

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

LEANDRO CAETANO GONÇALVES LUSTOSA

Arquitetura de Monitoração de Qualidade de Chamadas Telefônicas IP

RIO DE JANEIRO

2005

LEANDRO CAETANO GONÇALVES LUSTOSA

**ARQUITETURA DE MONITORAÇÃO DE QUALIDADE DE
CHAMADAS TELEFÔNICAS IP**

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DO INSTITUTO DE MATEMÁTICA / NÚCLEO DE COMPUTAÇÃO ELETRÔNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO – UFRJ, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO TÍTULO DE MESTRE EM CIÊNCIAS EM INFORMÁTICA.

Orientador: Paulo Henrique de Aguiar Rodrigues

RIO DE JANEIRO

2005

FICHA CATALOGRÁFICA

L974 Lustosa, Leandro Caetano Gonçalves.

Arquitetura de monitoração de qualidade de chamadas telefônicas IP / Leandro Caetano Gonçalves Lustosa. – Rio de Janeiro, 2005.
104 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, 2005.

Orientador: Paulo Henrique de Aguiar Rodrigues

1. Redes – Teses. 2. VoIP – Teses. 3. Monitoração – Teses. 4. Qualidade – Teses. I. Paulo Henrique de Aguiar Rodrigues (Orient.). II. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. III. Título.

LEANDRO CAETANO GONÇALVES LUSTOSA

**ARQUITETURA DE MONITORAÇÃO DE QUALIDADE DE CHAMADAS
TELEFÔNICAS IP**

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM
INFORMÁTICA DO INSTITUTO DE MATEMÁTICA / NÚCLEO DE COMPUTAÇÃO
ELETRÔNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO – UFRJ, COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO TÍTULO DE MESTRE EM
CIÊNCIAS EM INFORMÁTICA.

Aprovada em

Prof. Paulo Henrique de Aguiar Rodrigues, Ph.D.
(Universidade Federal do Rio de Janeiro)

Prof^a. Luci Pirmez, D.Sc.
(Universidade Federal do Rio de Janeiro)

Prof. Michael Anthony Stanton, Ph.D.
(Universidade Federal Fluminense)

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.
(Universidade Federal do Rio de Janeiro)

RIO DE JANEIRO

2005

A Deus, pela benção da vida, saúde e
oportunidades.

Aos meus pais, Elmo e Maria Aparecida, pela
educação, apoio, amor, estímulo e entusiasmo
em todas as fases da minha vida.

AGRADECIMENTOS

À minha carinhosa irmã Lísia e ao meu cunhado Rodrigo, pelo grande incentivo no decorrer do curso.

À minha namorada Carla, pelo carinho, amizade e compreensão.

À Margarida, pela amizade e presença em momentos significativos da minha vida.

Aos bons e velhos amigos de Cachoeiro de Itapemirim, Edter, Fernando, Glauber, Michel, Renato e Ana Luiza, e aos outros grandes amigos de Petrópolis, Roberto, Fabrício, Fábio, Rômulo e Evandro, pela amizade e bons momentos de descontração.

Ao Fabio e ao Douglas pela enorme contribuição no desenvolvimento deste trabalho, e a todos os outros colegas e ex-colegas do Laboratório de Voz Sobre IP, que também contribuíram direta e indiretamente neste desenvolvimento, Peixoto, Marcelo, Rodrigo, Anderson, César, Vitor e Maurício.

Aos colegas do curso de Pós-graduação do Instituto de Matemática / Núcleo de Computação Eletrônica (IM/NCE), Cristiano, Renata, Gilberto e Austeclynio, pelos enriquecedores e agradáveis momentos de estudo.

Ao Leandro da Universidade Federal do Amazonas (UFAM), pelas incontáveis horas de discussão relativas à VoIP.

Ao NCE/UFRJ pela excelente infra-estrutura e ambiente de trabalho.

À Tia Deise e aos outros funcionários da Secretaria de Informática e da Secretaria da Área de Ensino e Pesquisa, pela presteza e ajuda nos assuntos burocráticos.

À Rede Nacional de Ensino e Pesquisa (RNP), pelo suporte oferecido ao Laboratório de Voz Sobre IP.

Aos professores do programa de Pós-graduação do IM/NCE, pelo ensino de excelência oferecido.

Ao Laboratório Nacional de Computação Científica (LNCC), pelo apoio e disponibilidade dispensados na fase inicial do curso.

À Professora Luci Pirmez, pelo apoio, incentivo e valiosos ensinamentos.

Ao meu orientador Professor Paulo Aguiar, pelos ensinamentos precisos, oportunidade e confiança depositada.

RESUMO

ARQUITETURA DE MONITORAÇÃO DE QUALIDADE DE CHAMADAS TELEFÔNICAS IP

Leandro Caetano Gonçalves Lustosa

Orientador: Paulo Henrique de Aguiar Rodrigues

Resumo da Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática, Instituto de Matemática e Núcleo de Computação Eletrônica, da Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Uma nova arquitetura de monitoração de qualidade de chamadas para sistemas de comunicação baseados em Voz Sobre IP (VoIP) é proposta e implementada.

Esta arquitetura define um novo registro de detalhamento de chamada (CDR), denominado Voice Quality CDR (VQCDR), que é coletado diretamente de telefones IP e gateways de voz. O VQCDR faz uso do Modelo E da ITU (*International Telecommunications Union*) para oferecer indicadores detalhados para a monitoração da qualidade de chamadas VoIP, permitindo ainda o envio opcional de um histórico, que possibilita a análise da variação da qualidade ao longo da chamada, para uso em medições mais minuciosas.

Uma nova biblioteca multiplataforma, denominada Voice Quality (VQuality), capaz de emitir o VQCDR, foi produzida. A partir desta implementação, foram desenvolvidos dois softwares de telefone IP, o VQOpenPhone para Windows e o VQMeeting para Unix, que incorporam esta biblioteca para avaliação de qualidade das chamadas. Também foi implementado o VQCDR Server, responsável pela coleta, autenticação e encaminhamento dos VQCDRs para armazenamento.

Este trabalho apresenta ainda um estudo sobre o Modelo E e a extensão a esse modelo proposta pelo ETSI (*European Telecommunications Standards Institute*). Críticas são realizadas e uma proposta de revisão à forma de cálculo do modelo estendido é formulada.

ABSTRACT

VOICE QUALITY MONITORING ARCHITECTURE OF IP TELEPHONIC CALLS

Leandro Caetano Gonçalves Lustosa

Orientador: Paulo Henrique de Aguiar Rodrigues

Abstract da Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática, Instituto de Matemática e Núcleo de Computação Eletrônica, da Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

This thesis proposes a new call quality monitoring architecture for Voice Over IP (VoIP) communication systems.

This architecture defines a new call detail record (CDR), called Voice Quality CDR (VQCDR), which is collected directly from IP telephones and voice gateways. VQCDR uses the ITU (*International Telecommunications Union*) E-Model to offer detailed parameters for quality call monitoring. VQCDR also allows the transfer of the call history, which enables the analysis of the variation of quality during the call.

A new multiplatform library that supports VQCDR generation, called Voice Quality (VQuality) and two IP telephones incorporating this library, the VQOpenPhone for Windows and the VQMeeting for Unix, have been developed. The server responsible for collection, authentication and VQCDR storage, called VQCDR Server, has also been implemented.

This work also analyses the E-Model and its extension as proposed by ETSI (*European Telecommunications Standards Institute*) and revised formulas for calculating the extended model are presented.

LISTA DE ILUSTRAÇÕES

ILUSTRAÇÃO 1. ARQUITETURA DO SERVIÇO FONE@RNP	18
ILUSTRAÇÃO 2. REPRESENTAÇÃO DE ATRASOS EM UM SISTEMA DE COMUNICAÇÃO DE VOZ	28
ILUSTRAÇÃO 3. COMPARAÇÃO ENTRE A APROXIMAÇÃO PARAMÉTRICA SUGERIDA POR COLE E ROSENBLUTH E A CURVA DE <i>Id</i> RECOMENDADA PELA ITU-T G.107.....	31
ILUSTRAÇÃO 4. COMPORTAMENTO ALTERNANTE DE PERDAS MODELADO COM USO DE CADEIA DE MARKOV DE 4 ESTADOS	37
ILUSTRAÇÃO 5. REPRESENTAÇÃO DO EFEITO DA MEMÓRIA RECENTE	39
ILUSTRAÇÃO 6. INFRA-ESTRUTURA DE GERAÇÃO, COLETA E MONITORAMENTO DA QUALIDADE DE LIGAÇÕES VOIP	42
ILUSTRAÇÃO 7. PERDA DE PACOTES NA REDE	43
ILUSTRAÇÃO 8. DESCARTE DE PACOTES NO <i>BUFFER</i> DE COMPENSAÇÃO DE <i>JITTER</i>	43
ILUSTRAÇÃO 9. <i>Ie</i> OBTIDO SEGUNDO A REVISÃO SUGERIDA	44
ILUSTRAÇÃO 10. <i>Ie</i> OBTIDO SEGUNDO A ABORDAGEM DE CLARK	44
ILUSTRAÇÃO 11. <i>Id</i> (ITU-T G.107)	44
ILUSTRAÇÃO 12. MOS (ITU-T G.107)	44
ILUSTRAÇÃO 13. VARIAÇÃO DO MOS AO LONGO DA LIGAÇÃO OBTIDA SEGUNDO A REVISÃO SUGERIDA	45
ILUSTRAÇÃO 14. VARIAÇÃO DO MOS AO LONGO DA LIGAÇÃO OBTIDA SEGUNDO A ABORDAGEM DE CLARK	45
ILUSTRAÇÃO 15. H.323 BEACON	47
ILUSTRAÇÃO 16. ARQUITETURA DA BIBLIOTECA VQUALITY	51
ILUSTRAÇÃO 17. HIERARQUIA DE CLASSES PARA IMPLEMENTAÇÃO DE UM PM PARA O OPENH323	53
ILUSTRAÇÃO 18. HIERARQUIA DE CLASSES PARA INCLUSÃO DE VQEMs	53
ILUSTRAÇÃO 19. CÁLCULO DE RTT	55
ILUSTRAÇÃO 20. 32 BITS CENTRAIS DO LSR	56
ILUSTRAÇÃO 21. ALTERAÇÃO DO CÓDIGO DA OPENH323 NO INSTANTE T2	57
ILUSTRAÇÃO 22. ALTERAÇÃO DO CÓDIGO DA OPENH323 NO INSTANTE T3	57
ILUSTRAÇÃO 23. CONVERSÃO DE DLSR DE 1/65535 SEGUNDOS PARA MILISSEGUNDOS	58
ILUSTRAÇÃO 24. RECUPERAÇÃO DO TIMESTAMP DO INSTANTE T1	58
ILUSTRAÇÃO 25. CÁLCULO DO RTT	59
ILUSTRAÇÃO 26. COLETANDO PARÂMETROS DA OPENH323	60
ILUSTRAÇÃO 27. COLETANDO INFORMAÇÕES REFERENTES AO CODIFICADOR DE VOZ	60
ILUSTRAÇÃO 28. INTERAÇÃO ENTRE AS CLASSES CALLQUALITYSUMMARY, EXTENDED E MODELBASE E PARSEBASE	61
ILUSTRAÇÃO 29. INTEGRAÇÃO DO OPENPHONE COM A VQUALITY	61
ILUSTRAÇÃO 30. VQOPENPHONE	62
ILUSTRAÇÃO 31. VQMEETING	63
ILUSTRAÇÃO 32. EXEMPLO DE ARQUIVO VQLOG	68
ILUSTRAÇÃO 33. QUALIDADE DA CHAMADA EM RELAÇÃO AO TEMPO	69
ILUSTRAÇÃO 34. ESTATÍSTICAS DA CHAMADA EM RELAÇÃO AO TEMPO	70
ILUSTRAÇÃO 35. ARQUITETURA DE MONITORAÇÃO DE CHAMADAS TELEFÔNICAS IP	73
ILUSTRAÇÃO 36. ARQUITETURA DO VQCDR SERVER.....	81
ILUSTRAÇÃO 37. GNUMK AUTHENTICATOR MODULE	83
ILUSTRAÇÃO 38. DICIONÁRIO DE DADOS RADIUS PARA O VQCDR.....	84
ILUSTRAÇÃO 39. RADIUS STORAGE MODULE	85

ILUSTRAÇÃO 40. AMBIENTE WEB DE VISUALIZAÇÃO DE ESTATÍSTICAS.....	86
ILUSTRAÇÃO 41. FUNCIONAMENTO DA ARQUITETURA DE MONITORAÇÃO DE QUALIDADE	87
ILUSTRAÇÃO 42. ASTERISK COMO <i>GATEWAY</i> VOIP/RTFC E <i>COMO GATEWAY</i> SIP/H.323	90
ILUSTRAÇÃO 43. TRECHOS DE UMA CHAMADA ENTRE SIP E H.323 VIA ASTERISK	91
ILUSTRAÇÃO 44. REPRESENTAÇÃO VISUAL DA EQUAÇÃO F_1	101
ILUSTRAÇÃO 45. REPRESENTAÇÃO VISUAL DA EQUAÇÃO F_2	101
ILUSTRAÇÃO 46. REPRESENTAÇÃO VISUAL DA EQUAÇÃO I_2	102
ILUSTRAÇÃO 47. REPRESENTAÇÃO VISUAL DA EQUAÇÃO I_1	102

LISTA DE QUADROS

QUADRO 1. CATEGORIAS DE TRANSMISSÃO DA FALA	26
QUADRO 2. VALORES RECOMENDADOS PARA O CÁLCULO DE <i>ID</i>	30
QUADRO 3. LINHA DE IDENTIFICAÇÃO DE CODEC	64
QUADRO 4. LINHA DE INDICADORES DE QUALIDADE	65
QUADRO 5. LINHA DE SUMÁRIO	66
QUADRO 6. LINHA DE IDENTIFICAÇÃO DE CHAMADA.....	66
QUADRO 7. CAMPOS DO VQCDR.....	76
QUADRO 8. PROTOCOLOS DE SINALIZAÇÃO	79
QUADRO 9: CODIFICADORES DE VOZ	79
QUADRO 10: MODELOS OBJETIVOS DE AVALIAÇÃO.....	79
QUADRO 11: IDENTIFICAÇÃO DE ORIGINADOR.....	79

LISTAS DE ABREVIATURAS E SIGLAS

A	<i>Advanced Factor</i>
AM	<i>Authenticator Module</i>
API	<i>Application Programming Interface</i>
ASCII	<i>American Standard Code for Information Interchange</i>
CDR	<i>Call Detail Record</i>
CESNET	<i>Czech Republic National Research and Education Network</i>
CM	<i>Collector Module</i>
DLSR	<i>Delay Since Last Sender Report</i>
ECAS	<i>Enterprise Call Analysis System</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FEC	<i>Forward Error Correction</i>
GCC	<i>GNU Compiler Collection</i>
GNU	<i>GNU's not Unix</i>
GPS	<i>Global Position System</i>
GSM	<i>Global System for Mobile Communications</i>
GSM-EFR	<i>GSM – Enhanced Full Rate</i>
GSM-FR	<i>GSM – Full Rate</i>
HTTP	<i>HyperText Transfer Protocol</i>
IANA	<i>Internet Assigned Numbers Authority</i>
Id	<i>Delay Impairment</i>
Ie	<i>Equipment Impairment</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
Is	<i>Simultaneous Impairment</i>
ITEC	<i>Internet2 Technology Evaluation Center</i>
ITU-T	<i>International Telecommunications Union-Telecommunications Standards</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
LSR	<i>Last Sender Report</i>
MGCP	<i>Media Gateway Control Protocol</i>
MOBVEM	<i>Modified OpenH323 Based Voice Evaluation Module</i>
MOS	<i>Mean Opinion Score</i>
NAT	<i>Network Address Translation</i>
NTP	<i>Network Time Protocol</i>
OWD	<i>One-Way Delay</i>
PAMS	<i>Perceptual Analysis Measurement System</i>
PBX	<i>Private Branch Exchange</i>
PC	<i>Personal Computer</i>
PCM	<i>Pulse Code Modulation</i>
PESQ	<i>Perceptual Evaluation of Speech Quality</i>
PLC	<i>Packet Loss Concealment</i>
PM	<i>Parser Module</i>
PSQM	<i>Perceptual Speech Quality Measure</i>
QoS	<i>Quality of Service</i>
R	<i>R-Factor</i>
RADIUS	<i>Remote Authentication Dial In User Service</i>
RFC	<i>Request For Comments</i>

RLR	<i>Receive Loudness Rating</i>
RNP	<i>Rede Nacional de Ensino e Pesquisa</i>
Ro	<i>Basic Signal-to-Noise Ratio</i>
RR	<i>Receiver Report</i>
RTCP	<i>Real-Time Control Protocol</i>
RTFC	<i>Rede de Telefonia Fixa Comutada</i>
RTP	<i>Real-Time Protocol</i>
RTT	<i>Round-Trip Time</i>
SIP	<i>Session Initiation Protocol</i>
SM	<i>Storage Module</i>
SNMP	<i>Simple Network Management Protocol</i>
SNR	<i>Signal-to-Noise Ratio</i>
SQL	<i>Structured Query Language</i>
SR	<i>Sender Report</i>
TCP	<i>Transmission Control Protocol</i>
TDM	<i>Time-Division Multiplexing</i>
TELR	<i>Talker Echo Loudness Rating</i>
TIPHON	<i>Telecommunications and Internet Protocol Harmonization Over Networks</i>
UFAM	<i>Universidade Federal do Amazonas</i>
UFRJ	<i>Universidade Federal do Rio de Janeiro</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
VAD	<i>Voice Activity Detection</i>
VGM	<i>VQCDR Generation Module</i>
VoIP	<i>Voice Over IP</i>
VQCDR	<i>Voice Quality Call Detail Record</i>
VQEM	<i>Voice Quality Evaluation Module</i>
VQLog	<i>Voice Quality Log</i>
VQPlot	<i>Voice Quality Plot</i>
WELP	<i>Weighted Echo Path Loss</i>

SUMÁRIO

CAPÍTULO 1	15
INTRODUÇÃO	15
1.1 MÉTODOS PARA AVALIAÇÃO DE QUALIDADE DE VOZ	16
1.2 O SERVIÇO FONE@RNP	18
1.3 OBJETIVOS	20
1.4 OUTRAS SOLUÇÕES PARA MONITORAMENTO DE QUALIDADE DE CHAMADAS TELEFÔNCIAS IP	21
1.5 ORGANIZAÇÃO DO TRABALHO	23
CAPÍTULO 2	25
MODELO E PARA AVALIAÇÃO OBJETIVA DE QUALIDADE DE VOZ	25
2.1 DECOMPONDO O MODELO E	25
2.1.1 <i>Relação Sinal-ruído Básica, Ro</i>	27
2.1.2 <i>Fator de Perdas Simultâneas, Is</i>	27
2.1.3 <i>Fator de Perdas Associadas ao Atraso, Id</i>	27
2.1.4 <i>Fator de Perdas Associadas ao Equipamento, Ie</i>	28
2.1.5 <i>Fator de Vantagem, A</i>	29
2.2 DETERMINAÇÃO DOS FATORES DE PERDA <i>ID</i> E <i>IE</i>	29
2.2.1 <i>Determinação do Fator de Perda Id</i>	29
2.2.2 <i>Determinação do Atraso Absoluto, Ta</i>	32
2.2.3 <i>Determinação do Fator de Perda Ie</i>	33
2.3 MODELO E ESTENDIDO	35
2.3.1 <i>Modelagem de Comportamento Alternante de Perda de Pacotes e de Efeito de Memória Recente</i>	36
2.4 SUGESTÕES DE REVISÃO AO MODELO E ESTENDIDO	39
CAPÍTULO 3	41
MODIFIED BASED OPENH323 VOICE EVALUATION MODULE	41
3.1 AVALIAÇÃO DOS MODELOS	43
3.2 H.323 BEACON	46
CAPÍTULO 4	48
BIBLIOTECA VOICE QUALITY	48
4.1 FLEXIBILIDADE E EXTENSABILIDADE	49
4.2 PORTABILIDADE	49
4.3 EMISSÃO DE CDR	50
4.4 ARQUITETURA DA BIBLIOTECA VQUALITY	50
4.4.1 <i>Parser Module (PM)</i>	51
4.4.2 <i>Voice Quality Evaluation Module (VQEM)</i>	51
4.4.3 <i>VQCDR Generation Module (VGM)</i>	52
4.4.4 <i>Criando um Parser Module</i>	52
4.4.5 <i>Criando um Voice Quality Evaluation Module</i>	53
4.5 INTEGRAÇÃO DA VQUALITY COM A BIBLIOTECA OPENH323	54

4.5.1 Mecanismo de Cálculo de RTT do RTP	54
4.5.2 Alteração da Biblioteca OpenH323 para Cálculo do RTT	55
4.6 COLETANDO OS PARÂMETROS	59
4.7 INTEGRAÇÃO DA VQUALITY COM TELEFONES IP	60
4.7.1 VQOpenPhone	62
4.7.2 VQMeeting.....	63
4.8 VOICE QUALITY LOG	64
4.9 VOICE QUALITY PLOT	68
CAPÍTULO 5	73
ARQUITETURA DE MONITORAÇÃO DE CHAMADAS TELEFÔNICAS IP	73
5.1 TELEFONES IP E GATEWAYS DE VOZ COM CAPACIDADE DE EMISSÃO DE VQCDR	74
5.2 VOICE QUALITY CALL DETAIL RECORD (VQCDR).....	75
5.2.1 O Formato do VQCDR.....	76
5.3 VOICE QUALITY CDR SERVER	81
5.3.1 Collector Module (CM)	81
5.3.2 Authenticator Module (AM).....	82
5.3.3 Storage Module (SM).....	82
5.3.4 GnuGK Authenticator Module	83
5.3.5 RADIUS Storage Module	84
5.4 AMBIENTE DE VISUALIZAÇÃO	86
5.5 ARQUITETURA DE MONITORAÇÃO EM OPERAÇÃO	87
5.6 CLIENTES SIP COM SUPORTE A VQCDR	88
5.7 ASTERISK COM SUPORTE A EMISSÃO DE VQCDR	89
CAPÍTULO 6	92
CONCLUSÕES E TRABALHOS FUTUROS.....	92
APÊNDICE A	101
VALIDAÇÃO MATEMÁTICA DA EXPRESSÃO DE DETERMINAÇÃO I_1	
PROPOSTA	101

Capítulo 1

INTRODUÇÃO

Os sistemas de comunicação baseados em voz sobre IP (VoIP) se consolidaram com a padronização dos protocolos de sinalização e representam hoje uma solução tecnológica viável para a provisão de serviços de voz.

Todavia, o transporte de voz sobre o protocolo IP ainda continua sendo um desafio. O protocolo IP foi inicialmente concebido de forma a oferecer transporte baseado em comutação por pacotes e serviço de melhor esforço (*best effort*), sem estabelecer nenhuma garantia de entrega ou limite de atraso na entrega de pacotes. Em contrapartida, a Rede de Telefonia Fixa Comutada (RTFC), responsável pelo transporte de voz do sistema de telefonia convencional, opera sobre uma infra-estrutura TDM (*Time-Division Multiplexing*), que oferece um serviço de comutação de circuitos com latência limitada e índices de variação de atraso e de perdas extremamente baixos, totalmente adequados para o transporte de voz.

Para suportar a demanda de serviços de operação em tempo real e altamente sensíveis ao atraso, à variação de atraso (*jitter*) e à perda, como o transporte de voz, várias abordagens para prover Qualidade de Serviço (QoS) em redes IP vem sendo estudadas, tanto no meio acadêmico quanto na indústria. Dentre as diversas abordagens, destacam-se o IntServ (*Integrated Services*) [1] e o DiffServ (*Differentiated Services*) [2], que oferecem uma infra-estrutura para fornecer, além do tradicional serviço de melhor esforço, tratamento diferenciado e novos serviços aos fluxos associados às aplicações sensíveis a parâmetros de

qualidade de serviço. Embora não sejam aplicados amplamente na Internet pública, estas abordagens já se mostram presentes em redes corporativas e em *backbones* de operadoras de serviços de telecomunicações, oferecendo tratamento diferenciado aos fluxos prioritários, ao menos em suas redes privadas.

Com a popularização de equipamentos e soluções de comunicação baseados em VoIP, seja em redes corporativas ou na Internet pública, urge a necessidade de se mensurar a qualidade da fala fim-a-fim, seja para validar a infra-estrutura de rede e mecanismos de provisão de QoS implementados, ou mesmo para analisar e acompanhar o nível de satisfação dos usuários quanto ao serviço oferecido.

1.1 Métodos para Avaliação de Qualidade de Voz

Uma das primeiras abordagens para avaliação da qualidade da voz foi a utilização da Pontuação de Opinião Média, ou MOS (*Mean Opinion Score*), método subjetivo definido nas Recomendações ITU-T P.800 [3] e ITU-T P.830 [4], pelo qual um conjunto de avaliadores ouvintes atribuem uma pontuação de 1 (pobre) a 5 (excelente) à qualidade da fala reproduzida pelo sistema de comunicação em teste. Como expressa diretamente a opinião média dos usuários, o MOS representa um índice de referência para avaliação da qualidade da fala em sistemas de comunicação.

Contudo, como a obtenção da pontuação MOS é um procedimento de difícil implementação e inadequado de ser utilizado para medir a qualidade de conversas telefônicas, tem-se optado pelo desenvolvimento e implementação de métodos objetivos para estimar a qualidade de voz em sistemas de comunicação que utilizam a tecnologia de VoIP [5].

Os métodos objetivos utilizam recursos computacionais para inferir a qualidade da voz submetida a um sistema de transmissão ou a um codificador de áudio. Dentre os métodos

objetivos, destaca-se o Modelo E, originalmente proposto pelo ETSI (*European Telecommunications Standards Institute*), através do relatório técnico ETR250 [10] e posteriormente padronizado pelo ITU-T (*International Telecommunications Union-Telecommunications Standard Sector*), através da Recomendação G.107 [11].

Outros exemplos de métodos objetivos são o PSQM (*Perceptual Speech Quality Measure*) [6], o PAMS (*Perceptual Analysis Measurement System*) [7] e o PESQ (*Perceptual Evaluation of Speech Quality*) [8], conhecidos na literatura como métodos perceptuais. Estes métodos fazem uso do conhecimento do sistema auditivo humano para comparar um sinal de referência (trecho de voz previamente gravado) com um sinal degradado (sinal de referência submetido ao sistema de transmissão a ser avaliado), a fim de compor uma medida de distorção do sinal de voz [9].

As principais vantagens do Modelo E, em relação aos métodos perceptuais supracitados, são a capacidade de medição em tempo real, já que não há necessidade de comparação entre os sinais de referência e degradado, e a contabilização em separado de cada um dos fatores responsáveis pela degradação da qualidade da voz, como perdas, atraso fim-a-fim, distorções inerentes a codificadores de alta compressão, entre outros [11]. Estas funcionalidades permitem a avaliação da origem e do grau de influência de cada um destes fatores separadamente, tornando o diagnóstico de problemas de transmissão mais precisos.

Devido à facilidade de operação em tempo real e a capacidade de inferir a influência de fatores de degradação separadamente, o presente trabalho dá enfoque ao uso do Modelo E como base para a criação de uma ferramenta de avaliação de qualidade de voz. Detalhes do Modelo E serão apresentados e discutidos no Capítulo 2.

1.2 O Serviço fone@RNP

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, interconecta os sistemas privados de telefonia (PBX) de uma série de instituições acadêmicas nacionais e fornece uma plataforma para utilização de telefones IP (*hardwares* dedicados ou *softwares* rodando em PCs com capacidade multimídia), permitindo ainda o estabelecimento de chamadas com a RTFC, através dos PBXs das instituições participantes, e com redes de telefonia IP acadêmicas de outros países, via Internet2.

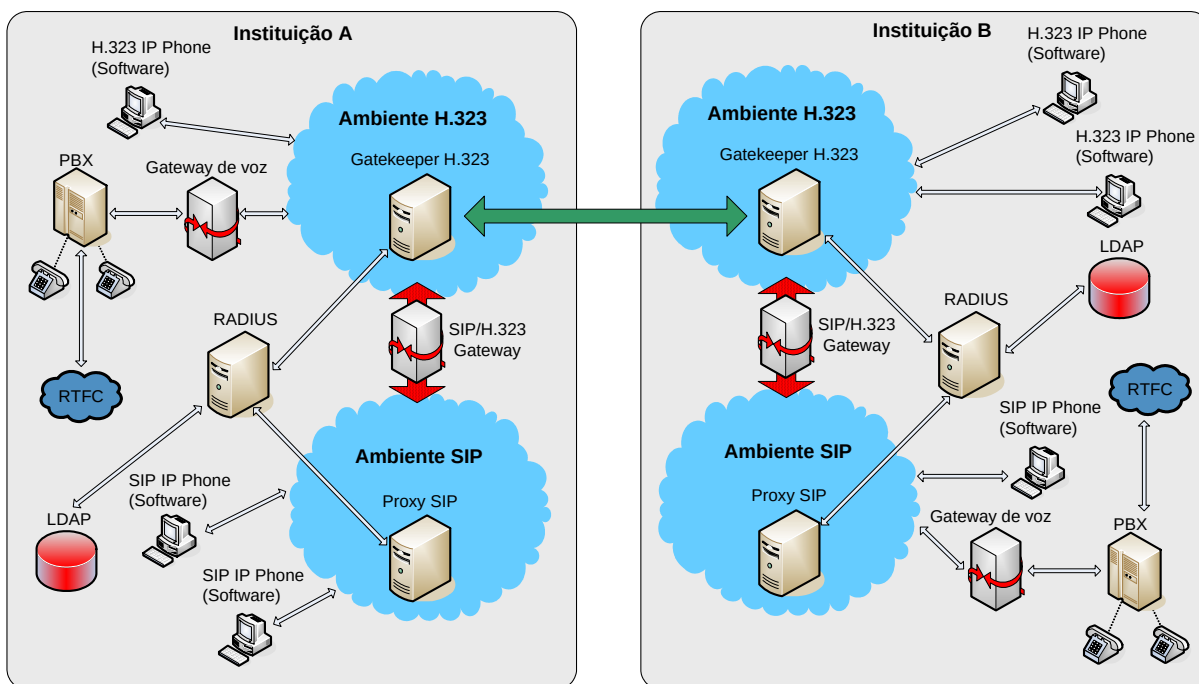


Ilustração 1. Arquitetura do serviço fone@RNP

O fone@RNP utiliza uma arquitetura heterogênea baseada nos protocolos de sinalização H.323 [12] e SIP [13]. Cada instituição usuária do serviço possui um ambiente H.323 e um ambiente SIP (como mostrado na Ilustração 1). Ambos os ambientes possuem um servidor para registro e localização de telefones IP, o *gatekeeper* GnuGK [14], para o

ambiente H.323, e o *Proxy* SER [15], para o ambiente SIP. A autenticação de terminais H.323 e de clientes SIP é realizada de forma unificada via um servidor de diretórios LDAP (*Lightweight Directory Access Protocol*) [16] e um servidor RADIUS [17] (com um banco de dados SQL associado), que também é responsável pela contabilização das chamadas e pela coleta de estatísticas de uso do serviço. Opcionalmente, cada instituição pode possuir um *gateway* de voz para a interconexão com o PBX. Este *gateway* pode ser, por exemplo, um equipamento Cisco [18] ou um *software* Asterisk [19] operando em um PC. A interoperação entre os ambientes H.323 e SIP é realizada com o uso do *software* Asterisk, que neste caso, opera como *gateway* de sinalização SIP/H.323.

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 (*Call Detail Records* ou simplesmente CDRs) emitidos pelos *gatekeepers* H.323, pelos *proxies* SIP e pelos *gateway* de voz. Entretanto, somente os CDRs gerados pelos *gateways* de voz Cisco [18] fornecem indicativos de qualidade que, embora sejam simplistas comparando-se aos fornecidos por modelos mais sofisticados como o Modelo E estendido [5,20,21,22], oferecem alguma estimativa da qualidade da chamada. Dessa forma, somente as chamadas que envolvem *gateways* Cisco têm sua qualidade mensurada, tornando as estatísticas pouco confiáveis para caracterizar o sistema como um todo, já que a qualidade das chamadas realizadas entre dois telefones IP, e entre um telefone IP e o PBX, via *gateway* Asterisk, não são contabilizadas.

1.3 Objetivos

Este trabalho apresenta uma implementação de uma nova arquitetura de monitoração de qualidade de chamadas telefônicas IP que torna a obtenção da qualidade das chamadas mais eficaz e abrangente, possibilitando inclusive, que as chamadas estabelecidas entre telefones IP também sejam avaliadas.

Esta arquitetura é implementada com base no desenvolvimento da biblioteca Voice Quality (VQuality) [23], que implementa uma nova especificação de CDR especializada para avaliação de qualidade de voz, denominada Voice Quality CDR (VQCDR), que possui indicadores extremamente detalhados e que mostram-se mais fiéis à percepção humana que os indicadores fornecidos pelo CDR dos *gateways* de voz Cisco [18,25], devido ao uso do Modelo E estendido [5,20,21,22].

Outra potencialidade da biblioteca VQuality é a sua arquitetura flexível e portátil. Considerando que telefones IP e *gateways* de voz operam em diferentes plataformas computacionais e com diferentes protocolos de sinalização VoIP, e também que novos modelos de avaliação de qualidade de voz venham a ser adotados, as premissas básicas para o desenvolvimento da biblioteca foram a capacidade de operação em variados sistemas operacionais, a independência de protocolo de sinalização VoIP e a facilidade para suporte a novos modelos de avaliação.

A disponibilização de telefones IP e *gateways* de voz com a biblioteca VQuality incorporada possibilitará uma eficiente monitoração de ambientes VoIP, auxiliando na detecção de problemas de transmissão de voz, na validação da infra-estrutura de rede e no acompanhamento do grau de satisfação dos usuários.

A arquitetura proposta vem sendo operacionalizada dentro do ambiente do serviço `fone@RNP` para permitir que, no futuro, a qualidade de todas as chamadas realizadas no ambiente sejam avaliadas, seja a chamada estabelecida entre dois telefones IP, entre um telefone IP e um *gateway* de voz (Cisco ou Asterisk), ou entre dois *gateways* de voz (Cisco ou Asterisk).

Em seu estado atual, o trabalho disponibiliza terminais H.323 para variados sistemas operacionais (FreeBSD, Linux e Windows) com suporte a emissão de VQCDR e um mecanismo para coleta e visualização das estatísticas de qualidade obtidas. Além disso, provê os recursos necessários para que, no futuro, outros telefones IP e *gateways* de voz, sejam estes baseados em SIP, ou qualquer outro protocolo de sinalização VoIP que venha a ser adotado no ambiente do serviço `fone@RNP`, forneçam indicadores de qualidade das chamadas via emissão de VQCDR.

Outro ponto coberto pela dissertação é o estudo detalhado do Modelo E e suas extensões [5,20,21,22]. Apesar de ter sido desenvolvido para o planejamento de sistemas de comunicação de voz [11], o modelo tem sido objeto de estudos para aplicá-lo como ferramenta de monitoração da qualidade da voz no cotidiano de redes VoIP [5,26,27]. Neste contexto, o Modelo E é decomposto em detalhes e críticas, bem como sugestões para seu aperfeiçoamento são apresentadas.

1.4 Outras Soluções para Monitoramento de Qualidade de Chamadas Telefônicas IP

A CESNET, rede de ensino e pesquisa da República Tcheca, implementou uma infraestrutura de monitoração e contabilização de chamadas por meio de utilização de servidores RADIUS [17] e coleta de CDRs emitidos por *gateways* de voz [28]. Em [29], é apresentada uma arquitetura denominada *Enterprise Call Analysis System* (ECAS), também baseada na

coleta e análise de CDRs gerados por *gateways* de voz. Ambas as soluções apresentam a mesma limitação de não permitirem a monitoração da qualidade das chamadas realizadas entre telefones IP.

Em [30], é apresentada uma solução pouco confiável para acompanhamento da qualidade oferecida por um ambiente Internet. 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 [31], assim como em diversos produtos comerciais [32,33,34], 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.

A arquitetura de monitoração proposta supera todas as limitações das soluções apresentadas acima, pois possibilita a monitoração de qualidade de todas as chamadas dos usuários realizadas em um ambiente VoIP, independentemente dos elementos envolvidos na chamada, da natureza do tráfego (ponto-a-ponto ou não) e do uso de criptografia, já que os indicadores de qualidade são emitidos diretamente dos telefones IP e dos *gateways* de voz.

1.5 Organização do Trabalho

Este trabalho está estruturado em seis capítulos. O primeiro capítulo apresenta conceitos introdutórios, uma descrição do ambiente do serviço *fone@RNP*, onde a arquitetura de monitoração de qualidade de chamadas telefônicas IP proposta está sendo aplicada, os objetivos e a organização do trabalho.

No segundo capítulo o Modelo E é discutido em detalhes e as diferentes abordagens encontradas na literatura para determinação das variáveis que compõem o modelo são analisadas e revisadas.

No terceiro capítulo é apresentado um módulo para avaliação objetiva de qualidade de voz baseado no Modelo E, denominado MOBVEM (Modified OpenH323 Voice Evaluation Module) [5], que é um dos resultados desta dissertação. Este módulo, que implementa as diferentes abordagens de cômputo do modelo e sugestões de revisão à estas abordagens, apresentadas no segundo capítulo, é então utilizado para análise destas revisões. Também é apresentado o H.323 Beacon [35,36,37], ferramenta que passou a utilizar o MOBVEM para avaliação objetiva da qualidade da voz.

No quarto capítulo é apresentada a VQuality [23], biblioteca multiplataforma para avaliação de qualidade de voz capaz de emitir CDRs com indicadores de qualidade diretamente de telefones IP e *gateways* de voz. Suas características de portabilidade e flexibilidade (como o mecanismo de adição de novos modelos de avaliação de qualidade e a facilidade de integração com telefones IP) são apresentadas. Também são descritos os detalhes de alteração no código da biblioteca OpenH323 [38] para adequação aos padrões dos protocolos RTP e RTCP [39] que foram realizados para possibilitar a disponibilização de terminais H.323 com suporte a avaliação de qualidade de voz e emissão de VQCDR.

No quinto capítulo, a arquitetura de monitoração de qualidade de chamadas telefônicas IP proposta por este trabalho é descrita [24]. As especificações do VQCDR e do VQCDR Server, servidor responsável pela coleta, armazenamento e autenticação de VQCDRs, são apresentadas, bem como mostradas considerações sobre a disponibilização de telefones IP e *gateways* de voz com capacidade de emissão de VQCDR. E finalmente, conclusões e sugestões de trabalhos futuros são apresentadas no sexto capítulo.

Capítulo 2

MODELO E PARA AVALIAÇÃO OBJETIVA DE QUALIDADE DE VOZ

O Modelo E implementa um mecanismo baseado na soma de termos que representam distorções na qualidade da voz, tais como atrasos de transmissão, eco, distorções introduzidas pelos equipamentos utilizados, entre outros fatores. O resultado do modelo é o fator escalar R , que pode ser convertido para a escala de pontuação MOS.

Neste capítulo, o Modelo E é descrito em detalhes e é apresentado como os termos de distorção da qualidade da voz que compõem o modelo podem ser determinados. Críticas e sugestões sobre as abordagens de determinação destes termos são então apresentadas e discutidas.

2.1 Decompondo o Modelo E

O princípio fundamental do Modelo E, definido na Recomendação ITU-T G.107 [11], baseia-se no conceito de que “fatores psicológicos numa escala subjetiva são aditivos” [11], ou seja, cada contribuição ocasionada por um fator de perda na degradação de um sistema de comunicação de voz pode ser computada separadamente, embora isso não implique que tais fatores não estejam correlacionados [26]. O resultado final do cômputo dos fatores de perda é um fator escalar R , que varia de 0 (péssimo) a 100 (excelente).

O fator R pode ser convertido para escala de pontuação MOS através da seguinte expressão de terceiro grau [11]:

Para $R < 0$: $MOS = 1$

Para $0 \leq R \leq 100$: $MOS = 1 + 0,035 R + 7 \cdot 10^{-6} R (R-60) (100-R)$ (1)

Para $R > 100$: $MOS = 4,5$

Normalmente, o fator R é descrito em categorias de valores, tal como pode ser consultado no Quadro 1. Sistemas cuja qualidade da fala seja avaliada em $R \leq 60$ não são recomendáveis [11], sendo desejável obter $R \geq 70$.

Quadro 1. Categorias de transmissão da fala

Fator R	MOS	Satisfação do usuário
$90 \leq R < 100$	4,34 – 4,50	Muito satisfeitos
$80 \leq R < 90$	4,03 – 4,34	Satisfeitos
$70 \leq R < 80$	3,60 – 4,03	Alguns insatisfeitos
$60 \leq R < 70$	3,10 – 3,60	Muitos insatisfeitos
$0 \leq R < 60$	1,00 – 3,10	Quase todos insatisfeitos

O fator R é obtido pela seguinte fórmula:

$$R = R_o - I_s - I_d - I_e + A, \quad (2)$$

onde R_o representa os efeitos da relação sinal-ruído (SNR); I_s representa as perdas simultâneas ao sinal de voz; I_d representa as perdas associadas ao atraso fim-a-fim; I_e representa as perdas associadas ao equipamento utilizado; e A corresponde ao fator de vantagem, ou fator de expectativa.

Vejamos a seguir quais os fatores e grandezas que influenciam cada uma dessas parcelas e quais delas podem ser reduzidas a uma constante.

2.1.1 Relação Sinal-ruído Básica, R_o

Na determinação da relação sinal-ruído básica, são computados o ruído acrescido pelos circuitos de transmissão, o ruído ambiente no lado do receptor e no lado do emissor e o teto de ruído correspondente à sensibilidade do sistema auditivo humano. A Recomendação ITU-T G.107 [11] apresenta as expressões matemáticas e os valores padrões para cada um desses ruídos, resultando num valor padrão de 94,77 para R_o .

2.1.2 Fator de Perdas Simultâneas, I_s

Entre as perdas que ocorrem mais ou menos simultaneamente ao sinal de voz, estão a queda na qualidade devido a uma conexão de volume demasiadamente alto, perdas causadas pela interferência da própria voz do locutor sobre o fone de ouvido do mesmo *headset* que utiliza para falar e a distorção de quantização causada pela digitalização do sinal de voz. As perdas causadas pelos *codecs*¹ de baixa taxa de transmissão de bits são computadas separadamente pelo fator de perda I_e . A Recomendação ITU-T G.107 [11] também apresenta as expressões e os valores padrões para as perdas que contribuem no cálculo do fator I_s . Sendo que a versão mais recente da recomendação apresenta uma modificação no cálculo de I_s em relação às versões anteriores. Porém, a atualização não alterou substancialmente o valor padrão de I_s , que reduziu de 1,43 para 1,41.

2.1.3 Fator de Perdas Associadas ao Atraso, I_d

O fator I_d é determinado pela seguinte expressão [11]:

$$I_d = I_{dte} + I_{dle} + I_{dd}, \text{ onde:} \quad (3)$$

I_{dte} representa as perdas devido ao eco no lado transmissor; I_{dle} representa as perdas devido ao eco no lado receptor; e I_{dd} representa as perdas relacionadas ao atraso superior a 100 ms.

¹ O codificador de voz é comumente chamado de *codec*.

Para o cálculo do fator Id , é necessária a medida de três diferentes tipos de atraso, ilustrados na Ilustração 2.

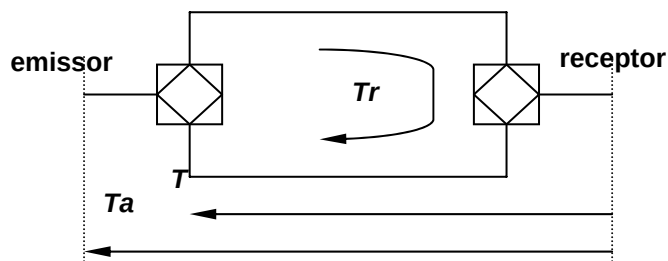


Ilustração 2. Representação de atrasos em um sistema de comunicação de voz

Ta é o atraso absoluto no sistema sem eco. T é o atraso médio percorrido pelo eco. Tr é o atraso de ida-e-volta no circuito a quatro fios. Como não é possível medir separadamente esses três atrasos [40], a seguinte aproximação é utilizada [27]:

$$T = Ta = Tr/2 \quad (4)$$

Na seção 2.2 são apresentadas considerações para determinação do fator Id .

2.1.4 Fator de Perdas Associadas ao Equipamento, Ie

O fator de perda Ie é um meio flexível de se computar a degradação da qualidade causada pelos *codecs* de alta complexidade e com baixa taxa de transmissão de bits. Seu valor para cada *codec* depende dos resultados de exaustivos testes de pontuação MOS sob diversas taxas de perdas de pacotes.

O Apêndice I da Recomendação ITU-T G.113 [41,42,43] fornece valores provisórios do Ie para alguns *codecs* (G.723.1-A, G.729-A, GSM-EFR e G.711) sob condições de perdas aleatórias de pacotes e de perdas em rajada. Na seção 2.2 também serão apresentadas algumas abordagens para o cálculo do fator de perda Ie .

2.1.5 Fator de Vantagem, A

O fator de vantagem A é empregado para definir o grau de tolerância que um usuário espera pela conveniência de utilização de uma determinada tecnologia. Segundo os valores provisórios apresentados na Recomendação ITU-T G.107 [11], este fator varia de 0 (telefonia fixa) a 20 (localidades de difícil acesso que necessitam de enlace de satélite, por exemplo).

Para a telefonia celular é recomendado o uso de A igual a 5 para redes *in-door* ou igual a 10 para redes geográficas. Para VoIP, a ITU-T G.107 [11] recomenda zero. No entanto, pode-se considerar que este fator é uma variável com influência cultural e que, para países que possuem serviço de telefonia precário e caro, como o Brasil, este fator pode ser aumentado. Para efeitos gerais, considera-se neste trabalho o fator A como zero.

Assim, temos que a equação (2) pode ser reduzida a:

$$R = 93,36 - Id(Ta) - Ie(codec, perda_{pacotes na rede no buffer}), \quad (5)$$

onde Id é uma função do atraso fim-a-fim e Ie é, simplificada, uma função que depende do tipo de *codec* utilizado e da taxa de perda de pacotes. Na seção a seguir veremos detalhadamente como esses dois fatores de perda podem ser determinados.

2.2 Determinação dos Fatores de Perda Id e Ie

Nesta seção são discutidos os procedimentos de determinação dos fatores de perda Id e Ie , necessários à equação (5), que simplifica a equação (2), para cômputo do Modelo E.

2.2.1 Determinação do Fator de Perda Id

Segundo a ITU-T G.107 [11] e versões anteriores, o fator de perda Id é determinado pela equação (3), cujas parcelas são determinadas pelas seguintes expressões descritas a seguir:

parcela correspondente às perdas devido ao eco no lado transmissor (*Idte*):

$$Idte = \left[(Roe - Re) / 2 + \sqrt{(Roe - Re)^2 / 4 + 100} - 1 \right] \cdot (1 - e^{-T}), \text{ onde:} \quad (6)$$

$$Roe = -1,5 \cdot (No - RLR) \quad (7)$$

$$Re = 80 + 2,5 \cdot \left(TELR - 40 \cdot \log_{10} \left(\frac{1 + T / 10}{1 + T / 150} \right) + 6 \cdot e^{-0,3T^2} - 14 \right) \quad (8)$$

parcela correspondente às perdas devido ao eco no lado receptor (*Idle*):

$$Idle = (Ro - Rle) / 2 + \sqrt{(Ro - Rle)^2 / 4 + 169}, \text{ onde:} \quad (9)$$

$$Rle = 10,5 \cdot (WELP + 7) \cdot (Tr + 1)^{-0,25} \quad (10)$$

parcela correspondente às perdas relacionadas ao atraso absoluto da voz (*Idd*):

$$Idd = 0, Ta > 100 \text{ ms};$$

$$Idd = 25 \cdot \left[\sqrt[6]{1 + \left(\log_2 \left(\frac{Ta}{100} \right) \right)^6} - 3 \cdot \sqrt[6]{1 + \left(\left(\log_2 \left(\frac{Ta}{100} \right) \right) / 3 \right)^6} + 2 \right], Ta > 100 \text{ ms}; \quad (11)$$

No Quadro 2, encontram-se os valores recomendados por [11] para os termos das equações de (6) a (11).

Quadro 2. Valores recomendados para o cálculo de *Id*

Grandeza	Significado	Valor padrão
No	Ruído total em dBm0p	-61,18 dBm0p
Ro	Relação sinal-ruído básica	94,77
RLR	Nível de intensidade no receptor	2,00 dB
TELR	Nível de intensidade do eco no emissor	65,00 dB
WELP	Caminho do eco ponderado	110,00 dB

Uma interpolação pode ser utilizada para aproximar as equações (3), (6), (7), (8), (9) e (10), combinadas com a consideração feita pela equação (4). Cole e Rosenbluth [27] propõem uma expressão interpolada para Id , dependente apenas de Ta :

$$\begin{aligned} Id &= 0,024 Ta, Ta \leq 177,3 \text{ ms}; \\ Id &= 0,134 Ta - 18,103, Ta > 177,3 \text{ ms} \end{aligned} \quad (12)$$

No trabalho apresentado por Clark [26] é proposta uma expressão que está nitidamente incorreta, pois penaliza o fator Id para valores de $Ta \leq 175$ ms:

$$\begin{aligned} Id &= 4 Ta, Ta \leq 175 \text{ ms}; \\ Id &= 0,111 Ta - 15,444, Ta > 175 \text{ ms} \end{aligned} \quad (13)$$

Este fato ocorre possivelmente devido a um erro tipográfico. Calculando o ponto de inflexão entre as duas curvas que deveriam ser expressas pela expressão (13) de Clark, concluí-se que a equação correta deveria ser expressa por:

$$\begin{aligned} Id &= 0,023 Ta, Ta \leq 175 \text{ ms}; \\ Id &= 0,111 Ta - 15,444, Ta > 175 \text{ ms} \end{aligned} \quad (14)$$

que apresenta grande semelhança com a expressão (12).

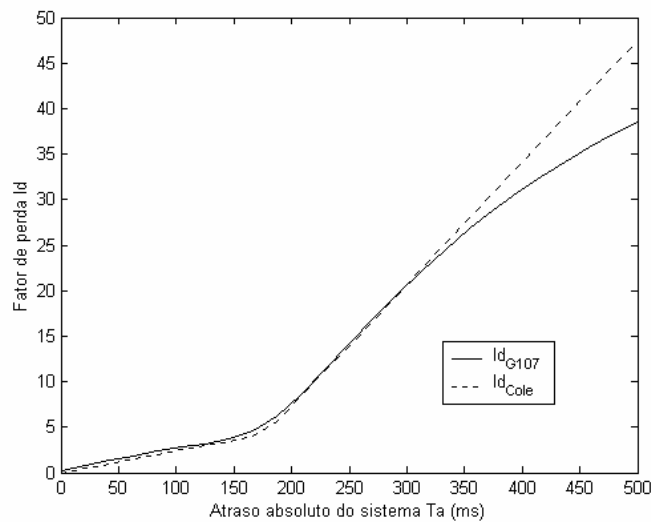


Ilustração 3. Comparação entre a aproximação paramétrica sugerida por Cole e Rosenbluth e a curva de Id recomendada pela ITU-T G.107

Na Ilustração 3, é representada a curva obtida a partir da Recomendação ITU-T G.107 [11] e a aproximação sugerida por Cole e Rosenbluth [27]. Esta última, ainda que matematicamente muito mais simples, retorna valores suficientemente precisos de Id para $Ta \leq 330$ ms. Para situações onde $Ta > 330$ ms, a aproximação piora progressivamente. Entretanto, para situações de Ta tão elevado, a qualidade da ligação já estará seriamente comprometida, exceto em casos específicos onde um *codec* de baixa compressão em condições mínimas de perdas e valor elevado de fator de vantagem A é utilizado.

2.2.2 Determinação do Atraso Absoluto, Ta

Baseado em [44], Ta pode ser obtido pela equação:

$$Ta = T_{codec} + T_{rede} + T_{buffer} \quad (15)$$

onde T_{codec} é o atraso inerente ao *codec* utilizado, T_{rede} é o atraso da rede e T_{buffer} é o atraso adicionado pelo *buffer* de compensação de *jitter*.

T_{rede} foi considerado simétrico e calculado como sendo a metade de RTT (*Round-Trip Time*), que representa o tempo de ida e volta na rede, já que para uma medição mais precisa, seria necessário que manter os relógios das máquinas de origem e destino sincronizados, o que exigiria uma infra-estrutura mais elaborada, como por exemplo, uso do protocolo NTP (*Network Time Protocol*) ou de GPS (*Global Position System*).

Uma observação importante é que a variação de Ta é eliminada pelo *buffer* de compensação de *jitter*, já que todos os pacotes são atrasados de modo que possuam um Ta correspondente ao atraso máximo tolerado [44]. Tais atrasos são necessários para se assegurar a cadência correta na reprodução da fala no dispositivo receptor. Conseqüentemente, quando um *buffer* fixo é utilizado, a influência do tempo é mantida constante durante toda a ligação. Entretanto, caso seja utilizado um *buffer* dinâmico, a influência do tempo pode oscilar no

momento em que o tamanho do *buffer* se altera, o que só deve ocorrer em períodos de silêncio, para não degradar a reprodução da fala.

Dessa forma, é utilizado:

$$T_a = T_{codec} + T_{primeiro\ pacote} + (T_{buffer\ total} / 2) \quad (16)$$

onde T_{codec} é o atraso inerente ao *codec* utilizado, que possui valor fixo conhecido [44], $T_{primeiro\ pacote}$ é o T_{rede} referente ao primeiro pacote recebido no período de fala (*talkspurt*) e $T_{buffer\ total}$ é o tamanho total do *buffer* de compensação de *jitter*.

2.2.3 Determinação do Fator de Perda I_e

A determinação do fator de perda I_e depende de testes subjetivos (pontuação MOS) executados pelo Grupo de Estudo 12 da ITU-T, cujos resultados são publicados periodicamente no Apêndice I da Recomendação ITU-T G.113 [41,42,43].

Segundo a ITU-T G.107 [11], o fator efetivo de perda associada ao equipamento $I_{e,eff}$ é calculado a partir da expressão:

$$I_{e,eff} = I_e + (95 - I_e) \cdot Ppl / (Ppl + Bpl) \quad (17)$$

onde I_e é o fator de perda associada ao equipamento a 0% de perda de pacotes, Bpl é o fator de robustez à perda de pacotes e Ppl é a probabilidade de perda de pacotes.

Os valores de I_e para perda de pacotes nula e de Bpl estão tabelados para alguns *codecs* em [41,42,43]. O valor de Ppl deve ser de conhecimento do projetista ou substituído pela informação de perda de pacote obtida a partir da monitoração do sistema de comunicação VoIP em estudo. Para determinar os valores de I_e e Bpl , a Recomendação ITU-T G.113 [43] leva em consideração que as condições de perdas são aleatórias, ou seja, que “a probabilidade de perda de um pacote independe da probabilidade de perda de qualquer outro pacote”.

Porém, a ITU-T G.113 [43] alerta que essa premissa é, na prática, insatisfatória para redes VoIP. Além disso, deve-se ressaltar que a determinação do $I_{e,eff}$ não consiste de apenas uma simples consulta de uma tabela, já que os fatores descritos a seguir devem ser considerados:

- Percentagem de perda de pacotes

Para que o resultado do Modelo E se aproxime da pontuação MOS, essa percentagem deve ser medida após o *buffer* de *jitter*, pois pacotes podem ser entregues à aplicação VoIP demasiadamente tarde para que sejam reproduzidos. A Recomendação ITU-T G.113 [43] alerta ainda que “implementações de *buffers* de *jitter* podem variar consideravelmente entre fabricantes e ainda mesmo entre versões diferentes de um mesmo dispositivo”. Em outras palavras, a taxa de perda de pacotes depende, além das condições da rede, do algoritmo implementado no *buffer* de compensação de *jitter*.

- Número de quadros (*frames*) de voz por pacote de dados

Após digitalizar e comprimir a voz, o *codec* entrega quadros à camada de transporte, que pode, ou não, aguardar a chegada de mais quadros antes de formar um pacote. Por isso, a perda de um pacote IP afeta mais ou menos trechos de voz, conforme a lógica do aplicativo VoIP utilizado. No entanto, as tabelas constantes na ITU-T G.113 [43] apresentam os valores de I_e e B_{pl} para tamanhos fixos de quadros, conforme o *codec* consultado. Assim, ao se consultar estas tabelas deve-se considerar que encapsular x quadros de voz de tamanho igual a y ms em um pacote IP é equivalente a se encapsular 1 quadro de voz de tamanho igual a xy ms em um pacote IP.

- Uso de algoritmos compensadores de perda (*Packet Loss Concealment* - PLC)

O uso de PLC torna o *codec* mais robusto a perdas de pacotes. Dessa forma, o projetista deve tomar conhecimento de qual mecanismo de compensação de perda o aplicativo

VoIP utiliza, pois na Recomendação ITU-T G.113 [43] os valores de I_e e B_{pl} são dependentes do tipo de PLC empregado.

- Uso de detectores de atividade de voz (*Voice Activity Detection* - VAD)

A detecção da atividade de voz pelo aplicativo VoIP não é imediata, ou seja, a fala está presente antes do início da execução da função de detector e pode ocorrer o corte nas primeiras sílabas da locução. Esse efeito é denominado *clipping* e seu impacto sobre a qualidade da voz varia de um *codec* para outro e, conseqüentemente, influencia no valor de I_e , mesmo em condições de perda de pacote nula.

2.3 Modelo E Estendido

Uma importante contribuição foi apresentada por Clark [26] para o cálculo do fator R , tendo sido adotada pelo grupo TIPHON do ETSI na especificação técnica ETSI TS 101 329-5 v1.1.1 [20] e revisão [21]. Trata-se de uma versão estendida do Modelo E apresentado na Recomendação ITU-T G.107 e que estabelece alguns novos conceitos descritos a seguir:

- Qualidade instantânea

Qualidade medida ou calculada utilizando-se a perda de pacotes ou outros fatores de degradação.

- Qualidade percebida

Qualidade percebida pelo usuário em algum instante durante a locução.

- Comportamento da perda de pacotes alternantes

Normalmente, as perdas de pacotes nas redes de dados IP ocorrem em períodos de rajadas (*burst*) que se alternam com períodos de perdas isoladas (*gap*) [26]. Mecanismos de

compensação de perdas (PLC) conseguem amenizar tais distorções em períodos de poucas perdas, mas não quando estas ocorrem em rajadas.

- Efeito de memória recente

Durante pesquisas conduzidas para determinação de pontuações MOS, notou-se que a qualidade percebida de uma chamada telefônica varia de acordo com a localização temporal das perdas. Dessa forma, distorções que ocorrem ao final de uma chamada têm um efeito psicológico negativo maior sobre os avaliadores do que distorções que acontecem no início de uma chamada [26,22].

Segundo estudos realizados pelos laboratórios da France Telecom [45], se a qualidade instantânea muda de “bom” para “ruim” em algum momento durante a chamada, o usuário não irá instantaneamente perceber tal mudança, mas aos poucos ficará incomodado com a degradação da qualidade da fala. Então, notou-se que um usuário leva em média 5 segundos para perceber completamente a piora da qualidade e 15 segundos para perceber completamente a melhoria da qualidade. Clark [26] modela esse efeito aproximando por uma curva exponencial de constante de tempo de 5 segundos na transição de “bom” para “ruim” e de 15 segundos na transição de “ruim” para “bom”.

2.3.1 Modelagem de Comportamento Alternante de Perda de Pacotes e de Efeito de Memória Recente

Segundo o Modelo E Estendido [26], a recepção de pacotes de voz podem ocorrer em períodos de perdas isoladas (*gap*) e perdas em rajadas (*burst*). Se o número de pacotes recebidos entre duas perdas de pacotes sucessivas for menor que a constante G_{min} (dado como 16), então estes dois pacotes perdidos e a sequência de pacotes recebidos entre eles são considerados como parte de um período de perdas em rajada (*burst*). Se uma sequência de

G_{min} pacotes ou mais são recebidos corretamente, esta sequência é considerada como parte de um período de perdas isoladas (gap).

Para representar o comportamento alternante da perda de pacotes, Clark [26] utiliza uma cadeia de Markov com 4 estados, os quais “representam as condições de recebimento e de perda de pacotes, nas condições de perda em rajada e de perdas isoladas”:

Estado 1 – pacote recebido durante período de perdas isoladas;

Estado 2 – pacote recebido durante período de perdas em rajadas;

Estado 3 – pacote perdido durante período de perdas em rajadas;

Estado 4 – pacote perdido durante período de perdas isoladas.

A Ilustração 4 apresenta, em forma de grafo, estes 4 estados e suas respectivas probabilidades de transição entre estados P_{11} , P_{13} , P_{14} , P_{22} , P_{23} , P_{31} , P_{32} , P_{33} e P_{41} .

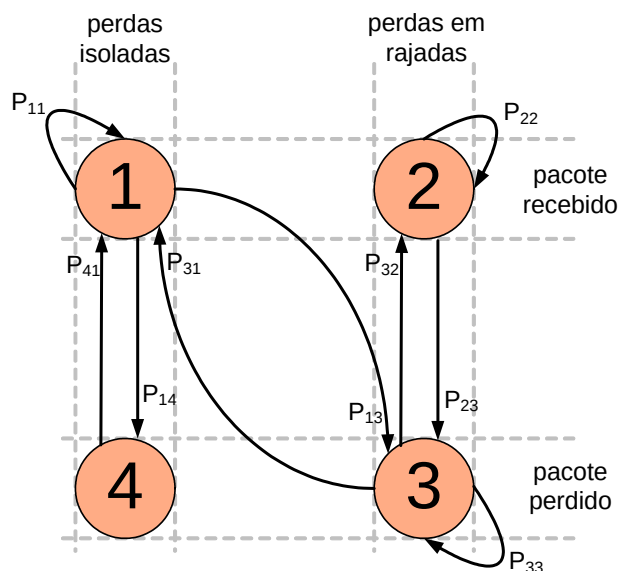


Ilustração 4. Comportamento alternante de perdas modelado com uso de cadeia de Markov de 4 estados

Baseado no cálculo destas probabilidades, o valor de I_e [43] pode ser obtido para a condição de perdas em rajada (I_{eb}) e para a condição de perdas isoladas (I_{eg}). Ainda de acordo com Clark [26], consideram-se as seguintes grandezas:

I_1 – nível de qualidade na mudança de condição de I_{eb} para I_{eg} ;

I_2 – nível de qualidade na mudança de condição de I_{eg} para I_{eb} ;

b – duração do estado de perdas em rajadas (em segundos);

g – duração do estado de perdas isoladas (em segundos).

Tais grandezas se relacionam da seguinte forma segundo Clark (Ilustração 5):

$$I_1 = I_{eb} - (I_{eg} - I_2)e^{-b/t_1} \quad (18)$$

$$I_2 = I_{eg} + (I_1 - I_{eg})e^{-g/t_2} \quad (19)$$

onde tipicamente $t_1=5$ segundos e $t_2=15$ segundos.

Clark [26] integra as expressões de I_1 e I_2 para obter um I_e médio, que corresponderia ao $I_e(perda)$ da equação (5):

$$I_e(av) = (bI_{eb} + gI_{eg} - t_1(I_{eb} - I_2)(1 - e^{-b/t_1}) + t_2(I_1 - I_{eg})(1 - e^{-g/t_2})) / (b + g) \quad (20)$$

$$I_e(\text{fim de chamada}) = I_e(av) + (k(I_1 - I_e(av)))e^{-y/t_3} \quad (21)$$

O efeito da memória recente para determinação de $I_e(av)$ dado pela expressão (20) pode ser modelado considerando que a qualidade percebida decai exponencialmente até I_{eb} , com constante de tempo t_1 , na transição entre um período de perdas isoladas para perdas em rajada, e aumenta exponencialmente até I_{eg} , com constante de tempo t_2 , na transição entre um período de perdas em rajada para um período de perdas isoladas, segundo as curvas I_1 e I_2 , respectivamente (Ilustração 5). Já para a determinação da qualidade final da ligação, $I_e(\text{fim da chamada})$ dado pela expressão (21), a qualidade percebida aumenta exponencialmente com

constante de tempo t_3 , de valor típico de 30 segundos, a partir da última perda significativa de pacotes em rajada. A variável y representa o tempo transcorrido desde a última perda em rajada e k é uma constante de valor igual a 0,7 [26].

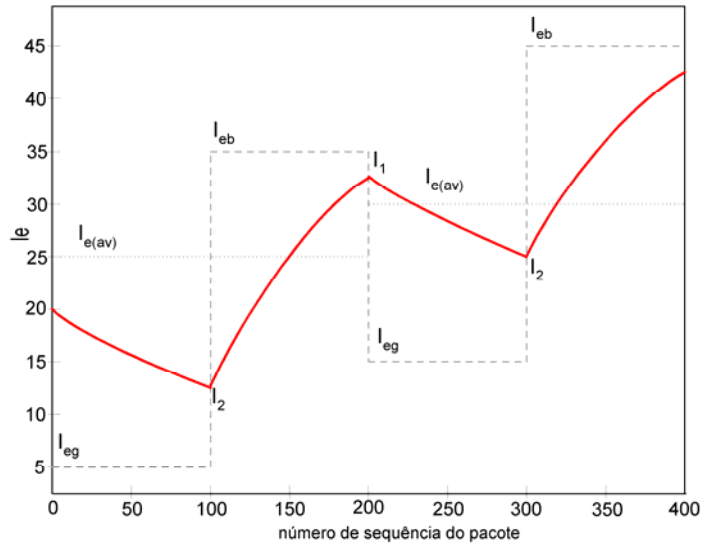


Ilustração 5. Representação do efeito da memória recente

2.4 Sugestões de Revisão ao Modelo E Estendido

Considerando que as variáveis I_{eg} , I_{eb} , b e g nas equações (18) e (19) podem ser alimentadas através de parâmetros colhidos do transporte de voz, Clark [26] combina as equações (18) e (19) isolando I_1 e I_2 , em função destas variáveis:

$$I_1 = \frac{I_{eb}(1 - e^{-b/t_1}) + I_{eg}(1 - e^{-g/t_2})e^{-b/t_1}}{1 - e^{-b/t_1 - g/t_2}} \quad (22)$$

$$I_2 = \frac{I_{eg}(1 - e^{-g/t_2}) + I_{eb}(1 - e^{-b/t_1})e^{-g/t_2}}{1 - e^{-b/t_1 - g/t_2}} \quad (23)$$

Todavia, refazendo os passos descritos por Clark [26], percebe-se que a partir das equações (18) e (19) não se consegue chegar às equações (22) e (23).

Analisando como as equações (18) e (19) foram derivadas da Ilustração 5 (ver também Apêndice A), verifica-se que a equação (18) possui um pequena incorreção que pode causar

problemas na determinação de $I_e(av)$ e de $I_e(fim\ da\ chamada)$. Todavia, apesar da incorreção da equação (18), as equações (20) e (21) estão corretas. Considerando ainda que a equação (18) foi transportada para a Especificação Técnica ETSI TS 101 329-5 [20] e revisões posteriores [21,22], é importante alertar para que tais especificações técnicas sejam corrigidas e que a equação (18) seja alterada para a expressão:

$$I_1 = I_{eb} - (I_{eb} - I_2) e^{-b/t_1} \quad (24)$$

Partindo das equações (24) e (19), todas as expressões subseqüentes são corretamente obtidas sem problemas.

A incorreção da equação (18) sugere que um erro relevante pode ser inserido no cômputo do Modelo E. Entretanto, comparando esta equação com sua expressão revisada, a equação (24), nota-se que, em certos casos específicos, onde os períodos de perdas em rajadas tendem a ser grandes (superiores a t_1 , que tipicamente equivale a 5 segundos), ambas equações tendem a convergir para $I_1 = I_{eb}$. Conseqüentemente, os valores de $I_e(av)$ e de pontuação MOS, obtidos com o uso de ambas equações, nestes casos, tendem a convergir para um valor aproximado.

No Capítulo 3 é apresentado um módulo para avaliação objetiva de qualidade de voz que implementa o cômputo do Modelo E original, suas extensões e revisões propostas neste capítulo. Objetivando uma melhor análise do modelo, são apresentadas medições e comparações entre as abordagens de cômputo discutidas neste capítulo.

Capítulo 3

MODIFIED BASED OPENH323 VOICE EVALUATION MODULE

O Laboratório de Voz sobre IP da Universidade Federal do Rio de Janeiro desenvolveu uma infra-estrutura [46] de geração, coleta e monitoramento da qualidade de ligações VoIP baseada na biblioteca OpenH323 [38] capaz de extrair, por meio de monitoração ativa, dados estatísticos referentes ao nível de transporte de ligações e apresentar graficamente informações de perdas de pacotes, *jitter* e RTT ao longo de uma ligação.

Com o intuito de avaliar as abordagens do cômputo do Modelo E, essa infra-estrutura foi estendida neste trabalho, implementando um módulo de avaliação analítica de qualidade de voz capaz de extrair, além das informações referentes ao nível de transporte, informações referentes ao nível de aplicação, tais como descarte de pacotes no *buffer* de compensação de *jitter* e atraso total fim-a-fim (Ta).

Este módulo denominado Modified OpenH323 Based Voice Evaluation Module (MOBVEM). O MOBVEM também oferece: análise da distribuição de descartes de pacotes no *buffer* e de perdas na rede, utilizando o modelo baseado em cadeia de Markov [26] capaz de distinguir períodos de perdas em rajadas de períodos de perdas isoladas; utilização de *buffer* de compensação de *jitter* com algoritmo adaptativo para ajuste dinâmico às condições da rede; e cômputo do Modelo E segundo as abordagens comentadas neste trabalho.

A infra-estrutura (Ilustração 6) utiliza o H.323 [12] como protocolo de sinalização de chamadas e possui sua arquitetura formada por dois programas: o *openam* (ou Open Answering Machine), um tipo de secretária eletrônica VoIP, e o programa *caller* (ou Call Generator), uma modificação do *openam* desenvolvida por [47]. O programa *caller* realiza ligações para o programa *openam*, enviando um arquivo pré-gravado de áudio durante a ligação. O *openam*, por sua vez, armazena o áudio recebido em um arquivo e registra em um log que possui, além de toda sinalização H.323, todos os eventos de envio e recepção de fluxos de mídia e relatórios RTCP e informações referentes à camada de aplicação, como por exemplo, eventos relacionados ao *buffer* de compensação de *jitter*. O arquivo de áudio recebido pelo *openam* poderia, eventualmente, ser submetido a outros métodos objetivos de avaliação, como o PESQ (*Perceptual Evaluation of Speech Quality*) [8] ou o PAMS (*Perceptual Analysis Measurement System*) [7].

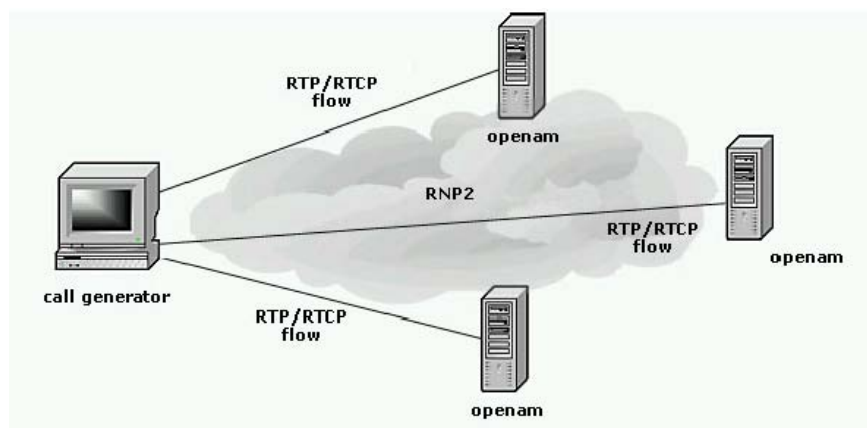


Ilustração 6. Infra-estrutura de geração, coleta e monitoramento da qualidade de ligações VoIP

Utilizando scripts desenvolvidos em Perl [48], o MOBVEM é capaz de extrair dos arquivos de log das ligações as informações necessárias para computar o Modelo E, permitindo ainda, que gráficos com os resultados obtidos sejam gerados a posteriori.

3.1 Avaliação dos Modelos

Com o objetivo de avaliar as diferentes abordagens do cômputo de Modelo E, foi utilizado o módulo de avaliação para extrair a qualidade de várias ligações VoIP entre a Universidade Federal do Rio de Janeiro (UFRJ) e a Universidade Federal do Amazonas (UFAM), interconectadas através do *backbone* da Rede Nacional de Pesquisa (RNP2). Para o experimento, é utilizado o *codec* G.723.1 de 6,3 kbps + VAD com quadro de tamanho de 30 ms.

Dentre as ligações originadas, foi selecionada uma que possuisse as características desejadas para poder observar o comportamento de perdas de pacotes, transição entre estados de perdas em rajada e perdas isoladas, efeito de memória recente, entre outros. Os resultados da avaliação da qualidade dessa ligação são apresentados a seguir.

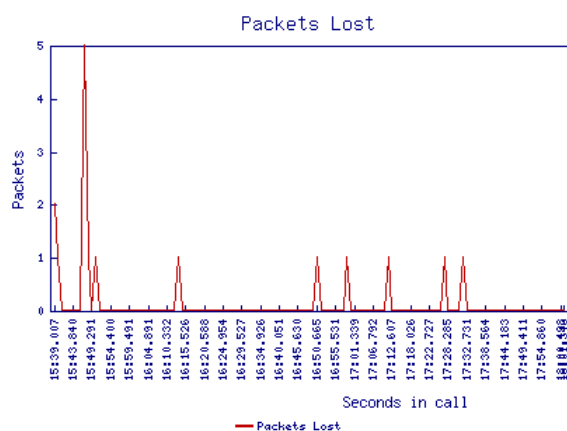


Ilustração 7. Perda de pacotes na rede

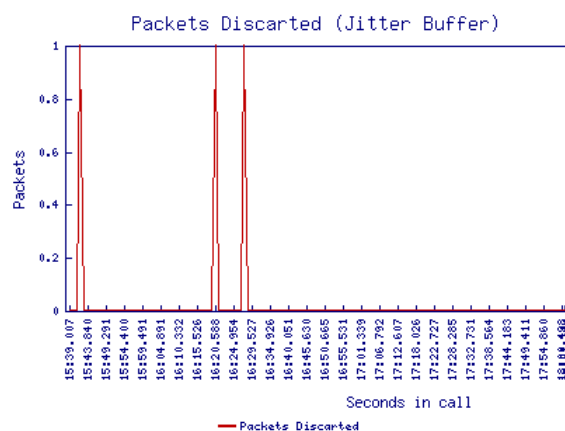


Ilustração 8. Descarte de pacotes no *buffer* de compensação de *jitter*

A Ilustração 7 apresenta a perda de pacotes na rede e a Ilustração 8 apresenta o descarte de pacotes no *buffer* de compensação de *jitter* ao longo da ligação selecionada. É importante notar que a análise da distribuição de perdas não distingue perdas de pacotes na rede de descartes de pacotes no *buffer*. Logo, é a soma destes dois fatores que influenciará na determinação do MOS da ligação.

A Ilustração 9 apresenta a variação do I_e ao longo da ligação. Os valores de I_e ilustrados nessa figura foram obtidos usando a equação (24), que substitui a equação (18) da abordagem de Clark [26] segundo nossas considerações (seção 2.4). A Ilustração 10 apresenta a variação de I_e ao longo da ligação segundo a abordagem original de Clark [26], que foi considerada incorreta. Consta-se neste trabalho que Clark obtém uma avaliação da qualidade da voz mais pessimista, já que o valor de I_e aumenta mais bruscamente após um período de perdas em rajada e decresce mais lentamente na transição de um período de perdas em rajada para um período de perdas isoladas.

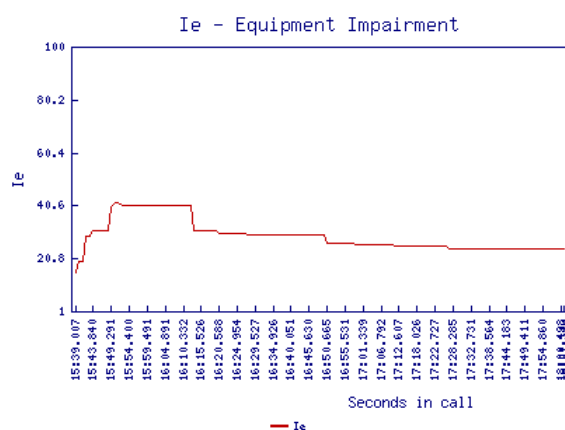


Ilustração 9. I_e obtido segundo a revisão sugerida

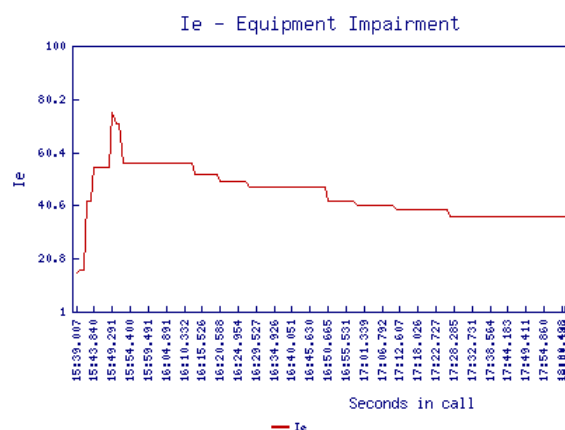


Ilustração 10. I_e obtido segundo a abordagem de Clark

A Ilustração 11 apresenta a variação do I_d ao longo da ligação. Seus valores expressos na curva foram determinados segundo a abordagem em ITU-T G.107 [11].

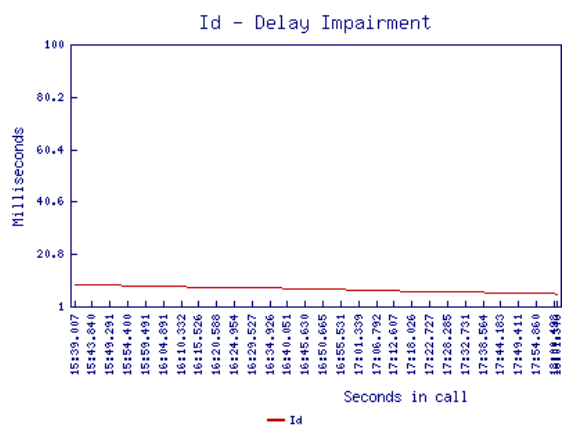


Ilustração 11. I_d (ITU-T G.107)

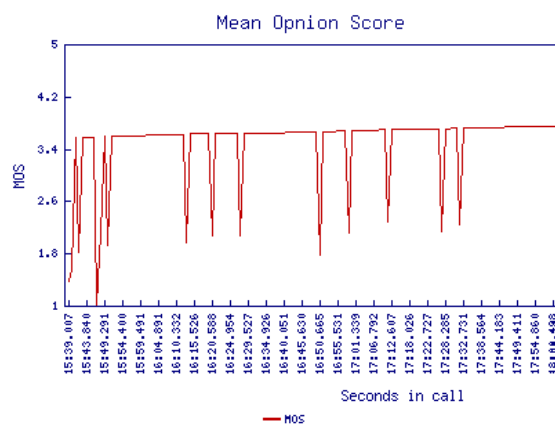


Ilustração 12. MOS (ITU-T G.107)

Observa-se que há uma ligeira queda da influência do atraso ao longo de toda ligação (Ilustração 11). Esta ligeira queda ocorre porque o tamanho inicial do *buffer* de compensação de *jitter* é superestimado (em 116 ms). A aplicação percebe que esse valor é superior ao valor ideal e vai diminuindo-o progressivamente, o que, conseqüentemente, reduz a influência do atraso na degradação da fala.

A Ilustração 12 apresenta a variação do MOS computado pelo Modelo E segundo a abordagem em ITU-T G.107 [11]. Essa abordagem não considera o efeito de memória recente e não faz distinção entre os períodos de perdas em rajadas e perdas isoladas. Assim, cada ponto da curva ilustrada nessa ilustração representa um valor instantâneo de MOS, que espera-se que sejam menos fiéis à percepção humana que o resultado apresentado na Ilustração 13.

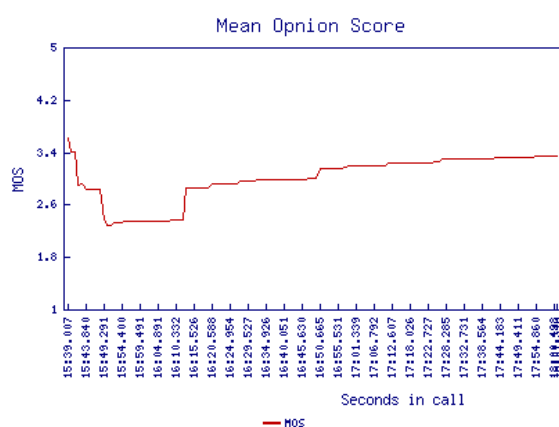


Ilustração 13. Variação do MOS ao longo da ligação obtida segundo a revisão sugerida

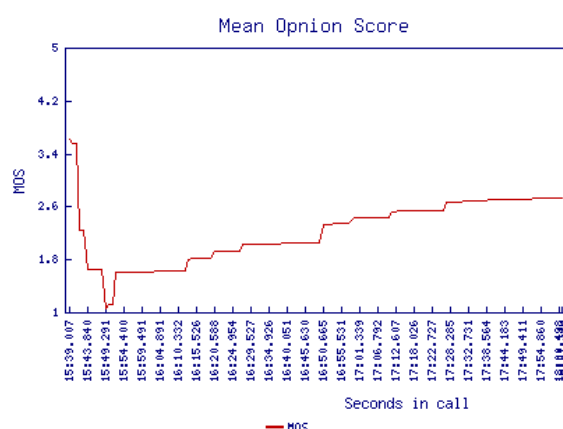


Ilustração 14. Variação do MOS ao longo da ligação obtida segundo a abordagem de Clark

A Ilustração 13 apresenta a variação do MOS computado segundo as considerações descritas nesse trabalho, ou seja, combinando os resultados de influência de I_e e I_d , apresentados na Ilustração 9 e na Ilustração 11, respectivamente. A curva apresentada nessa ilustração representa a qualidade perceptível ao usuário ao longo da ligação. A avaliação final da ligação, $I_e(\text{fim da chamada})$ (21), inferida pelo Modelo E, segundo esta abordagem, foi um MOS de 3,28.

A Ilustração 14 apresenta a variação do MOS computado segundo a abordagem de Clark [26], ou seja, utilizando a equação incorreta (18) na determinação dos valores de I_e (apresentados na Ilustração 10). A avaliação final da ligação, $I_e(\text{fim da chamada})$ (21), neste caso, foi um MOS de 2,66, bastante inferior a 3,28.

Observa-se que o erro introduzido pelo uso da equação (18) resulta em uma estimativa de qualidade bastante inferior em relação ao resultado obtido segundo as considerações desse trabalho. Se o cálculo correto de MOS do modelo estendido não refletir a percepção real dos usuários, o que pode vir a ser comprovado através de testes subjetivos, então os parâmetros padronizados usados no modelo estendido devem ser revistos, pois seus valores podem alterar o resultado do MOS estimado, ou o próprio modelo estendido ser questionado. Esta revisão fica como sugestão para o ETSI e trabalhos futuros.

3.2 H.323 Beacon

O H.323 Beacon [35,36,37] é uma ferramenta para depuração e diagnóstico de problemas (*troubleshooting*) com videoconferências H.323 desenvolvido pelo Internet2 Technology Evaluation Center – ITEC Ohio. Seu propósito é fornecer informações ao usuário final, engenheiro de rede ou operador de conferência, para auxiliar na detecção e na resolução de problemas relacionados à rede ou à aplicação.

O H.323 Beacon é um exemplo de ferramenta que utiliza o MOBVEM para avaliação objetiva de qualidade de chamadas VoIP. Sua integração com MOBVEM é fruto de um trabalho em conjunto com ITEC Ohio. Após a integração, a ferramenta incorporou a funcionalidade de cômputo do Modelo E e a capacidade de plotar graficamente dados estatísticos, entre outras informações de qualidade referentes a voz. A Ilustração 15 apresenta a tela do H.323 Beacon integrado ao MOBVEM.

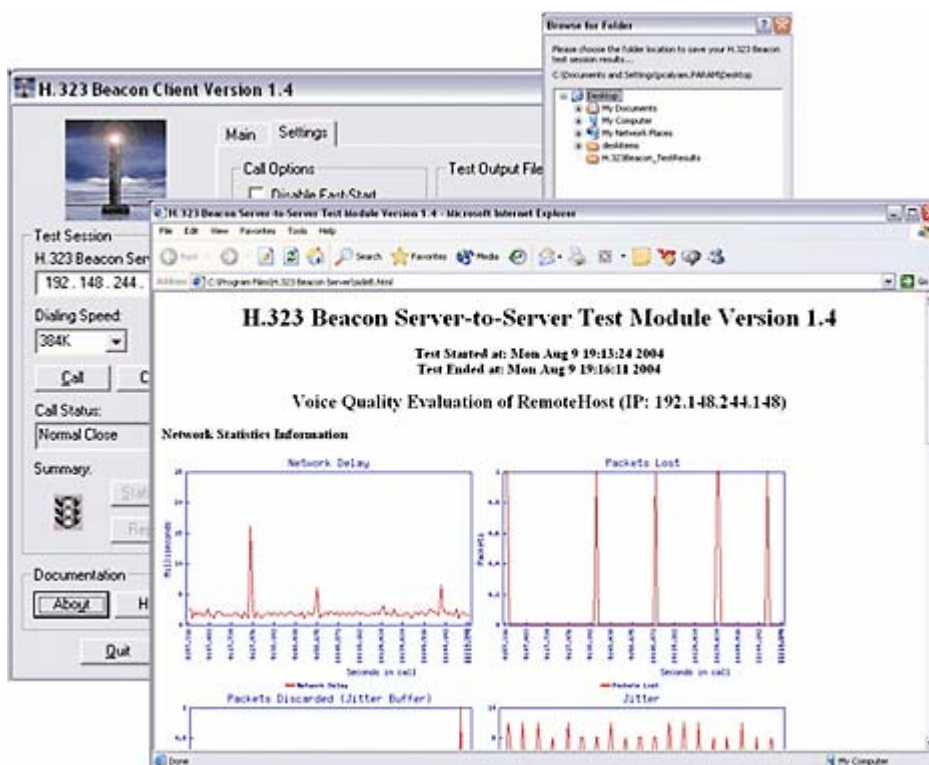


Ilustração 15. H.323 Beacon

Capítulo 4

BIBLIOTECA VOICE QUALITY

Embora ofereça os requisitos necessários para uma ferramenta de medições de qualidade de voz em ambientes de teste, o MOBVEM possui uma série de limitações que inviabilizam sua adoção para medições em ambientes de produção.

Implementada em Perl [48], o módulo sofre a ineficiência de desempenho inerente a uma linguagem interpretada e nem oferece uma *API (Application Programming Interface)* que disponibilize um mecanismo para fácil integração com outros *softwares*. Uma outra importante limitação, associada ao uso da biblioteca OpenH323 [38], é a dificuldade da obtenção do atraso na rede, já que os campos do protocolo RTCP, necessários para cálculo de RTT (*Round-Trip Time*) não são implementados pela pilha RTP da OpenH323. Para estimar o RTT, a MOBVEM necessita analisar simultaneamente os logs gerados pela OpenH323 do transmissor e receptor da mídia, inviabilizando, assim, a análise da qualidade de voz em tempo real e a possibilidade de realizar medições com clientes que não utilizem a biblioteca OpenH323 e que, por conseguinte, não geram o log de chamada no formato requerido.

Objetivando superar estas limitações e ter a perspectiva da criação de clientes VoIP com capacidade de obtenção de parâmetros de qualidade e geração de CDR apropriado, foi desenvolvida a biblioteca VQuality [23]. Escrita em C++, herdou todas as características positivas da implementação anterior, como o cálculo do Modelo E [11] e suas extensões [5,20,21,22]. Todavia, além da performance superior, é flexível, extensível e portátil.

A VQuality foi desenvolvida com a preocupação de facilitar a sua integração com clientes VoIP já existentes, sejam eles baseados em H.323, SIP, MGCP [49], ou em qualquer outra pilha de sinalização. No caso de clientes baseados na OpenH323, diversas modificações tiveram que ser realizadas no seu código para suprir deficiências de sua implementação.

4.1 Flexibilidade e Extensibilidade

A biblioteca VQuality foi desenvolvida segunda a 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.

4.2 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 [50], que permite que sistemas Windows compilem código baseado na API de Pthreads [51], padrão para sistemas Unix. O resultado é um código flexível, que pode ser compilado tanto em Unix e Windows, além de ser portátil para outras plataformas, se necessário.

Para o desenvolvimento da biblioteca foi utilizado o compilador GCC (*GNU Compiler Collection*) [52] para ambientes baseados em UNIX e o compilador *Microsoft Visual C++* [53] para ambientes baseados em Windows.

Embora o código seja independente de plataforma, os scripts que fornecem os mecanismos de compilação dos fontes são específicos para plataformas distintas. Para automatizar a compilação e a distribuição destes fontes, ferramentas de *autotools* [54] são utilizada para gerar scripts que padronizam o mecanismo de compilação entre sistemas operacionais baseados em Unix. Para compilação em ambiente Windows, foi padronizado o uso do *Microsoft Visual C++*. Entretanto, nada impede que outros procedimentos para compilação em outros ambientes de desenvolvimento sejam disponibilizados no futuro.

4.3 Emissão de CDR

Uma funcionalidade muito importante introduzida pela VQuality é sua capacidade de envio de CDR (*Call Detail Record*) ao término de uma chamada. Este CDR contém diversos indicadores referentes à qualidade de voz recebida, além de parâmetros de identificação da chamada e terminais envolvidos, bem como a possibilidade de inclusão de um relatório contendo o histórico da variação dos indicadores de qualidade ao longo de toda a duração da recepção. O CDR é enviado para uma entidade central, denominada VQCDR Server (ver seção 5.3), que coleta, autentica e encaminha o VQCDR para um servidor de armazenamento, que pode ser um banco de dados SQL ou um servidor RADIUS. A comunicação com o VQCDR Server é realizada, a princípio, via protocolo TCP na porta 80, facilitando sua convivência com *firewalls*. O CDR da VQuality, denominado VQCDR (Voice Quality Call Detail Record), possui um formato próprio, cuja estrutura é apresentada na seção 5.2.1.

4.4 Arquitetura da Biblioteca VQuality

A Ilustração 16 mostra os três módulos que integram a biblioteca VQuality. Estes módulos são descritos a seguir.

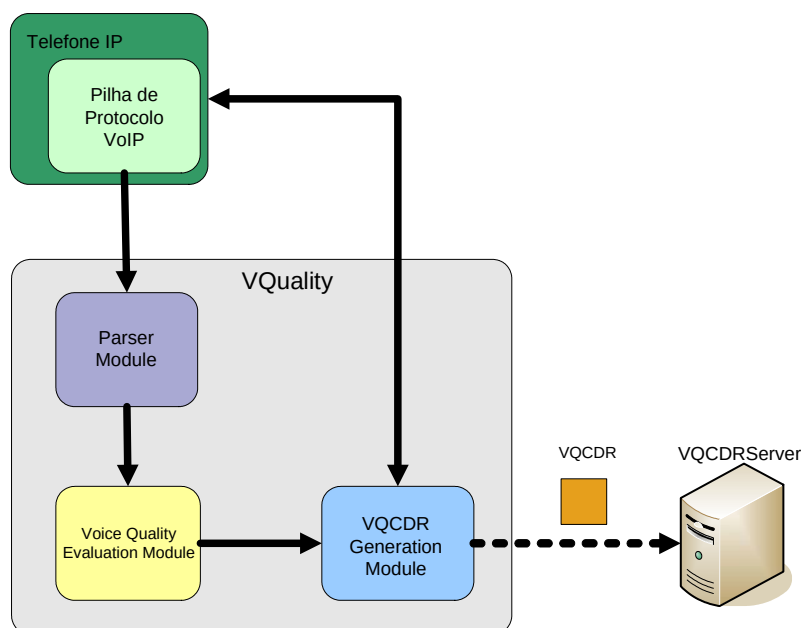


Ilustração 16. Arquitetura da biblioteca VQuality

4.4.1 Parser Module (PM)

O PM recebe da pilha de protocolos VoIP do cliente os parâmetros necessários para o cômputo dos modelos objetivos de avaliação de qualidade de voz. O PM deve ser apto a obter e interpretar todos os parâmetros necessários para o cômputo do modelo a ser utilizado. Considerações sobre a implementação de um PM e sua interação com a pilha de protocolos são apresentados na seção 4.4.4.

4.4.2 Voice Quality Evaluation Module (VQEM)

O VQEM aciona o PM para obter os parâmetros coletados da pilha de protocolos VoIP e realiza a avaliação da qualidade. Estes parâmetros são referentes ao tamanho do *buffer* de compensação de *jitter*, RTT, identificação de codificador de áudio e eventos de recepção, perdas e descarte de pacotes no *buffer* de compensação de *jitter*. Considerações sobre a implementação de um VQEM e sua interação com o PM são apresentados na seção 4.4.5.

4.4.3 VQCDR Generation Module (VGM)

O VGM aciona o VQEM para obtenção dos parâmetros referentes à qualidade da voz computados, e juntamente com os dados de identificação da chamada e dos terminais envolvidos, obtidos diretamente do cliente VoIP, monta o VQCDR e o envia para o VQCDR Server. O VGM pode ser acionado pelo telefone IP para disponibilizar ao usuário informações relativas à qualidade da chamada. Considerações sobre a interação do VGM com o VQEM e com o telefone IP são apresentadas na seção 4.5.

É importante ressaltar que deve existir um PM específico para cada implementação de pilha de sinalização VoIP, assim como deve existir um VQEM para cada modelo de avaliação objetiva. Dessa forma, para integrar a VQuality a um cliente VoIP, é necessário incluir o VGM em seu código, e criar um PM específico, caso não exista. De forma análoga, para adicionar um novo modelo de avaliação de qualidade, é requerido criar um novo VQEM.

4.4.4 Criando um Parser Module

A biblioteca VQuality oferece um mecanismo flexível de integração com implementações de pilhas de protocolos VoIP. Adotando técnicas de orientação a objetos, foi definida uma interface básica, denominada *ParserBase*, que descreve as assinaturas dos métodos que devem ser implementados para que a pilha de protocolos consiga repassar à VQuality os parâmetros necessários para a análise da qualidade de voz. Para criação de um PM específico para o OpenH323, por exemplo, foi implementada a classe *OpenH323LogParserBase*, a partir da interface *ParserBase*.

A interface *ParserBase* permite que os detalhes de implementação da VQuality sejam abstraídos no processo de integração. Desta forma, a preocupação do programador se restringe apenas na determinação dos parâmetros necessários para implementar a interface,

que depende apenas do conhecimento dos detalhes de implementação da pilha de protocolos a ser integrada. A Ilustração 17 apresenta o diagrama de classes UML (*Unified Modeling Language*) que representa interação entre a interface *ParserBase* e a classe *OpenH323LogPaser*.

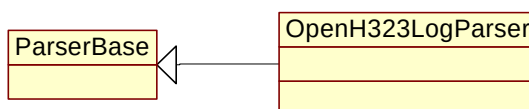


Ilustração 17. Hierarquia de classes para implementação de um PM para o OpenH323

4.4.5 Criando um Voice Quality Evaluation Module

A VQuality disponibiliza VQEMs específicos para cômputo do Modelo E original [11], sua extensão proposta pelo ETSI [20,21,22] e a revisão do ETSI proposta pela UFRJ/UFAM [5], além de oferecer uma estrutura flexível para criação de novos VQEMs para suportes a novos modelos de avaliação de qualidade.

A classe *ExtendedEmodelBase* define as assinaturas dos métodos básicos que devem ser implementados pelos VQEMs, permitindo que estes sejam adicionados sem a necessidade de nenhuma alteração na arquitetura já existente.

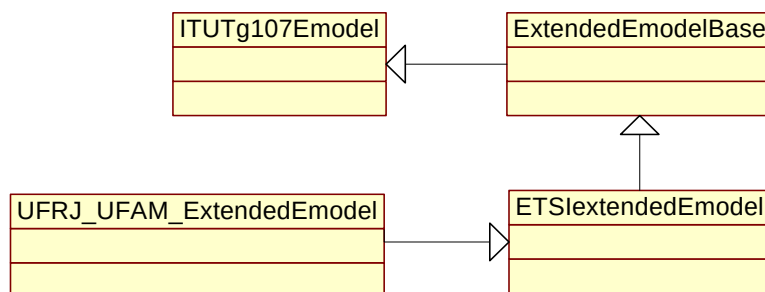


Ilustração 18. Hierarquia de classes para inclusão de VQEMs

A Ilustração 18 apresenta o diagrama de classes UML que representa o relacionamento da classe *ITUTg107Emodel*, referente ao VQEM do Modelo E original, com

as outras classes que implementam VQEMs específicos. A classe *ETSIextendedEmodel*, referente ao VQEM do Modelo E estendido do ETSI, estende a classe *ExtendedEmodelBase*. A classe *UFRJ_UFAM_ExtendedEmodel*, referente ao VQEM da revisão do Modelo E proposta pela UFRJ/UFAM, é um caso particular. Como este VQEM representa uma revisão ao Modelo E estendido do ETSI, ela não implementa a interface *ExtendedEmodelBase* diretamente, mas utiliza a classe *ETSIextendedEmodel* para realizar a sobrecarga (reimplementando) de todos métodos que necessitam ser alterados para a implementação do modelo revisito.

4.5 Integração da VQuality com a Biblioteca OpenH323

Visando incluir as funcionalidades da biblioteca VQuality ao OpenPhone e OhPhone, clientes VoIP H.323 de código aberto usados no serviço fone@RNP, foi desenvolvido um PM específico para a biblioteca OpenH323. Entretanto, a OpenH323 não fornece todos os parâmetros que o VQEM necessita para avaliar a qualidade da chamada, já que sua implementação da pilha RTP/RTCP não é completa.

Desta forma, foi realizada uma série de alterações na biblioteca OpenH323 para implementar o mecanismo de cálculo de RTT em sua pilha RTP/RTCP, conforme especificado na RFC 3550 [39].

4.5.1 Mecanismo de Cálculo de RTT do RTP

O cálculo do RTT é baseado em dois tipos de pacotes RTCP: SR (*Sender Report*) e RR (*Receiver Report*). Estes pacotes são usados pelos participantes da sessão para informar estatísticas relativas à transmissão/recepção do fluxo de dados. Na Ilustração 19 são mostrados os instantes de envio e recepção destes pacotes.

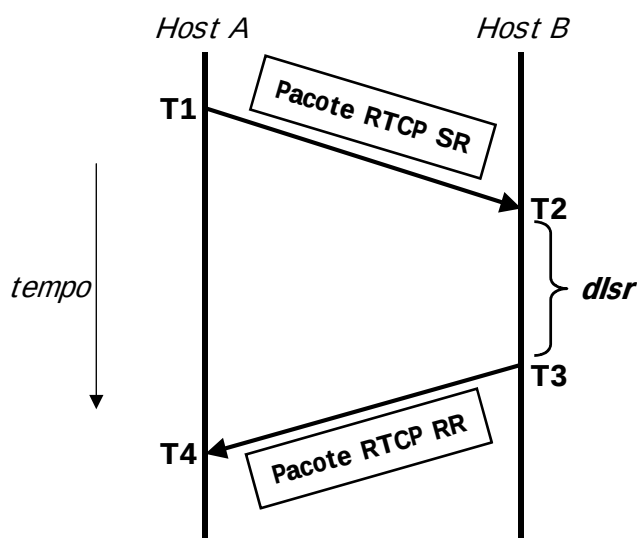


Ilustração 19. Cálculo de RTT

No instante T1, dentro do pacote RTCP SR, são enviados os 32 bits centrais do *timestamp* do *host A*, que são armazenados pelo *host B*, ao receber o pacote RTCP SR no instante T2. No próximo pacote RTCP RR enviado pelo *host B* para o *host A*, o campo LSR (*Last SR timestamp*) é preenchido com o valor recebido em T2, e no campo DLSR (*Delay since last SR*) é colocado o tempo decorrido entre T2 e T3. Finalmente em T4, o *host A* é capaz de estimar o tempo de ida e volta de um pacote entre o *host A* e o *host B*, através da seguinte expressão:

$$RTT = (T4 - T1) - DLSR \quad (25)$$

Todavia, a biblioteca OpenH323 não preenche os campos LSR e DLSR do pacote RR. A seguir, as alterações realizadas na biblioteca OpenH323 para corrigir estas deficiências são descritas.

4.5.2 Alteração da Biblioteca OpenH323 para Cálculo do RTT

A biblioteca OpenH323 foi toda desenvolvida em C++ e utiliza classes definidas na biblioteca PWLib [55] para dar total portabilidade de código entre os sistemas operacionais Unix-like (Linux, FreeBSD, etc.) e MS Windows.

Para armazenar os instantes T1, T2, T3 e T4 foram criadas instâncias da classe *PTime* definida na PWLib. A classe *PTime* define as funções de data e hora, com diversos métodos para manipulação de seus dados. Ao se criar uma nova instância, o objeto é inicializado com a hora corrente do sistema.

O código da OpenH323 preenche corretamente o campo NTP *timestamp* quando envia o pacote RTCP SR, no instante T1. O instante T2 acontece quando o evento de recepção de um pacote RTCP SR ocorre. Neste momento são armazenados os 32 bits centrais do NTP *timestamp* recebido na variável `ntp_lsr`, desprezando-se os 16 bits mais significativos e os 16 bits menos significativos dos 64 bits usados pelo *timestamp*, conforme mostrado na Ilustração 20. O instante da recepção do pacote RTCP SR é armazenado na variável T2, que será utilizado posteriormente para o cálculo do DLSR.

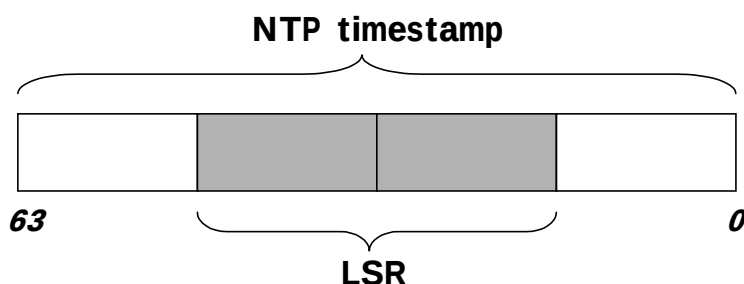


Ilustração 20. 32 bits centrais do LSR

Como pode ser observado no trecho de código da Ilustração 21, a linha 10 salva na variável `ntp_lsr` os 32 bits centrais do NTP *timestamp* recebido no pacote SR. Na linha 13, o instante de chegada do pacote SR é salvo na variável T2, para posteriormente ser utilizado no cálculo do DLSR.

No instante T3, o tempo decorrido desde o último instante T2 é calculado e armazenado em DLSR (linha 12 da Ilustração 22). É importante observar que o DLSR definido na RFC 3550 [39] utiliza 1/65535 segundos como unidade de tempo. O método

GetTimestamp() do objeto *PTime* retorna o valor armazenado em microssegundos, sendo necessário fazer a conversão da unidade de tempo. É interessante observar que no código original, as variáveis referentes aos campos LSR e DLSR do pacote RR eram preenchidas com zero (linhas 05 a 07 da Ilustração 22).

```

01 RTP_Session::SendReceiveStatus
02 TP_Session::OnReceiveControl(RTP_ControlFrame & frame)
03 {
04     ...
05     Switch (frame.GetPayloadType()) {
06         case RTP_ControlFrame::e_SenderReport :
07             ...
08             Const RTP_ControlFrame::SenderReport & sr =
09                 *(const RTP_ControlFrame::SenderReport *)payload;
10             ntp_lsr = (sr.ntp_sec << 16) | (sr.ntp_frac >> 16);
11             ...
12             PTime now;
13             T2 = now;
14         }
15     ...
16 }

```

Ilustração 21. Alteração do código da OpenH323 no instante T2

```

01 void RTP_Session::AddReceiverReport
02 RTP_ControlFrame::ReceiverReport & receiver)
03 {
04     ...
05     // The following have not been calculated yet.
06     receiver.lsr = 0;
07     receiver.dlsr = 0;
08     ...
09     PTime now;
10     t3 = now;
11     receiver.lsr = ntp_lsr;
12     receiver.dlsr = ((PInt64)
13         (t3.GetTimestamp()-t2.GetTimestamp()) << 16) / 1000000;
14     ...
15 }

```

Ilustração 22. Alteração do código da OpenH323 no instante T3

O momento de cálculo de RTT é realizado no instante de recepção de um pacote RTCP RR (T4). Neste ponto foi necessário fazer uma correção no código original da biblioteca: a variável *report->delay*, responsável por armazenar o DLSR em milissegundos, recebia o valor convertido de forma errada, como pode ser observado na linha 03 do trecho de código apresentado na Ilustração 23. A linha 05 mostra como ficou o código após a correção realizada.

```

01  ...
02  // Código original
03  //  report->delay = ((Pint64)rr->dlsr << 16)/1000;
04  // Código alterado
05  Report->delay = (Pint64)(rr->dlsr*1000) >> 16;
06  ...

```

Ilustração 23. Conversão de DLSR de 1/65535 segundos para milissegundos

Para se obter o *timestamp* do instante T1 associado a este pacote RTCP RR recebido, é necessário recuperar os bits mais significativos do *timestamp* original, usando a parte mais alta do próprio instante T4, como mostrado na Ilustração 24. Os 16 bits menos significativos são desprezados, gerando diferença no cálculo do RTT na ordem de microssegundos, que não afetam a avaliação da qualidade da voz.

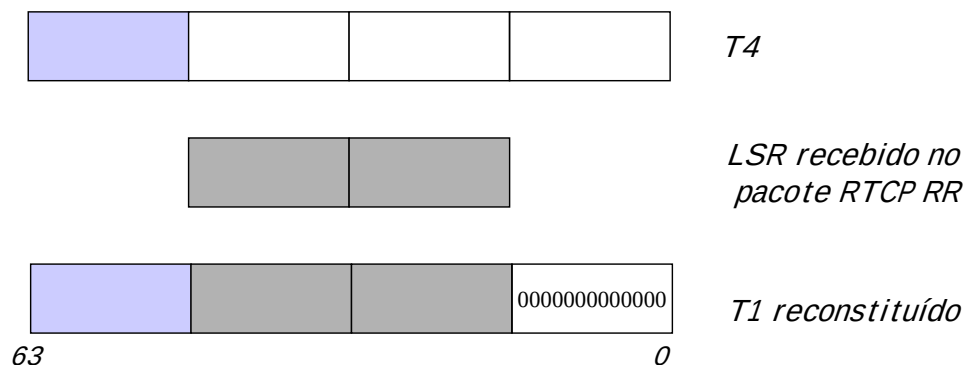


Ilustração 24. Recuperação do timestamp do instante T1

Finalmente, o cálculo do RTT é realizado conforme o trecho de código apresentado na Ilustração 25.

```

01 static RTP_Session::ReceiverReportArrayBuildReceiverReportArray
02     (const RTP_ControlFrame & frame, PINDEX offset)
03 { ...
04     PTime now;
05     T4 = now;
06     Report->delay = (PInt64)(rr->dlsr*1000) >> 16;
07
08     DWORD rtt_calculado;
09     Pint64 t1_refeito;
10     DWORD sec, frac;
11     DWORD parte_alta;
12     if (rr->lsr) {
13         // Pega a parte baixa do LSR e remonta o frac do ntp
14         Frac = (rr->lsr & 0xffff) << 16;
15         //Pega parte alta de LSR e adiciona a parte obtida de t4
16         parte_alta = (t4.GetTimeInSeconds() +
17             SecondsFrom1900to1970) & 0xffff0000;
18         sec = rr->lsr >> 16 | parte_alta;
19         // Calcula o t1 em microsegundos desde 01/01/1970
20         t1_refeito = (Pint64)(sec-SecondsFrom1900to1970)*1000000 +
            frac/4294;
21         rtt_calculado = (Pint64)(t4.GetTimestamp() -
22             t1_refeito)/1000 - (Pint64)(rr->dlsr*1000) >> 16);
23     }

```

Ilustração 25. Cálculo do RTT

4.6 Coletando os Parâmetros

Basicamente, foram realizadas chamadas para o cálculo da qualidade da ligação em dois módulos da biblioteca OpenH323: `rtp.cxx` e `channels.cxx`. No módulo `rtp.cxx` foi criada uma chamada que é executada a cada evento de chegada de pacote de dado RTP (Ilustração 26), permitindo obter informações como número de pacotes perdidos, *jitter*, pacotes descartados no *buffer* de compensação de *jitter*, tamanho do *buffer* de compensação de *jitter* e RTT. Estes dados são gravados em um arquivo temporário, que depois é utilizado pelo PM.

No módulo `channels.cxx`, após a sessão RTP ser estabelecida, é gerada uma chamada para armazenar, também no arquivo temporário, os parâmetros de identificação do codificador de voz (Ilustração 27), que também são utilizados pelo PM.

```

01 RTP_Session::SendReceiveStatus RTP_Session::OnReceiveData
02 (const RTP_DataFrame & frame)
03 {
04     ...
05     PString packets = packetsReceived;
06     PString lost = packetsLost;
07     PString tooLate = GetPacketsTooLate();
08     PString jitter = jitterLevel >> 7;
09     PString bufferSize = GetJitterBufferSize()/8;
10     PString rtt = rtt_calculado;
11     LogEvent(packets+" "+lost+" "+tooLate+" "+jitter+"
12 "+bufferSize+" "+rtt);
13     ... }

```

Ilustração 26. Coletando parâmetros da OpenH323

```

01 BOOL H323_RTPChannel::Open()
02 {
03     ...
04     if (GetDirection() == IsReceiver)
05     {
06         const OpalMediaFormat & mediaFormat =
            codec-> GetMediaFormat();
07         DWORD codecFrameRate = codec->GetFrameRate();
08         unsigned framesinpacket;
09         // try to discovery how many frames have in the packet
10         ( capability->GetRxFramesInPacket() > 6) ? framesinpacket = 1 :
11         framesinpacket = capability->GetRxFramesInPacket();
12         PString framesize(codecFrameRate/mediaFormat.GetTimeUnits());
13         PString framesinpacketstr(framesinpacket);
14         rtpSession.LogEvent("CODEC "+mediaFormat+"
15 "+framesize+" "+framesinpacketstr);
16     }
17     ...
18 }

```

Ilustração 27. Coletando informações referentes ao codificador de voz

4.7 Integração da VQuality com Telefones IP

As classes *CallQualitySummary* e *VQCDR* implementam um VGM. Ao instanciar um objeto da classe *CallQualitySummary*, é esperado que se passe um objeto do tipo *ExtendedEmodelBase* por parâmetro em seu construtor. Como a classe *ExtendedEmodelBase* generaliza o VQEM, pode-se determinar qual VQEM será utilizado para o cômputo da

qualidade da voz, em função do VQEM passado por parâmetro na construção da *CallQualitySummary*.

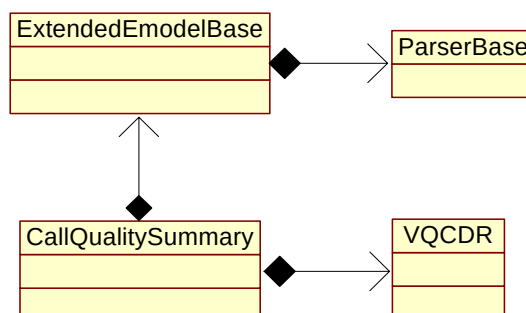


Ilustração 28. Interação entre as classes CallQualitySummary, ExtendedEmodelBase e ParserBase

Um objeto do tipo *ExtendedEmodelBase*, por sua vez, necessita de uma referência a um objeto do tipo *ParserBase*, que generaliza o PM. Dessa forma, pode-se determinar o tipo de PM que o VQEM irá utilizar para obtenção de dados, passando o PM a ser utilizado na construção do *ExtendedEmodelBase*. A Ilustração 28 apresenta o diagrama de classes UML que resume este mecanismo.

```

01  OpenH323LogParser pm (MODIFIED_OPEN_H323_CALL_QUALITY_LOG);
02  UFRJ_UFAM_ExtendedEmodel vqem (&pm);
03  CallQualitySummary vgm (&vqem);
04  vgm.setProtocol(H323_PROTOCOL);
05  vgm.setID((char *) confIDString);
06  vgm.setSource((char *) remoteIPString);
07  vgm.setDestination((char *) localIPString);
08  vgm.setUsername((char *) userString);
09  vgm.setLocalAlias( (char *) localNumberString);
10  vgm.setRemoteAlias( (char *) remoteNumberString);
11  (IsCaller) ? vgm.setDirection(CALLER_DIRECTION)
12              : vgm.setDirection(CALLED_DIRECTION);
13  vgm.analyze();
14  vgm.setVQCDRServer((char *) vqcdeServer);
15  vgm.sendVQCDR();
  
```

Ilustração 29. Integração do OpenPhone com a VQuality

O trecho de código da Ilustração 29 foi usado para incorporar a VQuality ao OpenPhone. O cliente resultante, capaz de realizar a avaliação da qualidade de voz recebida e

gerar VQCDR, foi chamado de VQOpenPhone. Na linha 01, um objeto *OpenH323LogParser* é instanciado (pm), representando um PM. Na linha 02, um VQEM é criado instanciando-se um objeto *UFRJ_UFAM_ExtendedEmodel* (vqem), que utiliza o objeto *parser* criado na linha 01. Na linha 03, é instanciado um VGM através de um objeto *CallQualitySummary* (vgm) e o objeto vqem é passado como referência. Nas linhas de 04 a 12, o VGM (objeto vgm) coleta os parâmetros referentes à chamada e identificação dos terminais envolvidos, que serão enviados no VQCDR e não são computados pelo VQEM. Na linha 13, a análise de qualidade da chamada é acionada. Na linha 14, é configurado o endereço do servidor VQCDR Server e, por fim, na linha 15 o VQCDR é enviado.

4.7.1 VQOpenPhone

A Ilustração 30 mostra um terminal H.323 VQOpenPhone após o fim de uma chamada. No visor é apresentado o nível de qualidade da voz recebida em MOS (*Mean Opinion Score*) [3].

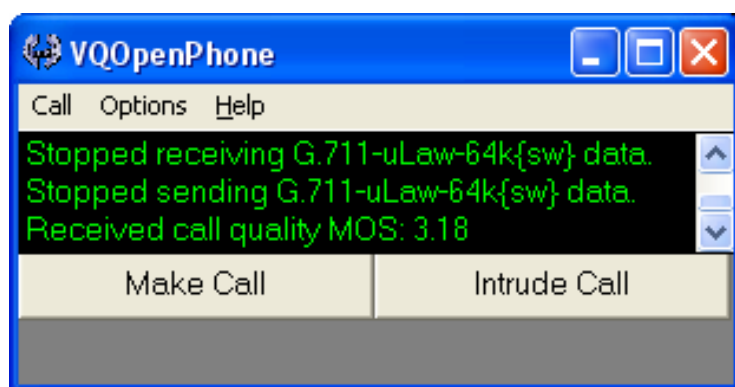


Ilustração 30. VQOpenPhone

A integração da VQuality ao OpenPhone foi simples e direta. Como a biblioteca OpenH323 já estava integrada à VQuality, foi necessário apenas coletar as informações de identificação da chamada e dos terminais envolvidos, necessárias para o preenchimento do VQCDR e fazer simples chamadas a biblioteca VQuality no código do OpenPhone.

4.7.2 VQMeeting

Embora a OpenH323 e a VQuality sejam multiplataforma e possuam versões tanto para o sistema Windows quanto para sistemas baseados em Unix, o OpenPhone, que é desenvolvido a partir destas duas bibliotecas, não é multiplataforma e possui apenas versões para o sistema Windows.

Objetivando atender os usuários Unix e demonstrar a capacidade de operação em arquiteturas diferenciadas, o GnomeMeeting [56], terminal H.323 baseado na OpenH323 e voltado para ambientes Unix foi integrado à VQuality resultando no VQMeeting (Ilustração 31), que hoje possui versões disponíveis para Linux e FreeBSD.



Ilustração 31. VQMeeting

O processo de integração da VQuality ao GnomeMeeting foi simples e direto como o processo de integração do OpenPhone. E assim como o VQOpenPhone, o VQMeeting aderiu a capacidade de avaliação de qualidade de voz, emissão de VQCDR e apresentação do MOS referente à qualidade de voz recebida.

4.8 Voice Quality Log

Os indicadores de qualidade fornecidos pelo VQCDR (ver seção 5.2.1) 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 de toda a duração da chamada.

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. Os campos de cada uma destas linhas estão descritos no Quadro 3, Quadro 4, Quadro 5 e Quadro 6, respectivamente. Maiores detalhes de alguns campos apresentados nestes quadros podem ser encontrados na seção 5.2.

Quadro 3. Linha de Identificação de Codec

Nº.	Campo	Descrição
01	codecString †	Descrição textual do <i>codec</i> .
02	framesPerPacket *	Unidades de quadros de voz empacotados no pacote IP.
03	frameSize **	Tamanho do quadro de voz em milisegundos.

04	codecDelay **	Atraso de codificação e empacotamento do pacote de voz em milisegundos.
----	---------------	---

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

* Valor inteiro.

** Valor real com duas casas decimais.

Quadro 4. Linha de Indicadores de Qualidade

Nº.	Campo	Descrição
01	time	Marcação de tempo no formato “HH:MM:SS.mmm”, sendo “H” hora, “M” minuto, “S” segundo e “m” microssegundos.
02	partialPacketsLost *	Número de pacotes perdidos desde a marcação anterior. No caso da primeira Linha de Identificação de Codec, representa o número de perdas desde o início da chamada.
03	partialPacketsDiscarded *	Número de pacotes descartados no <i>buffer</i> de compensação de <i>jitter</i> desde a marcação anterior. No caso da primeira Linha de Identificação de Codec, representa o número de descartes desde o início da chamada.
04	netDelay *	Valor instantâneo de atraso na rede em milisegundos.
05	owd **	Valor instantâneo de atraso total em um sentido em milisegundos.
06	jitter **	<i>Jitter</i> no formato RTP [39].
07	jitterBufferSize **	Tamanho instantâneo do <i>buffer</i> de compensação de <i>jitter</i> em milisegundos.
08	Id **	Valor instantâneo de <i>Delay Impairment (Id)</i> .
09	Ieav **	Valor instantâneo de <i>Equipment Impairment (Ie)</i> .
10	Rfactor **	Valor instantâneo de <i>R-Factor</i> .
11	MOS **	Valor instantâneo de MOS.

* Valor inteiro.

** Valor real com duas casas decimais.

Quadro 5. Linha de Sumário

Nº.	Campo	Descrição
01	model *	Modelo utilizado para análise da qualidade de voz. Ver Quadro 10.
02	avgNetDelay **	Atraso médio da rede. Equivalente ao campo 19 do VQCDR. Ver Nota 1 do Quadro 7.
03	avgJitterBufferDelay **	Tamanho médio do <i>buffer</i> de compensação de <i>jitter</i> em milisegundos. Equivalente ao campo 20 do VQCDR. Ver Nota 2 do Quadro 7.
04	avgJitter **	<i>Jitter</i> médio em milisegundos. Equivalente ao campo 21 do VQCDR. Ver Nota 3 do Quadro 7.
05	codecDelay **	Atraso de codificação e empacotamento do pacote de voz em milisegundos.
06	packetsReceived *	Total de pacotes recebidos.
07	packetsLost *	Total de pacotes perdidos.
08	packetsDiscarded *	Total de pacotes descartados no <i>buffer</i> de compensação de <i>jitter</i> .
09	gapLossDensity **	Densidade de perda em momentos de perdas isoladas.
10	burstLossDensity **	Densidade de perda em momentos de perdas em rajada.
11	IdEndOfCall **	<i>Id</i> ao final da chamada.
12	IeEndOfCall **	<i>Ie</i> final da chamada.
13	RfactorEndOfCall **	Fator R final da chamada.
14	MOSEndOfCall **	MOS final da chamada.

* Valor inteiro.

** Valor real com duas casas decimais.

Quadro 6. Linha de Identificação de Chamada

Nº.	Campo	Descrição
01	protocol	Protocolo de sinalização VoIP utilizado. Ver Quadro 8.
02	ID	Identificação da chamada.
03	username	Identificação do usuário.

04	localAlias	Identificador do cliente VoIP local. Pode ser um número E.164 ou uma URI.
05	remoteAlias	Identificador do cliente VoIP remoto. Pode ser um número E.164 ou uma URI.
06	direction	Identificação do originador da chamada. Ver Quadro 11.
07	mediaSource	Endereço (IP ou <i>hostname</i>) do cliente VoIP remoto, originador da mídia.
08	mediaDestination	Endereço (IP ou <i>hostname</i>) do cliente VoIP local.
09	duration	Duração da chamada em segundos.

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

* Valor inteiro.

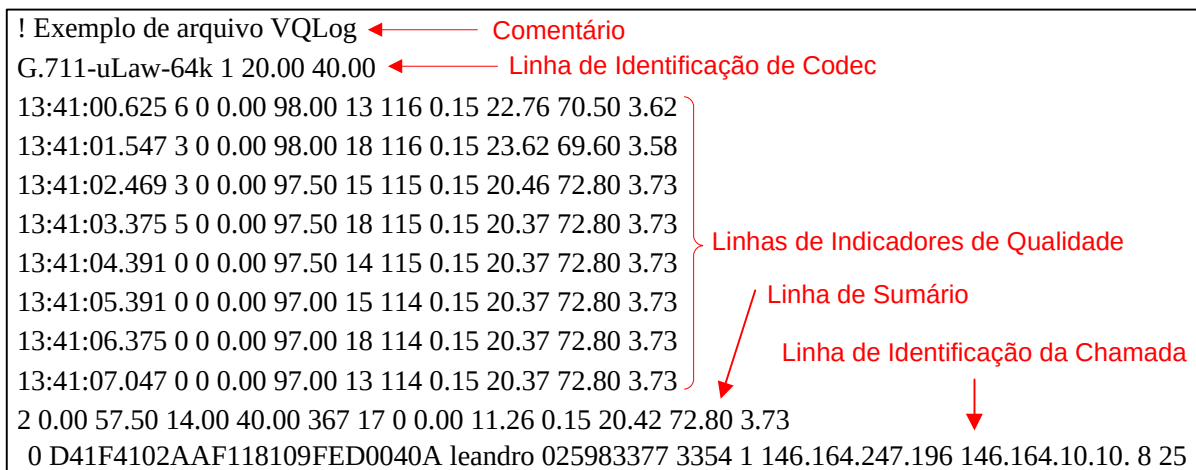
** Valor real com duas casas decimais.

Cada linha é composta por uma série de campos delimitados por um caractere de espaço. A ordem de cada campo deve ser estritamente obedecida e caso não seja possível obter o valor de algum campo, deve-se preenchê-lo com o caractere “#”.

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.

Também é possível que sejam acrescentados comentários a um arquivo VQLog. Para tanto, é necessário que se preceda o comentário pelo caractere “!”.

Um exemplo de um arquivo VQLog pode ser visualizado na Ilustração 32. A primeira linha deste arquivo VQLog é um comentário. Note que esta linha é iniciada por um caractere “!”. A segunda é a *Linha de Identificação de Codec*, as oito seguintes são *Linhas de Indicadores de Qualidade*, a penúltima é a *Linha de Sumário*, e, por fim, a última é a *Linha de Identificação de Chamada*.



```

! 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 D41F4102AAF118109FED0040A leandro 025983377 3354 1 146.164.247.196 146.164.10.10. 8 25
  
```

! Exemplo de arquivo VQLog ← Comentário

G.711-uLaw-64k 1 20.00 40.00 ← Linha de Identificação de Codec

Linhas de Indicadores de Qualidade

Linha de Sumário

Linha de Identificação da Chamada

Ilustração 32. Exemplo de arquivo VQLog

4.9 Voice Quality Plot

O VQPlot (*Voice Quality Plot*) 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, 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. Estas informações são visualizadas através de duas telas, uma para apresentação das saídas referentes ao Modelo E e dados sumarizados da qualidade da chamada, e outra para apresentação de dados estatísticos que influenciam o cômputo do modelo.

A Ilustração 33 e a Ilustração 34 apresentam a saída do VQPlot referente à uma chamada realizada por um *notebook* com interface sem fio (*wireless*).

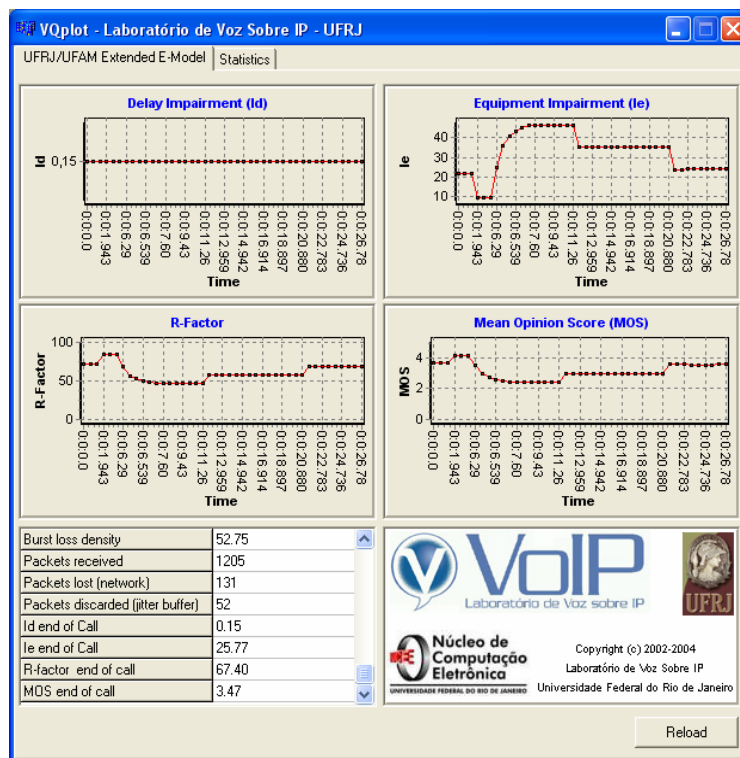


Ilustração 33. Qualidade da chamada em relação ao tempo

A Ilustração 33 apresenta a tela do VQPlot referente às saídas do Modelo E e aos dados sumarizados da qualidade da chamada. Nesta tela, podemos visualizar os seguintes gráficos descritos a seguir:

- *Delay Impairment (Id)*: representa a variação de *Id*. Permite a análise da degradação da interatividade da conversa ao longo da chamada.
- *Equipment Impairment (Ie)*: representa a variação de *Ie*. Permite a análise de como as perdas de pacotes na rede e os descartes de pacotes no *buffer* de compensação de *jitter* influenciaram a degradação da qualidade ao longo da chamada.

- *R-Factor*: representa a saída direta do Modelo E. Permite a análise da variação da qualidade através da combinação dos fatores de degradação *Id* e *Ie*.
- *Mean Opinion Score (MOS)*: também representa a saída do Modelo E, *R-Factor*, convertida para a escala MOS.

Ainda nesta tela, são apresentados todos os campos referentes à *Linha de Identificação de Codec* e à *Linha de Sumário*. Estas informações são úteis para verificar os parâmetros referentes ao *codec*, tais como *codec* utilizado na chamada, quantidade de quadros encapsulados por pacote IP, entre outras. Também são apresentadas informações sumarizadas relativas à qualidade da chamada, tais como fator de degradação *Ie(final da chamada)* e *MOS(final da chamada)*, entre outras.

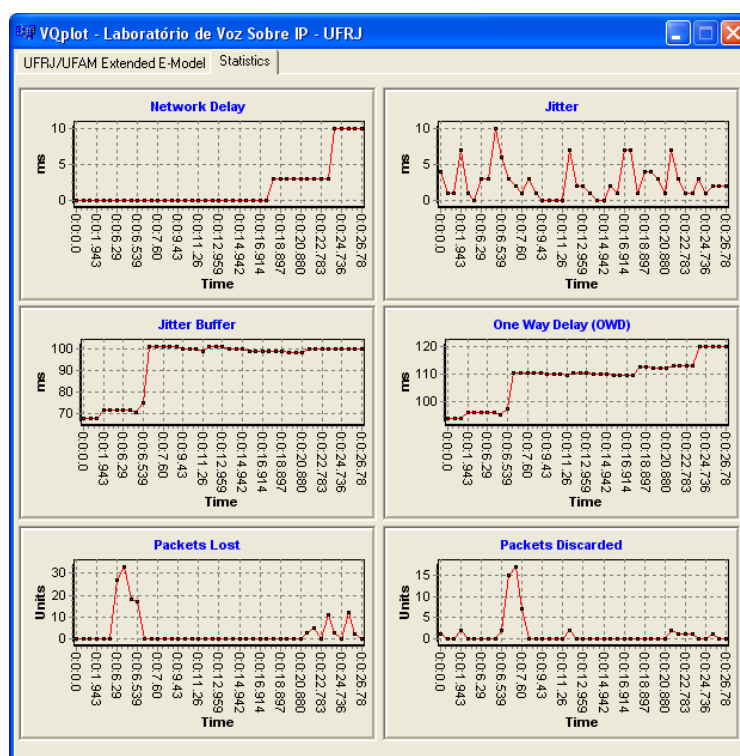


Ilustração 34. Estatísticas da chamada em relação ao tempo

A Ilustração 34 apresenta a tela do VQPlot referente aos dados estatísticos que influenciam no cômputo do Modelo E. Nesta tela, podemos visualizar os gráficos descritos a seguir:

- *Network Delay*: representa a variação do atraso na rede ao longo da chamada. É um dos fatores que influencia no cômputo de *Id*. Permite a detecção de atrasos excessivos na rede, que podem influenciar na perda de interatividade da conversação.
- *Jitter*: representa a variação de *jitter* ao longo da chamada segundo o cálculo de *jitter* oferecido pela pilha RTP/RTCP [39]. Auxilia na detecção de congestionamento da rede e de problemas de concorrência do tráfego de voz com outros tráfegos. Também auxilia na determinação de parâmetros relativos ao *buffer* de compensação de *jitter*, que podem ser configurados pelos usuários. Algumas aplicações permitem que os usuários determinem um tamanho mínimo e máximo de *buffer* a ser utilizado, ou até mesmo um valor fixo, caso a aplicação não trabalhe com *buffer* dinâmico.
- *Jitter Buffer*: representa a variação do tamanho do *buffer* de compensação de *jitter* ao longo da chamada. No caso de aplicações que não trabalhem com *buffer* dinâmico, este valor é mantido constante. No caso de aplicações que trabalhem com *buffer* dinâmico, esse valor permite analisar se o algoritmo para determinação do tamanho de *buffer* tem sido eficiente frente a variações de *jitter* e descartes de pacotes no *buffer*. É outro fator que influencia no cômputo de *Id*.

- *One Way Delay (OWD)*: combina os fatores relativos aos atrasos inseridos pelo *codec*, pela rede e pelo *buffer* de compensação de *jitter* (ver seção 2.2.2). Representa o atraso total em um sentido, da codificação do áudio no lado emissor até sua reprodução no lado do receptor. Este valor influencia diretamente no cálculo do fator I_d e permite a detecção de atrasos excessivos, que influenciam na perda de interatividade da conversa.
- *Packets Lost*: representa a perda de pacotes na rede ao longo da chamada. É outra informação que auxilia na detecção de congestionamento da rede e problemas de concorrência do tráfego de voz com outros fluxos. Este fator influencia diretamente no cálculo do fator I_e .
- *Packets Discarded*: representa o descarte de pacotes no *buffer* de compensação de *jitter* ao longo da chamada. Juntamente com as informações de *jitter*, auxilia na determinação dos parâmetros de configuração do *buffer* a ser utilizado pela aplicação. Também auxilia na detecção de congestionamento da rede e problemas de concorrência do tráfego de voz com outros tráfegos. Este fator também influencia diretamente no cálculo do fator I_e .

Capítulo 5

ARQUITETURA DE MONITORAÇÃO DE CHAMADAS TELEFÔNICAS IP

A arquitetura de monitoração proposta baseia-se na coleta de VQCDRs (Voice Quality CDR) emitidos por telefones IP e *gateways* de voz a fim de se realizar um levantamento de estatísticas de qualidade das chamadas realizadas em um ambiente VoIP (Ilustração 35). Estes VQCDRs são coletados por uma entidade central, denominada VQCDR Server, que possui a responsabilidade de interpretar, autenticar e encaminhar os VQCDRs coletados para serem armazenados em uma base de dados. Para realizar a autenticação, o VQCDR Server deve interagir com alguma aplicação ou acessar alguma base de dados que permita verificar a legitimidade do VQCDR.

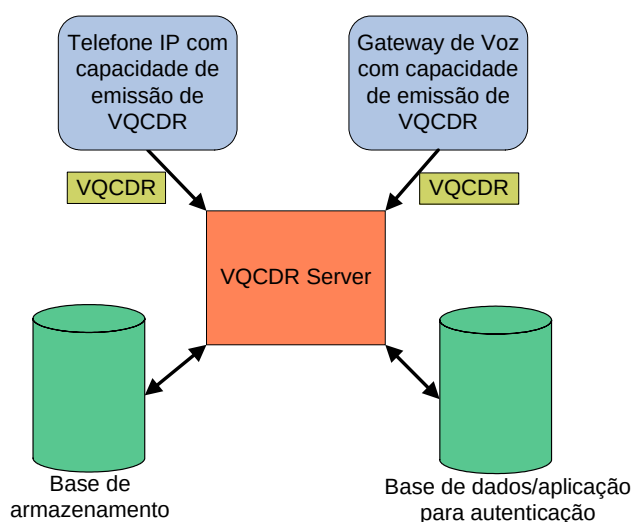


Ilustração 35. Arquitetura de monitoração de chamadas telefônicas IP

A arquitetura especifica apenas o formato e o mecanismo de envio do VQCDR, e as funcionalidades (ou módulos, ver seção 5.3) que devem ser implementadas pelo VQCDR

Server. A base de armazenamento e a base de dados/aplicação para autenticação não são especificadas pela arquitetura, a fim de torná-la mais versátil e flexível.

Neste capítulo são apresentadas as especificações referentes ao VQCDR e ao VQCDR Server, bem como são mostradas considerações sobre a disponibilização de telefones IP e *gateways* de voz com capacidade de emissão de VQCDR.

5.1 Telefones IP e Gateways de Voz com Capacidade de Emissão de VQCDR

A biblioteca VQuality oferece todos os recursos necessários para que telefones IP e *gateways* de voz agreguem a funcionalidade de emissão de VQCDR. Entretanto, é importante notar que é preciso ter acesso ao código fonte de tais elementos para que a VQuality seja integrada à seus respectivos códigos. Para facilitar este processo de integração, é que a VQuality foi projetada para ser flexível e multiplataforma.

O VQOpenPhone e o VQMeeting, terminais H.323 em uso no serviço fone@RNP, possuem capacidade de emissão de VQCDR e, no futuro, clientes SIP de código aberto e o *gateway* de voz Asterisk serão integrados à VQuality para que também agreguem esta funcionalidade. As seções 5.6 e 5.7 apresentam considerações sobre a integração da VQuality à clientes SIP e ao Asterisk, respectivamente.

O *gateway* de voz Cisco [18], também em uso no serviço, possui capacidade de emissão de CDR, que embora não forneça informações de qualidade detalhadas como o VQCDR, oferece indicadores de qualidade baseados no Modelo E original, que também auxiliam no levantamento de estatísticas de qualidade das chamadas [25].

No serviço fone @RNP são também utilizados telefones IP com *hardware* proprietário [57,58], que não possuem capacidade de emissão de CDR e nem de avaliação de qualidade.

Não há expectativa então, em curto prazo, de coleta de qualidade referente a tais equipamentos, já que estes não possuem código aberto.

Entretanto, quando o processo de integração da VQuality com clientes SIP e com o Asterisk estiver concluído, quase todas as chamadas realizadas no ambiente do serviço `fone@RNP` poderão ser avaliadas em ambos os sentidos da comunicação. As exceções referem-se às chamadas estabelecidas estritamente entre dois telefones IP baseados em *hardware* proprietário. É importante notar que quando ao menos um dos pontos da chamada possui a capacidade de emissão de CDR com indicação de qualidade, é possível avaliar a chamada em um dos sentidos da comunicação. Mas nada impede que, no futuro, parcerias com fabricantes possibilitem a disponibilização de equipamentos proprietários com capacidade de emissão de VQCDR, possibilitando assim, que todas as chamadas sejam sempre avaliadas em ambos os sentidos da comunicação.

5.2 Voice Quality Call Detail Record (VQCDR)

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 Detail 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.

5.2.1 O Formato do VQCDR

O formato do VQCDR versão 1 e todos seus atributos são apresentados no Quadro 7.

Quadro 7. 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. Ver Quadro 8.
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 Quadro 10.
07	codec	1 *	Codificador de voz utilizado. Ver Quadro 9.
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 (Id)</i> 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 (Ie)</i> final da chamada. Indicador de degradação associado à codificadores de alta compressão, perda de pacotes na rede e descarte no <i>buffer</i> de compensação de <i>jitter</i> . Este valor é multiplicado por 100.
12	MOSEndOfCall	2 *	<i>Mean Opinion Score (MOS)</i> final da chamada. Representa a percepção humana da qualidade. Este valor é 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 *	Percentagem de pacotes descartados no <i>buffer</i> de compensação de <i>jitter</i> , desde o início da recepção. Este valor é multiplicado por 100.
16	lossRate	2 *	Percentagem 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 <i>buffer</i> de compensação de <i>jitter</i> em microssegundos. Ver Nota 2.
21	avgJitter	4 *	<i>Jitter</i> 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 <i>buffer</i> de compensação de <i>jitter</i> .
26	mediaSource	256 †	Endereço (IP ou <i>hostname</i>) do cliente remoto, originador da mídia.
27	mediaDestination	256 †	Endereço (IP ou <i>hostname</i>) 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 Quadro 11.

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 4.8).

* 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 [39]. Caso não haja 16 variações disponíveis, esta média é calculada apenas com base nas últimas variações disponíveis.

É importante ressaltar que o Quadro 8, o Quadro 9 e o Quadro 10 possuem valores pré-estabelecidos. Entretanto, novos valores podem ser especificados para suporte a novos protocolos de sinalização, novos *codecs* e novos modelos objetivos de avaliação de qualidade, se necessário.

Com a criação de novos modelos de avaliação e com a adição de novas funcionalidades, campos do VQCDR talvez venham a ser alterados, inseridos, ou até mesmo, excluídos. Para tornar o VQCDR versátil e extensível, o campo version é utilizado para identificar a versão do formato de VQCDR. Assim, novas versões podem ser projetadas e funcionalidades podem ser alteradas, incluídas ou excluídas.

Os campos protocol, IDSize e ID, por sua vez, possibilitam a independência de protocolo de sinalização. O campo protocol é utilizado para identificação do protocolo, o

campo IDSize especifica o tamanho do código de identificação de chamada, que pode ser diferente para cada protocolo, e o campo ID é utilizado para armazenar o código de identificação de chamada propriamente dito.

Quadro 8. Protocolos de sinalização

Protocolo	Valor
H.323	0
SIP	1
MGCP	2

Quadro 10: Modelos objetivos de avaliação

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

Quadro 11: Identificação de originador

Direção	Valor
chamado	0
chamador	1

Quadro 9: 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.729 ^a	10
GSM FR (6.10)	11

O campo username identifica o usuário VoIP emissor do VQCDR, o campo model identifica o modelo de avaliação utilizado e os campos codec, frameSize, framesPerPacket e codecDelay são responsáveis pela identificação de parâmetros relativos ao codificador utilizado na chamada.

Os campos IdEndOfCall, IeEndOfCall, MOSEndOfCall e RfactorEndOfCall fornecem as saídas do Modelo E para a avaliação da qualidade da chamada. Os campos gapLossDensity, burstLossDensity, discardRate, lossRate, packetsLost, packetsReceived e packetsDiscarded oferecem indicadores de perdas de pacotes na rede, perdas de pacotes no

buffer de compensação de *jitter* e total de pacotes recebidos, informando ainda a densidade de perdas em rajada e a densidade perdas isoladas.

Indicadores de atraso médio da rede, tamanho médio do *buffer* de compensação de *jitter* e *jitter* médio são oferecidos pelos campos `avgNetDelay`, `avgJitterBufferDelay` e `avgJitter`. Estes indicadores podem não representar valores médios para a duração total da chamada, já que são calculados com base nas últimas variações de atraso, tamanho do *buffer* e *jitter*, respectivamente (ver Nota 1, Nota 2 e Nota 3 do Quadro 7). Entretanto, representam informações estatísticas importantes para a análise do sistema como um todo, indicando tendências de atraso e *jitter* da rede e como os *buffers* das aplicações reagem frente a estas tendências. É importante ressaltar que as últimas variações de atraso e *jitter* são mais representativas para o sistema, já que se espera que se estes valores aumentarem em demasia, os usuários provavelmente ficarão insatisfeitos com a qualidade e encerrarão a chamada, e caso estes valores se estabilizem em índices aceitáveis, a chamada será finalizada normalmente ao final da conversa.

Indicações de origem e destino da mídia são fornecidas através dos campos `mediaSource` e `mediaDestination`, que podem ser endereços IP ou *hostnames*. É importante comentar que o destino da mídia será sempre o telefone IP emissor do VQCDR e que a origem da mídia pode ser o outro telefone IP envolvido na chamada (caso a mídia tenha sido estabelecida ponto-a-ponto), ou um *proxy* RTP (caso a mídia tenha sido estabelecida via um *proxy* de mídia).

O campo `localAlias` identifica o telefone emissor do VQCDR, e campo `remoteAlias` identifica o telefone IP remoto envolvido na chamada. Estes campos podem ser números E.164 ou URIs. O campo `direction` indica qual dos telefones IP iniciou a chamada e o campo `duration` indica a duração da chamada.

O mecanismo de envio de arquivo VQLog referente à chamada (ver seção 4.8) é fornecido pelo campo VQLogSize, que indica o tamanho do arquivo, e pelo campo VQLog, que é o conteúdo do arquivo VQLog propriamente dito.

Os campos appExtensionSize e appExtension são reservados para uso futuro. Eles foram inseridos para que os telefones IP possam enviar ao VQCDR Server informações específicas de aplicação.

5.3 Voice Quality CDR Server

O Voice Quality CDR Server (VQCDR Server) é a entidade da arquitetura de monitoração responsável por coletar, autenticar e encaminhar os VQCDRs a uma base de armazenamento. O VQCDR Server é composto por três módulos (Ilustração 36): Collector Module, Authenticator Module e Storage Module.

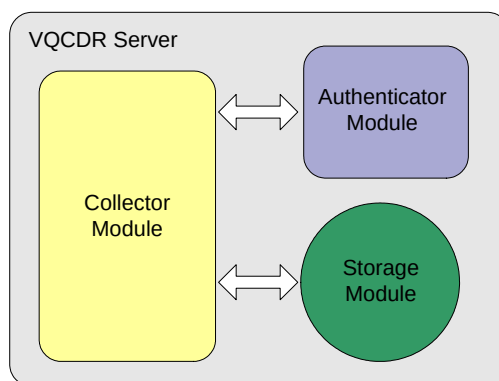


Ilustração 36. Arquitetura do VQCDR Server

5.3.1 Collector Module (CM)

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

5.3.2 Authenticator Module (AM)

O Authenticator Module (AM) é 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.

5.3.3 Storage Module (SM)

O Storage Module (SM) é 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.

O RADIUS é um protocolo IETF padrão para autenticação, autorização e contabilidade de serviços de conexão através de acesso discado. Entretanto, tem sido utilizado também como protocolo padrão de fato para outros serviços Internet, como por exemplo, contabilização de serviços VoIP [25]. A contabilidade baseada em RADIUS permite a independência do banco de dados utilizado. Assim, ao invés de usar um mecanismo determinado para acesso a um banco de dados específico, as aplicações que geram informações a serem contabilizadas precisam apenas implementar o mecanismo de acesso ao RADIUS, deixando por conta do RADIUS a interface com as implementações específicas de banco de dados.

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.

5.3.4 GnuGK Authenticator Module

Este AM utiliza a porta de gerência remota do GnuGK [14] (ver Ilustração 37) 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 [59]).

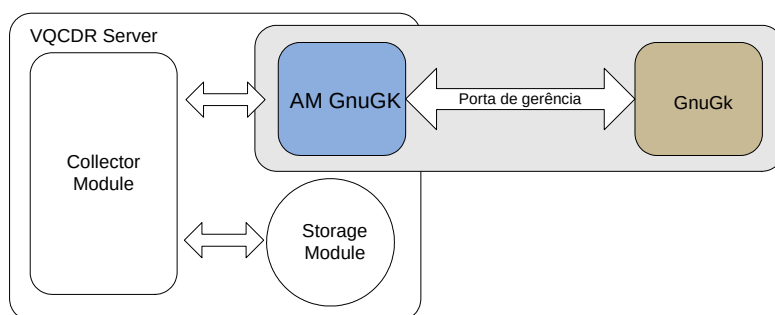


Ilustração 37. GnuGK Authenticator Module

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 coincidirem 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 seja executado na mesma máquina que o GnuGK, otimizando a interação do módulo de autenticação.

5.3.5 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*) [60] deve ser solicitado à IANA [61]. De posse desse número, o fabricante/aplicação pode definir um dicionário de dados que permite aos servidores RADIUS interpretarem estes atributos específicos.

# 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

Ilustração 38. Dicionário de dados RADIUS para o VQCDR

Para mapear os campos do VQCDR, foi solicitado à IANA um número privativo de fabricante, que atribuiu o código 21715 à UFRJ. Com posse deste código, foi criado um dicionário de dados próprio para o VQCDR, como mostrado na Ilustração 38.

Todos os campos do dicionário de dados RADIUS para o VQCDR são obtidos diretamente do VQCDR (ver Quadro 7), exceto pelos campos callStart e callStop. O campo

callStop, que representa o horário de término da chamada, é preenchido com instante de recebimento do VQCDR pelo VQCDR Server. O campo callStart, que representa o horário de início da chamada, é obtido subtraindo-se o campo duration do campo callStop, ou seja, do instante de recebimento do VQCDR.

Considerando a dificuldade de garantir que todos os telefones IP de um ambiente VoIP estejam sempre sincronizados, o procedimento de preenchimento dos campos callStart e callDuration no VQCDR Server é mais seguro do que o preenchimento destes campos no emissor do VQCDR (telefone IP). De qualquer forma, os campos callStart e callStop devem ser utilizados somente para fins de análise temporal e histórica da qualidade das chamadas realizadas ao longo de um período de tempo (como, por exemplo, horas do dia, dias da semana, meses do ano, etc), e não para efeitos de bilhetagem, já que espera-se que outros componentes do ambiente VoIP, tais como o *gatekeeper* H.323, o *proxy* SIP e o *gateway* de voz, possibilitem mecanismos mais especializados e eficientes para contabilização do uso do sistema. Tanto o *gatekeeper* H.323 GnuGK [14], o *proxy* SIP SER [15] e os *gateways* de voz, Cisco [18] e Asterisk [19], oferecem mecanismos mais precisos para contabilização e bilhetagem do sistema.

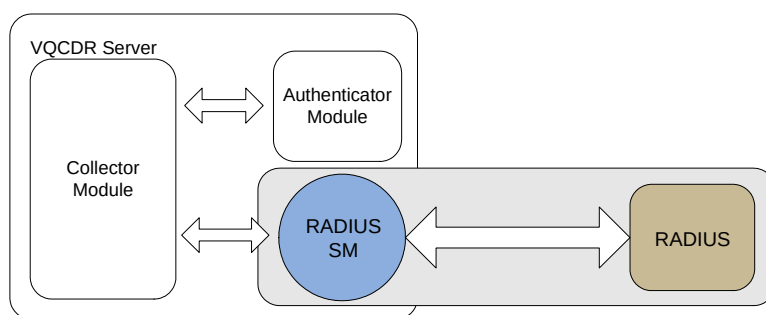


Ilustração 39. RADIUS Storage Module

A interface do SM com o RADIUS (ver Ilustração 39) foi implementada com o auxílio da biblioteca RadiusClient [62], 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.

5.4 Ambiente de Visualização

No serviço fone@RNP, todos os CDRs gerados, sejam eles provenientes do *gateway* de voz ou de clientes IP (VQCDR), são repassados aos servidores RADIUS, que por sua vez, armazenam-nos em um banco de dados SQL. Posteriormente, um sistema de visualização de estatísticas, desenvolvido pelo Laboratório de VoIP da UFRJ [63], processa e consolida os CDRs armazenados no banco de dados, e, por meio de uma interface *Web* desenvolvida em PHP [64], 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 (Ilustração 40).

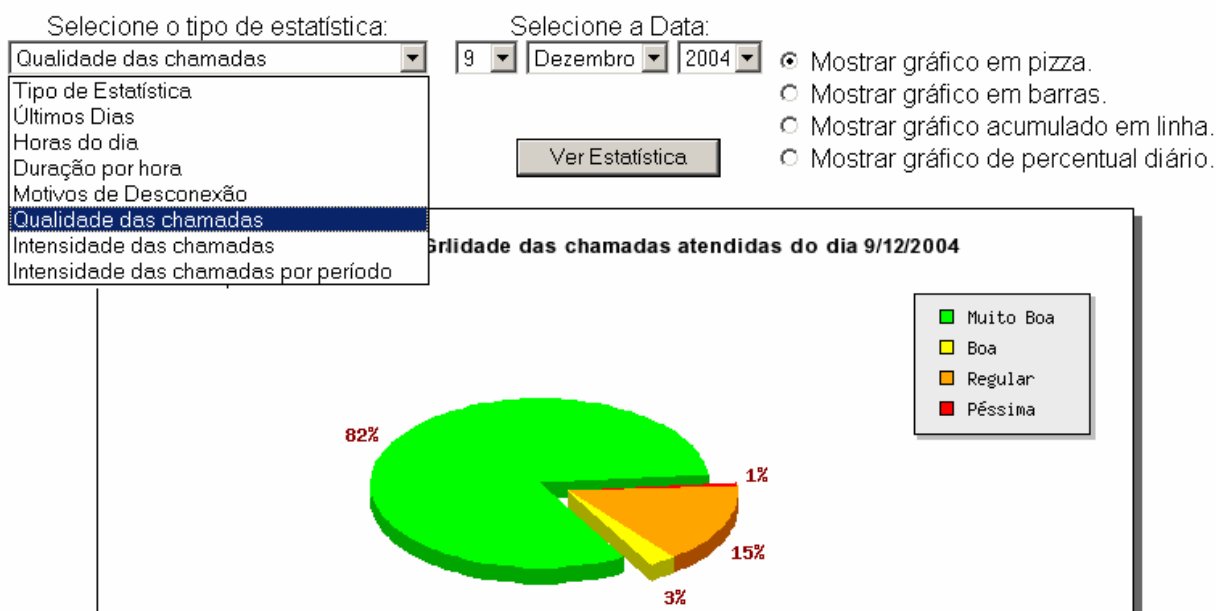


Ilustração 40. Ambiente Web de visualização de estatísticas

5.5 Arquitetura de Monitoração em Operação

A Ilustração 41 sumariza o funcionamento da arquitetura de monitoração de qualidade implementada. É importante notar que esta ilustração representa a arquitetura operando em um ambiente baseado em H.323, com uso do *Gatekeeper* GnuGK e RADIUS. Entretanto, um AM e um SM específicos podem ser implementados para suporte a outros ambientes, operando, por exemplo, com SIP.

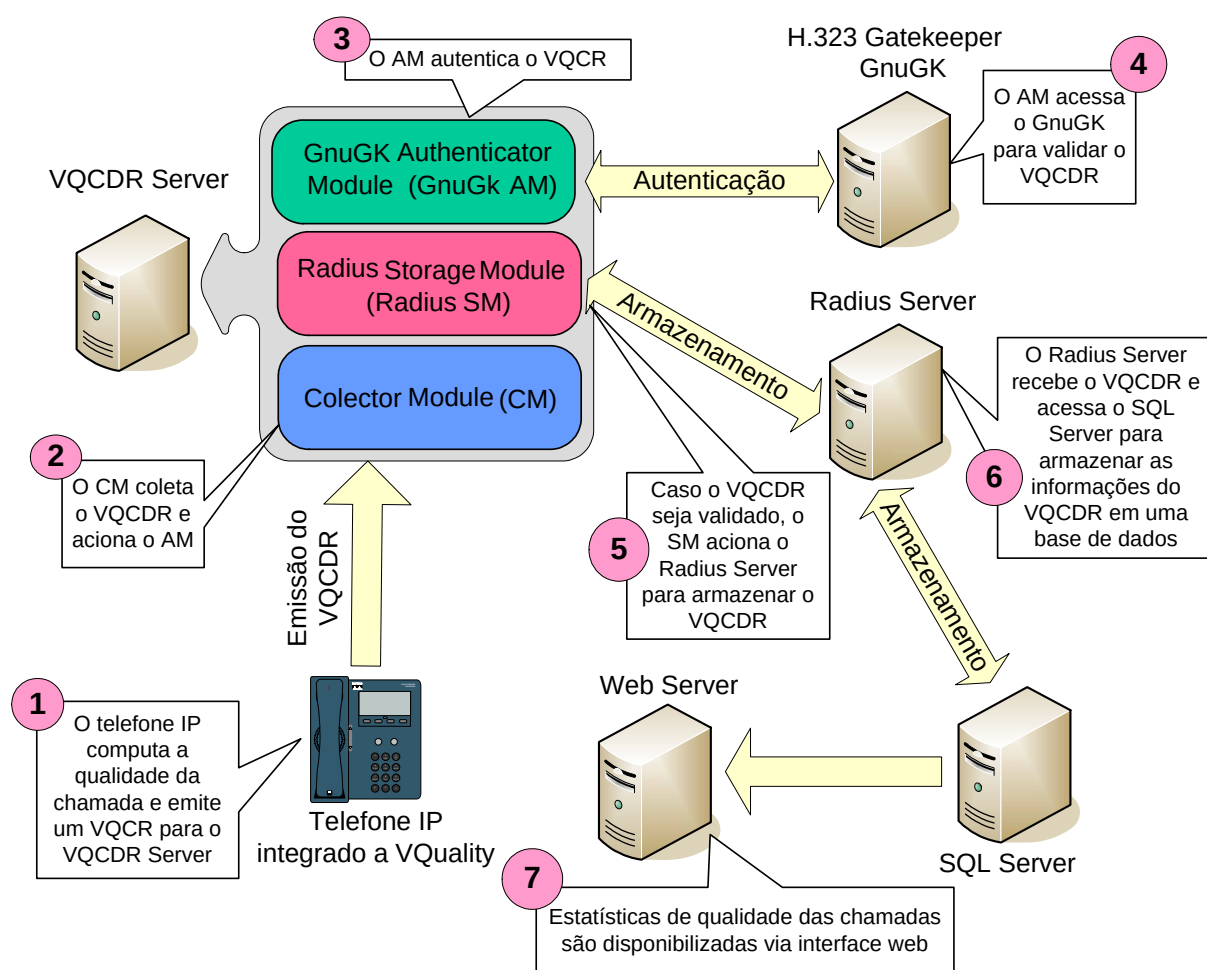


Ilustração 41. Funcionamento da arquitetura de monitoração de qualidade

Ao final da chamada, o telefone IP integrado à VQuality avalia a qualidade de voz recebida segundo sua perspectiva, e então, emite um VQCDR contendo indicadores de qualidade e identificadores de chamada (Ilustração 41-1). Este VQCDR é coletado pelo CM do VQCDR Server (Ilustração 41-2), que aciona o AM para verificar a legitimidade do

VQCDR (Ilustração 41-3). No caso do GnuGK AM, a autenticação é realizada via consulta ao GnuGK (Ilustração 41-4). Caso o VQCDR seja validado, o SM é acionado para armazenar o VQCDR. No caso do RADIUS SM, o VQCDR é enviado via protocolo RADIUS para um Servidor RADIUS (Ilustração 41-5). O Servidor RADIUS então armazena as informações do VQCDR em uma base de dados, que pode ser, por exemplo, um servidor SQL (Ilustração 41-6). Estas informações são por fim disponibilizadas em forma de gráficos e estatísticas via interface *Web* (Ilustração 41-7).

5.6 Clientes SIP com Suporte a VQCDR

Devido às suas características de flexibilidade e facilidade de integração com os serviços Internet, há hoje uma grande tendência para o uso do SIP como protocolo de sinalização VoIP.

Esta tendência pode ser observada, inclusive, no ambiente do serviço *fone@RNP*. Inicialmente, o serviço possuía uma arquitetura integralmente baseada em H.323, e hoje evoluiu para uma plataforma heterogênea com suporte a SIP e H.323. Desta forma, é desejável que a arquitetura de monitoração de qualidade de chamadas seja também capaz de monitorar chamadas SIP, assim como é capaz de monitorar chamadas H.323, com o uso do VQOpenPhone e do VQMeeting.

Para que a arquitetura de monitoração seja capaz de avaliar a qualidade de chamadas realizadas entre telefones SIP, duas metas devem ser atingidas: A primeira é a disponibilização de clientes SIP com suporte a avaliação de qualidade e emissão de VQCDR, que pode ser atingida através da integração de telefones SIP com a biblioteca VQuality. A segunda é a implementação de um AM específico para o *proxy* SIP SER, para que os

VQCDRs emitidos pelos telefones SIP do ambiente do serviço `fone@RNP` possam ser autenticados.

A implementação de um AM para o SER não exigirá grandes esforços de implementação, já que os dados necessários para autenticação do VQCDR (ver seção 5.3.4) são disponibilizados facilmente pelo SER, via consulta SQL.

A integração da VQuality a um cliente SIP também não deverá oferecer maiores dificuldades. A experiência adquirida com a integração do OpenPhone e a característica de flexibilidade da VQuality, que é independente de protocolo de sinalização e foi projetada visando a futura integração com SIP, são facilitadores para o processo de integração.

5.7 Asterisk com Suporte a Emissão de VQCDR

O Asterisk possui um importante papel no ambiente do serviço `fone@RNP`, operando como *gateway* VoIP/RTFC e como *gateway* SIP/H323. A Ilustração 42 apresenta os três possíveis cenários de uso do Asterisk no ambiente do serviço.

Como o Asterisk possui uma pilha SIP e uma pilha H.323, ambas as pilhas precisam ser integradas à VQuality para que todas as chamadas realizadas nos três cenários mostrados na Ilustração 42 sejam avaliadas.

Para que o Asterisk seja capaz de avaliar a qualidade de uma chamada estabelecida entre o ambiente H.323 e a RTFC (Ilustração 42-a), é necessário que a VQuality seja integrada à pilha H.323. Como esta pilha é baseada na biblioteca OpenH323, este processo de integração será imediato, já que a OpenH323 já se encontra integrada à VQuality.

De modo equivalente, para que o Asterisk seja capaz de avaliar a qualidade de uma chamada estabelecida entre o ambiente SIP e a RTFC (Ilustração 42-b), é necessário que a VQuality seja integrada à pilha SIP.

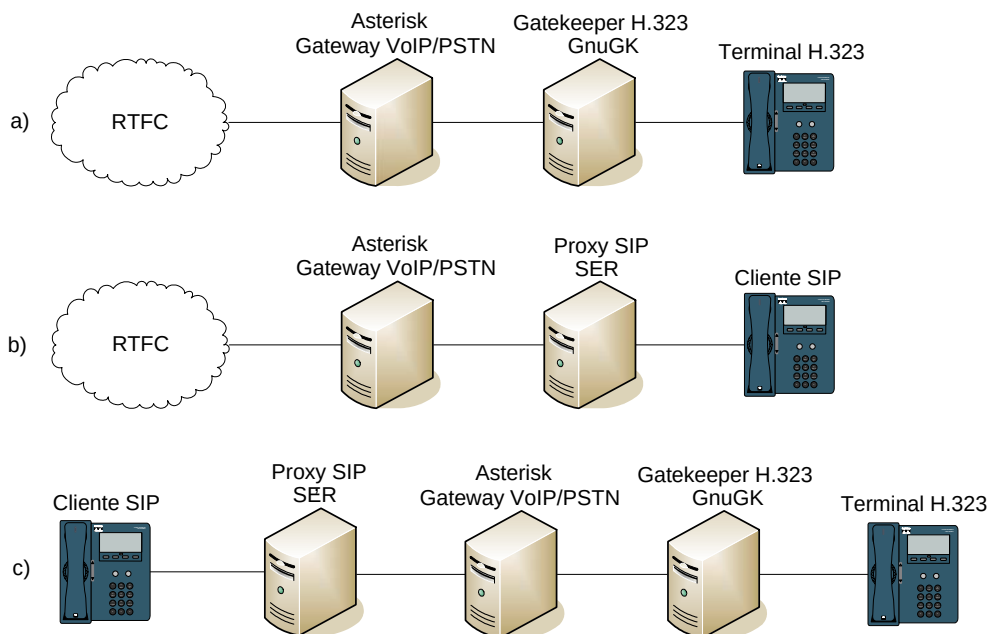


Ilustração 42. Asterisk como gateway VoIP/RTFC e como gateway SIP/H.323

A integração da VQuality às pilhas SIP e H.323 do Asterisk permitirá também que as chamadas estabelecidas entre o ambiente H.323 e SIP sejam avaliadas (Ilustração 42-c). Entretanto, é importante ressaltar que estas chamadas serão quebradas em dois trechos, um entre o cliente SIP e o Asterisk e outro entre o Asterisk e o terminal H.323, como mostrado na Ilustração 43.

A quebra da chamada em dois trechos é necessária, pois o Asterisk não opera como um *proxy* RTP transparente. Assim, ao realizar o mecanismo de tradução, seja de SIP para H.323 ou de H.323 para SIP, o RTP é regenerado, fato que impede que o RTP/RTCP ocorra fim-a-fim e que impossibilita que parâmetros necessários para o cálculo do Modelo E, como o RTT, não possam ser determinados pelos terminais finais da comunicação.

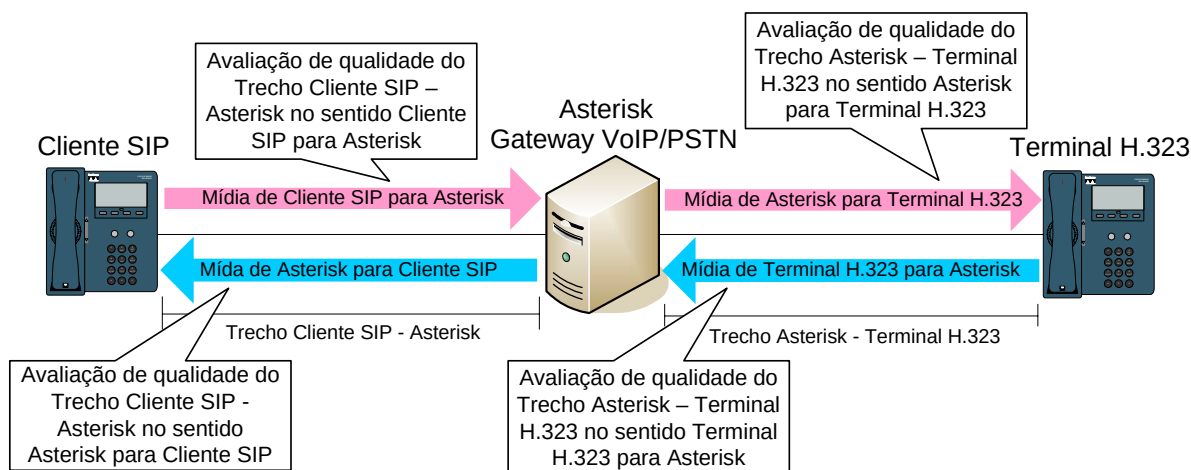


Ilustração 43. Trechos de uma chamada entre SIP e H.323 via Asterisk

Desta forma, cada um destes trechos deve ter uma avaliação de qualidade independente. Assim, ao invés de dois VQCDRs, quatro VQCDRs seriam emitidos ao final da chamada (Ilustração 43):

- um relativo à avaliação do trecho cliente SIP – Asterisk no sentido cliente SIP para Asterisk;
- outro relativo à avaliação do trecho cliente SIP – Asterisk no sentido Asterisk para cliente SIP;
- outro relativo à avaliação do trecho Asterisk – terminal H.323 no sentido Asterisk para terminal H.323;
- e outro relativo à avaliação do trecho Asterisk – terminal H.323 no sentido terminal H.323 para o Asterisk.

Capítulo 6

CONCLUSÕES E TRABALHOS FUTUROS

Nesta dissertação, uma arquitetura para monitoração de qualidade de chamadas telefônicas IP, com base em avaliações objetivas de qualidade de voz, é desenvolvida e implementada.

No primeiro capítulo, é feita uma introdução sobre a problemática da utilização do protocolo IP para transporte de voz, sobre os métodos objetivos de avaliação de qualidade e sobre a importância da monitoração de qualidade das chamadas de um serviço de telefonia IP. São apresentadas uma série de arquiteturas de monitoração encontradas na literatura [28,29,30,31] e em soluções comerciais [32,33,34], bem como é descrito o ambiente do serviço fone@RNP, onde a arquitetura de monitoração proposta por esta dissertação está sendo implementada e validada.

No segundo capítulo, o Modelo E da ITU (*International Telecommunications Union*) [11] e uma extensão a esse modelo, formulada por Clark [26] e adotada pelo ETSI (*European Telecommunications Standards Institute*) [20,21,22], são analisados em detalhes e é apresentada uma correção à forma de cálculo do Modelo E estendido definida por Clark [26].

No terceiro capítulo, o MOBVEM (Modified OpenH323 Voice Evaluation Module), módulo de avaliação analítica de qualidade de voz, é descrito. Este módulo possibilitou a comparação entre a forma de cálculo do modelo segundo a recomendação do ETSI, e a forma revisada, apresentando grande discrepância entre os resultados obtidos. Considerando que no

Capítulo 2 é demonstrado que a forma de cálculo do ETSI está incorreta, espera-se que a forma revisada apresente resultados mais fiéis à percepção humana. Devido a grande infraestrutura requerida, trabalhos mais detalhados com o uso de avaliação subjetiva foram deixados como sugestões para trabalhos futuros.

No quarto capítulo, os detalhes de implementação de uma nova biblioteca de avaliação de qualidade de voz, denominada Voice Quality (VQuality) [23], são apresentados. Esta biblioteca suporta a operação em múltiplas plataformas computacionais, é independente de protocolo de sinalização VoIP e tem facilidade para agregar novos modelos de avaliação de qualidade. Estas funcionalidades são decorrentes da concepção da arquitetura da biblioteca, que disponibiliza alto grau de portabilidade, flexibilidade e facilidade de extensão.

A VQuality é utilizada como base para operacionalização de uma nova proposta de arquitetura de monitoração de qualidade de chamadas telefônicas IP [24], que é descrita no Capítulo 5. Esta arquitetura define um mecanismo de emissão de um novo formato de registro de detalhamento de chamada (CDR), denominado Voice Quality CDR (VQCDR). Com base no Modelo E e em suas extensões e revisões [5,11,20,21,22], o VQCDR oferece indicadores de qualidade extremamente detalhados e especializados para a monitoração de qualidade de chamadas telefônicas IP, permitindo ainda, o envio opcional de um histórico de qualidade, chamado de Voice Quality Log (VQLog), que possibilita a análise da variação da qualidade ao longo da chamada, para uso em medições mais minuciosas realizadas em ambientes de teste. Esta arquitetura está sendo implantada e validada no ambiente do fone@RNP, serviço de telefonia IP da Rede Nacional de Ensino e Pesquisa (RNP).

O aplicativo Voice Quality Plot (VQPlot), que interpreta o VQLog e apresenta gráficos da variação da qualidade e dos fatores que influenciaram a qualidade da voz ao longo da chamada, foi mostrado. Este aplicativo representa uma poderosa ferramenta de

identificação de problemas na rede e em configurações do *buffer* de compensação de *jitter* do telefone IP.

O VQCDR Server, elemento da arquitetura de monitoração proposta responsável pela coleta, autenticação e encaminhamento dos VQCDRs para armazenamento, foi especificado. Foi desenvolvido, então, um módulo para autenticação de VQCDRs específico para o *gatekeeper* H.323 GnuGK [14] (GnuGK Authenticator Module), um módulo de armazenamento específico para o RADIUS [17] (RADIUS Storage Module) e um módulo para coleta de VQCDRs (Collector Module).

Objetivando o desenvolvimento de terminais H.323 com suporte a avaliação de qualidade de voz e emissão de VQCDR, a VQuality foi integrada à biblioteca OpenH323 [38]. Os detalhes deste processo de integração foram mostrados, bem como relatadas as alterações no código da biblioteca OpenH323 para sua adequação aos padrões do protocolo RTP/RTCP, necessários para prover o cômputo do Modelo E. Como resultado, foram disponibilizados os terminais H.323 VQOpenPhone e VQMeeting, para operação em ambientes baseados na plataforma Windows e Unix, respectivamente.

Como trabalho futuro, a biblioteca VQuality será integrada a clientes SIP, tanto em ambiente Windows como em ambientes Unix. Em especial, não há nada que impeça a incorporação desta biblioteca em telefones IP, implementados em produtos de *hardware* ou de *software*.

Com relação ao suporte a SIP na coleta de VQCDR, um módulo de autenticação para o *proxy* SER [15] deverá ser desenvolvido. Outro plano de desenvolvimento futuro é a integração da VQuality ao Asterisk, como descrito na seção 5.7.

A disseminação de telefones IP e *gateways* de voz integrados à VQuality no serviço fone@RNP possibilitará uma eficiente monitoração da qualidade das chamadas no ambiente do serviço, onde anteriormente, apenas as chamadas estabelecidas entre *gateways* de voz Cisco [18,25] possuíam alguma qualidade aferida, como também, possibilitará uma monitoração mais abrangente e confiável que outros trabalhos correlatos [28,29,30,31] e soluções comerciais [32,33,34] discutidas neste trabalho.

Com o crescente interesse no uso de VoIP e a ausência no mercado de telefones IP com a capacidade de gerar medições de qualidade, um leque grande de oportunidades de parcerias com fabricantes é aberto. Assim, espera-se que trabalhos futuros de integração da VQuality à telefones IP proprietários sejam desenvolvidos.

Com a disseminação de VoIP na Internet, é possível antever o aparecimento de vários provedores de serviços de telefonia IP e a situação em que uma chamada atravessasse a rede de vários provedores para ser estabelecida. Neste cenário, além da obtenção da qualidade fim-a-fim, será muito importante determinar o quanto cada provedor está contribuindo para a perda da qualidade da chamada. Com o uso da biblioteca VQualily, módulos de avaliação poderão ser desenvolvidos e posicionados em pontos de troca de tráfego entre provedores VoIP para possibilitar a determinação destas informações.

Assim como a monitoração de chamadas reais, a arquitetura proposta nesta dissertação também pode ser utilizada para a medição de qualidade de chamadas simuladas. Logo, uma ferramenta automatizada para simulação e avaliação de chamadas deverá ser desenvolvida e utilizada na determinação da capacidade de chamadas e no estudo de políticas de Qualidade de Serviço (QoS) para priorização de tráfego de voz, aplicadas em ambientes de redes com fio e redes sem fio (*wireless*). Esta ferramenta será uma evolução da infra-estrutura de geração, coleta e medição de chamadas utilizada no Capítulo 3 [46].

Outra possível linha de ação é a integração da VQuality com ferramentas de simulação de rede, como o Network Simulator [65], que poderia propiciar um ambiente para avaliação de sistemas complexos de implementação VoIP.

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 determinar os atrasos em cada direção de uma chamada VoIP. A partir da base de dados das chamadas no serviço `fone@RNP` pode-se, como trabalho complementar futuro, gerar uma análise da assimetria de tráfego e atraso na RNP e suas redes coligadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BRADEN, R., ZHANG L., BERSON, S., HERZOG S., JAMIN S., *Resource ReSerVation Protocol (RSVP) – Version I Functional Specification*, RFC2205, set. 1997.
- [2] BLAKE, S., BLACK, D., CARLSON M., DAVIES, E., WANG Z., WEISS, W., *An Architecture for Differentiated Service*, RFC2475, dez. 1998.
- [3] ITU-T Recommendation P.800, *Methods for subjective determination of transmission quality*. Genève, ago. 1996.
- [4] ITU-T Recommendation P.830, *Subjective performance assessment of telephoneband and wideband digital codecs*. Genève, Fev. 1996.
- [5] LUSTOSA, L.C.G., CARVALHO, L.S.G., 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.
- [6] ITU-T Recommendation P.861, *Objective quality measurement of telephone-band speech codecs*. Genève, 1996.
- [7] RIX, A. W., HOLLIER, M. P., *The perceptual analysis measurement system for robust end-to-end speech quality assessment*, IEEE ICASSP, jun 2000.
- [8] ITU-T Recommendation P.862, *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*. Genève, 2001.
- [9] HALL, Timothy A., *Objective Speech Quality Measures for Internet Telephony*, in Proceedings of SPIE Vol. 4522. Petros Mouchtaris Editor, [s.1] 2001. 9 p.
- [10] ETSI, *Speech Communication Quality for Mouth to Ear for 3.1 kHz Handset Telephony across Networks*, Technical Report ETR250, 1996.
- [11] ITU-T Recommendation G.107, *The E-Model, a computational model for use in transmission planning*. Genève, mar 2003.
- [12] ITU-T Recommendation H.323, *Packet-Based Multimedia Communications Systems*. Genève, nov 2000.
- [13] ROSENBERG, J., SCHULZRINNE, H., CAMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, M., SCHOOLER, E., *SIP: Session Initiation Protocol*. RFC 3261, jun. 2002.
- [14] OPENH323 GATEKEEPER. *The GNU Gatekeeper*. Disponível em <http://www.gnugk.org/>. Acesso em jun. 2005.
- [15] IPTEL.ORG, *SIP Server: SIP Express Router*. Disponível em <http://www.iptel.org/ser>. Acesso em jun 2005.
- [16] HODGES, J., MORGAN, R., *Lightweight Directory Access Protocol (v3): Technical Specification*. RFC 3377, September 2002
- [17] RIGNEY, C., WILLENS, S., RUBENS, A., SIMPSON, W., *Remote Authentication Dial In User Service (RADIUS)*. RFC 2865, jun. 2000.

- [18] 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 jun. 2005.
- [19] Asterisk™ – *The Open Source Linux PBX*, Disponível em <http://www.asterisk.org>. Acesso em jun. 2005.
- [20] ETSI TS 101 329-5 v1.1.1. 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*. France, nov. 2000.
- [21] 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.
- [22] 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.
- [23] LUSTOSA, L.C.G., RODRIGUES, P.H.A., DAVID, F., QUINELLATO, G. D., *VQuality: Uma Biblioteca Multiplataforma para Avaliação de Qualidade de Chamadas Telefônicas IP*, in: Anais do XXIII Simpósio Brasileiro de Redes de Computadores. Fortaleza, maio 2005.
- [24] LUSTOSA, L.C.G., RODRIGUES, P.H.A., DAVID, F., QUINELLATO, G. D., *Arquitetura de Monitoração de Qualidade de Chamadas Telefônicas IP*, in: Anais do XXIII Simpósio Brasileiro de Redes de Computadores. Fortaleza, maio 2005.
- [25] 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 jun. 2005.
- [26] CLARK, Alan D., *Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality*. IP – Telephony Workshop. [USA], 2001.
- [27] COLE, Robert G., ROSENBLUTH, J. H., *Voice over IP performance monitoring*. ACM SIGCOMM Computer Communication Review, vol.31, issue 2. abr. 2001.
- [28] UBIK, S., *IP Telephone Accounting and WAN Deployment Experience*. EUA, abr. 2001.
- [29] LIN, M., LO, C., W., S., *Design and Implementation of an Enterprise Call Analysis System for VoIP Deployments*. 2003 Australian Telecommunications, Networks and Applications Conference (ATNAC), dez. 2003.
- [30] HUANG, C., CHAO, C., LIU, A., *A Distributed Management Framework for H.323-Based VoIP System*. Communications in Computing (CIC) 2003. EUA, jun. 2003.
- [31] BROOM, S., HOLLIER, M., *Speech Quality Measurement Tools for Dynamic Network Management*. Measurement of Speech and Audio Quality in Networks (MESAQIN) 2003, maio 2003.
- [32] TELCHEMY, *SQmon - Service Quality Monitoring for Voice over IP*. Disponível em <http://www.telchemy.com/sqmon.html>. Acesso jun. 2005.

- [33] EMPIRIX, *Hammer XMS*. Disponível em <http://www.empirix.com/Empirix/Network+IP+Storage+Test/hammer+xms.html>. Acesso em jun. 2005.
- [34] BRIX NETWORKS, *Advanced VoIP Test Suites*. Disponível em http://www.brixnet.com/products/voip_testsuite.html. Acesso em jun. 2005.
- [35] H.323 BEACON TOOL, *H.323 Beacon Project*. Disponível em <http://www.itecoho.org/beacon/>. Acesso em jun. 2005.
- [36] CALYAM, P, MANDRAWA, W, SRIDHARAN, M, KHAN, A, SCHOPIS, P., *H.323 Beacon: An H.323 application related end-to-end performance troubleshooting tool*. Proceedings of ACM SIGCOMM Network Troubleshooting Workshop (NetTs' 04), Portland, Oct. 2004.
- [37] CALYAM, P, SRIDHARAN, M, MANDRAWA, W, SCHOPIS, P., *Performance Measurement and Analysis of H.323 Traffic*, Proceedings of Passive and Active Measurement Workshop (PAM '04), França, Abril 2004.
- [38] OPENH323 PROJECT, Disponível em <http://www.openh323.org/>. Acesso em jun. 2005.
- [39] SCHULZRINNE, H. CASNET, S., FREDERICK, R., JACOBSON, V. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550, jul. 2003.
- [40] PSYTECHNICS GROUP, *Estimating E-model Id within a VoIP network*. Psytechnics Technical note. United Kingdom, 2002.
- [41] ITU-T Recommendation G.113 Appendix I, *Provisional planning values for the equipment impairment factor Ie*. Genève, set. 1999.
- [42] ITU-T Recommendation G.113 Appendix I, *Provisional planning values for the equipment impairment factor Ie*. Genève, out. 2001.
- [43] ITU-T Recommendation G.113 Appendix I, *Provisional planning values for the equipment impairment factor Ie and packet-loss robustness factor Bpl*. Genève, maio 2002.
- [44] ITU-T Recommendation G.114, *One-way transmission time*. Genève, maio 2000.
- [45] FRANCE TELECOM, *Study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences: influence of a recency effect*. ITU Study Group 12 D.139, maio 2000.
- [46] 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.
- [47] STOYCHEV, M., <http://kgb.cnsys.bg/voice/>. Acesso em dez. 2004.
- [48] THE PERL FOUNDATION, *The Perl Directory at Perl.org*. Disponível em <http://www.perl.org/>. Acesso em jun. 2005.
- [49] ANDRESEN, F., FORTER, B., *Media Gateway Control Protocol (MGCP) Version 1.0*. RFC 3435, jan. 2003.
- [50] PTHREADS-WIN32, *Open Source POSIX Threads for Win32*. Disponível em <http://sources.redhat.com/pthreads-win32/>. Acesso em jun. 2005.
- [51] THE OPEN GROUP, *Single UNIX Specification Version 3*, IEEE Std 1003.1, 2004 Edition. 2004.

- [52] GCC, *The GNU Compiler Collection*. Disponível em <http://gcc.gnu.org/>. Acesso em jun. 2005.
- [53] MICROSOFT CORPORATION, *Microsoft Visual C++ Developer Center*. Disponível em <http://msdn.microsoft.com/visualc/>. Acesso em jun. 2005.
- [54] VAUGHAN, G. V. ELLISTON, B. TROMEY, T., TAYLOR, I. L., *GNU Autoconf, Automake and Libtool*. Disponível em <http://sources.redhat.com/autobook/>. Acesso em jun. de 2005.
- [55] PWLIB, *A Portable Text and GUI C/C++ Class Library*. Disponível em <http://www.openh323.org/docs/PWLib/>. Acesso em jun. 2005.
- [56] GNOME MEETING.ORG, *GnomeMeeting*. Disponível em <http://www.gnomemeeting.org/>. Acesso em jun. 2005.
- [57] CISCO SYSTEMS, *Cisco IP Phone 7940G*. Disponível em http://www.cisco.com/warp/public/cc/pd/tlhw/prodlit/7940_ds.htm. Acesso em jun. 2005.
- [58] CISCO SYSTEMS, *Cisco ATA 188 Analog Telephone Adapter*. Disponível em http://www.cisco.com/warp/public/cc/pd/tlhw/prodlit/7940_ds.htm. Acesso me jun. 2005.
- [59] SRISURESH, P., EGEVANG, K., *Traditional IP Network Address Translator (Traditional NAT)*, RFC 3022, jan. 2001.
- [60] REYNOLDS, J., Postel, J., *Assigned Numbers*, RFC 1700, out. 1994.
- [61] IANA, *Internet Assigned Numbers Authority*. Disponível em <http://www.iana.org/> Acesso em jun. 2005.
- [62] RADIUSCLIENT, Disponível em <http://freshmeat.net/projects/radiusclient/>. Acesso em jun. 2005.
- [63] COSTA, J.C.P., RODRIGUES, P.H.A., SILVEIRA, R. G., COSTA, M. A. L., *Contabilização de Chamadas em uma Infra-estrutura de Voz sobre IP (VoIP)*, in: Anais do XXIII Simpósio Brasileiro de Redes de Computadores. Fortaleza, maio 2005.
- [64] THE PHP GROUP, *PHP - Hypertext Preprocessor*. Disponível em <http://www.php.net>. Acesso jun. 2005.
- [65] THE UNIVERSITY OF SOUTHERN CALIFORNIA, *Network Simulator – ns2*. Disponível em <http://www.isi.edu/nsnam/ns/>. Acesso em jun. 2005.

APÊNDICE A

Validação Matemática da Expressão de Determinação I_1 Proposta

A substituição da equação de obtenção de I_1 (18) (apresentada por Clark [26], na recomendação do ETSI [20] e em revisões a esta recomendação [21,22]) pela equação (24) (sugerida por esta dissertação) pode ser facilmente validada através da determinação visual de I_1 e I_2 .

Para isso, iremos partir das equações F_1 (25) e F_2 (26), que possuem comportamento similar às equações de determinação de I_1 e I_2 (18), (19) e (24).

$$F_1 = b + a.e^{-x} \quad (25)$$

$$F_2 = b - a.e^{-x} \quad (26)$$

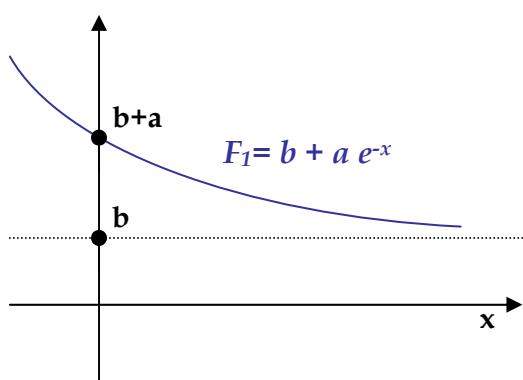


Ilustração 44. Representação visual da equação F_1

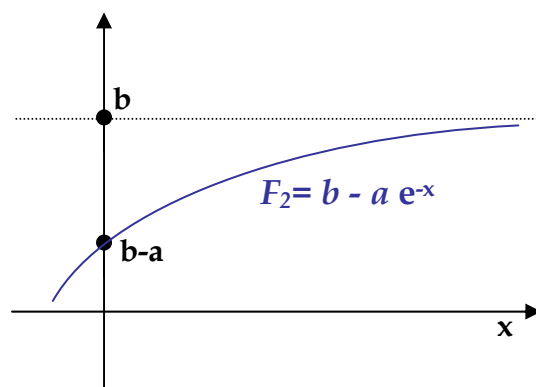


Ilustração 45. Representação visual da equação F_2

A Ilustração 44 representa visualmente a equação (25). Para x igual à zero, F_1 é igual à $b + a$, tendendo a b progressivamente quando x aumenta.

Já a Ilustração 45 representa visualmente a equação (26). Para x igual à zero, F_2 é igual à $b - a$, tendendo a b progressivamente quando x aumenta.

Utilizando as equações (25) e (26) como base de referência, podemos visualizar mais facilmente a determinação de I_1 e I_2 , apresentada nas Ilustrações 44 e 45.

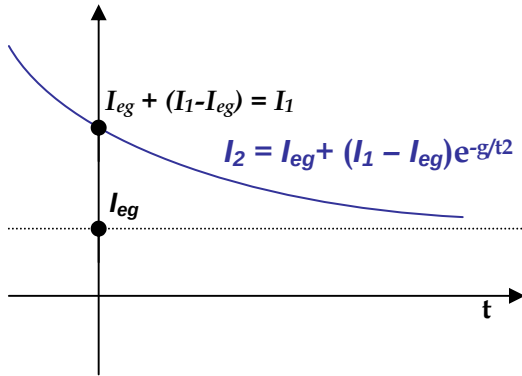


Ilustração 46. Representação visual da equação I_2

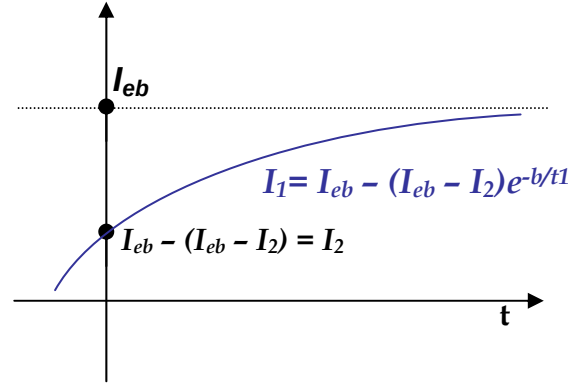


Ilustração 47. Representação visual da equação I_1

A Ilustração 46 apresenta visualmente a obtenção da equação I_2 , que representa a transição de um estado de perda em rajadas (*burst*) para um estado de perdas isoladas (*gap*). I_2 tende a I_{eg} (nível de degradação em *gap*) progressivamente à medida que a duração do *gap* (g) aumenta, partindo de um ponto onde a duração do *gap* (g) é igual à zero. Este ponto equivale a uma transição anterior de um estado de *gap* para um estado de *burst*, representado por I_1 , que é igual à $I_{eg} + (I_1 - I_{eg})$. Logo se chega à mesma equação (19) apresentada por Clark [26] para determinação de I_2 ($I_2 = I_{eg} + (I_1 - I_{eg})e^{-g/t2}$).

A Ilustração 47 apresenta visualmente a obtenção da equação I_1 , que representa a transição de um estado de perdas isoladas (*gap*) para um estado de perda em rajadas (*burst*). I_1 tende a I_{eb} (nível de degradação em *burst*) progressivamente à medida que a duração do *burst* (b) aumenta, partindo de um ponto onde a duração do *burst* (b) é igual à zero. Este ponto equivale a uma transição anterior de um estado de *burst* para um estado de *gap*, representado por I_2 , que é igual à $I_{eb} - (I_{eb} - I_2)$. Logo se chega à equação (24) ($I_1 = I_{eb} - (I_{eb} - I_2)e^{-b/t1}$) que difere da equação original apresentada por Clark (18) ($I_1 = I_{eb} - (I_{eg} - I_2)e^{-b/t1}$).

Clark [26] combina as equações de (18) e (19) para determinar equações de I_1 e I_2 dependentes apenas de I_{eg} , I_{eb} , b e g , (22) e (23) respectivamente:

$$I_1 = \frac{I_{eb}(1 - e^{-b/t_1}) + I_{eg}(1 - e^{-g/t_2})e^{-b/t_1}}{1 - e^{-b/t_1 - g/t_2}}$$

$$I_2 = \frac{I_{eg}(1 - e^{-g/t_2}) + I_{eb}(1 - e^{-b/t_1})e^{-g/t_2}}{1 - e^{-b/t_1 - g/t_2}}$$

Entretanto, nota-se que não é possível se chegar às equações (22) e (23) com o uso das equações (18) e (19), mas combinando a equação (19) com a equação (24) (sugerida por esta dissertação) pode-se chegar às equações (22) e (23) sem problemas, como expressam os cálculos demonstrativos abaixo:

Equações de partida I_1 (24) e I_2 (19):

$$I_1 = I_{eb} - (I_{eb} - I_2)e^{-b/t_1}$$

$$I_2 = I_{eg} + (I_1 - I_{eg})e^{-g/t_2}$$

Obtendo I_1 em função I_{eg} , I_{eb} , b e g (equação (22)):

$$I_1 = I_{eb} - \{I_{eb} - [I_{eg} + (I_1 - I_{eg})e^{-g/t_2}]\}e^{-b/t_1}$$

$$I_1 = I_{eb} - \{I_{eb} - [I_{eg} + I_1.e^{-g/t_2} - I_{eg}.e^{-g/t_2}]\}e^{-b/t_1}$$

$$I_1 = I_{eb} - \{I_{eb} - I_{eg} - I_1.e^{-g/t_2} + I_{eg}.e^{-g/t_2}\}e^{-b/t_1}$$

$$I_1 = I_{eb} - \{I_{eb}.e^{-b/t_1} - I_{eg}.e^{-b/t_1} - I_1.e^{-g/t_2 - b/t_1} + I_{eg}.e^{-b/t_1 - g/t_2}\}$$

$$I_1 = I_{eb} - I_{eb}.e^{-b/t_1} + I_{eg}.e^{-b/t_1} + I_1.e^{-b/t_1 - g/t_2} - I_{eg}.e^{-b/t_1 - g/t_2}$$

$$I_1 = I_{eb}(1 - e^{-b/t_1}) + I_{eg}(1 - e^{-g/t_2})e^{-b/t_1} + I_1.e^{-b/t_1 - g/t_2}$$

$$I_1 - I_1.e^{-b/t_1 - g/t_2} = I_{eb}(1 - e^{-b/t_1}) + I_{eg}(1 - e^{-g/t_2})e^{-b/t_1}$$

$$I_1(1 - e^{-b/t_1 - g/t_2}) = I_{eb}(1 - e^{-b/t_1}) + I_{eg}(1 - e^{-g/t_2})e^{-b/t_1}$$

$$I_1 = \frac{I_{eb}(1 - e^{-b/t_1}) + I_{eg}(1 - e^{-g/t_2})e^{-b/t_1}}{1 - e^{-b/t_1 - g/t_2}}$$

Obtendo I_2 em função de I_{eg} , I_{eb} , b e g (equação (23)):

$$\begin{aligned}
I_2 &= I_{eg} + \left\{ \left[I_{eb} - (I_{eb} - I_2)e^{-b/t_1} \right] - I_{eg} \right\} e^{-g/t_2} \\
I_2 &= I_{eg} + \left\{ \left[I_{eb} - I_{eb}e^{-b/t_1} + I_2e^{-b/t_1} \right] - I_{eg} \right\} e^{-g/t_2} \\
I_2 &= I_{eg} + \left\{ I_{eb} - I_{eb}e^{-b/t_1} + I_2e^{-b/t_1} - I_{eg} \right\} e^{-g/t_2} \\
I_2 &= I_{eg} + \left\{ I_{eb}e^{-g/t_2} - I_{eb}e^{-b/t_1-g/t_2} + I_2e^{-b/t_1-g/t_2} - I_{eg}e^{-g/t_2} \right\} \\
I_2 &= I_{eg} + I_{eb}e^{-g/t_2} - I_{eb}e^{-b/t_1-g/t_2} + I_2e^{-b/t_1-g/t_2} - I_{eg}e^{-g/t_2} \\
I_2 &= I_{eg}(1 - e^{-g/t_2}) + I_{eb}(1 - e^{-b/t_1})e^{-g/t_2} + I_2e^{-b/t_1-g/t_2} \\
I_2 - I_2e^{-b/t_1-g/t_2} &= I_{eg}(1 - e^{-g/t_2}) + I_{eb}(1 - e^{-b/t_1})e^{-g/t_2} \\
I_2(1 - e^{-b/t_1-g/t_2}) &= I_{eg}(1 - e^{-g/t_2}) + I_{eb}(1 - e^{-b/t_1})e^{-g/t_2} \\
I_2 &= \frac{I_{eg}(1 - e^{-g/t_2}) + I_{eb}(1 - e^{-b/t_1})e^{-g/t_2}}{1 - e^{-b/t_1-g/t_2}}
\end{aligned}$$

O uso da equação (18) pode inserir erro nas equações de determinação de $I_e(av)$ (20) e $I_e(fim\ de\ chamada)$ (21), resultando em uma obtenção errônea do MOS.