323 Based Voice Evaluation Module*).

O MOBDEM é um módulo objetivo para avaliação de qualidade de voz que utiliza uma biblioteca OpenH323 modificada para gerar logs detalhados das chamadas com os parâmetros necessários para o cômputo do Modelo E e de suas extensões [2,4,5]. Embora ofereça os requisitos necessários para uma ferramenta de medições de qualidade de voz em ambientes de teste, a MOBDEM possui uma série de limitações que inviabilizam sua adoção para medições em ambientes de produção.

Desenvolvida em Perl [6], não dispõe da mesma performance de uma linguagem compilada, não oferece uma API (*Application Programming Interface*) que disponibilize um mecanismo, via biblioteca estática ou compartilhada, para integração com outros softwares, além de somente operar no escopo do OpenH323. Outra limitação relevante, mais precisamente atribuída ao OpenH323, é a dificuldade da obtenção do atraso na rede, já que o cálculo de RTT não é implementado por sua pilha RTP (*Real Time Protocol*) [8]. Para estimar o RTT, o MOBDEM necessita analisar simultaneamente os logs gerados pelo transmissor e pelo receptor da mídia, inviabilizando, assim, a análise da qualidade de voz em tempo real e a possibilidade de realizar medições com clientes que não implementem a biblioteca OpenH323 modificada, e que, por conseguinte, não geram o log no formato requerido.

Objetivando superar estas limitações, foi desenvolvida a biblioteca VQuality. Escrita em C++, herdou todas as características positivas da implementação anterior, como o cômputo do Modelo E [3] e suas extensões [2,4,5]. Todavia, além da performance bastante superior, é flexível, extensível e portátil.

2.1. Flexibilidade e Extensibilidade

A biblioteca VQuality foi desenvolvida com técnica de orientação a objetos, oferecendo ampla facilidade de extensão, seja para suportar novos modelos de avaliação objetiva de qualidade de voz, seja para ser portada para novos sistemas operacionais ou para ser integrada a diferentes pilhas de sinalização VoIP.

2.2. Portabilidade

A portabilidade para diferentes sistemas operacionais é conseguida com a implementação em C/C++ padrão, exceto pelas funções de *sockets TCP* e *threads*, que são implementadas diferentemente em cada arquitetura. Foi desenvolvida uma implementação própria da função *sockets TCP*, compatível tanto com sistemas Windows quanto com sistemas baseados em Unix. Para o tratamento de *threads* é utilizada a biblioteca Pthreads-win32 [9], que permite que sistemas Windows compilem código baseado na API de Pthreads [10], padrão para sistemas Unix. O resultado é um código flexível, que pode ser compilado tanto em Linux, FreeBSD e Windows, além de ser portátil para outras arquiteturas, se necessário.

2.3. Integração com a Biblioteca OpenH323

Sendo o H.323 o protocolo de sinalização do serviço fone@RNP, a integração com a biblioteca OpenH323 foi um dos primeiros objetivos da biblioteca VQuality. O principal desafio, neste sentido, foi a alteração da OpenH323 para implementar o

mecanismo de cálculo de RTT de sua pilha RTP. A integração permitiu o desenvolvimento de clientes H.323 com capacidade de análise de qualidade de voz e envio de CDR (ver seção 3.4).

3. Arquitetura de Coleta

Uma das funcionalidades mais importantes da biblioteca VQuality é a capacidade de envio de CDR, em tempo real. A biblioteca deve, ao término de uma chamada, realizar a avaliação objetiva da qualidade da voz recebida, levantar parâmetros de identificação da chamada e dos terminais envolvidos, e reportar estas informações a uma entidade centralizadora, responsável pela coleta dos CDRs das chamadas.

A biblioteca VQuality define um mecanismo de envio de CDR denominado VQCDR (*Voice Quality Call Detailed Record*). O VQCDR é enviado via TCP/porta 80, podendo ser selecionada outra porta, caso apropriado. O uso do TCP assegura a confiabilidade e integridade da transferência. O uso da porta 80 (HTTP) facilita a operação em redes protegidas por sistemas de *firewall*, já que dificilmente estes sistemas bloqueiam o uso desta porta. O formato do VQCDR e todos seus atributos são apresentados na Tabela 1.

Tabela 1: Campos do VQCDR

Nº.	Campo	Tamanho (bytes)	Descrição
01	version	1 *	Versão do VQCDR. A versão atual é 1.
02	protocol	1 *	Protocolo de sinalização VoIP utilizado. Ver Tabela 2.
03	IDSize	2 *	Especifica o tamanho em bytes do campo ID.
04	ID	0 a 65535	Identificação da chamada. No H.323 este campo representa o ConfID; no SIP este campo representa o Call-ID.
05	username	64 †	Identificação do usuário.
06	model	1 *	Modelo para análise da qualidade da voz. Ver Tabela 4.
07	codec	1 *	Codificador de voz utilizado. Ver Tabela 3.
08	frameSize	4 *	Tamanho do quadro de voz em microssegundos.
09	framesPerPacket	1 *	Número de quadros de voz empacotados no pacote IP.
10	IdEndOfCall	2 *	<i>Delay Impairment</i> (Id) ao final da chamada. Indicador de degradação associado ao atraso fim-a-fim e interatividade. Este valor é multiplicado por 100.
11	IeEndOfCall	2 *	<i>Equipment Impairment</i> (Ie) final da chamada. Indicador de degradação associado à codificadores de alta compressão, perda de pacotes na rede e descarte no buffer de compensação de jitter. Este valor é multiplicado por 100.
12	MOSEndOfCall	2 *	<i>Mean Opinion Score</i> (MOS) final da chamada. Representa a percepção humana da qualidade. Valor multiplicado por 100.
13	gapLossDensity	2 *	Densidade de perda em momentos de perdas isoladas (gap). Este valor é multiplicado por 100.
14	burstLossDensity	2 *	Densidade de perda em momentos de perdas em rajada (burst). Este valor é multiplicado por 100.
15	discardRate	2 *	Porcentagem de pacotes descartados no buffer de compensação de jitter, desde o início da recepção. Este valor é multiplicado por 100.
16	lossRate	2 *	Porcentagem de pacotes perdidos, desde o início da recepção. Este valor é multiplicado por 100.
17	RfactorEndOfCall	2 *	Fator R final da chamada (fator de saída do Modelo E). Este valor é multiplicado por 100.
18	duration	4 *	Duração da chamada em segundos.

19	avgNetDelay	4 *	Atraso médio da rede em microssegundos. Ver Nota 1.
20	avgJitterBufferDelay	4 *	Tamanho médio do buffer de compensação de jitter em microssegundos. Ver Nota 2.
21	avgJitter	4 *	Jitter médio em microssegundos. Ver Nota 3.
22	codecDelay	4 *	Atraso de codificação e empacotamento do pacote de voz em microssegundos.
23	packetsReceived	4 *	Total de pacotes recebidos.
24	packetsLost	4 *	Total de pacotes perdidos.
25	packetsDiscarded	4 *	Total de pacotes descartados no buffer de compensação de jitter.
26	mediaSource	256 †	Endereço (IP ou hostname) do cliente remoto, originador da mídia.
27	mediaDestination	256 †	Endereço (IP ou hostname) do cliente local, receptor da mídia.
28	localAlias	64 †	Identificador do cliente emissor do VQCDR. Pode ser um número E.164 ou uma URI.
29	remoteAlias	64 †	Identificador do cliente VoIP remoto. Pode ser um número E.164 ou uma URI.
30	direction	1 *	Identificação do originador da chamada. Ver Tabela 5.
31	appExtensionSize	1 *	Tamanho em bytes do campo appExtension. Seu valor é 0 quando o campo appExtension não está presente.
32	appExtension	0 a 255 †	Informação adicional de aplicação.
33	VQLogSize	2 *	Tamanho em bytes do campo VQLog. Seu valor é 0 quando o campo VQLog não está presente.
34	VQLog	0 a 65535 †	Log detalhado da avaliação da qualidade da chamada no formato VQLog (Ver seção 3.5).

* Valor inteiro sem sinal (byte mais significativo primeiro). † Arquivo texto.

† Texto em codificação ASCII com delimitador (hex 00).

Nota 1: Média aritmética dos últimos 16 cálculos de atraso da rede (netDelay). Se menos de 16 cálculos de netDelay estão disponíveis, esta média é calculada apenas com base nos últimos cálculos disponíveis.

Nota2: Média aritmética das últimas 16 variações de atraso do buffer de compensação de jitter. Caso não haja 16 variações disponíveis, esta média é calculada apenas com base nas últimas variações disponíveis. Caso a aplicação não opere com buffer dinâmico, este valor representa um valor constante.

Nota 3: Média aritmética das últimas 16 variações jitter no formato RTP [8]. Caso não haja 16 variações disponíveis, esta média é calculada apenas com base nas últimas variações disponíveis.

Tabela 2: Protocolos de sinalização

Protocolo	Valor
H.323	0
SIP	1
MGCP	2

Tabela 4: Modelos objetivos de avaliação

Modelo	Valor
ITU-T G.107 E-Model	0
ETSI Extended E-Model	1
UFRJ/UFAM Extended E-Model	2

Tabela 5: Identificação de originador

Direção	Valor
Chamado	0
Chamador	1

Tabela 3: Codificadores de VOZ

Codec	Valor
G.711	0
G.711 PLC	1
G.723.1 5.3 kbps	2
G.723.1 6.3 kbps	3
G.726 16 kbps	4
G.726 24 kbps	5
G.726 32 kbps	6
G.726 40 kbps	7
G.728 16 kbps	8
G.729	9
G.729A	10
GSM FR (6.10)	11

3.1. Voice Quality CDR Server

O Voice Quality CDR Server (VQCDR Server) é a entidade central responsável por coletar os VQCDRs dos clientes, verificar sua legitimidade e encaminhá-los a uma base de armazenamento. Para validar os CDRs, o VQCDR Server precisa interagir com alguma base de dados ou aplicação que permita verificar se os VQCDRs coletados foram emitidos por usuários/clientes válidos para o sistema.

O VQCDR Server é composto por três módulos, como mostra a Figura 2.

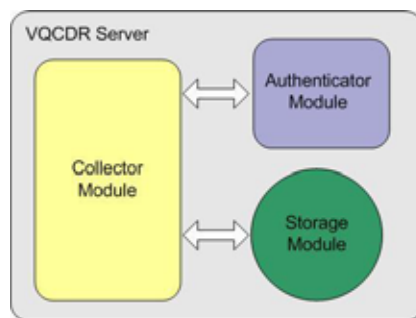


Figura 2: Arquitetura do VQCDR Server

Collector Module (CM): responsável por coletar e interpretar o VQCDR, além de acionar o Authenticator Module (AM) e o Storage Module (SM).

Authenticator Module (AM): responsável por validar os VQCDRs recebidos pelo CM. O VQCDR Server pode operar com zero ou mais AMs. Para ambientes controlados de testes e medições, o VQCDR Server oferece uma lista de acesso simples baseada em endereços IP, que pode substituir o uso de AMs. Já em ambientes de produção, que requerem uma verificação mais sofisticada, o uso de AM é mais adequado.

Em sistemas heterogêneos de telefonia, operando com SIP e H.323, por exemplo, pode se ter a necessidade de um AM específico para cada ambiente. Nestes casos, é necessário que o CM seja capaz de identificar qual AM deve ser acionado. O campo *protocol* do VQCDR pode auxiliar nesta identificação.

Storage Module (SM): responsável por armazenar os VQCDRs coletados pelo CM em uma base de armazenamento. Esta base pode ser, por exemplo, um banco de dados SQL ou um servidor RADIUS.

Para o escopo do serviço *fone@RNP*, foi implementado um VQCDR Server com um AM específico para o *gatekeeper* GnuGK e um SM baseado em RADIUS.

3.2. GnuGK Authenticator Module

Este AM utiliza a porta de gerência remota do GnuGK [12] (ver Figura 3) para validação do emissor do VQCDR. A porta de gerência remota do GnuGK oferece um canal de comunicação que permite verificar os usuários registrados (ativos) no sistema e uma série de informações referentes a cada usuário, como: identificador do usuário, endereço IP de registro (endereço IP público originador do registro do usuário) e o endereço IP privado (presente caso o usuário esteja atrás de um NAT Box [15]).

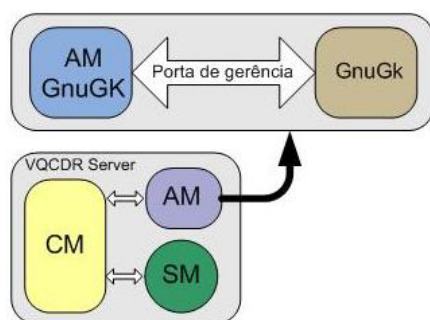


Figura 3: GnuGK Authenticator Module

Para o procedimento de autenticação, o AM confere os campos *username* (identificação do usuário) e *destinationMedia* (endereço IP do cliente - corresponde ao endereço IP privado, caso o usuário esteja atrás de um NAT Box) do VQCDR e o endereço IP do cabeçalho do pacote IP do VQCDR (endereço de registro). Caso o emissor do VQCDR esteja atrás de um NAT Box, o VQCDR será validado somente se os dados de identificação de usuário, endereço IP privado e endereço IP de registro casarem com um registro do GnuGK. Caso o emissor não esteja atrás de um NAT Box, basta os dados de identificação de usuário e endereço IP de registro coincidirem com um registro do GnuGK, para que o VQCDR seja validado.

No serviço fone@RNP está sendo recomendado que o VQCDR server rode na mesma máquina que o GnuGK, otimizando a interação do módulo de autenticação.

3.3. RADIUS Storage Module

O mecanismo de contabilidade do RADIUS oferece um conjunto padrão de atributos, que não é suficiente para detalhar a chamada. Entretanto, o RADIUS permite o envio de atributos extras, específicos de aplicação, chamados de VSAs (*Vendor Specific Attributes*). Para o uso de atributos específicos, um número privativo de fabricante (*Private Enterprise Number*) [14] deve ser solicitado à IANA [27]. De posse desse número, o fabricante/aplicação pode definir um dicionário de dados que permite aos servidores RADIUS a interpretação destes atributos específicos.

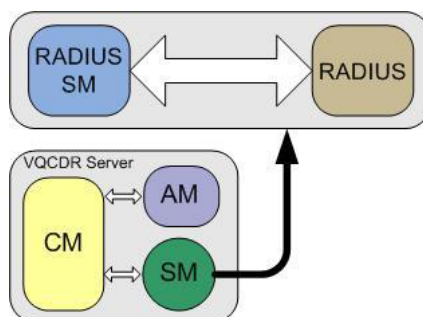


Figura 4: RADIUS Storage Module

Para mapear os campos do VQCDR, foi solicitado a IANA [27] um número privativo de fabricante e criado o nosso próprio dicionário de dados (Figura 5).

A interface do SM com o RADIUS (ver Figura 4) foi implementada com o auxílio da biblioteca RadiusClient [13], que oferece os mecanismos necessários para comunicação entre cliente e servidor RADIUS. Um ponto de limitação desta biblioteca

é a restrição de compatibilidade para com sistemas operacionais baseados em Unix. Este fato tem nos motivado sua reimplementação. Entretanto, no presente, o RADIUS SM encontra-se disponível apenas para sistemas Unix.

# UFRJ Vendor Specific Attributes				ATTRIBUTE	discardRate	15	integer	ufrj
VENDOR ufrj 21715				ATTRIBUTE	lossRate	16	integer	ufrj
# VQCDR attributes				ATTRIBUTE	duration	18	integer	ufrj
ATTRIBUTE protocol	1	string	ufrj	ATTRIBUTE	avgNetDelay	19	integer	ufrj
ATTRIBUTE callStart	2	string	ufrj	ATTRIBUTE	avgJitterBufferDelay	20	integer	ufrj
ATTRIBUTE callStop	3	string	ufrj	ATTRIBUTE	avgJitter	21	integer	ufrj
ATTRIBUTE ID	4	string	ufrj	ATTRIBUTE	codecDelay	22	integer	ufrj
ATTRIBUTE username	5	string	ufrj	ATTRIBUTE	packetsReceived	23	integer	ufrj
ATTRIBUTE model	6	string	ufrj	ATTRIBUTE	packetsLost	24	integer	ufrj
ATTRIBUTE codec	7	string	ufrj	ATTRIBUTE	packetsDiscarded	25	integer	ufrj
ATTRIBUTE frameSize	8	integer	ufrj	ATTRIBUTE	mediaSource	26	string	ufrj
ATTRIBUTE framesPerPacket	9	integer	ufrj	ATTRIBUTE	mediaDestination	27	string	ufrj
ATTRIBUTE IdEndOfCall	10	integer	ufrj	ATTRIBUTE	localAlias	28	string	ufrj
ATTRIBUTE IeEndOfCall	11	integer	ufrj	ATTRIBUTE	remoteAlias	29	string	ufrj
ATTRIBUTE MOSEndOfCall	12	integer	ufrj	ATTRIBUTE	direction	30	string	ufrj
ATTRIBUTE gapLossDensity	13	integer	ufrj	ATTRIBUTE	RFactorEndOfCall	31	integer	ufrj
ATTRIBUTE burstLossDensity	14	integer	ufrj					
								1/2
								2/2

Figura 5: Dicionário de dados RADIUS para o VQCDR

3.4. Clientes H.323 Modificados

Foram desenvolvidos clientes com capacidade de avaliar a qualidade de voz recebida e gerar VQCDRs, a partir de clientes H.323 de código aberto usados no serviço fone@RNP. O OpenPhone, cliente gráfico para Windows, originou o VQOpenPhone; e o OhPhone, cliente textual multiplataforma (Windows e Unix), deu origem ao VQOhPhone. Ambos utilizam as bibliotecas OpenH323 modificada e VQuality.

A Figura 6 mostra um cliente VQOpenPhone ao término de uma chamada. Em seu visor é informado o MOS estimado para a voz recebida.

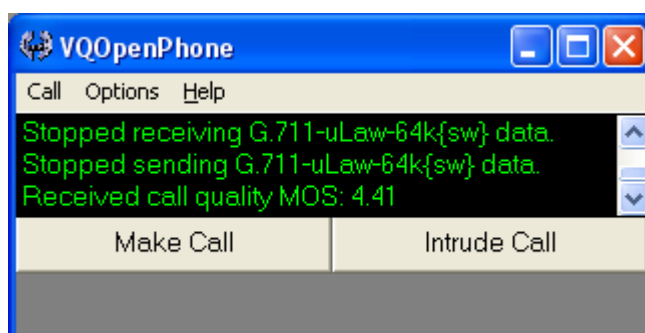


Figura 6: VQOpenphone

3.5. Voice Quality Log

Os indicadores de qualidade fornecidos pelo VQCDR oferecem um sumário da qualidade da chamada pela perspectiva do emissor do VQCDR, informação suficiente para um sistema VoIP operando com centenas ou milhares de chamadas diárias. Entretanto, não permitem a análise da variação da qualidade em relação ao tempo, que pode ser imprescindível para medições ou testes mais minuciosos e específicos.

A necessidade de efetuar medições mais detalhadas, juntamente com a motivação para o compartilhamento de relatórios de avaliação de qualidade entre aplicações baseadas na biblioteca VQuality, resultou na definição de um formato VQLog (*Voice Quality Log*), que além dos indicadores de qualidade do VQCDR, também possibilita o envio de outras informações. Para permitir uma análise temporal da chamada, são enviados adicionalmente os valores das principais variáveis de qualidade ao longo do tempo.

Um arquivo no formato VQLog é completamente codificado em ASCII e deve conter uma ou mais *Linhas de Identificação de Codec*, uma ou mais *Linhas de Indicadores de Qualidade*, uma *Linha de Sumário* e uma *Linha de Identificação da Chamada*, obrigatoriamente nesta ordem. Cada linha é composta por uma série de campos delimitados por um caractere de espaço. Durante uma chamada, mais de um codec pode ser usado, necessitando de uma linha para cada. Cada linha de indicadores de qualidade está associada a um campo instante de tempo, de forma que temos a necessidade de enviar várias linhas para o acompanhamento da evolução temporal da qualidade. Um exemplo de um arquivo VQLog pode ser visualizado na Figura 7.

```
! Exemplo de arquivo VQLog
G.711-uLaw-64k 1 20.00 40.00
13:41:00.625 6 0 0.00 98.00 13 116 0.15 22.76 70.50 3.62
13:41:01.547 3 0 0.00 98.00 18 116 0.15 23.62 69.60 3.58
13:41:02.469 3 0 0.00 97.50 15 115 0.15 20.46 72.80 3.73
13:41:03.375 5 0 0.00 97.50 18 115 0.15 20.37 72.80 3.73
13:41:04.391 0 0 0.00 97.50 14 115 0.15 20.37 72.80 3.73
13:41:05.391 0 0 0.00 97.00 15 114 0.15 20.37 72.80 3.73
13:41:06.375 0 0 0.00 97.00 18 114 0.15 20.37 72.80 3.73
13:41:07.047 0 0 0.00 97.00 13 114 0.15 20.37 72.80 3.73
2 0.00 57.50 14.00 40.00 367 17 0 0.00 11.26 0.15 20.42 72.80 3.73
0 D41F4102AAF118109FED0040A70 leandro 025983377 3354 1 146.164.247.196 146.164.10.10. 8
```

Figura 7: Exemplo de arquivo VQLog

3.6. Voice Quality Plot

O VQPlot é um aplicativo que lê e apresenta graficamente as informações de um arquivo VQLog. Junto a um cliente integrado à biblioteca VQuality, como o VQOpenPhone ou o VQOhPhone, constitui uma sofisticada ferramenta de análise. Após o término de uma chamada, o usuário pode acionar o VQPlot. Este abrirá automaticamente o VQLog e permitirá a análise temporal da qualidade da chamada, possibilitando a identificação de quais fatores degradaram a qualidade da chamada e com que grau. Esta análise pode auxiliar, por exemplo, que um técnico ajude um usuário remoto a identificar problemas de configuração em sua rede ou computador pessoal.

O VQplot, apresenta todos os campos da *Linha de Identificação de Codec* e da *Linha de Sumário*. Além disso, apresenta os gráficos referentes a todos os campos da *Linha de Indicadores de Qualidade*, que também podem ser visualizados em *zoom* para facilitar a análise.

As Figuras 8 e 9 ilustram a saída do VQPlot referente a uma chamada realizada por um *notebook* com interface sem fio (*wireless*).

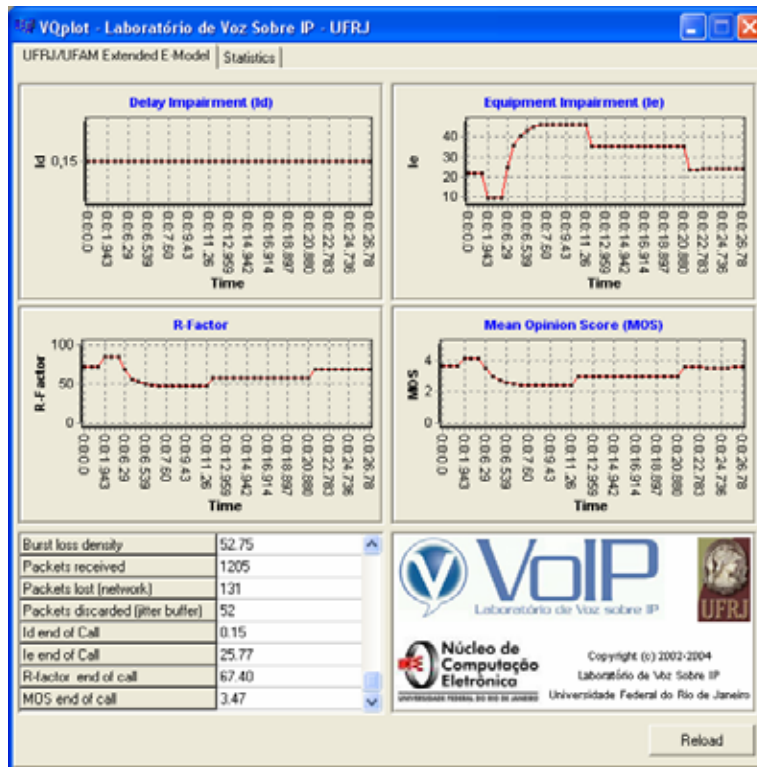


Figura 8: Qualidade da chamada em relação ao tempo

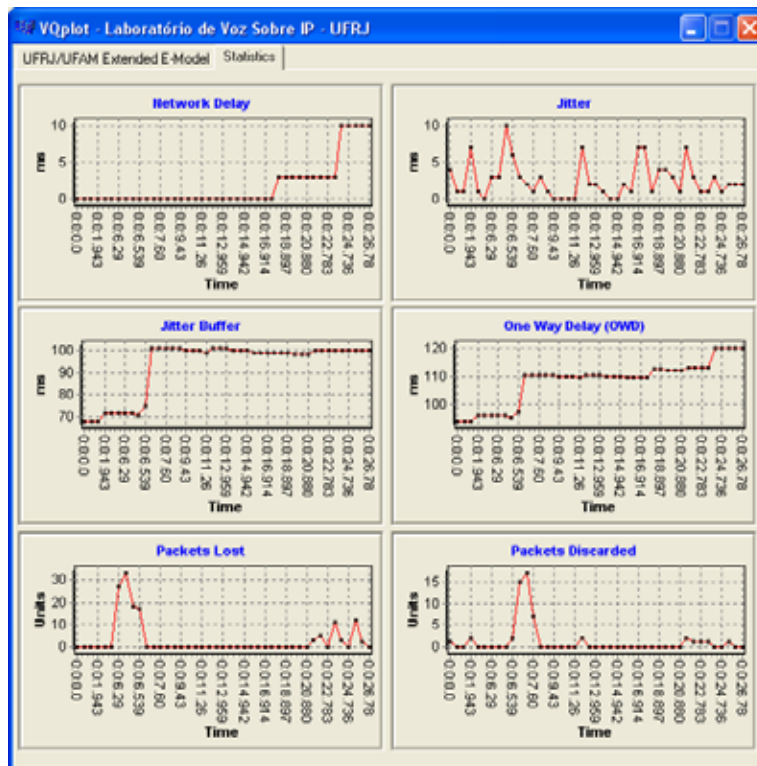


Figura 9: Estatísticas da chamada em relação ao tempo

3.7. Ambiente de Visualização

Todos os CDRs gerados pelo sistema, sejam eles provenientes do *gateway* de voz ou de clientes IP (VQCDR), são repassados aos servidores RADIUS, que por sua vez, armazena-os em um banco de dados SQL. Posteriormente, os CDRs são processados e consolidados, e, por meio de uma interface WEB desenvolvida em PHP [26], são disponibilizados relatórios com informações estatísticas e gráficos tais como: número de chamadas realizadas por período, intensidade de chamadas, nível de qualidade das chamadas, entre outros (Figura 10).

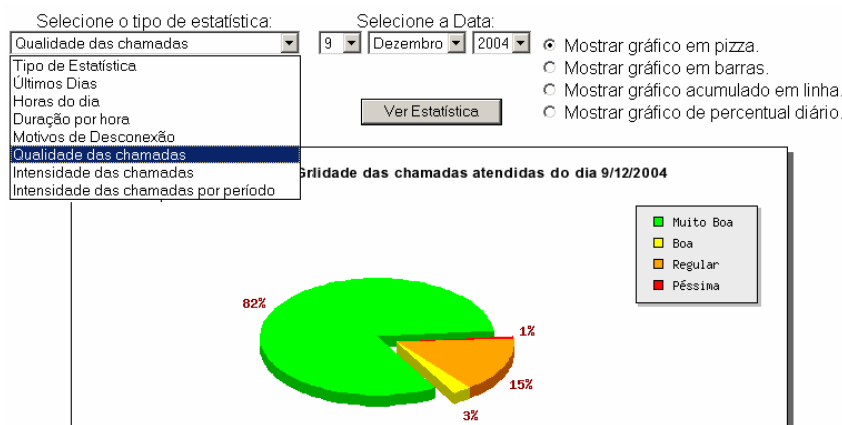


Figura 10: Ambiente Web de visualização de estatísticas

4. Conclusões e Trabalhos Futuros

Neste artigo apresentamos uma arquitetura para monitoração da qualidade de chamadas de voz sobre IP, baseada no desenvolvimento de uma biblioteca chamada VQuality, que implementa a medição objetiva calculada pelo modelo E e por suas extensões. Esta biblioteca, escrita em C/C++ padrão, incorpora funcionalidades para portabilidade e suporte a diferentes sinalizações de VoIP. A biblioteca VQuality incorpora ainda a geração de CDR próprio e um mecanismo para a coleta do CDR via RADIUS e extensão para envio de arquivos de informações complementares.

Com base no uso desta biblioteca e no uso da biblioteca OpenH323 modificada, foram desenvolvidos os clientes VQOpenphone e VQOphone, com a capacidade de gerar CDRs ao final das chamadas. O uso destes novos clientes na arquitetura desenvolvida irá permitir que a qualidade das chamadas geradas entre estes clientes H.323 seja contabilizada nos relatórios de produção do serviço fone@RNP.

Como a VQuality foi desenvolvida pensando em diferentes pilhas de sinalização VoIP, nosso passo seguinte será a incorporação desta biblioteca em clientes SIP de código aberto e em telefones IP de fabricantes abertos a parceria tecnológica.

Outra possível linha de ação é a integração da VQuality com ferramentas de simulação de rede, como o Network Simulator [28], que poderia propiciar um ambiente para avaliação de sistemas complexos de implementação de voz sobre IP.

Com as modificações implantadas na biblioteca OpenH323, principalmente no preenchimento correto dos campos dos relatórios RTCP, e com o mecanismo de coleta detalhada de informações de qualidade, é possível determinarmos os atrasos em cada

direção de uma chamada VoIP. A partir da base de dados das chamadas no serviço `fone@RNP` poderemos, como trabalho complementar futuro, gerar uma análise da assimetria de tráfego e atraso na RNP e suas redes coligadas.

Com o crescente interesse no uso de VoIP e a ausência no mercado de clientes com a capacidade de gerar medições de qualidade, um leque grande de oportunidades de parcerias com fabricantes e operadoras é aberto.

5. Referências

- [1] Marcondes, C.A.C., Rodrigues, P.H.A., David, F., Costa, J.C.P. *Ambiente para Simulação e Monitoração de Ligações Telefônicas IP*, in: Anais do XXI Simpósio Brasileiro de Redes de Computadores. Natal, maio 2003.
- [2] Lustosa, L.C.G., Carvalho, Rodrigues, P.H.A., Mota, S. E., *Utilização do Modelo E para avaliação da qualidade da fala em sistemas de comunicação baseados em voz sobre IP*, in: Anais do XXII Simpósio Brasileiro de Redes de Computadores. Gramado, maio 2004.
- [3] ITU-T Recommendation G.107. *The E-Model, a computational model for use in transmission planning*. Genève, mar. 2003.
- [4] ETSI TS 101 329-5 v1.1.2. *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON systems; Part 5: Quality of Service (QoS) measurement methodologies*. França, jan. 2002.
- [5] ETSI TS 102 024-5 v4.1.1. *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; End-to-end Quality of Service in TIPHON systems; Part 5: Quality of Service (QoS) measurement methodologies*. França, set. 2003.
- [6] The Perl Foundation. *The Perl Directory at Perl.org*. Disponível em <http://www.perl.org/>. Acesso em dez. 2004.
- [7] OpenH323 Project. Disponível em <http://www.openh323.org/>. Acesso em dez. 2004.
- [8] Schulzrinne, H. Casnet, S., Frederick, R., Jacobson, V. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550, jul. 2003.
- [9] PThreads-Win32. *Open Source POSIX Threads for Win32*. Disponível em <http://sources.redhat.com/pthreads-win32/>. Acesso em dez. 2004.
- [10] The Open Group. *Single UNIX Specification Version 3*, IEEE Std 1003.1, 2004 Edition. 2004.
- [11] Rigney, C., Willens, S., Rubens, A., Simpson, W. *Remote Authentication Dial In User Service (RADIUS)*. RFC 2865, jun. 2000.
- [12] OpenH323 Gatekeeper. *The GNU Gatekeeper*. Disponível em <http://www.gnugk.org/>. Acesso em dez. 2004.
- [13] RadiusClient. Disponível em <http://freshmeat.net/projects/radiusclient/>. Acesso em dez. 2004.
- [14] Reynolds, J. e Postel, J. *Assigned Numbers*, RFC 1700, out. 1994.

- [15] Srisuresh, P., Egevang, K. *Traditional IP Network Address Translator (Traditional NAT)*, RFC 3022, jan. 2001.
- [16] Cisco Systems, *RADIUS VSA Voice Implementation Guide*. Disponível em http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.pdf. Acesso em dez. 2004.
- [17] Cisco Systems, *Managing Voice Quality with Cisco Voice Manager (CVM) and Telemate*. Disponível em: <http://www.cisco.com/warp/public/788/AVVID/cvmtelemate.pdf>. Acesso em dez. 2004.
- [18] ITU-T Recommendation H.323. *Packet-Based Multimedia Communications Systems*. Genève, nov 2000.
- [19] Ubik, S. *IP Telephony Accounting and WAN Deployment Experience*. EUA, abr. 2001.
- [20] Lin, M., LO, C., e W., S. *Design and Implementation of an Enterprise Call Analysis System for VoIP Deployments*. 2003 Australian Telecommunications, Networks and Applications Conference (ATNAC), dez. 2003.
- [21] Huang, C., Chao, C., e Liu, A. *A Distributed Management Framework for H.323-Based VoIP System*. Communications in Computing (CIC) 2003. EUA, jun. 2003.
- [22] Broom, S., Hollier, M. *Speech Quality Measurement Tools for Dynamic Network Management*. Measurement of Speech and Audio Quality in Networks (MESAQIN) 2003, maio 2003.
- [23] Telchemy, SQmon - *Service Quality Monitoring for Voice over IP*. Disponível em: <http://www.telchemy.com/sqmon.html>. Acesso dez. 2004.
- [24] Empirix, Hammer XMS. Disponível em:
<http://www.empirix.com/Empirix/Network+IP+Storage+Test/hammer+xms.html>.
Acesso em dez. 2004.
- [25] Brix Networks, *Advanced VoIP Test Suites*. Disponível em: http://www.brixnet.com/products/voip_testsuite.html. Acesso dez. 2004.
- [26] The PHP Group, PHP - Hypertext Preprocessor. Disponível em: <http://www.php.net>. Acesso dez. 2004.
- [27] IANA. *Internet Assigned Numbers Authority*. Disponível em: <http://www.iana.org/>
Acesso em dez. 2004.
- [28] The University Of Southern California. *Network Simulator – ns2*. Disponível em: <http://www.isi.edu/nsnam/ns/> Acesso em dez. 2004.