

Italo Tiago da Cunha

Balanceamento de Chamadas VoIP a Transcodificar

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre em Ciência da Computação pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Uberlândia



Programa de Pós-Graduação em Ciência da Computação – PPGCC

Faculdade de Computação – FCOM

Universidade Federal de Uberlândia – UFU

Uberlândia – Minas Gerais – Brasil

Fevereiro de 2009

Italo Tiago da Cunha

Balanceamento de Chamadas VoIP a Transcodificar

Dissertação apresentada ao programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Uberlândia, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Redes de Computadores.

Orientador: Prof. Dr. Jamil Salem Barbar.

UBERLÂNDIA – MG

Fevereiro de 2009

Dados Internacionais de Catalogação na Publicação (CIP)

C972b Cunha, Italo Tiago da, 1982-
Balanceamento de chamadas VoIP a transcodificar / Italo Tiago da
Cunha. - 2009.
200 f. : il.

Orientador: Jamil Salem Barbar.
Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Redes de computação - Teses. 2. Telefonia pela internet - Teses. 3. Algoritmo de balanceamento - Teses. 4. Codificador de voz - Teses. I. Barbar, Jamil Salem. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU: 681.3.02

Italo Tiago da Cunha

Balanceamento de Chamadas VoIP a Transcodificar

Dissertação apresentada ao programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Redes de Computadores.

Uberlândia – MG, 04 de fevereiro de 2009

Banca examinadora

Prof. Dr. Jamil Salem Barbar – FACOM/UFU (orientador)

Prof. Dr. João Cândido Lima Dovicchi – INE/CTC/UFSC

Prof. Dr. Luís Fernando Faina – FACOM/UFU

Dedicatória

Àqueles que todos os dias rezam por nossa família e que nada esperam em troca, que só querem que sejamos felizes, que já passaram e são capazes de tantos sacrifícios; que tantas e tantas vezes já se levantaram de madrugada para preparar o café da manhã e o lanche ou a marmita, a fim de que saíssemos bem alimentados para trabalhar ou estudar; que quantos dias já dormiram na boléia de um caminhão, tomando banho gelado de corote, passando meses sem retornar ao lar; que quantos e quantos bordados, doces e conservas já fizeram; ... e tudo isso, com um único intuito: o bem-estar da família.

Em suma, dedico esta dissertação a esses que sempre estiveram ao meu lado, apoiando-me, transmitindo-me segurança, fiéis exemplos de que não precisamos usar os outros como degraus e, com isso, guiam-me a fim de que seja um homem digno.

Pai “João Batista da Cunha”, mãe “Maria Tiago de Queiroz Cunha”, eu os amo.

Seu filho, Italo.

*Dissertação consagrada à mãe de
Nosso Senhor Jesus Cristo, a Virgem
Maria, sob o título de Nossa Senhora da
Conceição Aparecida, padroeira do Brasil.*

Agradecimentos

A Deus, pelas inúmeras oportunidades de crescimento que me proporciona.

A todos os amigos que me apoiaram para a realização deste trabalho.

À CAPES, pelo apoio financeiro concedido.

Ao Programa de Pós-Graduação em Ciência da Computação, bem como à Faculdade de Computação pela oportunidade.

Resumo

O presente trabalho trata da tecnologia VoIP (*Voice over Internet Protocol*), que utiliza o protocolo de rede IP (*Internet Protocol*) para prover comunicação por meio das redes de dados, proporcionadora de economia e mobilidade às chamadas. Para que uma chamada seja estabelecida e gerenciada, faz-se necessário o uso de Módulo Servidor de Sessão. Para tanto, pressupõe-se que os participantes disponham de idênticos *codecs*, funcionalidade responsável por transformar um fluxo binário de áudio em outro com a mesma propriedade da informação inicial. Caso sejam distintos, para que a chamada seja estabelecida, é necessária a transcodificação, que consiste na conversão de um fluxo binário de áudio em outro, requerendo o uso de *Media Gateway*, que a realiza, mas com considerável demanda de processador. Isso pode culminar em gargalo, quando for elevado o número de chamadas simultâneas que necessitam ser transcodificadas. Assim sendo, o objetivo do presente trabalho é prover o balanceamento de chamadas VoIP a transcodificar, o que se deu pelo desenvolvimento do Módulo Balanceador de Chamadas. Tal solução é original e proporciona resultados substanciais.

Palavras-chave: VoIP, transcodificação, *Media Gateway*, balanceamento de chamadas, Módulo Balanceador de Chamadas.

Abstract

The present work relates to the VoIP (Voice over *Internet* Protocol) technology that uses the network protocol, IP (*Internet* Protocol) to provide communication through data networks, giving economy and mobility to the calls. In order to establish and manage call it is necessary the use a Session Server Module. For this, it is supposed that the participants have identical codecs as well there is a functionality responsible to transform an audio binary stream into another with the same property of initial information. However, in order to establish calls in case of distinct codecs it is necessary the transcoding operation, which consists in the conversion of an audio stream into another, which requires the use of Media Gateways, that are able to do this, but with considerable processor demand. This may become a problem when the number of simultaneous calls to be transcoded increases. Knowing this, the aim of this work is to provide the balancing of VoIP calls that need to be transcoded that is possible by means of developing a Call Balancing Module. This is an original solution that provides concrete results.

Keywords: VoIP, transcoding, Media Gateway, call balancing, Call Balancing Module.

Lista de Acrônimos e Siglas

a – Attributes

A/D – Analógico-digital

ACD – Automatic Call Director

ACF – Admission Confirm

ACK – Acknowledge

ADPCM – Adaptive Differential PCM

API – Application Programming Interface

ARQ – Admission Request

ASN.1 – Abstract Syntax Notation One

ATA – Analog Telephone Adaptor

ATM – Asynchronous Transfer Mode

b – Bandwidth

B2BUA – Back-To-Back User Agent

c – Connection Data

CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CC – CSRC Count

CELP – Code Excited Linear Prediction

CNAME – canonical name

Codec – COder/DECoder

CSRC – Contributing Source

CTC – Centro Tecnológico

D/A – Digital-analógico

DAC – Distribuidor Automático de Chamadas

DDD – Discagem Direta a Distância

DDoS – *Distributed Denial of Service*
DNS – *Domain Name System*
DoS – *Denial of Service*
DPCM – *Differential PCM*
e – *E-mail Address*
e-mail – *Electronic Mail*
FACOM – *Faculdade de Computação*
FXO – *Foreign eXchange Office*
FXS – *Foreign eXchange Subscriber*
GB – *Gigabyte*
GHz – *Gigahertz*
HLEN – *Header Length*
HTTP – *Hypertext Transfer Protocol*
Hz – *Hertz*
i – *Session Information*
IAX – *Inter-Asterisk eXchange*
IETF – *Internet Engineering Task Force*
IHL – *IP Header Length*
INE – *Departamento de Informática e de Estatística*
IP – *Internet Protocol*
IPTV – *Internet Protocol Television*
ISDN – *Integrated Services Digital Network*
ISO – *International Organization for Standardization*
ITSP – *Internet Telephony Service Provider*
ITU – *International Telecommunication Union*
ITU-T – *ITU's Telecommunication Standardization Sector*
IVR – *Interactive Voice Response*
k – *Encryption Key*
kbps – *kilobits por segundo*
KHz – *Kilohertz*
LAN – *Local Area Network*
LARC – *Laboratório Avançado de Redes de Computadores*

LCF – *Location Confirm*
LPC – *Linear Predictive Coding*
LRJ – *Location Reject*
LRQ – *Location Request*
M – *Marker*
m – *Media Descriptions*
MAC – *Media Access Control*
Malware – *Malicious Software*
MB – *Megabytes*
MBC – *Módulo Balanceador de Chamadas*
Mbps - *Megabits por segundo*
MG – *Media Gateway*
MGCP – *Media Gateway Control Protocol*
MHz – *Megahertz*
MIB – *Management Information Base*
MITM – *Man in the Middle*
MOS – *Mean Opinion Score*
ms – *Milissegundos*
NAT – *Network Address Translation*
o – *Origin*
OpenSER – *Open SIP Express Router*
OSI – *Open Systems Interconnection*
P – *Padding*
p – *Phone Number*
P2P – *Peer-to-Peer*
PABX – *Private Automatic Branch eXchange*
PBX – *Private Branch eXchange*
PC – *Personal Computer*
PCI – *Peripheral Component Interconnect*
PCM – *Pulse Code Modulation*
PDA – *Personal Digital Assistants*
PPGCC – *Programa de Pós-Graduação em Ciência da Computação*

PPP – *Point-to-Point Protocol*
pps – pacotes por segundo
PSTN – *Public Switched Telephone Network*
PT – *Payload Type*
QoS – *Quality of Service*
r – *Repeat times*
RAM – *Random Access Memory*
RAS – *Registration, Admission and Status*
RFC – *Request for Comments*
RR – *Receiver Reports*
RS-232 – *Recommended Standard 232*
RSVP – *ReSerVation Protocol*
RTCP – *RTP Control Protocol*
RTP – *Real-time Transport Protocol*
RTSP – *Real Time Streaming Transport Protocol*
s – *Session Name*
SAP – *Session Announcement Protocol*
SCCP – *Skinny Client Control Protocol*
SDES – *Source Description*
SDP – *Session Description Protocol*
SIP – *Session Initiation Protocol*
SMTP – *Simple Mail Transfer Protocol*
SNMP – *Simple Network Management Protocol*
SPIT – *Spam over Internet Telephony*
SR – *Sender Reports*
SSRC – *Synchronization Source*
t – *Timing*
TCP/IP – *Transmission Control Protocol/Internet Protocol*
TDM – *Time Division Multiplexing*
TTS – *Text-To-Speech*
u – *URI*
UA – *User Agent*

UAC – *User Agent Client*
UAS – *User Agent Server*
UDP – *User Datagram Protocol*
UFU – Universidade Federal de Uberlândia
URA – Unidade de Resposta Audível
URI – *Uniform Resource Indicator*
URL – *Uniform Resource Locator*
v – *Protocol Version*
V – *Version*
VAD – *Voice Activity Detection*
VCB – Valor Calculado para o Balanceamento
VLAN – *Virtual LAN*
VoIP – *Voice over Internet Protocol*
WWW – *World Wide Web*
X – *Extension*
xDSL – *Digital Subscriber Line*
z – *Time Zone Adjustments*

Lista de Figuras

Figura 2.1 – PABX.	15
Figura 2.2 – Ilustração de um Cenário Utilitário de Centrex.....	16
Figura 2.3 – Arquitetura do Asterisk PBX.	18
Figura 2.4 – Telefonia IP.	21
Figura 2.5 – Cenários Iniciais da PSTN e das Redes de Dados.....	22
Figura 2.6 – Comunicação Unificada.	23
Figura 2.7 – Placa Analógica Digium TDM410.....	24
Figura 2.8 – Sinalização FXS e FXO.	24
Figura 2.9 – Placa Digital Digium TE205P.....	25
Figura 2.10 – Rede sem <i>Firewall</i>	35
Figura 2.11 – Rede com <i>Firewall</i> , Dados Permitidos.....	36
Figura 2.12 – Rede com <i>Firewall</i> , Dados Negados.....	36
Figura 2.13 – Processo de Criptografia.	37
Figura 2.14 – ATA.....	39
Figura 2.15 – Telefone IP.	40
Figura 2.16 – <i>Softphone X-Lite</i>	40
Figura 3.1 – Representação das Camadas do Modelo OSI.....	44

Figura 3.2 – Protocolos e Aplicativos Pertinentes a VoIP em Associação ao Modelo OSI.	47
Figura 3.3 – Campos do Pacote IPv4.	48
Figura 3.4 – Encapsulamento UDP.	51
Figura 3.5 – Datagrama UDP.	52
Figura 3.6 – Pseudocabeçalho UDP.	53
Figura 3.7 – Multiplexação e Demultiplexação UDP.	54
Figura 3.8 – Formato de um Pacote TCP.	56
Figura 3.9 – Cabeçalho RTP.	58
Figura 3.10 – Encapsulamento do Áudio.	61
Figura 3.11 – Arquitetura do Protocolo H.323.	63
Figura 3.12 – Sinalização H.323 Direta entre Terminais.	64
Figura 3.13 – Formato de uma Mensagem do Tipo <i>REQUEST</i>	68
Figura 3.14 – Formato de uma Mensagem do Tipo <i>RESPONSE</i>	69
Figura 3.15 – Estabelecimento de uma Chamada Utilizando o SIP.	71
Figura 3.16 – Descrição de uma Sessão SDP.	74
Figura 4.1 – Ilustração do Uso de um <i>Codec</i> em VoIP.	76
Figura 4.2 – Processo de Conversão Analógico para Digital.	77
Figura 4.3 – Processo de Conversão Digital para Analógico.	78
Figura 4.4 – Digitalização do Sinal.	78
Figura 4.5 – Representação da Faixa de Frequência.	79
Figura 4.6 – Sinal de Voz Amostrado.	80
Figura 4.7 – Sinal de Voz Quantizado.	80
Figura 4.8 – Sinal de Voz Codificado.	81

Figura 4.9 – Comparação dos Três Principais Tipos de Codificadores.....	82
Figura 4.10 – Codificação DPCM.	84
Figura 4.11 – Decodificação DPCM.....	85
Figura 4.12 – Ilustração de uma Comunicação VoIP.	88
Figura 4.13 – Efeito do <i>Jitter</i> na Percepção da Fala.	90
Figura 4.14 – Chamada VoIP não Completada.	93
Figura 4.15 – Chamada VoIP Estabelecida por Meio de <i>Media Gateway</i>	94
Figura 5.1 – Esquema Ilustrativo das Etapas do Trabalho.	99
Figura 5.2 – Ilustração de Parte da Árvore da MIB.....	101
Figura 5.3 – Ramo <i>systemStats</i> da Árvore da MIB.....	103
Figura 5.4 – Caminho do Objeto <i>ssCpuIdle (11)</i> na Árvore da MIB.....	104
Figura 5.5 – Obtenção do Percentual de Carga Disponível no Processador.....	104
Figura 5.6 – Ramo <i>memory</i> da Árvore da MIB.	105
Figura 5.7 – Caminho do Objeto <i>memAvailReal (6)</i> na Árvore da MIB.	105
Figura 5.8 – Obtenção do Total de Memória Física Disponível.....	106
Figura 5.9 – Cenário de Aplicabilidade do Módulo Balanceador de Chamadas.....	107
Figura 5.10 – Ilustração do Sub-módulo Acionador.....	109
Figura 5.11 – Ilustração do Sub-módulo Verificador.	110
Figura 5.12 – Ilustração do Sub-módulo Atualizador.....	110
Figura 5.13 – Ilustração do Sub-módulo Indicador de MG.....	111
Figura 5.14 – Esquema de Funcionamento do Módulo Balanceador de Chamadas.....	112
Figura 5.15 – Cenário de Implementação do Módulo Balanceador de Chamadas.....	117
Figura 5.16 – Tela de Configuração de Conta do <i>X-Lite</i>	118
Figura 5.17 – Servidor de Chamadas.....	119

Figura 5.18 – Módulo Servidor de Sessão.	120
Figura 5.19 – Ilustração do Estabelecimento de Chamada.	121
Figura 5.20 – Parte do Código Fonte para Registro dos Participantes nos <i>Media Gateways</i>	122
Figura 5.21 – Parte do Código Fonte para Acionar o Módulo Balanceador de Chamadas.	122
Figura 5.22 – Código Fonte do Sub-módulo Acionador.	123
Figura 5.23 – Código Fonte do Sub-módulo Verificador.	124
Figura 5.24 – Código Fonte do Sub-módulo Atualizador.	125
Figura 5.25 – Código Fonte do Sub-módulo Indicador de MG.	126
Figura 5.26 – <i>Media Gateway</i>	127
Figura 5.27 – Código Fonte do Arquivo <i>res_mysql.conf</i>	128
Figura 5.28 – Código Fonte do Arquivo <i>extconfig.conf</i>	128
Figura 5.29 – Código Fonte do Arquivo <i>extensions.conf</i>	129
Figura 5.30 – Cenário com <i>Media Gateway</i> Inativo.	130
Figura 5.31 – Cenário de <i>Media Gateway</i> com Elevado Percentual de Memória Disponível.	131
Figura 5.32 – Cenário de <i>Media Gateway</i> com Elevado Percentual Disponível de Processador.	133
Figura 6.1 – Cenário de Testes do Módulo Balanceador de Chamadas.	136
Figura 6.2 – Teste 1, Participantes com Idênticos <i>Codecs</i>	139
Figura 6.3 – Linha de Tempo para o Teste 1.	140
Figura 6.4 – Teste 2, Participantes com <i>Codecs</i> Distintos.	142
Figura 6.5 – Linha de Tempo para o Teste 2.	143
Figura 6.6 – Teste 3, com <i>Media Gateway</i> Ativo e Inativo.	148

Figura 6.7 – Linha de Tempo para o Teste 3.....	149
Figura 6.8 – Teste 4, com <i>Media Gateways</i> Inativos.....	154
Figura 6.9 – Linha de Tempo para o Teste 4.....	155

Lista de Tabelas

Tabela 4.1 – Medidas de Qualidade em VoIP.	91
Tabela 4.2 – Pontuação de MOS.	91
Tabela 4.3 – Alguns <i>Codecs</i> e suas Respectivas Taxas de Transmissão e MOS.	92
Tabela 4.4 – Tempos de Transcodificação pelo <i>Media Gateway</i> 1 (<i>asterisk1</i>).	95
Tabela 4.5 – Tempos de Transcodificação pelo <i>Media Gateway</i> 2 (<i>asterisk2</i>).	95
Tabela 6.1 – Chamadas do Teste 1, com Participantes de Idênticos <i>Codecs</i>	139
Tabela 6.2 – Momentos e Intervalos de Tempo do Teste 1.	141
Tabela 6.3 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 1.	141
Tabela 6.4 – Chamadas do Teste 2, Participantes com <i>Codecs</i> Distintos.	143
Tabela 6.5 – Momentos, Intervalos de Tempo e <i>Media Gateway</i> do Teste 2.	145
Tabela 6.6 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 2.	146
Tabela 6.7 – Chamadas do Teste 3, com <i>Media Gateway</i> Ativo e Inativo.	149
Tabela 6.8 – Momentos, Intervalos de Tempo e <i>Media Gateway</i> do Teste 3.	151
Tabela 6.9 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 3.	152
Tabela 6.10 – Teste 4, com <i>Media Gateways</i> Inativos.	155
Tabela 6.11 – Momentos e Intervalos de Tempo do Teste 4.	156
Tabela 6.12 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 4.	157
Tabela 6.13 – VCBs do <i>Media Gateway</i> 2 nos Testes 2 e 3.	158

Sumário

1 Introdução.....	1
2 Tecnologia VoIP.....	8
2.1 Aplicabilidade da VoIP.....	10
2.1.1 OpenSER	11
2.1.2 Asterisk PBX	13
2.1.2.1 PBX VoIP	14
2.1.2.2 Arquitetura do Asterisk PBX	17
2.1.2.3 Funcionalidades e Características	19
2.1.2.4 Telefonia IP.....	21
2.1.2.5 <i>Hardware</i>	23
2.1.2.6 Protocolos de Sessão e <i>Codecs</i> Suportados	26
2.1.3 Interação entre o OpenSER e o Asterisk PBX.....	26
2.2 Segurança em VoIP	28
2.2.1 Vulnerabilidades em VoIP.....	30
2.2.2 Ataques Decorrentes das Vulnerabilidades	31
2.2.3 Medidas de Segurança	33
2.2.3.1 VLAN.....	33

2.2.3.2 <i>Firewall</i>	34
2.2.3.3 Criptografia em VoIP	37
2.3 Utilitários da VoIP	38
2.3.1 ATA.....	39
2.3.2 Telefone IP	39
2.3.3 <i>Softphone</i>	40
2.4 Conclusão.....	41
3 Protocolos Pertinentes a Tecnologia VoIP	43
3.1 Protocolo IP	47
3.2 UDP.....	50
3.2.1 Formato do Datagrama UDP	51
3.2.2 Pseudocabeçalho UDP	53
3.2.3 Multiplexação e Demultiplexação do UDP.....	54
3.3 TCP.....	55
3.4 Protocolo RTP	58
3.5 RTCP	60
3.6 H.323	62
3.7 SIP	65
3.7.1 SIP URIs.....	67
3.7.2 Elementos da Arquitetura SIP	67
3.7.3 Requisições SIP.....	68
3.7.4 Respostas SIP	70
3.7.5 Estabelecimento de uma Sessão SIP	70

3.8 SDP	71
3.8.1 Sintaxe e Campos do SDP	72
3.9 Conclusão.....	74
4 Codificação da Voz	76
4.1 Representação dos Sinais A/D e D/A	77
4.1.1 Amostragem.....	78
4.1.2 Quantização	80
4.1.3 Codificação	81
4.1.3.1 Codificação por Forma de Onda	82
4.1.3.1.1 PCM	83
4.1.3.1.2 DPCM e ADPCM	84
4.1.3.2 Codificação Paramétrica	85
4.1.3.2.1 Codificação LPC	86
4.1.3.3 Codificação Híbrida	86
4.1.3.3.1 CELP	87
4.2 Avaliação de Áudio	87
4.2.1 Qualidade do Som.....	88
4.2.2 Qualidade da Voz.....	89
4.2.3 MOS.....	91
4.3 Transcodificação.....	93
4.4 Conclusão.....	97
5 Balanceamento de Chamadas VoIP a Transcodificar	98
5.1 SNMP.....	100

5.1.1 Processador.....	103
5.1.2 Memória	104
5.2 O Módulo Balanceador de Chamadas	106
5.2.1 Cálculo do VCB	114
5.3 Implementação do Módulo Balanceador de Chamadas	116
5.3.1 Participantes A e B.....	118
5.3.2 Servidor de Chamadas.....	119
5.3.2.1 Módulo Servidor de Sessão	119
5.3.2.2 Implementação do Módulo Balanceador de Chamadas	123
5.3.2.2.1 Sub-módulo Acionador	123
5.3.2.2.2 Sub-módulo Verificador.....	124
5.3.2.2.3 Sub-módulo Atualizador	125
5.3.2.2.4 Sub-módulo Indicador de MG.....	126
5.3.3 <i>Media Gateway</i>	127
5.4 Cenários de Atuação do Módulo Balanceador de Chamadas	129
5.5 Conclusão	134
6 Validação da Proposta	135
6.1 Teste 1, Participantes com Idênticos <i>Codecs</i>	138
6.2 Teste 2, Participantes com <i>Codecs</i> Distintos	142
6.3 Teste 3, com <i>Media Gateway</i> Ativo e Inativo.....	147
6.4 Teste 4, com <i>Media Gateways</i> Inativos	153
6.5 Conclusão	157

7 Considerações Finais.....	159
Referências.....	163
Anexo A - Códigos Fonte dos Arquivos de Configuração.....	175
A.1 - <i>openserctrlrc</i>	175
A.2 - <i>openser.cfg</i>	178
A.3 - <i>snmpd.conf</i>	186
Anexo B - Logs das Chamadas Realizadas nos Testes.....	188
B.1 - Teste 1, Participantes com Idênticos <i>Codecs</i>	188
B.2 - Teste 2, Participantes com <i>Codecs</i> Distintos	190
B.3 - Teste 3, com <i>Media Gateway</i> Ativo e Inativo	192
B.4 - Teste 4, com <i>Media Gateways</i> Inativos	195
Anexo C - Logs dos VCBs nos Testes.....	197
C.1 - Teste 1, Participantes com Idênticos <i>Codecs</i>	197
C.2 - Teste 2, Participantes com <i>Codecs</i> Distintos	198
C.3 - Teste 3, com <i>Media Gateway</i> Ativo e Inativo	199
C.4 - Teste 4, com <i>Media Gateways</i> Inativos	200

Capítulo 1

Introdução

A *Internet*, até o início da década de 1990, era um verdadeiro reduto de pesquisadores ligados às universidades, ao governo e à indústria. A troca de informações começava a tornar-se indispensável entre as pessoas e as corporações. Por exemplo, o correio eletrônico foi uma das primeiras aplicações que aos poucos começou a substituir as correspondências, de forma rápida, segura e confiável, se comparado com os outros meios de comunicação existentes na época. Aos poucos, a *Internet* passava a ocupar um lugar fundamental, de destaque, na área da comunicação, abrangendo os aspectos pessoais, profissionais e comerciais da sociedade. Assim, a realidade cotidiana mudou [15] [61].

A disponibilização de informações ou conteúdos fez-se mais presente e uma nova abordagem dessa disponibilização ocorria mediante hipertextos, permitindo o uso de imagens, sons e sofisticadas interfaces multimídia, por meio de localizadores URL (*Uniform Resource Locator*) e levando a *Internet* a uma nova denominação, conhecida como teia de alcance mundial ou WWW (*World Wide Web*). Consequentemente, milhares de novos usuários foram atraídos para a rede. A *Internet* nunca foi ou será apenas um único serviço. Atualmente, existe uma diversidade muito grande de aplicações que a utilizam. Basicamente, estas aplicações, que são constituídas de um ou mais serviços, trabalham sob o paradigma cliente-servidor [3].

Na *Internet*, as aplicações para disponibilizarem os serviços se comunicam entre si por meio dos protocolos, que são um conjunto de regras padronizado que especifica o

formato, a sincronização, o sequenciamento e a verificação de erros na comunicação dos dados. Portanto, é uma descrição formal de formatos de mensagem e das regras que dois ou mais processos, no mesmo computador ou em computadores diferentes, devem obedecer, ao trocarem mensagens. Uma aplicação da *Internet* pode fazer uso de um ou mais protocolos, adequando-se, basicamente, ao modelo de referência TCP/IP (*Transmission Control Protocol/Internet Protocol*) [30].

A diversidade de serviços que a *Internet* disponibiliza é extremamente grande, entre os quais, aqueles relacionados à troca de mensagens escritas por pessoas, as comunicações envolvendo mídias de vídeo e áudio, e as mensagens de dados para transferências de arquivos. Essas trocas podem ser feitas em tempo real ou não. Assim, as aplicações para a *Internet*, compostas de um ou mais serviços, foram sendo desenvolvidas de acordo com a demanda do seu uso, com um crescimento vertiginoso.

No início da década de 2000, no ápice da disseminação da *Internet*, mesma época em que a modalidade de *Internet* banda larga foi disponibilizada no Brasil, uma variedade de aplicações surgiu para usufruir do aumento da largura de banda de rede decorrente do aprimoramento tecnológico. Essas aplicações permitiram o uso de serviços de comunicação multimídias, tais como rádio *online*, *Skype*, *NetMeeting*, *Net2Phone*, *Dialpad* e outras. Entre elas estão as que fazem uso da tecnologia VoIP (*Voice over Internet Protocol*), que consiste na transmissão de voz sobre as redes de dados, e que demandam considerável largura de banda de rede [3] [4] [36] [64].

O tráfego de uma mensagem em rede de dados, no caso das redes de comutação de pacotes, é constituído, basicamente, por pacotes, com uma diversidade de tipos de serviços a que eles se originam, nas quais a tecnologia VoIP se faz presente. Assim, a competição pela banda de rede é grande, pois, a princípio, não há prioridade entre os pacotes.

Quanto maior o tamanho de uma mensagem que trafega pela rede de dados, maior será o número de pacotes criados, que serão encaminhados à rede em consonância com a possibilidade por demanda. Proporcionalmente, quanto maior for o número de pacotes de uma mensagem, maior será a possibilidade de ocorrer algum descarte por intervenção do controle de fluxo e de congestionamento da rede, decorrente da limitação das capacidades disponíveis de seus recursos, tais como largura de banda, *buffers* dos roteadores entre outros.

As comunicações de voz entre pessoas distantes, em tempo real, foram inicialmente realizadas por meio de terminais telefônicos da PSTN (*Public Switched Telephone Network*). Com o advento da *Internet* e da tecnologia VoIP, surgiram aplicações que possibilitaram uma forma alternativa para essas comunicações de voz. O presente trabalho trata da tecnologia VoIP, cujo princípio de funcionamento tem como base a conversão da voz em uma sequência binária, por meio de amostragens e respectivas conversões em pacotes de dados para serem posteriormente transmitidos até a aplicação receptora de destino, via rede de dados. Tais transmissões ocorrem em ambos os sentidos, ou seja, ora um dos participantes constitui-se no transmissor, ora no receptor [75].

Quanto maior for a sequência binária convertida da voz, maiores serão as mensagens a serem transmitidas, as quantidades de pacotes necessários, a banda de rede requerida e, conseqüentemente, os descartes de pacotes que podem causar falha no entendimento da mensagem de voz entregue pela aplicação receptora.

O uso de um algoritmo de codificação e decodificação, denominado *codec* (*COder/DECoder*), nas sequências binárias de voz pode diminuir consideravelmente o número de pacotes, com o ganho de praticidade no fluxo de transmissão e manutenção relevante da qualidade da voz reproduzida pelo áudio na parte receptora. Tal *codec*, conseqüentemente, possibilita uma redução significativa da banda de rede pela redução da probabilidade de perda ou descarte. Já na parte receptora, ocorre o processo inverso com a aplicação de um algoritmo de decodificação para a reconstrução da mensagem de áudio [49] [75].

Ao utilizar-se o *codec*, a redução ocorre na fase de amostragem em conformidade com ele. Com isso, a transmissão da sequência binária de voz ocorre de forma compactada até a parte receptora, que, por intermédio do mesmo *codec* utilizado para a codificação, realiza a respectiva decodificação e apresentação da amostragem capturada no início do processo, mantendo, praticamente, a mesma qualidade do áudio reproduzido no destino [49] [52].

Usualmente, os participantes de um sistema de comunicação de voz via *Internet*, usam a tecnologia VoIP e contam com aplicações clientes que se relacionam entre si por intermédio de um Módulo Servidor de Sessão, modo pelo qual se permite a mobilidade e a escalabilidade. A comunicação de fato, que transporta a informação relativa à voz, é

feita entre os participantes por meio de protocolos, com funções de tempo real, sendo o RTP (*Real-time Transport Protocol*), definido pela RFC (*Request for Comments*) 1889, o mais utilizado. Já a comunicação entre as aplicações cliente e o Módulo Servidor de Sessão é feita por um protocolo de sessão. Nesta pesquisa utiliza-se o protocolo de sessão SIP (*Session Initiation Protocol*), o qual é o mais utilizado atualmente [30] [68] [52] [69].

Um sistema de comunicação via VoIP, ou seja, um ambiente VoIP, é caracterizado pela existência de dois elementos bem distintos: o Módulo Servidor de Sessão e o participante, com possibilidade, ainda, de haver ou não o terceiro, o *Media Gateway*, os quais, exceto o Módulo Servidor de Sessão, podem ter mais de uma instância. Em geral, existem duas ou mais instâncias de participantes e cada qual pode figurar em máquinas distintas.

Uma chamada em VoIP caracteriza-se por ser uma comunicação entre participantes desse sistema. No entanto, as conceituações de chamada e comunicação diferem, visto que, mesmo quando não haja atendimento e, por conseguinte, não se obtenha efetividade da interação propositada, não se pode desconsiderar que a chamada de fato ocorreu, pois recursos foram alocados para tal, apesar de, usualmente, considerar-se comunicação quando do atendimento à chamada, com o estabelecimento do respectivo fluxo de áudio entre os comunicantes.

O estabelecimento de uma chamada preconiza que os participantes já tenham sido registrados no Módulo Servidor de Sessão, ao qual o participante de origem requisita inicialmente autorização a fim de estabelecer a comunicação. Satisfeitas as permissões e restrições, servidor retorna-lhe com o endereço do participante de destino, para que ocorra o processo de comunicação. O Módulo Servidor de Sessão é também responsável pela bilhetagem da chamada, cujo término é feito por qualquer um dos participantes, a qualquer momento, após seu início, com o envio ao mesmo do sinal de término, que executa as funções pertinentes ao evento e comunica, em seguida, aos outros participantes.

Assim, num cenário de comunicação por meio do uso da VoIP, os participantes, necessariamente, devem registrar-se em um Módulo Servidor de Sessão que centralize as respectivas e pertinentes informações. A comunicação pode ocorrer entre dois ou

mais participantes, quando se configura uma conferência, em conformidade com as características do sistema, a exemplo do *Skype* [4].

Com o uso da tecnologia VoIP, há ocasiões em que se faz necessária a utilização de *Media Gateways*, pela sua capacidade de interconexão de chamadas, a exemplo da VoIP com a PSTN, e vice-versa. E, ainda, quando nem todas as aplicações-clientes dos participantes de um sistema de comunicação via VoIP possuam *codecs* compatíveis entre si, o que lhes impede a comunicação, problema que se resolve com o emprego dos *Media Gateways*, por serem servidores especializados na transcodificação, ou seja, na tradução de fluxos de áudio encapsulados por diferentes *codecs* em outros [68] [69] [75] [79].

Num sistema de comunicação via VoIP, em que se utilizam *Media Gateways*, estes podem constituir-se em gargalo, por requererem consideráveis recursos do sistema computacional como um todo, em especial o processador, a exemplo da transcodificação e interconexão com a PSTN [44] [47] [77].

Logo, em um ambiente VoIP, os recursos computacionais podem exaurir-se e acarretar a parada do sistema de comunicação, quando constituído por um único Módulo Servidor de Sessão e vários *Media Gateways* submetidos a elevado número de chamadas simultâneas com necessidade de transcodificação. Tal fato pode ocorrer devido ao fato de as chamadas não serem distribuídas equitativamente e carecerem de específico e adequado monitoramento.

O objetivo geral desta dissertação é apresentar uma solução para o balanceamento de chamadas entre os *Media Gateways* quando houver a necessidade de transcodificação. Como objetivos específicos tem-se o planejamento e a implementação de um Módulo Balanceador de Chamadas, que interage com o Módulo Servidor de Sessão.

Para tanto, o Módulo Servidor de Sessão foi preparado para acionar o Módulo Balanceador de Chamadas, cuja finalidade é indicar qual *Media Gateway* é o mais adequado ao atendimento dos participantes de um sistema VoIP, quando não conseguem comunicar-se devido à incompatibilidade entre os *codecs*.

Com o uso do Módulo Balanceador de Chamadas, ocorre maior disponibilidade no sistema VoIP, bem como distribuição equitativa das chamadas carentes de

transcodificação entre os *Media Gateways* que compõem o sistema, evitando-se, assim, a sobrecarga.

O Módulo Balanceador de Chamadas possibilita a avaliação dos recursos computacionais dos *Media Gateways*, para, a partir da ponderação dos recursos neles disponíveis, selecionar-se o mais adequado à chamada entrante no sistema, o que corresponde a uma métrica para a carga ou esforço a que um *Media Gateway* deva estar submetido. Dessa forma, o Módulo Servidor de Sessão ao receber a chamada que necessita de transcodificação, seleciona o *Media Gateway* que deve atendê-la, o que é feito por intermédio do Módulo Balanceador de Chamadas, proporcionando aos *Media Gateways* serem submetidos a uma carga em valores assemelhados.

Para que o Módulo Balanceador de Chamadas identifique os percentuais dos recursos computacionais disponíveis nos *Media Gateways*, é utilizado o protocolo de gerência de redes, o SNMP (*Simple Network Management Protocol*), descrito na RFC 1157 [31].

Para o desenvolvimento da proposta, criou-se um ambiente em que os testes foram realizados reproduzindo o cenário real de um sistema via VoIP, com a utilização de cinco máquinas tipo PC (*Personal Computer*), entre as quais duas para atuarem como *Media Gateways*, uma como Módulo Servidor de Sessão e as demais como participantes de origem e destino das chamadas VoIP, com respectivos *codecs*, com a ênfase da função autenticadora pelo Módulo Servidor de Sessão aos usuários do sistema VoIP, além dos *Media Gateways* a ele subordinados.

Em suma, o Módulo Balanceador de Chamadas permite sistemas de alta escalabilidade, além do balanceamento de chamadas o mais ponderado possível, por possibilitar que *Media Gateways* sejam acrescentados e ou substituídos de forma rápida, prática e simples ao sistema VoIP; ainda propicia o mínimo de descarte de chamadas que necessitem de transcodificação, apresentando-se como solução inovadora, visto não ter sido encontrado na literatura pesquisada informação pertinente à problemática do balanceamento de chamadas que necessitem de transcodificação, o que torna justificável o presente trabalho [39] [59].

Quanto a trabalhos correlatos, a única referência encontrada na literatura diz respeito ao balanceamento de chamadas VoIP com destino à PSTN, para um ambiente em específico, e quando do encaminhamento das chamadas, não leva em consideração

qualquer métrica de monitoramento de carga dos *Media Gateways*, haja vista que apenas um número conhecido de chamadas é atendido nesse ambiente, o qual é inerente ao número de interfaces disponíveis com a PSTN [40].

A presente dissertação está estruturada em sete capítulos. A esta introdução seguem-se as descrições dos demais seis capítulos resumidamente.

O Capítulo 2 apresenta a tecnologia VoIP, suas peculiaridades e empregabilidade. É apresentado o estado da arte da VoIP, considerando os aspectos de implementação e segurança.

No Capítulo 3 são mostrados os protocolos pertinentes à tecnologia, desde os protocolos da camada de rede aos da camada de aplicação. Os principais protocolos são abordados, entre eles, o IP (*Internet Protocol*), o RTP e o SIP.

Já o Capítulo 4 introduz os conceitos referentes à codificação da voz, em todos os processos a que ela deve ser submetida para que possa ser encaminhada por meio das redes de dados até ser apresentada ao destino. Aborda ainda as principais formas de codificação.

No Capítulo 5 é apresentado descritivamente o Módulo Balanceador de Chamadas, abordando sua concepção e implementação como uma solução ao balanceamento de chamadas VoIP a transcodificar. Os detalhes para implementação são abordados.

O Capítulo 6 elenca os testes realizados para comprovar a eficácia do Módulo Balanceador de Chamadas. Os resultados dos testes são comentados.

Já o Capítulo 7 encerra o trabalho com as considerações finais, apontando os pontos positivos e negativos da pesquisa realizada, incluindo as contribuições à comunidade científica, bem como sugestões de melhorias para este trabalho e para trabalhos futuros.

O Anexo A apresenta os códigos fonte dos arquivos de configuração. Já o Anexo B traz os *logs* das chamadas realizadas nos testes. E por fim, o Anexo C apresenta os *logs* dos VCBs nos testes.

Capítulo 2

Tecnologia VoIP

A VoIP (*Voice over Internet Protocol*) consiste numa tecnologia, e não em um serviço, para a comunicação de voz por meio das redes de dados, pela utilização de um conjunto de *software*, *hardware* e padrões que habilitam o transporte da voz por meio do protocolo *Internet*, o IP (*Internet Protocol*) [1] [2] [62] [68].

Essa tecnologia já se tornou muito comum e utilizada por 43% das empresas mundiais, de acordo com levantamento realizado com 1,5 mil profissionais de 45 países, feito pela *Systimax Solutions*, cujo resultado traduz um incremento de 152% em relação à pesquisa realizada anteriormente, em 2002, quando apenas 17% das empresas participantes a adotavam [43].

A VoIP, em comparação com a telefonia convencional, utiliza a comutação por pacotes, via *Internet*, e não por circuitos, o que a provê de maior potencial de aproveitamento dos recursos existentes, maximizando os canais anteriormente usados apenas para dados. Em outras palavras, pode-se ter diferentes formas de ser interconectar com operadoras de várias partes do planeta, sem a necessidade de conexão física com elas, pela ampliação das opções para envio do seu tráfego de telefonia.

Com o surgimento da *Internet* e as possibilidades de telefonia a ela pertinentes, bem como com o uso de tecnologias mais acessíveis, a transmissão de voz dessa forma tem-se tornado mais viável do que pelo modelo convencional utilizado na PSTN [3].

Tal inovação tem alterado o setor de telefonia de maneira a fragilizar as empresas tradicionalmente constituídas, pois se trata de um fenômeno de mercado

oportunizador à diversificação de opções de serviços aos usuários com a disponibilização de benefícios facilmente absorvíveis por eles [17].

O crescimento acelerado do mercado mundial das plataformas VoIP e a integração de voz, vídeo e dados discretos em um único sistema amplia a potencialidade dessa tecnologia, pois uma rede convergente IP pode transmitir diversos tipos de dados, sem a necessidade de redes distintas para cada serviço, como, por exemplo, de telefonia, TV ou acesso à *Internet*.

Com isso, grandes operadoras de serviços de comunicação vislumbram ampliar sua participação nesse segmento com inestimáveis oportunidades de exploração, em razão de sua capacidade de convergência, o que é confirmado por estudos apresentados pelo instituto de pesquisas *Infonetics* de que, somente na América do Norte, a previsão de receitas para as operações via VoIP, apenas no mercado doméstico, constitua US\$ 23,4 bilhões em 2009, com perspectiva de crescimento para o patamar de 75% dos serviços globais de transporte de voz, justificada pela redução geral de custos, melhor gerenciamento e suporte a aplicações multimídia avançadas [2] [68].

As operadoras têm, gradativamente, disponibilizado a tecnologia VoIP em seus serviços, haja vista sua maturidade e convergência a múltiplas operações, como auxiliar imprescindível à competitividade contemporânea no setor de telefonia. Disso decorre o fato de as empresas e consumidores disporem das mais amplas opções para serviços de comunicação.

Nesse sentido, um dos desafios da tecnologia VoIP consiste em proporcionar o mesmo nível de confiabilidade dos sistemas de transmissão convencionais, com disponibilidade equivalente a 99,99% da PSTN, o que garante a realização da comunicação com qualidade e segurança, sem interceptações, modificações ou distorções [73].

Assim, em VoIP faz-se necessário impedir à utilização do sistema por pessoas não autorizadas, o que é possível por meio de autenticação. Para tanto, a VoIP requer alta disponibilidade e tempo de manutenção atualizada da rede de dados, que necessita prover qualidade ao respectivo serviço, visto que o atraso de apenas um segundo pode comprometer acentuadamente sua qualidade [2] [62] [73].

É importante ressaltar que a rede necessita suportar altos volumes de tráfego e integrar os conceitos multimídia de seus serviços. Deste modo, os provedores VoIP

passam a ser competitivos, por oferecerem serviços inteligentes e simplificados. Por conseguinte, pela utilização de tal tecnologia, torna-se possível proporcionar ao usuário condições de optar convenientemente pela maneira e pelo equipamento mais apropriado para realizar ou receber uma chamada de posse de uma única identificação [21].

Apesar de a VoIP já ser conhecida há tempos, somente a partir de 2004, com sua viabilização econômica e a crescente demanda por mobilidade corporativa, as redes de dados e voz iniciaram o processo que culminou em sua veloz convergência e implantação em larga escala; em consequência, iniciaram-se as preocupações referentes ao tráfego de pacotes com áudio, que possuem especificidades distintas daquelas peculiares aos pacotes de dados [5] [29] [64].

A convergência como fator imperativo em plataformas híbridas, ou seja, aquelas que mesclam VoIP com PSTN, potencializou investimentos em soluções na tecnologia VoIP, em decorrência de sua competitividade pela oportunização de uma gama ampliada de serviços aos usuários.

De outro modo, pode-se afirmar que os sistemas híbridos têm-se destacado na provisão de recursos IP e na compatibilização com a infraestrutura de troncos e ramais, pelo que se vislumbra que a convergência de dados, vídeo e voz que já norteia os investimentos na transmissão de dados sobre IP, afigura-se, sobretudo, com excepcional perspectiva em futuro próximo, pela aplicabilidade da VoIP [34] [62] [75]. A seção seguinte aborda a referida aplicabilidade da VoIP.

2.1 Aplicabilidade da VoIP

Com a evolução tecnológica, tornou-se possível o aprimoramento das redes de dados, outrora apenas transmissoras de dados, para contemplarem também o transporte de voz. A *Internet*, atualmente, é amplamente difundida e o usuário, ao se valer da comunicação por seu intermédio, beneficia-se dos respectivos custos reduzidos a valores ínfimos, possibilitados pela usualidade e aplicabilidade da VoIP para a respectiva comunicação [75].

A VoIP é amplamente empregada, em razão de possibilitar larga aplicabilidade, que provê funcionalidades inerentes à transmissão da voz, a exemplo de realização de

conferência, de correio de voz, de Telefonia IP, entre outras, inclusive aquelas usuais ao modelo de PBX (*Private Branch eXchange*) convencional, se já portado para a VoIP como um PBX VoIP [77].

É peculiaridade da VoIP sua abertura à interação ou integração com tecnologias já existentes, a exemplo da VoIP com a PSTN, do que decorre a Telefonia IP. Isso demonstra sua consonância com a dinâmica do processo de evolução das inovações no campo da comunicação, possibilitador do refinamento ou incremento de funcionalidades da VoIP, observado no *software* Asterisk PBX [68] [77].

Para a efetiva utilização da tecnologia VoIP, conta-se com recursos viabilizadores de vários fins que a potencializam, a exemplo do *software* OpenSER para o gerenciamento de sessão, cuja especificidade concerne em proporcionar acentuado volume de chamadas em curto espaço de tempo, especificamente no âmbito de VoIP, com usuários de *codecs* idênticos e que façam uso indispensavelmente do protocolo de sessão SIP (*Session Initiation Protocol*) [38]. Na próxima seção é apresentado o *software* OpenSER.

2.1.1 OpenSER

O OpenSER (*Open SIP Express Router*) é um servidor SIP robusto, flexível, em *software* livre, escrito em C puro, criado para manusear infraestruturas de grande porte, para gerenciar centenas de milhares de usuários [38].

Pode ser usado em sistemas com recursos limitados, assim como em servidores de grande porte, tornando possível uma escalabilidade rápida para milhares de ligações por segundo, a exemplo de testes já realizados, em que o OpenSER registrou mais de quatro milhões de usuários e atendeu a mais de 28 milhões de chamadas por hora [38].

Embora especialmente desenvolvido para sistemas operacionais Unix/Linux, com otimizações específicas, o OpenSER pode ser utilizado também com Solaris e com o FreeBSD, com oferta de alto desempenho e estabilidade [38].

O OpenSER pode ser indicado para provedores VoIP, também conhecidos como ITSP (*Internet Telephony Service Provider*) e serviços de revenda de minutos; ainda

conta com capacidade de rápido crescimento em número de assinantes, com reduzida necessidade de investimento em *hardware*.

O OpenSER gerencia todas as ligações entrantes e saíntes de uma rede VoIP, bem como intermedeia controles de acesso e preferências de usuários e ou domínios. Com isso, tarefas essenciais, como integração com banco de dados e contabilidade das ligações completadas, também são por ele executadas.

São funcionalidades do OpenSER atuar como [38] [45] [79]:

- *SIP proxy server*: entidade responsável por receber mensagens e encaminhá-las. Possui as funcionalidades de autorização, tradução de endereços, autenticação, aplicação de regras de segurança por meio do controle de acesso à rede, bem como roteamento de chamadas;
- *SIP registrar server*: entidade responsável por manter atualizadas as informações de registro sobre os UAs (*User Agents*) e ou autenticá-los, bem como compartilhá-los com outros servidores da rede;
- *SIP location server*: entidade utilizada para identificar as possíveis localizações dos destinos chamados;
- *SIP application server*: entidade que permite a novas aplicações estenderem as funcionalidades do OpenSER;
- *SIP dispatcher server*: entidade que implementa um distribuidor de endereços de destino, o que é feito computando *hashes* de partes da requisição e selecionando respectivo endereço para destino.

Não obstante as atribuições referenciadas do OpenSER, ele se mostra inviável como [38] [45]:

- *Media server*: o OpenSER não trabalha com o fluxo de mídia, mas apenas com a sinalização, pois não executa transcodificação nem mesmo a interconexão com a PSTN;
- *Back-To-Back User Agent* (B2BUA): entidade que mantém o completo estado da transação, e faz-se participativo em todas as requisições, razão pela qual pode desempenhar várias funções impraticáveis ao *SIP proxy server*, que apenas mantém seu estado.

Em suma, pelo fato de o OpenSER atuar única e exclusivamente com a sinalização, sua abrangência quantitativa assume cifras consideráveis, o que o torna referência no que tange ao trato de chamadas VoIP em larga escala, utilitárias do protocolo de sessão SIP e de mesmos *codecs* [38].

Apesar da elevada potencialidade do OpenSER para chamadas em larga escala, ele carece de funcionalidades supridas pelo Asterisk PBX, abordado na seção seguinte [38].

2.1.2 Asterisk PBX

O Asterisk PBX constitui-se em um PBX VoIP completo em *software*, baseado no paradigma *Open Source*. Foi criado em 1999 por Mark Spencer e, nos dias atuais, desponta como um dos principais agentes da revolução nas telecomunicações [6] [79].

No final do século passado, mais precisamente em 1999, um programador americano, Mark Spencer, decidiu seguir na contramão das grandes empresas, pois sua pretensão era apenas criar um sistema para sua, até então, pequena empresa, que pudesse atender e armazenar as chamadas não atendidas. Atualmente o Asterisk PBX é uma das grandes promessas no que diz respeito ao *software* livre e, tem e está mudando os conceitos conhecidos em telefonia, haja vista ser de código aberto, ultra flexível e gratuito [6] [79] [105].

Seu diferencial está em sua natureza, totalmente aberta e customizável, complementada por uma série de padrões atendidos. Nenhum outro equipamento de PBX pode implementar as funcionalidades operadas pelo Asterisk PBX nem conta com sua respectiva diversidade [79].

O Asterisk PBX suporta uma vasta gama de protocolos TDM (*Time Division Multiplexing*) para o gerenciamento e transmissão de voz nas interfaces convencionais de telefonia. Suporta também os padrões de sinalização americanos e europeus, usados em grande parte dos sistemas de comunicação, permitindo com isto um elo entre a próxima geração de telefonia, que integra dados e voz, com a infraestrutura atual [79] [80] [81].

Desse modo, não somente suporta a telefonia tradicional como lhe agrega novas funções, por possuir um núcleo central de chaveamento, constituído por quatro APIs (*Application Programming Interface*), cada qual com funções distintas, desde manuseio de formatos de arquivos e *codecs* ao carregamento modular das aplicações de telefonia e interfaces de *hardware*. Com isso, é possível comutação transparente entre todas as interfaces suportadas, o que por conseguinte possibilita a disponibilização de uma série de sistemas de telefonia numa única rede de comutação [80] [81].

Como qualquer *software* baseado na filosofia *Open Source*, o Asterisk PBX recebe milhares de contribuições de programadores de todo o mundo diariamente, o que auxiliou no desenvolvimento de várias de suas funções, entre as quais se destacam o correio de voz, as salas de audioconferências, as unidades de resposta audível e o atendimento automático.

O correio de voz é responsável por armazenar recados quando uma chamada não pode ser atendida e avisar ao usuário de destino sobre o acontecimento. Já as salas de audioconferência criam ambientes para “reuniões” por telefone, enquanto as unidades de resposta audível atendem às ligações e, por meio de menus, direcionam-nas para os locais mais apropriados.

Em síntese, o Asterisk PBX consiste em um sistema convergente de telefonia baseado em código livre, que foi concebido inicialmente para ser suportado pelo sistema operacional Linux. O mesmo ainda reúne mais de cem anos de conhecimento em telefonia, em uma suíte de *softwares* altamente integrados.

Embora, originalmente desenvolvido para Linux, atualmente suporta vários outros sistemas operacionais, como OpenBSD, FreeBSD, OS X e Solaris. Seu nome é uma alusão ao caractere * (asterisco), a máscara que representa qualquer nome de arquivo no sistema Linux [79] [80].

2.1.2.1 PBX VoIP

O telefone convencional é, na grande maioria das vezes, conectado diretamente à operadora local de telefonia, o que permite por meio dele a realização de chamadas pela discagem do número de destino desejado. No entanto, já em um ambiente corporativo,

usualmente existem mais ramais do que linhas telefônicas, principalmente devido ao custo, o que suscita a necessidade de um ponto central para o gerenciamento e a distribuição das chamadas, o que é realizado pelo PABX (*Private Automatic Branch Exchange*) [52] [69]. A Figura 2.1 ilustra o exemplo de um PABX.

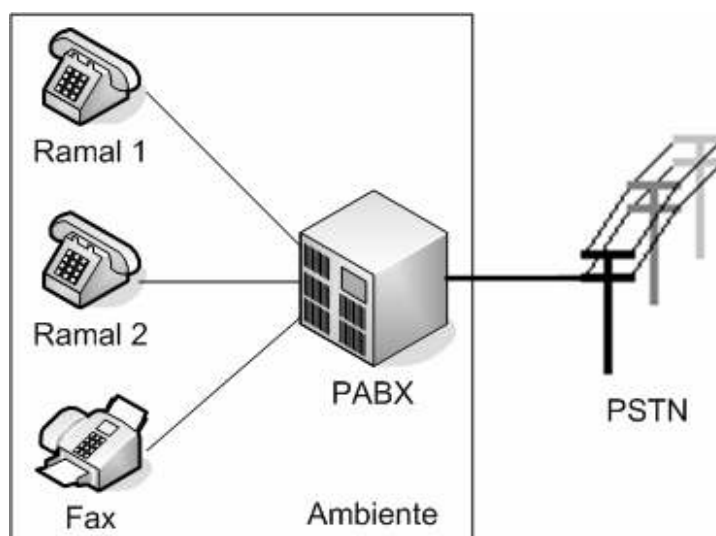


Figura 2.1 – PABX.

Pela análise da Figura 2.1 depreende-se que qualquer aparelho contido no ambiente de abrangência do PABX pode interconectar-se tanto interna quanto externamente à PSTN via sua respectiva central.

Pode-se afirmar que um PABX constitui uma central telefônica, cuja função, em essência, baseia-se em comutação de circuitos e consiste na possibilidade de distribuição de uma ou mais linhas telefônicas num ambiente, ou seja, ligar uma série de telefones a uma linha externa como um elemento de controle, permitindo efetuar ligações entre ramais internos sem intervenção manual, ou mesmo realizar chamadas para PSTN ou dela recebê-las e, ainda gerencia permissões de uso dos ramais individualmente ou por grupos. O PABX oferece vários outros serviços, incluindo chamada em espera, conferência, transferência de chamadas, autoatendimento e correio de voz, entre outros [52] [69].

O termo original para a designação das centrais telefônicas usadas nas empresas era PBX (*Private Branch eXchange*), atribuído aos equipamentos que necessitavam da intervenção manual de um operador para completar chamadas. Com a evolução

tecnológica, eles foram modernizados, alcançando a automação, a partir da qual passaram a ser denominados PABX. No entanto, é necessário atenção quanto ao uso da terminologia, pelo fato de referir-se ao mesmo conceito, significando um centro de distribuição telefônica, de propriedade de uma empresa cuja atividade não incluía o fornecimento de serviços telefônicos ao público em geral, com *modos operandus* distintos, manuais obsoletos, o PBX, ou automáticos, o PABX [7] [106].

Em aprimoramento ao referido modelo, surgiu na década de 1990 o PABX Virtual, quando a indústria de telecomunicações instigou as operadoras de telefonia a substituírem as centrais telefônicas, até então eletromecânicas, pelas emergentes centrais digitais, que propiciavam maiores benefícios aos seus usuários, por agregarem mais funcionalidades que as anteriores.

Exemplo disso foi a possibilidade da oferta aos assinantes de linhas telefônicas convencionais caracterizadas como se fossem ramais de um PABX, distribuídas a partir da central telefônica da operadora, ou seja, um Centrex. A Figura 2.2 ilustra um cenário utilitário de Centrex [37] [52].

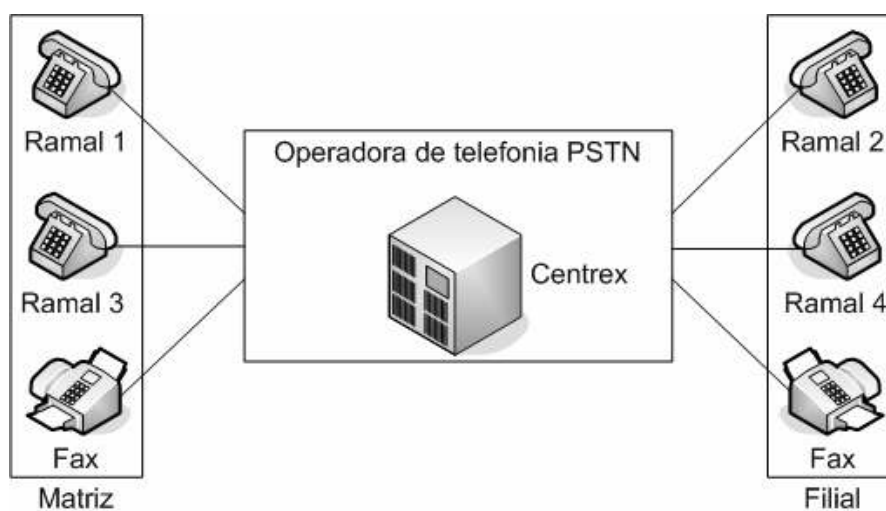


Figura 2.2 – Ilustração de um Cenário Utilitário de Centrex.

A Figura 2.2 exemplifica um modelo de PABX virtual Centrex, que se situa na operadora de telefonia PSTN, e não nos ambientes componentes do cenário, como no modelo convencional.

Mais recentemente, com o aprimoramento tecnológico e, em especial, o advento da tecnologia VoIP, ela foi incorporada ao modelo então usual, configurando-se uma inovadora modalidade denominada PBX VoIP, que mantém a topologia de um PABX centralizado, não na operadora de telefonia, mas em um ambiente de rede, a exemplo de um *data center*, cujos ramais não mais são convencionais, mas utilizadores da VoIP [69].

Por suas características, ele se encaixa muito bem em alguns cenários como empresas em ascensão, na abertura de novas filiais, ou nas quais o PABX se saturou, e as que requeiram necessidades especiais, tais como mobilidade ou serviços de correio de voz, URAs (Unidade de Resposta Audível) configuráveis e acesso *online* a bilhetagem das chamadas, entre outros [37] [69].

Pelas substanciais potencialidades agregadas ao PBX VoIP, na atualidade, ocorre sua prevalência em decorrência das funcionalidades da aplicação da tecnologia VoIP, que disponibiliza vantagens, a exemplo de:

- Mobilidade: desta forma, basta conectar-se à rede de dados, a exemplo da *Internet*, para ter disponível um ramal VoIP e, assim, estar apto à comunicação;
- Flexibilidade: a realocação de ramais é facilmente resolvida, conectando-se o dispositivo provedor da interface de comunicação à rede de dados;
- Escalabilidade: para acrescentar novos ramais, basta habilitá-los ao sistema, ao contrário do que ocorre no modelo PABX convencional, cuja solução é onerosa.

O *software* Asterisk PBX é um excelente exemplo do uso da VoIP, o qual foi utilizado nesta dissertação com a função de *Media Gateway*, pela sua flexibilização e por implementar um fiel cenário de PBX VoIP [65] [66].

2.1.2.2 Arquitetura do Asterisk PBX

A arquitetura do Asterisk PBX é composta, basicamente, de [80] [81]:

- Canais: entendidos como toda entrada ou saída de um sinal, quer seja de natureza digital, analógica ou VoIP, de quaisquer de suas interfaces;

- Protocolos de comunicação: SIP, MGCP (*Media Gateway Control Protocol*), SCCP (*Skinny Client Control Protocol*), H.323 e IAX (*Inter-Asterisk eXchange*), responsáveis pela sinalização das chamadas;
- *Codecs*: constituem-se os elementos que codificam e decodificam as informações das comunicações VoIP, responsáveis por adaptarem a quantidade de *bits* que serão transmitidos para a rede, sempre com a meta de deixar a qualidade da voz o mais próximo possível do natural, ou, pelo menos, com o padrão já alcançado pela telefonia tradicional;
- Aplicações: funcionalidades acopladas ao sistema, responsáveis pela operacionalidade do PBX, a exemplo de correio de voz, conferência e URA.

A arquitetura do Asterisk PBX é ilustrada na Figura 2.3 [80] [81].

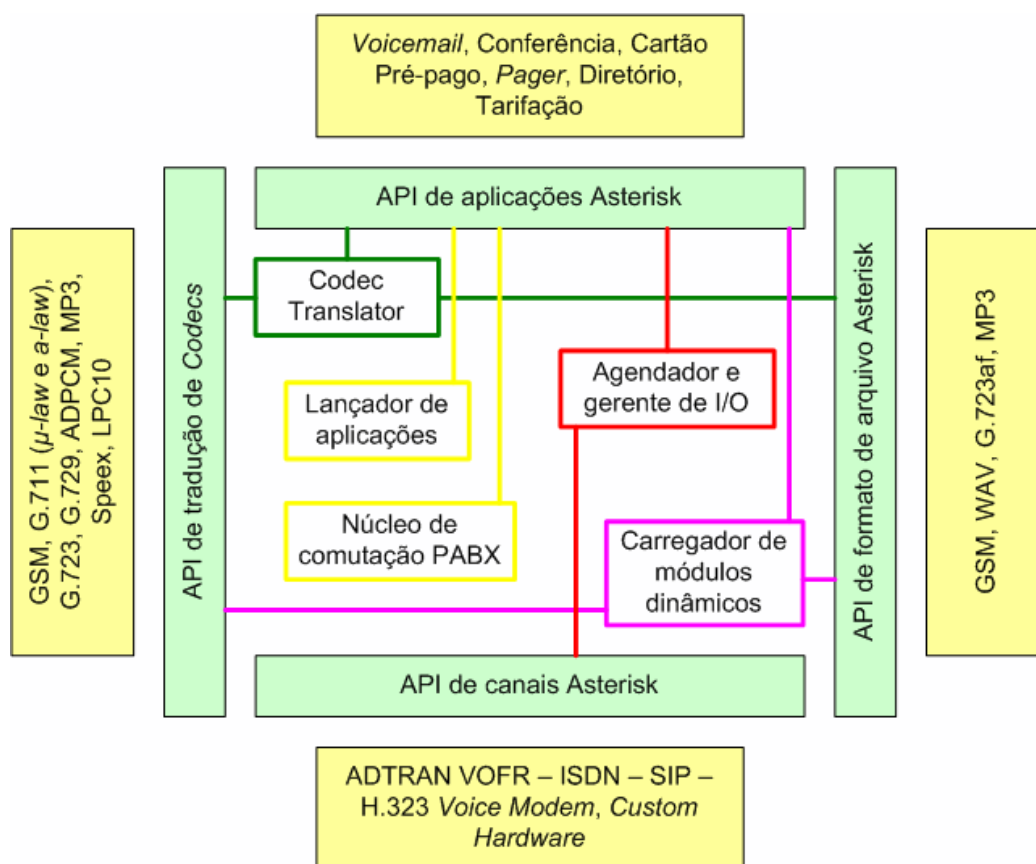


Figura 2.3 – Arquitetura do Asterisk PBX.

Visualiza-se, na Figura 2.3, a arquitetura geral do Asterisk PBX em que se explicitam os seus elementos arcabouço, descritos anteriormente.

Enfatiza-se ser o Asterisk PBX essencialmente um PBX VoIP que suporta múltiplos protocolos utilizados em VoIP, tais como SIP, MGCP, SCCP, H.323, IAX, que permite tanto a utilização de telefones em *software* “*softphone*” como equipamentos e dispositivos telefônicos VoIP, como ATA (*Analog Telephone Adaptor*) e Telefone IP [79].

O Asterisk PBX também possui a capacidade de interoperar com os sistemas de telefonia tradicionais, PSTN, o que somente é possível pela adição de *hardware* peculiar. Pode ser usado em inúmeras aplicações, desde um PBX VoIP para uma pequena empresa a sistemas de resposta automática de alta densidade [64] [68].

2.1.2.3 Funcionalidades e Características

As principais funcionalidades e características incorporadas e, portanto, inerentes ao Asterisk PBX são [79] [80] [81]:

- Correio de voz “*Voicemail*”: semelhante a uma caixa postal tradicional, entretanto, ao invés de cartas, armazena mensagens de voz, inclusive, com a opção de enviá-las aos usuários via *e-mails* (*Electronic Mail*);
- Unidade de Resposta Audível (URA), ou IVR (*Interactive Voice Response*): atendimento eletrônico das ligações com a possibilidade de integração com outras aplicações e bancos de dados, de prover a criação de menus de navegação e de reconhecimento de voz. Permite programação de forma flexível ao atendimento de chamadas, em vista a dar-lhes algum tipo de resposta sem interação humana, por intermédio da escolha entre as opções expostas pelo respectivo emissor. Possibilidade de reprodução de áudio, leitura de texto e devolução de informações a partir de uma base de dados. Interação com o sistema por intermédio do toque em teclas do telefone, a exemplo do sistema de votação via telefone, ou o simples acesso a uma conta bancária, pela inserção do número da conta e respectiva senha via teclado, seguindo ordens pré-definidas no sistema;

- Distribuidor Automático de Chamadas (DAC) ou *Automatic Call Director* (ACD): distribui as chamadas entrantes no dispositivo, grupo ou serviço, utilizando algoritmos de distribuição aos agentes que estão logados no respectivo serviço, com maior tempo livre ou menor tempo acumulado. Quando utilizado em *call centers*, possibilita o gerenciamento de filas, prioridades das chamadas e direcionamento para agentes específicos;
- Conferência de áudio: cria salas de conferência que possibilitam a vários usuários conversarem simultaneamente;
- Discador automático: gera chamadas a partir de uma base de dados, que contém números de telefones e as direcionam para respectivos ramais ou agentes;
- Servidor de música de espera “*Music On Hold*”: constituído por vários formatos de arquivos de áudio reproduzidos de forma síncrona ou assíncrona, que ocorrem apenas ao se colocar uma chamada corrente em espera, ou antes do seu atendimento, de modo que quem originou a chamada possa ouvi-los enquanto aguarda o atendimento;
- Registro detalhado das ligações: relatórios completos inerentes às ligações, como duração, origem, destino e custo, a partir de cujas informações pode-se monitorar a utilização do Asterisk PBX e detectar padrões e ou possíveis anomalias;
- Ramal local e remoto: ramais podem estar geograficamente distantes, entretanto participando do mesmo serviço VoIP, desde que haja acesso via rede de dados, por possuírem independência geográfica e
- Interoperabilidade entre diferentes protocolos: o Asterisk PBX suporta, praticamente, todos os protocolos utilizados atualmente na telefonia convencional e VoIP. Portanto, a migração e interligação com sistemas híbridos é altamente facilitada;
- Distribuição de chamada recebida: o Asterisk PBX permite receber uma chamada telefônica, olhar para seus atributos e tomar decisões de encaminhamento com base no seu conteúdo. Exemplos de como encaminhar uma chamada podem ser seu envio para uma extensão única, para um grupo de extensões e gravá-la.

As funcionalidades e características descritas constituem em seu conjunto o Asterisk PBX.

2.1.2.4 Telefonia IP

A utilização da VoIP em PBX VoIP ilustra o desenvolvimento tecnológico da transmissão de voz por meio das redes de dados. Deste modo, é suplantada a comutação tradicional pela utilização consonante da VoIP com a PSTN na Telefonia IP, bem como em outras usualidades [68] [69].

Ou seja, a VoIP, como alternativa de comunicação, é buscada por usuários de linhas telefônicas convencionais, por meio da modalidade de Telefonia IP, que se constitui na associação da respectiva tecnologia à PSTN, em cenários nos quais até recentemente a telefonia convencional constituía a única alternativa. A Figura 2.4 ilustra a Telefonia IP.

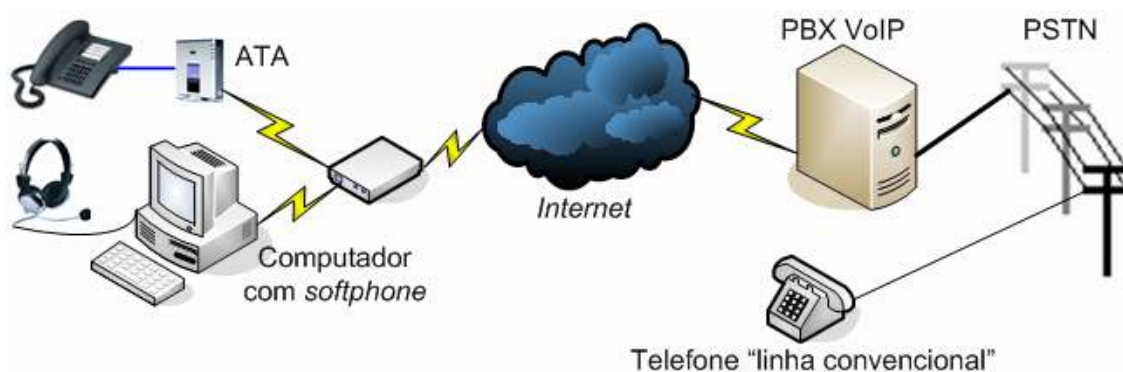


Figura 2.4 – Telefonia IP.

Se incorporada a aparelhos de PABX, ou aos *Media Gateways* por meio de placas dotadas de interfaces analógicas ou digitais, torna possível a realização da comunicação da VoIP para PSTN, de PSTN para VoIP ou mesmo da VoIP para VoIP conforme já ilustrado na Figura 2.4 [55].

Empresas de diversos segmentos, pelo uso da Telefonia IP alcançam economia nas comunicações, com cifras entre 70% e 80%, por reduzir custos com deslocamentos

e integrar infraestruturas de dados, vídeo e áudio, com controle eficaz na segmentação de chamadas [8] [34] [43].

Sua peculiar natureza integradora permite a redução do número de fornecedores de equipamentos na oferta de serviços, tanto de telefonia, rede de dados, conferência e videoconferência, com segurança e monitoramento, do que decorre maior produtividade, em função de mobilidade, coordenação e colaboração, ilustrados respectivamente nos cenários das Figuras 2.5 e 2.6 [23] [88].

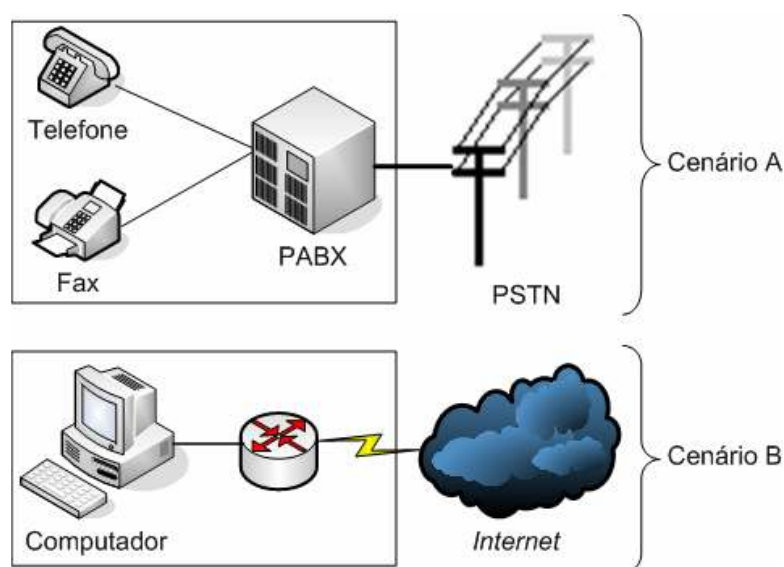


Figura 2.5 – Cenários Iniciais da PSTN e das Redes de Dados.

Constata-se, pela análise do cenário A da Figura 2.5, que a comunicação de voz, em seu início, restringia-se única e exclusivamente à PSTN, enquanto, no cenário B da Figura 2.5, as redes de dados, como a *Internet*, disponibilizavam apenas o suporte à transmissão de dados.

Com o advento da VoIP, a comunicação de voz, que até então era restrita a PSTN, passa a ser possível, quer também via VoIP, ou por sua associação à PSTN. Isso acarretou um melhor aproveitamento dos recursos inerentes à transmissão de voz, proporcionando mobilidade e, em especial, a redução de custos. A Figura 2.6 é ilustra a evolução mencionada.

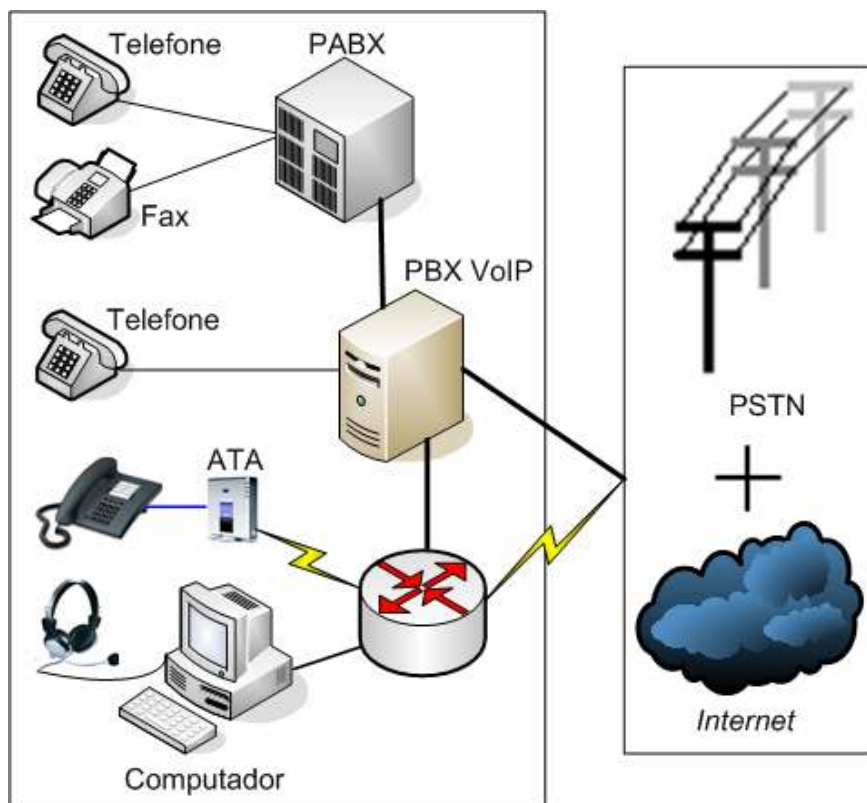


Figura 2.6 – Comunicação Unificada.

A Figura 2.6 mostra que a comunicação passa a ser unificada, haja vista que a VoIP vale-se das redes de dados, pelo que estas passam a transmitir não apenas dados mas também voz. Deste modo, funcionalidades, como receber o próprio correio de voz personalizado via *e-mail* ou ter ramal IP no *smartphone* ou *notebook*, tornam-se realidades corriqueiras.

2.1.2.5 Hardware

A rede PSTN, de início, oferecia apenas linhas em modo analógico. Com a evolução tecnológica, contudo, tem passado a oferecer também linhas de natureza digital. Em Telefonia IP, para que chamadas VoIP possam alcançar a PSTN, ou esta alcançar a VoIP, tem-se o recurso da utilização do *software* Asterisk PBX, que proporciona por meio de placas analógicas ou digitais suporte à rede PSTN em ambas as modalidades de transmissão, ou seja, analógica e ou digital.

As placas analógicas são aquelas dotadas de interfaces RJ-11 e módulos FXO (*Foreign eXchange Office*), FXS (*Foreign eXchange Subscriber*), ou ambos. Os módulos FXO ou FXS associam-se às referidas interfaces em uma relação de um para um. A Figura 2.7 ilustra uma placa analógica [89].

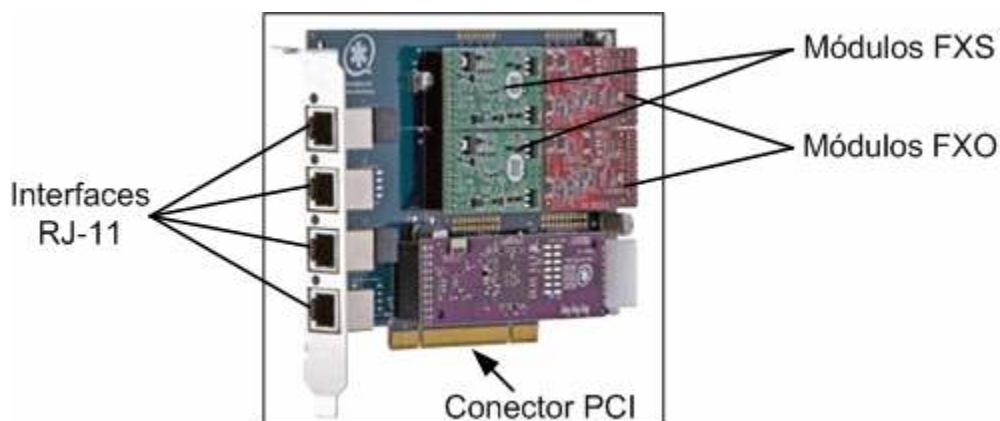


Figura 2.7 – Placa Analógica Digium TDM410.

Tal placa é instalada no barramento de expansão PCI (*Peripheral Component Interconnect*) do computador e suporta quatro interfaces RJ-11, que tanto podem servir módulos FXO quanto de FXS.

Os módulos FXS e FXO constituem-se nas portas utilizadas por linhas telefônicas analógicas da PSTN. O FXS constitui-se o fornecedor da linha analógica ao assinante, ou seja, fornece o tom de discagem, corrente de energia e som. O FXO, ao contrário, constitui-se no receptor da linha analógica. Ambos são ilustrados na Figura 2.8 [71].



Figura 2.8 – Sinalização FXS e FXO.

A Figura 2.8 ilustra um aparelho telefônico analógico conectado diretamente à linha da operadora de telefonia PSTN, que fornece sinalização FXS, recebida pelo cliente pelo módulo FXO.

Já as placas digitais são aquelas que permitem a comunicação com a PSTN por meio de canais digitais. Estes canais, por sua vez, podem ser tanto E1 quanto T1, com interface RJ-48. A Figura 2.9 ilustra uma placa digital [89].

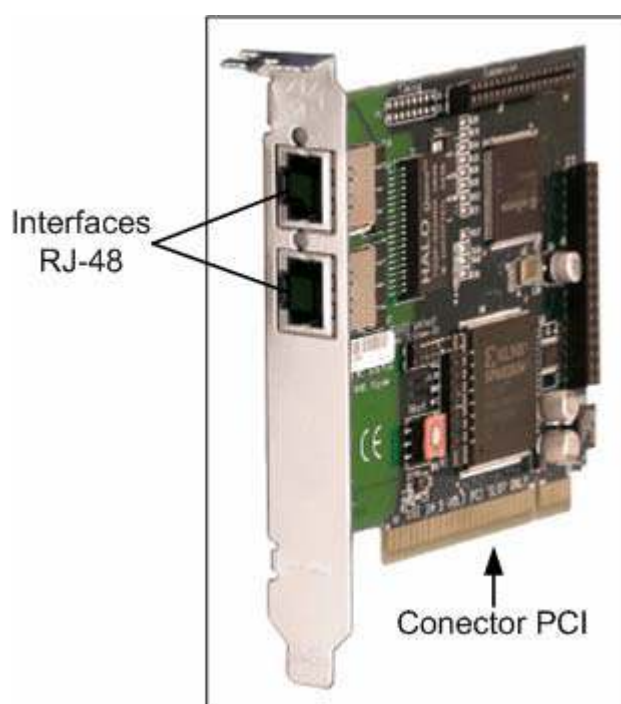


Figura 2.9 – Placa Digital Digium TE205P.

Em telefonia, é frequente o uso de linhas com alta velocidade e, conseqüentemente, de grande capacidade designadas pelo termo tronco, usando PCM (*Pulse Code Modulation*), conjuntamente com a multiplexagem TDM (*Time Division Multiplexing*). Constituem-se padrões de sistemas de transmissão por linhas digitais os enlaces T1, utilizado nos Estados Unidos e Japão, bem como o E1, adotado na Europa e Brasil [52] [90].

O meio de transmissão T1 possui taxa de transmissão de 1544 Mbps (*Megabits* por segundo), constitui-se por 24 canais de voz digitalizados (PCM24), cada qual com velocidade respectiva de 64 kbps (*kilobits* por segundo). O E1 é também chamado de

“Link E1”, “enlace digital” ou “2 mega”, um meio de transmissão de 2048 Mbps, 32 canais digitais, cada qual com velocidade respectiva de 64 kbps, sendo 30 canais de voz digitalizados (PCM30) ou dados, um para sincronismo e outro para sinalização [52] [82].

2.1.2.6 Protocolos de Sessão e *Codecs* Suportados

Quanto aos protocolos de sessão, o Asterisk PBX suporta os principais, senão todos os utilizados atualmente nas comunicações via VoIP, tais como o H.323, IAX e o SIP, descritos no Capítulo 3. Assim sendo, o Asterisk PBX possui peculiar versatilidade, o que lhe possibilita a realização de pontes entre os diversos protocolos de sessão por ele suportados, do mesmo modo com os *codecs* de áudio.

Em referência aos *codecs*, o Asterisk PBX dota-se da peculiaridade de suportar grande diversidade destes, o que o torna capaz de viabilizar a compatibilidade entre modalidades já em desuso, obsoletas, ou mesmo que venham a ser criadas.

O Asterisk PBX é um *software* que integra todas as peculiaridades inerentes à VoIP, o que lhe confere aptidão especial para aplicabilidade da VoIP em geral. Contudo, tal característica apesar de potencializá-lo, apresenta-se, por outro modo, como uma potencial restrição à sua usualidade, pela limitação ao atendimento de grandes volumes de chamadas. Ou seja, sua excelência qualitativa não condiz com a sua potencialidade quantitativa [38].

Tal dificuldade, contudo, pode ser solucionada pelo OpenSER, quando da utilização do protocolo de sessão SIP e mesmos *codecs* pelos usuários da VoIP, pois é apto à realização de funções específicas, no caso, o tratamento da sinalização [38].

2.1.3 Interação entre o OpenSER e o Asterisk PBX

O Asterisk PBX é um B2BUA (*Back-To-Back User Agent*), enquanto o OpenSER consiste em um *SIP Proxy*, o que os diferencia, pois a arquitetura de um *SIP Proxy* é mais rápida do que a de um B2BUA, por adequar-se à utilização da sinalização SIP [79].

De outro modo, um B2BUA, mesmo sendo um pouco mais lento, é capaz de gerenciar a mídia e vários serviços não disponibilizados em um *SIP Proxy*, tais como tradução entre *codecs*, a exemplo da transcodificação do G.729 para G.711, bem como de SIP para H.323. Ainda propicia serviços relacionados à mídia, tais como URA, DAC, TTS (*Text-To-Speech*) e reconhecimento de voz [38] [45] [80] [81].

O OpenSER permite acesso de baixo nível ao protocolo de sessão, neste caso, o SIP, pois, pode gerir todos os seus respectivos pedidos e respostas, razão pela qual lhe é possível, na maioria das vezes realizar traduções de versões incompatíveis de SIP. Contudo, não é capaz de atender nenhum serviço relacionado a mídia [38].

Dessa forma, não lhe é possível possibilitar funcionalidades, tais como correio de voz, URA, TTS e reconhecimento de voz. No entanto, pode valer-se de tais serviços pela utilização de um servidor de mídia em separado para esse fim, a exemplo do Asterisk PBX, do *Yate*, do *FreeSWITCH* ou do *SEMS*. Isso é decorrente da concepção estrutural desenvolvida para o protocolo SIP, definida pela norma RFC (*Request for Comments*) 3261 [38] [45] [70] [83] [84] [85].

O OpenSER sempre carecerá de um *gateway* para se conectar à rede pública, pois não existe a possibilidade de conectar-se a interfaces, tais como placas digitais e ou analógicas para provimento de Telefonia IP. Assim, o Asterisk PBX é largamente utilizado como seu *gateway* [38] [45] [80] [81].

O fato de o Asterisk PBX e o OpenSER serem complementares faculta ao OpenSER proporcionar serviços robustos referentes ao protocolo SIP, assim como atendimento de grande número de chamadas e balanceamento da respectiva carga, por dotar-se de excelência incomparável no trato com a sinalização SIP. Além disso, torna-se capaz de resolver cenários avançados de NAT (*Network Address Translation*).

Já o Asterisk PBX é um B2BUA muito forte no mercado de PBX VoIP, por sua simplicidade de configuração, além de estar apto a gerenciar volumes de chamadas de pequenos a moderados. Por suas funcionalidades, o Asterisk PBX pode ser considerado um verdadeiro *canivete suíço*, no que diz respeito a VoIP, pela sua multiplicidade funcional [38] [45] [80] [81].

O OpenSER tem sido muito utilizado por provedores com alta demanda e por universidades. Já o Asterisk PBX, em segmentos de menor demanda de chamadas VoIP, passíveis de serem satisfeitas por médios e pequenos provedores, nos quais, em geral, o

volume de usuários é inferior a 1000. Acima de tal cifra, costuma-se usar o OpenSER [38] [79].

2.2 Segurança em VoIP

A VoIP como qualquer outra tecnologia, apesar da eficácia, é suscetível a problemas de segurança e vulnerabilidades, quer peculiares quer inerentes à rede de dados e PSTN, que a tornam vulnerável a ataques [74].

Um descuido corriqueiro, ao se avaliar a segurança em VoIP, consiste, por exemplo, em tratá-la como uma tecnologia comum, pois se trata de um serviço complexo, que funciona em tempo real, e portanto, necessita de cuidados especiais. Dessa forma, implementar as medidas de segurança clássicas de um ambiente de comunicação de dados caracteriza a falta de domínio ao padrão de segurança adequado, já que a VoIP não é somente mais uma aplicação ativa sobre a infraestrutura IP, como o correio eletrônico e os serviços *web* [19] [76].

Assim, a utilização das redes de dados pode ocasionar riscos de segurança a que as mesmas estão suscetíveis e, por conseguinte, impactam de modo comprometedor e significativo a VoIP. Em razão disso, uma análise acurada da segurança em VoIP faz-se imprescindível, para possibilitar a disponibilidade de seus serviços, com a garantia da qualidade esperada pelo usuário.

Para alcançar os requisitos de confiabilidade próximos a 99,99% do tempo de disponibilidade, equivalente a menos de cinco minutos de *downtime* por ano, cerca de menos de um segundo por dia, usual à PSTN, a VoIP deve ser capacitada para reconhecer e impedir ataques à sua infraestrutura em segundos. Isso em virtude de sua elevada sensibilidade, o que, contudo, lhe permite implementar ações preventivas e automatizadas de resposta e bloqueio a ataques em tempo real, previamente, pela detecção deles, evitando impactos na qualidade e disponibilidade do serviço de voz, ou seja, medidas de segurança [19] [73].

Conforme o exposto, a segurança tem assumido cada vez maior relevância aos usuários e prestadores de serviços de VoIP e requer maiores cuidados preventivos e mitigadores a ataques à disponibilidade, integridade e confidencialidade, com

interferências respectivas nos âmbitos de indisponibilidade ou degradação do serviço, pela fraude de identidades e vazamento de informações, entre outros, a exemplo de *malware*.

O *malware* (*Malicious Software*), ou código malicioso, é uma expressão genérica que contempla amplamente os tipos de programas especificamente desenvolvidos para executar ações maliciosas em computadores. Pode danificar e comprometer dados e servidores e, com isso, constituir-se em possível ponto de partida para novos ataques. Exemplificam-no: vírus, *worms* e *bots*, *backdoors*, cavalos de tróia, *rootkits*, *keyloggers* e outros programas *spyware* [67].

A garantia de segurança em VoIP requer uma política de informação que leve em conta a conscientização dos envolvidos para êxito na implementação das medidas que visem a alcançar as soluções de possíveis problemas. Deve-se definir de forma clara o que pode ou não ser feito com as informações, bem como as regras de utilização dos sistemas pelos usuários. E pautar-se no envolvimento conscientizador dos integrantes do processo, para de fato ser seguido, de acordo com as regras pré-definidas, de modo plausível.

Os usuários do sistema devem ser conscientizados sobre a relevância da segurança da informação e treinados para o sucesso de sua implementação, pois, conscientes, tornar-se-ão auxiliares partícipes na consolidação da segurança.

Um computador, ou sistema computacional, é dito seguro, se atende aos requisitos básicos relacionados com os recursos que o compõem: confidencialidade, integridade e disponibilidade [30] [67].

A confidencialidade requer que a informação só seja disponibilizada para quem devidamente autorizado; a integridade, que a informação não seja destruída ou corrompida bem como o sistema tenha um desempenho correto; e a disponibilidade, a acessibilidade dos serviços ou recursos do sistema sempre que necessários [67].

A vulnerabilidade é definida como uma falha no projeto, implementação ou configuração de um *software* ou mesmo do sistema operacional que, quando explorada por um atacante, resulta na violação da segurança de um computador [67].

Existem casos em que um *software* ou sistema operacional instalado em um computador pode ter uma vulnerabilidade que permita sua exploração remota, ou seja, por meio da rede. Portanto, um atacante conectado à *Internet*, ao explorar tal

vulnerabilidade, pode obter acesso não autorizado ao computador vulnerável, tema de discussão em VoIP da próxima seção.

2.2.1 Vulnerabilidades em VoIP

As vulnerabilidades possibilitadoras de ameaças à infraestrutura de uma rede VoIP, muitas das vezes inerentes a falhas humanas, entre outras, segundo [78] [91]:

- Falhas de execução: grande parte dos problemas referentes a falhas de execução decorrem da má implementação de filtros e de programação;
- verificação insuficiente de dados: permite que intrusos acessem a rede com o intuito de atacá-la;
- pouca largura de banda: necessidade de compatibilidade de serviço, banda e demanda que proveja a funcionalidade do sistema, mesmo que todos os usuários utilizem-no simultaneamente;
- poucos recursos: sistemas com recursos escassos estão mais propensos à interrupção do respectivo serviço VoIP, em razão da pouca memória e baixa capacidade de processamento;
- falhas de manipulação de arquivos: necessidade de proteção dos arquivos, a fim de evitar erros ocasionados pela manipulação dos mesmos;
- qualidade da conexão física: quando há perda de pacotes durante a chamada, sua qualidade fica comprometida;
- gerenciamento de senhas: para a utilização da VoIP, são imprescindíveis identificação e senha próprias, que, se não forem bem protegidas, podem ser alvo de intrusos e utilizadas indevidamente;
- permissões e privilégios: decorrem da precaução necessária à proteção dos recursos em geral para evitar a acessibilidade e oferta dos mesmos indevidamente;
- redes homogêneas: em que uma falha consiste vulnerabilidade passível de ser estendida a todos os seus componentes pelo que redes heterogêneas asseguram maior eficácia na proteção a ataques, devido à diversidade de sua composição;

- falta de sistema de restabelecimento: ocasiona o comprometimento do sistema VoIP, ou seja, a sua suspensão.

Em virtude de alguns *gateways* VoIP, ou seja, *Media Gateways* poderem possuir interfaces com a PSTN, a exemplo do Asterisk PBX, ameaças à VoIP também podem incidir na PSTN, pela interconexão entre ambas. Assim, um atacante, ao comprometer um *Media Gateway*, pode causar interferências ao serviço de telefonia convencional, ou seja, a PSTN [76].

Deve-se considerar a segurança em relação ao serviço de DNS, pois, caso um intruso obtenha seu controle, poderá alterar entradas e, assim, promover o redirecionamento de chamadas remetendo-as para destinos indevidos.

Da mesma forma que a VoIP possibilita novos serviços e aplicações, com funcionalidades proporcionadoras de produtividade e eficiência, a sua má aplicabilidade implica ameaças e vulnerabilidades comprometedoras à segurança e impactantes na confidencialidade, integridade, disponibilidade e autenticidade das informações do sistema [19] [76].

2.2.2 Ataques Decorrentes das Vulnerabilidades

Uma vulnerabilidade em VoIP, ao ser explorada, possibilita ataques que ocasionam desde o controle de telefones IP e *Media Gateways*, à culminância em golpes como a falsificação da identificação do usuário (*CallerID Spoofing*), grampos das chamadas (*Eavesdropping*), sequestro de sessão (*Media Session Hijacking*), fraudes na tarifação (*Toll Fraud*), *spam* na forma de mensagem de voz (*Spam over Internet Telephony* – SPIT), negação de serviço (*Denial of Service* – DoS), entre outros, a exemplo dos descritos a seguir [19] [63] [67] [72] [75] [76]:

- DoS, ou *Denial of Service* (ataque por negação de serviço), constitui-se no ataque à rede de dados, ocasionando a indisponibilidade ou degradação de seus serviços para os seus usuários, pela alteração nas configurações, interrupção do funcionamento de componentes da rede, bem como consumo dos recursos como memória, processamento e banda de rede, que pode culminar no esgotamento do

sistema. Em razão de a VoIP ser sensível ao excesso de tráfego, sua vulnerabilidade a este risco acentua-se e pode ser gerado, por exemplo, por um simples vírus. Sua decorrência é especialmente notória pelos prejuízos econômicos;

- DDoS, ou *Distributed Denial of Service* (ataque distribuído por negação de serviço), consiste em um ataque, basicamente, DoS simultâneo, procedente de várias fontes, para uma cuja potencialização basta ao intruso comprometer maior número de máquinas. É utilizado para tirar de operação um ou mais serviços ou computadores conectados à *Internet*. Normalmente esses ataques procuram ocupar toda a banda disponível para o acesso a um computador ou rede, causando grande lentidão ou até mesmo indisponibilizando-lhes qualquer comunicação;
- *Eavesdropping* (escuta): consiste na interceptação de mensagens e chamadas por intrusos, o que permite quebra da confidencialidade. Isso é possível pela captura de pacotes RTP (*Real-time Transport Protocol*), posteriormente reagrupados para a obtenção do áudio. Ressalta-se que redes que se valem de *switch* (comutadores de dados) em vez de *hub*, possuem mais segurança, pois, os pacotes são encaminhados diretamente entre as partes envolvidas na comunicação, com o que há maior dificuldade na sua captura por intrusos. No entanto, essa segurança pode ser burlada por um ataque do tipo *ARP Spoofing*;
- MITM (*Man in the Middle* - homem no meio): consiste no ataque em que o invasor se coloca entre o emissor e o receptor sem ser notado, o que lhe possibilita interferir nas mensagens de ambos, sem que os mesmos não percebam;
- *Call Hijack* (sequestro de chamada): consiste em um tipo de ataque em que o atacante consegue fazer-se passar por um dos *user agents* (UA) da chamada, o qual, em geral, evolui para o ataque do tipo MITM. Atua sobre o protocolo de sessão, a exemplo do SIP, que utiliza mensagens de texto para iniciar, configurar e finalizar chamadas, que o intruso pode manipular e direcioná-las para onde queira, sem que seja percebido;

- *Spoofing* (imitação): consiste no ataque em que a pessoa ou programa consegue fazer-se passar por outra, em termos usuais de falsidade ideológica;
- *Call Fraud* (ligações fraudulentas): consiste no uso indevido das redes PSTN e ou VoIP para efetuar chamadas, pela utilização de falhas do sistema ou por configurações inadequadas. Isso pode acarretar ônus exorbitantes e surpreendentes;
- SPIT (*Spam over Internet Telephony*): consiste no envio não autorizado ou indesejado de mensagens de voz, em geral, com finalidades comerciais, analogamente ao *spam*, que consiste no envio de mensagens de texto. Ressalta-se que o SPIT consome muitos recursos do sistema, pois mensagens de voz, em geral, necessitam de maior capacidade de armazenamento e de consumo de tempo dos usuários.

Em suma, no caso de um ataque à rede de dados, a exemplo da técnica de negação de serviço “DoS”, o que ocorre, em geral, é a interrupção dos mencionados serviços, como *e-mail*, servidor *web*, ou ainda, da própria rede por um período de tempo, porquanto a VoIP é altamente sensível a perdas e atrasos. Desse modo, um ataque similar, mesmo que em menor grau, pode acarretar total perda da qualidade do serviço e, se mais acentuado culminar na sua indisponibilidade [19].

2.2.3 Medidas de segurança

É impossível garantir segurança perfeita aos sistemas, notadamente à VoIP. Contudo, existe a possibilidade de precaução especificamente no seu âmbito, pela adoção de medidas preventivas a exemplo de *firewalls*, criptografia, entre outras, descritas nas subseções a seguir.

2.2.3.1 VLAN

Em uma topologia de rede em que haja apenas *switches ethernet* ou segmentos com muitas portas, a mesma é usualmente conhecida como rede simples, na qual há

domínio de *broadcast*. Isso implica que todos os dispositivos conectados ao *switch* receberão pacotes de *broadcast*, o que em uma rede com poucos dispositivos, não é problema, ao contrário de quando se aumenta a quantidade de conexões [3].

A fim de prover uma solução a esse problema, foi desenvolvida a técnica conhecida como VLAN (*Virtual LAN*), utilizada na segmentação de redes de dados, por meio da criação de LANs (*Local Area Network*) virtuais num único equipamento de rede ou pilha de equipamentos, o que é favorável a VoIP por segmentar dados e áudio em redes distintas [9] [13]. Assim, as VLANs separam logicamente grupos de dispositivos que compartilham uma rede de nível 2 [52].

Caracteriza-se tal rede por ser logicamente independente, cujos pacotes de *broadcast* só são recebidos pelos dispositivos pertencentes respectivamente a uma determinada VLAN, que pode ser estática ou dinâmica:

- VLAN estática: é baseada em portas que representam redes de dados específicas, integradas por dispositivos a elas conectados;
- VLAN dinâmica: é baseada em endereço MAC (*Media Access Control*), composto por seis octetos que representam um identificador único da respectiva interface de rede, que é previamente cadastrado e associado a uma VLAN. Com isso, quando se conecta um dispositivo na rede, independentemente da porta, ele será direcionado para a VLAN correta, pois, ao contrário da VLAN estática, a dinâmica não se restringe a portas específicas.

As VLANs operam na camada 2 do modelo OSI (*Open Systems Interconnection*), abordado no capítulo seguinte. No entanto, uma VLAN é, em geral, configurada para mapear diretamente uma rede ou sub-rede IP, o que dá falsa impressão de que também envolva a camada 3 do referido modelo [3].

2.2.3.2 Firewall

Segundo [107], um *firewall* é definido como um sistema designado para prevenir acessos não autorizados a redes de computadores e, podem ser implementados tanto *hardware*, *software*, ou em combinação de ambos. Esse sistema é utilizado

frequentemente em redes privadas conectadas com a *Internet*, especialmente nas *Intranets*, para evitar que usuários não-autorizados tenham acesso a elas.

Ainda, segundo [107], o controle é feito por meio da checagem das mensagens que entram e saem da *Intranet*, passando pelo *firewall*, que as examina, uma a uma, e bloqueia aquelas que não obedecem aos critérios de segurança especificados pelo administrador da rede. A Figura 2.10 ilustra uma rede sem *firewall*, na qual o computador externo à LAN pode conectar-se a qualquer dos seus dispositivos por meio do roteador. Desta forma, a princípio, o computador pode obter controle de acesso às máquinas e informações armazenadas nessa rede. Assim, pessoas não autorizadas podem acessá-las, bem como lê-las, modificá-las ou apagá-las.

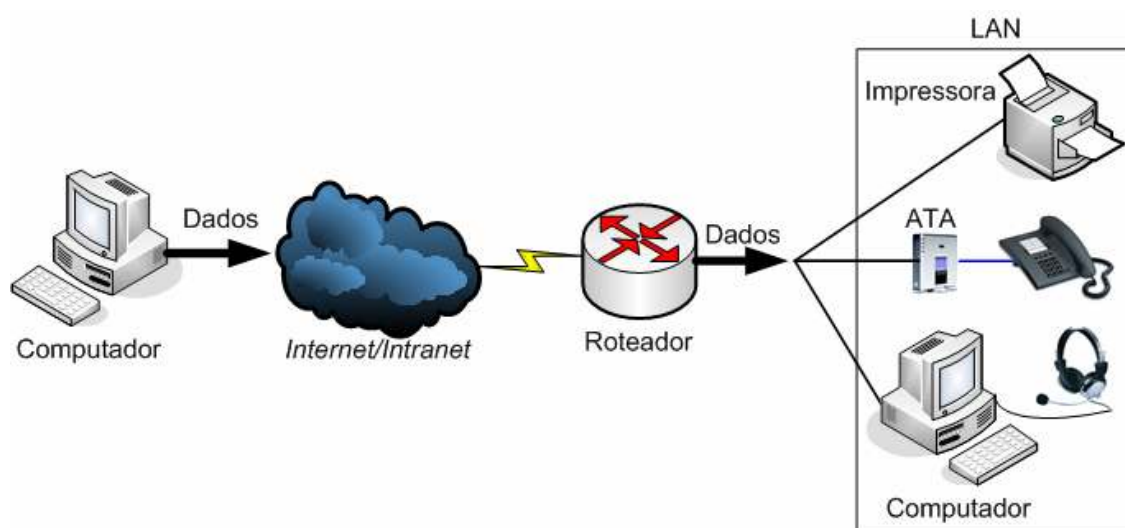


Figura 2.10 – Rede sem *Firewall*.

Conforme já mencionado, é explícito na Figura 2.10 que o roteador permite o tráfego de informações entre o computador externo à LAN e os seus dispositivos, sem qualquer tipo de segurança.

Como medida de segurança, a fim de evitar que os dados sejam acessados por intrusos, utiliza-se um *firewall*, pelo qual os dados transmitidos pelo computador externo continuam sendo trafegados pela rede, desde que sejam permitidos pelos seus filtros. A Figura 2.11 ilustra tal exemplo [107].

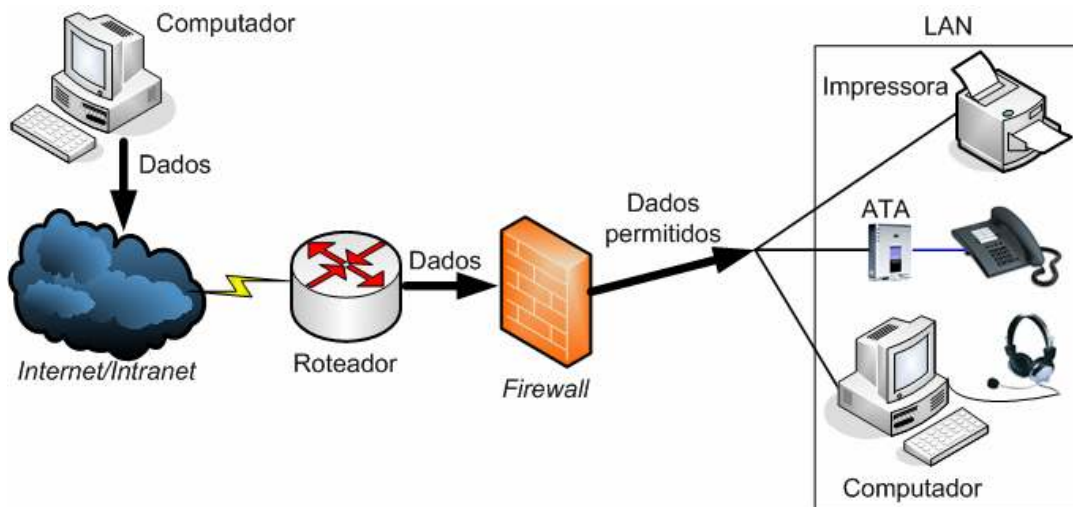


Figura 2.11 – Rede com *Firewall*, Dados Permitidos.

A partir da análise da Figura 2.11, constata-se que, com a implementação do *firewall*, os dados continuam a ser transmitidos, desde que sejam permitidos. Isso é imprescindível em uma aplicação de tempo real, como a VoIP.

Ao contrário, se os dados não forem permitidos, serão então bloqueados e uma mensagem será enviada pelo *firewall* ao computador de origem, informando-lhe que a transmissão não foi completada. Dessa forma, protege a LAN de eventuais ataques. Isso é ilustrado pela Figura 2.12 [107].

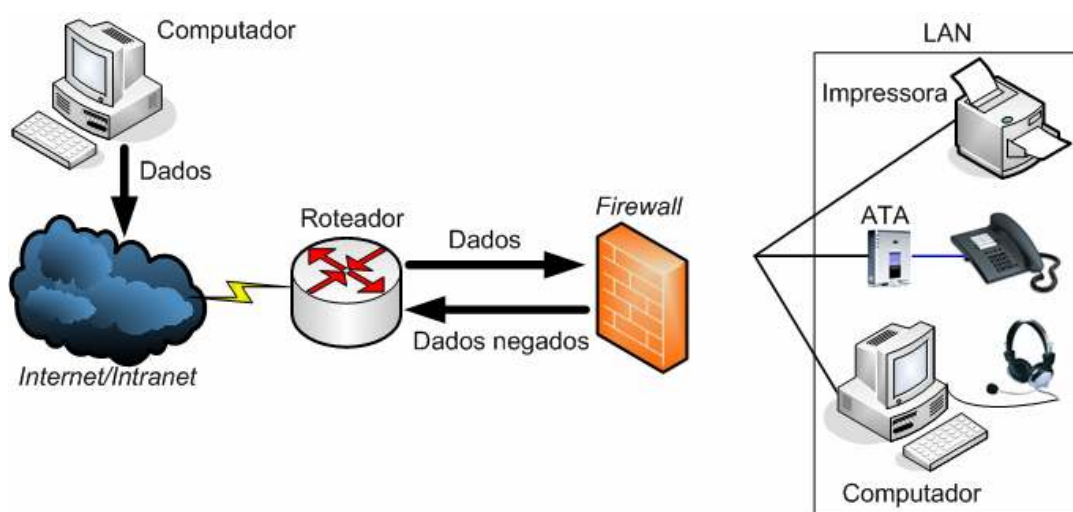


Figura 2.12 – Rede com *Firewall*, Dados Negados.

Depreende-se, pela análise da Figura 2.12, que caso a transmissão dos dados seja negada pelo *firewall*, uma mensagem é transmitida para o computador externo à LAN avisando que a transmissão não foi completada.

2.2.3.3 Criptografia em VoIP

Segundo [67], criptografia é a ciência e arte de escrever mensagens em forma cifrada ou em código. É parte de um campo de estudos que trata das comunicações secretas, usadas, entre outras finalidades, para autenticar a identidade de usuários, autenticar e proteger o sigilo de comunicações pessoais e de transações comerciais e bancárias e proteger a integridade de transferências eletrônicas de fundos. A Figura 2.13 é ilustrativa do processo de criptografia.



Figura 2.13 – Processo de Criptografia.

Ainda, segundo [67], uma mensagem codificada por um método de criptografia deve ser privada, ou seja, somente aquele que a enviou e aquele que a recebeu devem ter acesso ao conteúdo da mensagem, que deve poder ser assinada. Ou seja, de forma que a pessoa que a recebeu possa verificar se o remetente é mesmo a pessoa que diz ser e também ser capaz de identificar possíveis alterações na mensagem, o que se almeja por segurança em VoIP [30].

Atualmente os métodos de criptografia são seguros e eficientes e baseiam-se no uso de uma ou mais chaves, que são constituídas por uma respectiva sequência de caracteres, quer letras, dígitos ou símbolos, como por exemplo uma senha, que é convertida em um número, os quais são criptografados para codificar mensagens, ou, ao contrário, descriptografados para decodificá-las. Contemporaneamente, a criptografia

pode ser feita por chave única ou pela modalidade de chaves pública e privada [30] [67].

A criptografia que faz uso de chave única, utiliza a mesma chave tanto no processo de codificação quanto no de decodificação das mensagens, ainda, apresenta eficácia quanto ao tempo de processamento. Contudo, afigura-se vulnerável em meio não seguro em que a chave possa ser compartilhada [67].

Já a criptografia de chaves pública e privada utiliza duas chaves distintas: a pública, que pode ser divulgada livremente, bem como a privada, de caráter restrito e sigiloso. Em tal processo, a chave pública tem por função a codificação, e a privada, a decodificação. Se, por um lado, requer mais tempo para processamento, por outro, oportuniza maior segurança, por não necessitar de meio seguro para antecipadamente combinar chaves [30] [67].

Um exemplo da utilização do método de dupla chave consiste na geração de assinatura digital, por intermédio de um código pela chave privada, em que a mensagem apenas pode ser lida de posse da chave pública; a segurança do método baseia-se no fato de a chave privada ser de conhecimento apenas do próprio dono [67].

Em suma, em VoIP, a relevância de criptografia é notória quanto à segurança por ela propiciada na transmissão da voz. Contudo, apesar dos benefícios proporcionados, pode ocasionar perdas na qualidade da comunicação, por despender de maior tempo de latência.

2.3 Utilitários da VoIP

A aplicabilidade da VoIP proporciona mobilidade, realização de conferências e integração, para o que requer diversos tipos de utilitários da VoIP, que podem ser aplicações, dispositivos ou equipamentos, os quais constituem-se imprescindíveis à comunicação. Os utilitários da VoIP podem ser genericamente classificados em ATA, Telefone IP e *Softphone*, apresentados nas seções subsequentes.

2.3.1 ATA

O ATA (*Analog Telephone Adaptor*) possibilita que um aparelho telefônico analógico convencional seja utilizado com VoIP, sem a necessidade de um computador, por realizar a interface entre o referido telefone e a rede de dados e, para seu uso; basta apenas que esteja conectado à referida rede e configurado a um provedor VoIP. A Figura 2.14 ilustra um modelo de ATA da *Linksys* [79] [86].



Figura 2.14 – ATA.

A Figura 2.14 de um ATA, exemplifica um dispositivo simples por meio do qual se estabelece uma chamada VoIP, pelo fato do mesmo dispor de uma interface FXS, que provê a sinalização ao aparelho telefônico analógico, além de uma interface de rede, a qual lhe permite conexão à rede de dados.

2.3.2 Telefone IP

O Telefone IP, ou *hardphone*, dispõe das mesmas funções de um aparelho telefônico analógico convencional e é dotado de uma interface de rede, o que faz com que possa ser diretamente ligado à rede de dados, a fim de se utilizar VoIP, o que o faz um Telefone IP. A Figura 2.15. ilustra um modelo de Telefone IP da Cisco [75] [79] [87].



Figura 2.15 – Telefone IP.

O Telefone IP ilustrado pela Figura 2.15 é um equipamento propiciador do uso da VoIP.

2.3.3 *Softphone*

O *softphone* é um *software* que possibilita a utilização da VoIP em computadores ou PDAs (*Personal Digital Assistants*), entre outros. A Figura 2.16 ilustra o *X-Lite*, um dos modelos mais utilizados, haja vista suas peculiaridades de ser simples e prático [56] [79].



Figura 2.16 – *Softphone X-Lite*.

A utilização de aplicativo *softphone*, tal como o apresentado na Figura 2.16, possibilita não apenas chamadas VoIP, bem como conferência, videochamadas, chamada em espera, entre outras funcionalidades.

2.4 Conclusão

Uma das grandes vantagens propiciadas pela VoIP consiste na liberdade de escolha da operadora VoIP que interconecte com a PSTN, quer localizada no país ou no exterior, o que proporciona tarifas competitivas e diferentes opções de níveis de qualidade de serviço, consonantes às necessidades dos usuários.

Outra grande vantagem pouco difundida consiste no método de endereçamento por identificação em que os ramais dos usuários possuem endereços semelhantes aos respectivos endereços de *e-mails*. Ou seja, quando se quer falar com outrem, no mesmo sistema ou em independentes sistemas VoIP, os endereços dos domínios são consultados e uma rota ligando-os estabelece-se via rede de dados [75].

Apesar das vantagens propiciadas pela tecnologia VoIP, ela não deixa de incorrer em algumas dificuldades, a exemplo do despreparo técnico dos profissionais ou empresas responsáveis pela instalação e ou manutenção da solução VoIP adotada, causador de restrições à aplicabilidade do potencial funcional passível de ser disponibilizado que, por tal razão, permanece atrelado a operadora de telefonia convencional e carente de pertinente mobilidade.

A segurança em redes de computadores deve ter sempre um lugar de destaque, não apenas para pessoas que trabalham diretamente na área de informática, tais como técnicos, analistas, pesquisadores, bem como usuários que, de alguma forma, utilizam a rede como ferramenta para comunicação, estudo, entretenimento ou trabalho.

É prudente que, ao se implementar VoIP, previamente, atente-se para questões relativas à segurança, de modo preventivo que possibilite reduzir significativamente a efetividade de acometimentos prejudiciais ao serviço. Dessa forma conclui-se que a segurança em VoIP será cada vez mais requisitada conforme o seu crescimento e expansão.

Em suma, é interessante ressaltar que apesar da VoIP utilizar *software* e *hardware* peculiares, a eles não se atrela, a exemplo do sistema operacional e do *softphone*, e mesmo do protocolo de sessão utilizado, que são abordados no próximo capítulo.

E ainda, no tocante à VoIP, a implementação de balanceamento de chamadas, em função da característica de equidade proporcionada, por si só pode afigurar-se como eficaz na redução dos efeitos dos ataques; uma vez que caso ocorram, não, necessariamente, acarretarão o comprometimento total do sistema.

Capítulo 3

Protocolos Pertinentes a Tecnologia VoIP

A utilização de protocolos, que definem o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, é necessária para o controle e ações decorrentes do envio e recebimento de informações em redes de dados, a exemplo da *Internet* e das *Intranets* ou LANs, bem como ações realizadas na transmissão e no recebimento de uma mensagem. Ou seja, constitui-se na definição das regras para a troca de mensagens [30].

As entidades utilitárias dos protocolos variam conforme sua empregabilidade. Por exemplo, humanos utilizam protocolos a todo instante, desde um simples pedido de informação em que um *oi* enviado é respondido por outro *oi*, permitindo assim o estabelecimento de uma comunicação, até mesmo em cerimoniais, em que protocolos devem ser seguidos.

Com os computadores não é diferente, estes se comunicam mediante protocolos, que são organizados em camadas, cada qual responsável por dar suporte à camada adjacente, tanto superior quanto inferior. A estruturação dos protocolos em camadas visa a prover funcionalidades específicas a esses, por exemplo, um protocolo responsável pelo transporte de informação não seria incumbido da apresentação da mesma.

Uma normatização quanto à organização dos protocolos em camadas, para a troca de informações em redes de computadores, é apresentada pela ISO (*International Organization for Standardization*) por meio do modelo de referência OSI (*Open*

Systems Interconnection), desenvolvido no início da década de 1980. O modelo OSI é, hoje, o padrão para o desenvolvimento de protocolos que permitam a comunicação entre computadores. Embora nem todo protocolo utilize esse modelo, a maior parte dos novos protocolos segue o modelo de camadas proposto pelo mesmo [52].

O modelo OSI é dividido em sete camadas distintas, para tratar da comunicação intermáquinas, em que cada camada lida especificamente com a camada correspondente em outro dispositivo de rede. A Figura 3.1 ilustra o modelo de referência OSI.



Figura 3.1 – Representação das Camadas do Modelo OSI.

A Figura 3.1 apresenta as sete camadas componentes do modelo de referência OSI, que são segundo [108] e [3] sucintamente descritas a seguir:

- Camada 7, de Aplicação: dentro do processo de comunicação é representada pelo usuário final para o modelo OSI. Ou seja, com base em pedidos de usuários da rede, essa camada seleciona serviços que podem ser fornecidos por funções das camadas mais baixas, além de prover serviços como transferência de informações, emulação de terminais entre outros;
- Camada 6, de Apresentação: tem por função a interpretação, ou seja, transformação e formatação dos dados, bem como a manutenção da sintaxe e semântica quando da execução de aplicações remotas e, estabelece um formato de dados comum entre os comunicantes;

- Camada 5, de Sessão: tem por função estabelecer, gerenciar e terminar sessões de todas as atividades das camadas inferiores, o que é feito por meio de conexões virtuais, que são estabelecidas quando há troca de mensagens entre uma estação transmissora e uma receptora. Tal processo é análogo ao que acontece quando alguém se conecta a uma rede, em que, uma vez estabelecido o *login*, a conexão é mantida até o *logout*, mesmo sem acesso contínuo à rede;
- Camada 4, de Transporte: é a responsável pela transferência eficiente, confiável e econômica dos dados entre as máquinas de origem e de destino, independente do tipo, da topologia ou da configuração das redes físicas existentes entre elas, assegurando que os dados cheguem sem erros e na sequência correta. Ou seja, trata das questões de transporte entre *hosts*, propiciando confiabilidade no transporte dos dados, além de controlar o fluxo dos mesmos, bem como a recuperação e detecção de falhas;
- Camada 3, de Rede: estabelece uma conexão lógica entre dois pontos, cuidando do tráfego e roteamento dos dados da rede, ou seja, a escolha do caminho pelo qual datagramas serão enviados. A camada de Rede também controla o congestionamento das subredes, ou seja, evita os gargalos quando muitos pacotes estão presentes na subrede ao mesmo tempo. Outros problemas são inerentes a essa camada tais como: pacotes que não chegam ao destino devido ao tamanho, incompatibilidade de endereços em redes distintas, protocolos diferentes etc;
- Camada 2, de Enlace: cuida do acesso ao meio, pois providencia maneiras funcionais e procedimentos para estabelecimento, manutenção e liberação de enlace de dados entre as entidades da rede. Seus objetivos são: providenciar a transmissão de dados para a camada de rede, além de detectar e, possivelmente, corrigir erros que possam ocorrer no meio físico. São peculiaridades dessa camada: topologia de rede, endereçamento físico, notificação de erros e controle de fluxo;
- Camada 1, Física: provê características físicas, elétricas, funcionais e mecanismos para ativar, manter e desativar conexões entre duas partes, pois é a única que possui acesso físico ao meio de transmissão da rede. Ela está ligada

diretamente à transmissão de *bits* informação por um canal de comunicação. As tarefas de planejamento dessa camada devem garantir que, quando um lado envia um *bit*, este seja recebido do outro lado sem modificação. Ou seja, a camada física tem como função básica a adaptação da informação ao meio de transmissão.

A *Internet* vale-se de um conjunto de protocolos para a transmissão de dados. Esses protocolos formatam os dados a partir de definições recomendadas pela IETF (*Internet Engineering Task Force*), que é formada por diversos grupos de trabalho que divulgam documentos denominados RFCs (*Request for Comments*). Assim, as RFCs definem de fato os padrões utilizados pelas aplicações criadas para a rede mundial de computadores.

Ainda, a *Internet* possibilita o desenvolvimento de novas tecnologias num amplo leque de opções e abrangências nos âmbitos de televisão, rádio e telefonia, tais como IPTV (*Internet Protocol Television*), VoIP e rádio *online*. Ao contrário das tecnologias convencionais citadas, ela, por meio de um tráfego único “comutação de pacotes”, possibilita integrar e convergir as referidas funcionalidades, que, para funcionamento convencional, requerem soluções específicas.

O que capacita a *Internet* a tal tráfego, é o fato de ser baseada na comutação de pacotes. Isso permite que um caminho ou parte dele seja compartilhado ao mesmo tempo por vários sistemas comunicantes [30].

Por sua vez, a VoIP como utilitária da *Internet*, requer a utilização de várias modalidades de protocolos, com funções distintas, que, em conjunto, proporcionam-lhe especificidade na transmissão de voz, através das redes de dados. Entre esses protocolos, os principais que proporcionam a utilização no âmbito da tecnologia VoIP, seguindo o modelo de referência OSI são:

- Camada de rede: IP;
- Camada de transporte: TCP, UDP, RTP e RTCP;
- Camada de sessão: H.323, SIP e SDP.

A Figura 3.2 ilustra esquematicamente os protocolos enumerados, associados a camadas do modelo de referência OSI no âmbito da VoIP, bem como aplicativos e *codecs* [81].

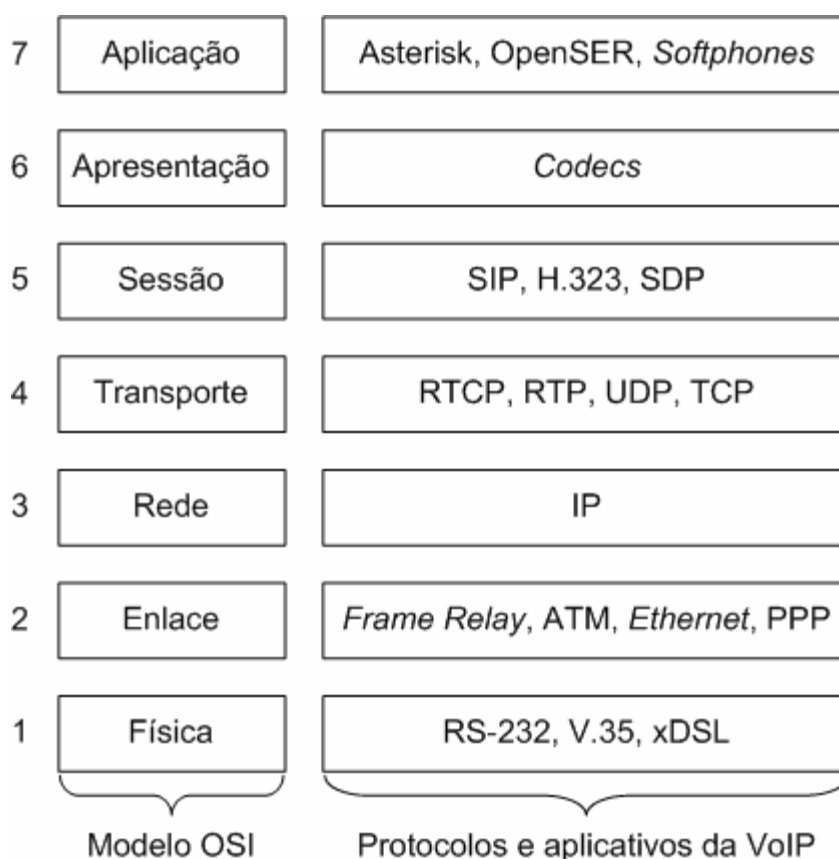


Figura 3.2 – Protocolos e Aplicativos Pertinentes a VoIP em Associação ao Modelo OSI.

Na Figura 3.2, pode-se observar a que camada corresponde determinado protocolo, bem como os *codecs* e as aplicações que são pertinentes à tecnologia VoIP. As seções seguintes contemplam os principais protocolos pertinentes a VoIP.

3.1 Protocolo IP

O protocolo IP baseia-se na comutação de pacotes, utilizando datagramas e consiste na base da estrutura de comunicação da *Internet*. Desde sua concepção, foi

implementado como um protocolo de comunicação sem conexão para controle de tráfego, baseado na regra do melhor esforço (*best-effort service*), contudo, sem nenhum mecanismo de qualidade de serviço [48].

O protocolo IP foi projetado para possibilitar a interconexão de redes de computadores que fazem uso da tecnologia de comutação de pacotes. Realiza a transmissão dos dados por intermédio de blocos de informação denominados datagramas, nos quais a origem e o destino são identificados por endereços de tamanho fixo [11].

O protocolo IP trabalha em conjunto com protocolos de camadas superiores, a fim de tornar possível a operação de aplicações ou serviços e, no modelo OSI, encontra-se na camada 3, denominada de camada de rede. Atualmente, o IP conta com duas versões, a IPv4, que é a mais usada e, a IPv6, que ainda não é amplamente utilizada.

Cada lado participante de uma comunicação envolve certo número de camadas, entre as quais a superior representa a informação a ser passada ao destino final. Entretanto, para que isso ocorra, ou seja, a informação seja transferida, ela necessita ser empacotada apropriadamente, e os respectivos pacotes serem roteados por intermédio de um meio físico [11].

Os pacotes IP são definidos pela RFC 791, como mostrado na Figura 3.3 [30].

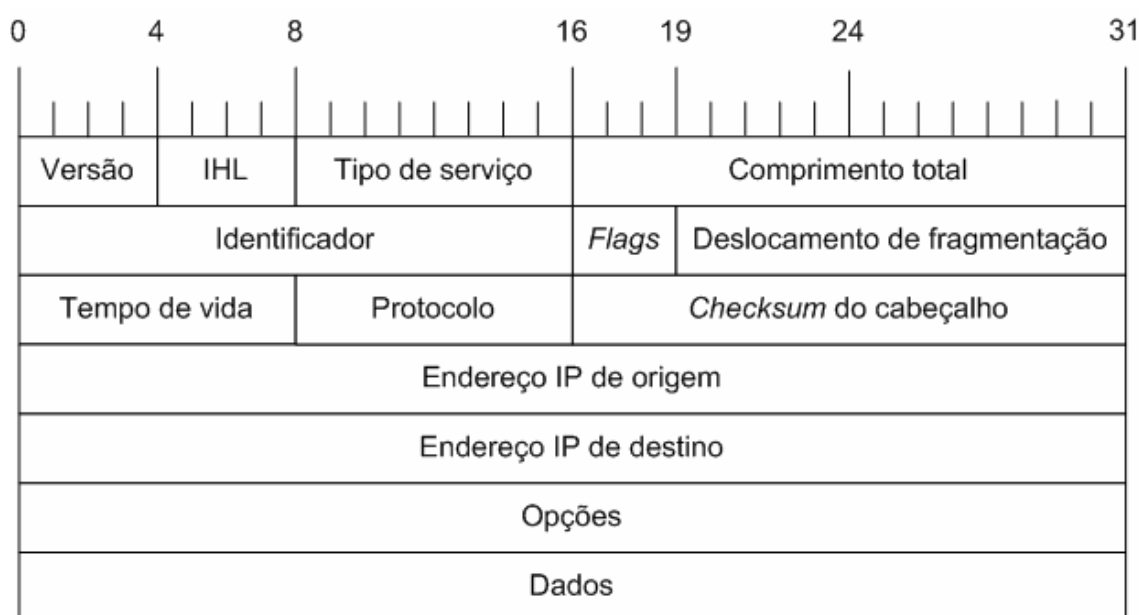


Figura 3.3 – Campos do Pacote IPv4.

A Figura 3.3 apresenta os campos de um pacote IP, os quais segundo [52] são descritos a seguir:

- Versão: indica se a versão IPv4 ou IPv6 está sendo utilizada;
- *IHL* (*IP Header Length* – comprimento do cabeçalho IP): indica o comprimento do cabeçalho do datagrama em múltiplos de palavras de 32 *bits*;
- Tipo de serviço: especifica como um protocolo em particular da camada superior deseja que o pacote em questão seja tratado. Pode-se atribuir aos pacotes diversos níveis de qualidade de serviço (QoS) baseados neste campo;
- Comprimento total: especifica o comprimento total do pacote IP inteiro, incluindo dados e cabeçalho, em *bytes*;
- Identificador: contém um número inteiro que identifica o datagrama atual. Este campo é utilizado para auxiliar a união de datagramas fragmentados;
- *Flags*: um campo de três *bits* em que os dois *bits* menos significativos controlam a fragmentação. O *bit* mais significativo neste campo não é utilizado. Um dos *bits* especifica se é ou não possível fragmentar o pacote; o segundo *bit* especifica quando o pacote é o último fragmento de uma série de pacotes fragmentados;
- Deslocamento de fragmentação: Informa o posicionamento do fragmento em relação ao datagrama do qual faz parte, para o reagrupamento no destino, na ordem correta. Quando o datagrama não é fragmentado o conteúdo deste campo é zero;
- Tempo de vida: mantém o contador que é gradualmente decrementado até zero, ponto no qual o datagrama é então descartado. Isso evita que pacotes fiquem circulando infinitamente na rede;
- Protocolo: indica qual protocolo da camada superior deve receber os dados após o final do processamento no nível IP;
- *Checksum* do cabeçalho: usado para verificar se o cabeçalho não foi corrompido;
- Endereço IP de origem: contém o endereço de origem;
- Endereço IP de destino: contém o endereço de destino;
- Opções: permite que o IP suporte diversas opções, a exemplo de segurança;

- Dados: contém os dados de aplicação, assim como informações de protocolos de nível superior.

Um pacote IP contém, entre outros dados, três informações fundamentais: endereço de origem, endereço de destino e dados. Os endereços de origem e de destino são utilizados pelos roteadores que compõem a infraestrutura da *Internet* para transportar os pacotes de um lado para o outro. Os dados constituem qualquer tipo de informação inserida pelo cliente do protocolo IP, no respectivo pacote, apresentam tamanhos variáveis.

O protocolo IP não é, em geral, diretamente utilizado pelas aplicações para a transmissão de dados, em virtude de sua simplicidade, por ser a plataforma básica da *Internet*. Não possui, por exemplo:

- a propriedade de conexão: cada pacote é isolado, ou seja, independente;
- nenhuma garantia de confiabilidade: o transmissor não percebe o erro caso um pacote não seja entregue para o destinatário;
- propriedade de multiplexar conexões entre computadores;
- determinar o destinatário de um pacote, caso existam em computadores conectados, em cada qual duas aplicações, o que ocorre devido ao fato de o pacote IP apenas definir os endereços do emissor e receptor, e não a específica aplicação destinatária no receptor da mensagem.

As referidas carências do protocolo IP, são supridas pelos protocolos TCP e UDP da camada de transporte, tratados a seguir.

3.2 UDP

Uma transmissão confiável, com conexão, é análoga a uma “chamada telefônica”. Os *softwares* de rede dos sistemas operacionais de ambos os lados trocam informações para verificarem se o receptor e o transmissor aceitam a “chamada” e se ambos estão prontos para ela. Assim que todos os detalhes forem estabelecidos, a conexão é efetivada e pode-se dar início à transferência de dados, durante a qual os

protocolos de ambas as máquinas continuam a comunicar-se a fim de averiguar se os dados foram recebidos corretamente. Caso ocorra alguma falha, detectarão e informarão os aplicativos correspondentes.

O UDP (*User Datagram Protocol*), definido na RFC 768, fornece um serviço de transmissão sem conexão, não confiável. Apresenta como característica fundamental a utilização de portas para a troca de dados via *Internet*, ou seja, o UDP utiliza o IP para o transporte das mensagens entre as máquinas, acrescentando a habilidade de fazer a distinção entre múltiplos destinos em um determinado *host*, ou seja, hospedeiro, por meio das portas [12] [35].

Os pacotes IP possuem apenas os endereços de origem e destino e não fazem distinção entre as aplicações contidas em um mesmo computador, ou seja, um *host* ao qual a mensagem se dirige, problema que o UDP resolve com a utilização de portas, por ser um protocolo da camada de transporte.

Uma aplicação que utiliza o UDP se sujeita à perda de mensagens, ordenação, duplicação e atraso, pois o mesmo não trata desses aspectos. Para receber dados via UDP, é necessário especificar um porto para a qual estes dados devam ser enviados. Ressalta-se que um porto, ou seja, uma porta, constitui-se em uma sequência binária de 16 *bits*, com variação entre 1 e 65535 [30].

3.2.1 Formato do Datagrama UDP

Denomina-se datagrama de usuário a mensagem UDP, que é encapsulada em um único datagrama IP, conforme esquematizado na Figura 3.4.

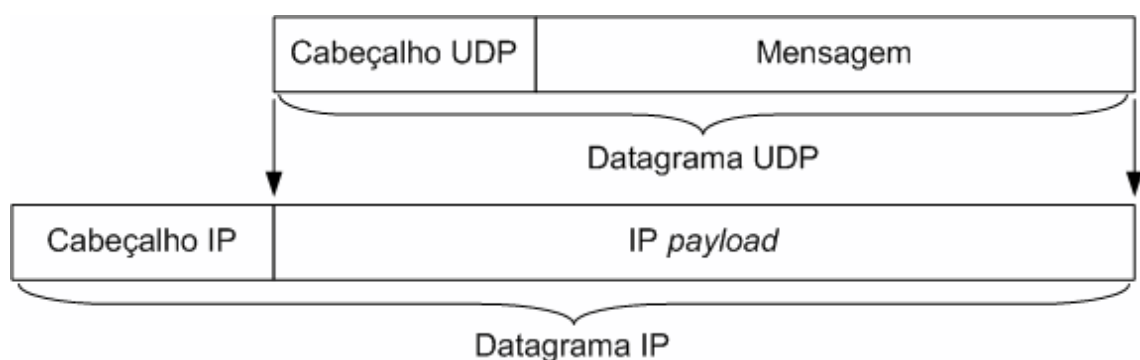


Figura 3.4 – Encapsulamento UDP.

A Figura 3.4 ilustra o encapsulamento de um datagrama UDP em um pacote IP. É importante salientar que o cabeçalho do UDP é inserido no início do campo de dados do pacote IP. O formato do datagrama UDP é apresentado na Figura 3.5.

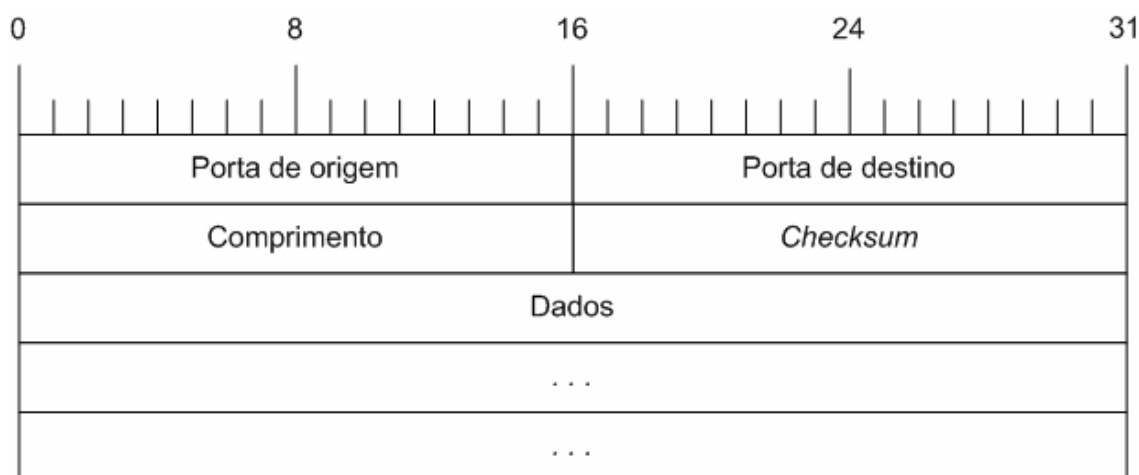


Figura 3.5 – Datagrama UDP.

São aspectos relevantes da Figura 3.5 os campos descritos a seguir:

- Porta de origem: campo responsável pela especificação da porta no *host* de origem para qual as respostas devam ser remetidas, em que o uso é opcional;
- Porta de destino): campo responsável pela especificação da porta no *host* de destino para a qual a mensagem é enviada;
- Comprimento: indica a contagem em *bytes*, incluindo o cabeçalho e os dados do usuário;
- Checksum (soma de verificação): Consiste na soma de verificação do UDP, a qual abrange mais informações do que as contidas no seu datagrama. Para seu cálculo, é adicionado um pseudocabeçalho ao datagrama UDP, que é acrescido por um octeto de zeros a fim de torná-lo um múltiplo exato de dezesseis *bits*, utilizado com função de preenchimento. No entanto, tanto octeto quanto o pseudocabeçalho não são enviados e nem mesmo utilizados no cálculo do Comprimento. No destino, para o cálculo do *checksum*, extrai-se do cabeçalho IP os endereços IP de origem e destino nele contidos, remonta-os no formato de

pseudocabeçalho e recalcula o *checksum*. Ou seja, os dezesseis *bits* de *checksum* são preenchidos com o resultado de uma função aplicada aos dados transportados, de forma que o receptor possa aplicar a mesma função ao receber a mensagem e, assim, confirmar a integridade dos dados transmitidos, pois caso algum *bit* tenha sido corrompido, o resultado da função aplicada pelo receptor diferirá do *checksum*, o que implicará na percepção do erro e consequente descarte do pacote. Ressalta-se que o uso de *checksum* é opcional [35].

3.2.2 Pseudocabeçalho UDP

O cálculo do *checksum* do datagrama UDP não utiliza apenas os dados do cabeçalho, mas requer que seja adicionado um pseudocabeçalho, que não é transmitido no datagrama UDP. O pseudocabeçalho é ilustrado na Figura 3.6.

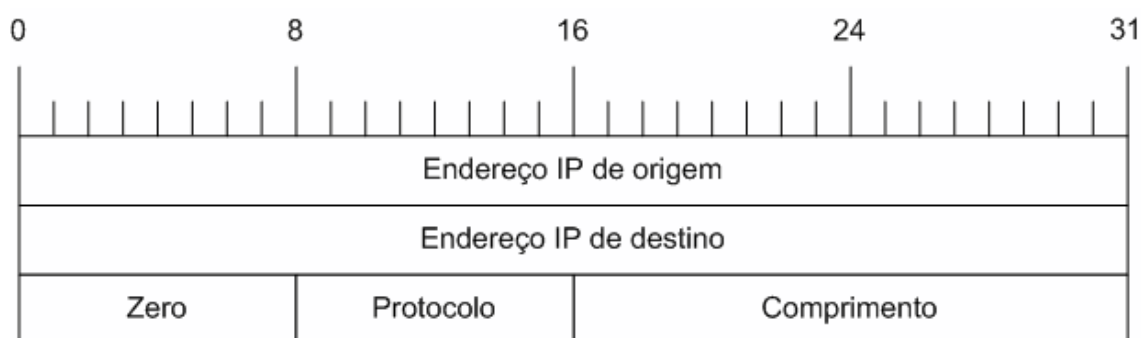


Figura 3.6 – Pseudocabeçalho UDP.

Em conformidade a Figura 3.6, observa-se que o pseudocabeçalho é formado pelos campos descritos a seguir:

- **Endereço IP de origem:** contém o endereço IP do *host* que gerou a mensagem;
- **Endereço IP de destino:** contém o endereço IP do *host* de destino para o qual a mensagem foi enviada;
- **Zero:** contém oito *bits* zero;
- **Protocolo:** contém o código do protocolo utilizado, no caso do UDP o 17;
- **Comprimento:** contém o tamanho do datagrama UDP, sem incluir o pseudocabeçalho.

O uso do pseudocabeçalho tem como objetivo verificar se o datagrama UDP atingiu seu destino correto, formado pelo endereço IP acrescido da porta UDP. Ressalta-se que essa opção deve ser utilizada somente em aplicações que requeiram um pouco mais de confiabilidade e não velocidade, pois isso deixará o processo mais lento [12].

3.2.3 Multiplexação e Demultiplexação do UDP

Cada aplicativo obtém do sistema operacional uma porta para envio de pacotes UDP, que portarão os números das respectivas portas, dentro do campo *source port*. Ao transmitir mensagens, o UDP aceita datagramas de diversos aplicativos e transmite-os à camada IP, propriedade denominada multiplexação, ilustrada na Figura 3.7.

Já ao receber mensagens, o UDP aceita datagramas oriundos da camada IP e os repassa aos aplicativos de destino por meio da *destination port*, endereçadas em cada pacote, o que consiste na propriedade de demultiplexação, também ilustrada na Figura 3.7 [12].

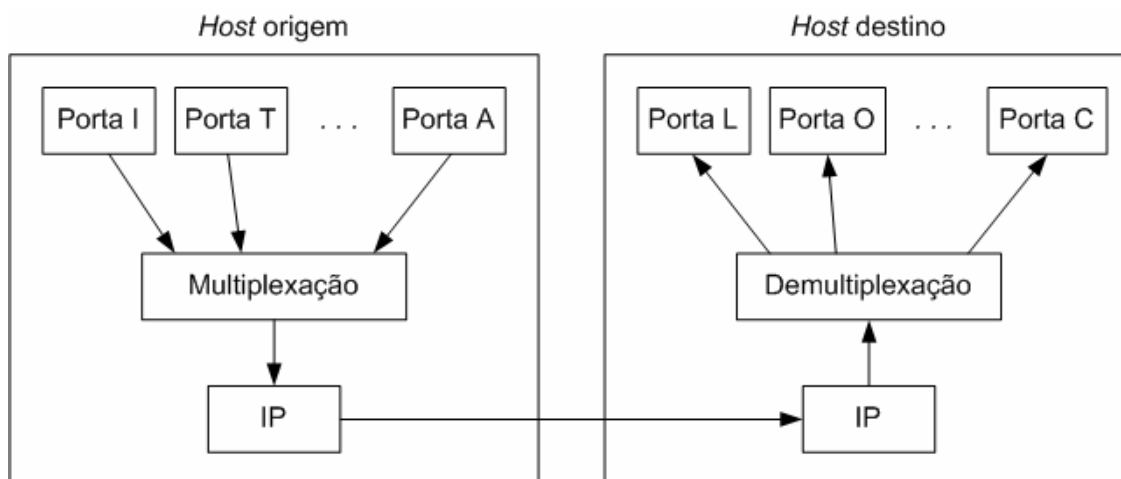


Figura 3.7 – Multiplexação e Demultiplexação UDP.

Compreende-se pela análise da Figura 3.7, que aplicações *multicast*, a exemplo de conferências multimídia, somente são possíveis em decorrência da multiplexação, pelo fato de basear-se em porta e não em conexões.

Embora dotado de avanços, o protocolo UDP ainda persiste com problemas, pois como o pacote IP, não é orientado à conexão, bem como não possui mecanismo de confirmação de recebimento, pelo que o transmissor não toma conhecimento quanto à entrega de um pacote enviado. A fim de suprir tais dificuldades, foi desenvolvido o TCP, tratado a seguir.

3.3 TCP

O protocolo TCP (*Transmission Control Protocol*), da camada de transporte, é definido pela RFC 793 e seus pacotes também devem ser encapsulados dentro de um pacote IP. Possui a peculiaridade de ser orientado a conexão, o que significa que os pacotes não são interpretados isoladamente e que fornece um serviço confiável de transferência de dados fim-a-fim [50].

Para troca de dados entre dois computadores via protocolo TCP, de início realiza-se o procedimento denominado *Three-Way Handshake* (cumprimento de três vias), que sincroniza ambos os lados. Isso ocasiona a manutenção dos pacotes, ainda que cheguem diversamente ao envio.

Ainda envia um pacote de confirmação ACK (*Acknowledge* - reconhecimento) para cada pacote de dados recebido, bem como outro de *keep-alive* “mantenha vivo” periódico, para informá-lo da manutenção da conexão entre ambos. Para o término da comunicação, é enviado por um dos lados e confirmado pelo outro um pacote de finalização e, com isso, fecha-se o canal.

O TCP está localizado na camada de transporte, acima do IP, ou seja, na quarta camada do modelo OSI e tem como função básica assegurar que todos os pacotes sejam entregues às aplicações de destino em sequência, sem perdas, erros ou duplicações. Como o IP é um protocolo do tipo *best-effort*, ou seja, não garante que todos os pacotes serão entregues ao destino, o TCP tem a função de compensar esta falta de confiabilidade do IP por meio de uma confirmação fim-a-fim do recebimento de cada pacote.

No caso em que pacotes não sejam recebidos no destino, o TCP garante uma retransmissão, pois também inclui um controle de fluxo, de forma que uma aplicação na

origem, não sobrecarregue uma aplicação mais lenta no destino, com o envio de uma quantidade maior de pacotes do que ela pode tratar, além de um controle de congestionamento [11].

O TCP cria um canal bi-direcional, ou seja, *full-duplex*, e o seu cabeçalho contém as adições necessárias à implementação das peculiares funcionalidades, como pode ser observado na Figura 3.8 [35].

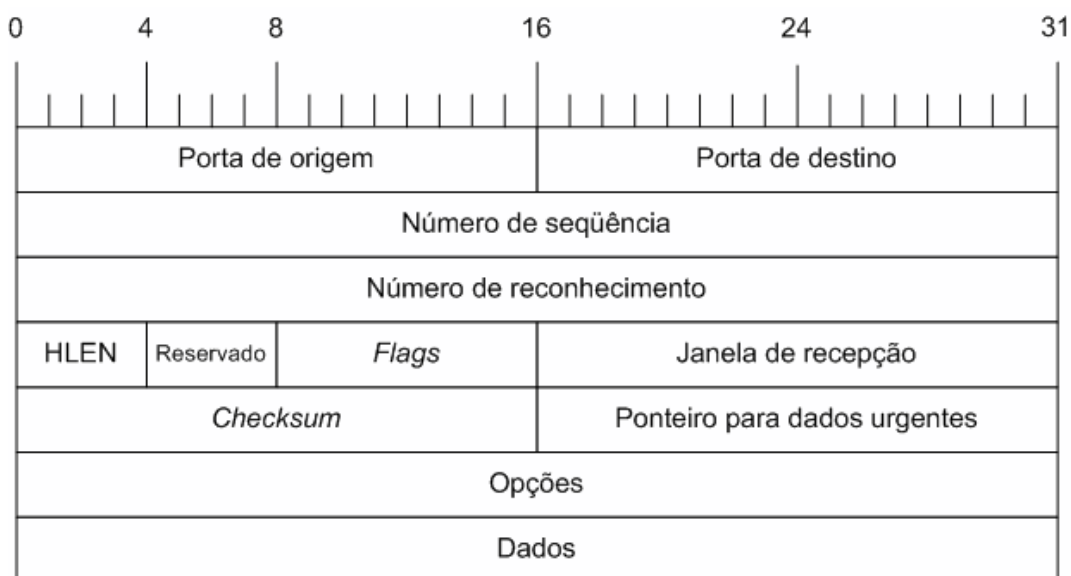


Figura 3.8 – Formato de um Pacote TCP.

Como comprovado pela Figura 3.8, integram o pacote TCP os campos abaixo descritos segundo [52]:

- Portas de origem e destino: identificam os pontos nos quais os processos de origem e de destino da camada superior recebem serviços TCP;
- número de sequência: geralmente especifica o número associado ao primeiro *byte* de dados da mensagem atual. Dadas certas circunstâncias, também pode ser usado para identificar um número de sequência inicial a ser usado na transmissão que está iniciando;
- número de reconhecimento: contém o número de sequência do próximo *byte* de dados que o emissor do pacote espera receber;
- HLEN (*Header Length* - comprimento do cabeçalho): indica o número de palavras de 32 *bits* que constituem o cabeçalho do segmento;

- reservado: deve conter zeros e poderá ser usado em implementações futuras;
- *flags*: transportam uma série de informações de controle;
- janela de recepção: especifica o tamanho da janela de recepção do emissor, isto é, o espaço disponível para dados ingressando;
- *checksum*: indica quando o cabeçalho e dados foram danificados em trânsito;
- ponteiro para dados urgentes: aponta para o primeiro *byte* de dados urgentes no pacote;
- opções: especificam várias opções TCP;
- dados: contém informações da camada superior.

A transmissão confiável não é apropriada para dados sensíveis a atrasos como áudio e vídeo em tempo real, pois, o TCP sempre tentará reenviar pacotes perdidos, causando atrasos e esperas por parte do usuário. Além de não ser *multicast*, ou seja, não viabilizar conferência e seu controle de congestionamento diminuir a janela de congestionamento assim que detectar perda de pacotes, diminuindo o fluxo de pacotes recebidos. Com isso, pacotes perdidos causariam graves problemas com áudio e vídeo, pois necessitam de fluxo contínuo de dados.

Quando se deseja transportar voz sobre IP, utiliza-se o UDP como protocolo de transporte, apesar de ele não ser confiável. Em uma conversação, a perda ocasional de até 5% dos pacotes de voz, embora não seja desejável, não prejudica de forma grave a transmissão [12].

Por outro lado, o tráfego de voz é altamente sensível ao atraso, e a rotina de estabelecimento e confirmação de mensagens do TCP introduz atrasos, bem como no caso de perda de pacotes, haveria ainda mais retardo na transmissão, devido à retransmissão. Assim, a tolerância pela perda de alguns pacotes é mais uma estratégia adequada do que introdução de atrasos na transmissão de voz [11].

O protocolo UDP, entretanto, não foi originalmente criado para o transporte de voz, apesar de afigurar-se melhor alternativa ao TCP no referido âmbito. Todavia para a utilização do UDP no tráfego de voz, necessita recorrer a outros protocolos, a exemplo do RTP, objeto da próxima seção.

3.4 Protocolo RTP

O protocolo RTP (*Real-time Transport Protocol*), definido pela RFC 1889 do IETF, consiste em um protocolo que provê o transporte fim-a-fim das informações multimídia (áudio e vídeo) em tempo real, podendo ser visto como uma subcamada da camada de transporte. Apesar de poder operar sobre qualquer protocolo de transporte, é usualmente utilizado sobre o UDP, com o qual, se combinado, é capaz de multiplexar os diversos fluxos de informações multimídia sobre um único fluxo de seus pacotes [12] [75].

O protocolo RTP provê a especificação de requisitos referentes a tempo, tanto na transmissão quanto na recepção dos pacotes, por possibilitar aplicações de tempo real. Na transmissão dos pacotes de dados, podem ocorrer perdas, atrasos ou entregas fora de ordem, problemas que o RTP permite ao receptor detectar, bem como ainda informações sobre o tipo de dado transportado, ou seja, a identificação do conteúdo, *timestamps* ou reconstrução temporal dos pacotes recebidos e respectivos números de sequência [16] [48].

Portanto, o RTP foi concebido para compensar aos receptores o *jitter*, variação do tempo de atraso definido no *timestamp* e a perda de sequência dos pacotes introduzidos pelas redes de dados. Um pacote RTP é bastante simples de ser entendido. Ele consiste de um cabeçalho de 12 *bytes*. A Figura 3.9 ilustra o seu cabeçalho.

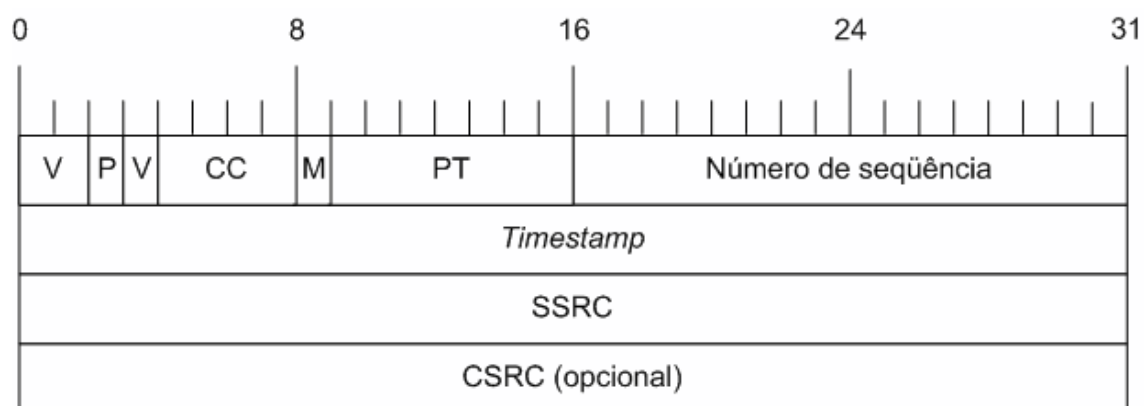


Figura 3.9 – Cabeçalho RTP.

A Figura 3.9, ilustrativa do cabeçalho do protocolo RTP apresenta-se composta por campos, com funções respectivas, descritas a seguir segundo [12] e [48]:

- V (*version*): descreve a versão do RTP, na atualidade, a 2;
- P (*padding*): informa à aplicação se o pacote sofreu algum tipo de completamento para fazer com que o mesmo possuísse um tamanho múltiplo de 32. O *padding* pode ser necessário para assegurar o alinhamento correto para criptografia ou para carregar muitos pacotes RTP em um único pacote;
- X (*extension*): sinaliza a presença de um cabeçalho entendido entre o cabeçalho fixo e os dados;
- CC (*CSRC Count*): descreve a quantidade de identificadores CSRC (*Contributing Source*) incluídos no cabeçalho, variando de 0 a 15;
- M (*Marker*): indica a presença da supressão de silêncio nos pacotes para codificações de áudio com essa capacidade;
- PT (*Payload Type*): informa o tipo dos dados, ou seja, o tipo de conteúdo que o campo dados contém;
- Número de sequência: O número de sequência é iniciado randomicamente no início da sessão e incrementado para cada pacote RTP que é enviado. Permite ao receptor detectar perda de pacotes ou restaurar a sequência original que porventura cheguem fora de ordem;
- *Timestamp*: contém a frequência do *clock* utilizada por determinado tipo de dado. É incrementado de acordo com a quantidade de amostras de mídia contidas nos pacotes;
- SSRC (*Synchronization Source*): corresponde a entidade responsável por configurar o número de sequência e o *timestamp*. Normalmente é transmissor do pacote RTP e seu identificador é escolhido randomicamente e, não se vincula a endereços de rede, bem como deve ser único em uma sessão e preferencialmente gerado pela aplicação;
- CSRC (*Contributing Source*): utilizada quando os pacotes são oriundos de um *mixer*, referindo-se ou a ele indicando, bem como informa ainda a SSRC geradora da mídia.

Para maior eficácia do controle, todos os participantes da comunicação enviam entre si, periodicamente, pacotes RTCP (*RTP Control Protocol*), sendo que o consumo de banda dos mesmos não deve superar 5% da utilizada na sessão. A combinação RTP e RTCP não garante a entrega de pacotes e nenhum mecanismo de qualidade de serviço. No entanto, possibilita observar o controle da qualidade na rede, acompanhar o fluxo de *bits*, bem como a quebra dos seus blocos de dados em pacotes e a transmissão, e a respectiva reprodução do fluxo de *bits* no receptor. A fim de minimizar o número de pacotes perdidos, o protocolo transporta informações de temporização possibilitando que o receptor tente compensar o atraso [30].

De forma resumida, o RTP cuida do transporte da mídia com o mínimo de sobrecarga possível, *overhead*, muito semelhante ao UDP, enquanto o RTCP cria um canal de controle de tráfego RTP, gerando estatísticas de qualidade e garantia parcial de entrega, possibilitando assim algum controle sobre a mídia, tal como no TCP [7].

Em suma, pelo fato de o protocolo RTP ser destituído de um mecanismo de controle sobre a conexão, referentes à transmissão e recepção, requer para tanto utilizar-se de outro protocolo que o proveja, o RTCP, objeto de análise na próxima seção.

3.5 RTCP

Com a finalidade de suprir o mecanismo de controle sobre a conexão do qual carece o RTP, foi desenvolvido o protocolo RTCP (*RTP Control Protocol*), também definido pela RFC 1889 como um protocolo no qual uma aplicação de rede multimídia pode ser utilizada conjuntamente com o RTP [54].

O RTCP permite a troca periódica de informações de controle entre os participantes de uma sessão, com o objetivo de propiciar respectivo *feedback* das informações, que podem ser utilizadas para detectar e corrigir problemas. Deste modo, com a utilização do RTCP, um operador da rede, por exemplo, pode monitorar a qualidade da sessão e detectar problemas na existentes na rede [11].

Os protocolos RTP e RTCP permitem aos receptores compensarem a variação do atraso dos pacotes na rede de dados, que é denominada de *jitter*, por meio do controle de *buffer* e sequenciamento apropriado para que medidas corretivas possam ser

tomadas. Os mesmos distinguem-se pela utilização de números de portas distintas, em que o RTCP recebe número de porta sequente à do RTP.

Os participantes da sessão recebem, periodicamente, pacotes RTCP de controle relativos a uma respectiva sessão RTP, de diferentes tipos, para cada modalidade de informação [12] [48]. São eles:

- SR (*Sender Reports*): contêm informações de transmissão e recepção para transmissores ativos e quantifica os dados enviados até o momento;
- RR (*Receiver Reports*): contêm informações de recepção para participantes que não sejam transmissores ativos, cada qual com referências a *timestamp* e atraso do último relatório recebido;
- SDES (*Source Description*): são pacotes utilizados para controle de sessão e descrição dos parâmetros da fonte, que contêm o CNAME (*Canonical Name*), identificador global para associação de diferentes fluxos de mídia gerados pelo mesmo usuário, possui formato similar ao de um endereço de *e-mail*;
- BYE: enviado pelo participante ao término da sessão;
- APP: adiciona funções específicas de uma aplicação ao pacote RTP.

Em síntese, com referência ao transporte de voz em redes de dados, a Figura 3.10 ilustra o processo de encapsulamento do fluxo de áudio em um pacote IP [11].

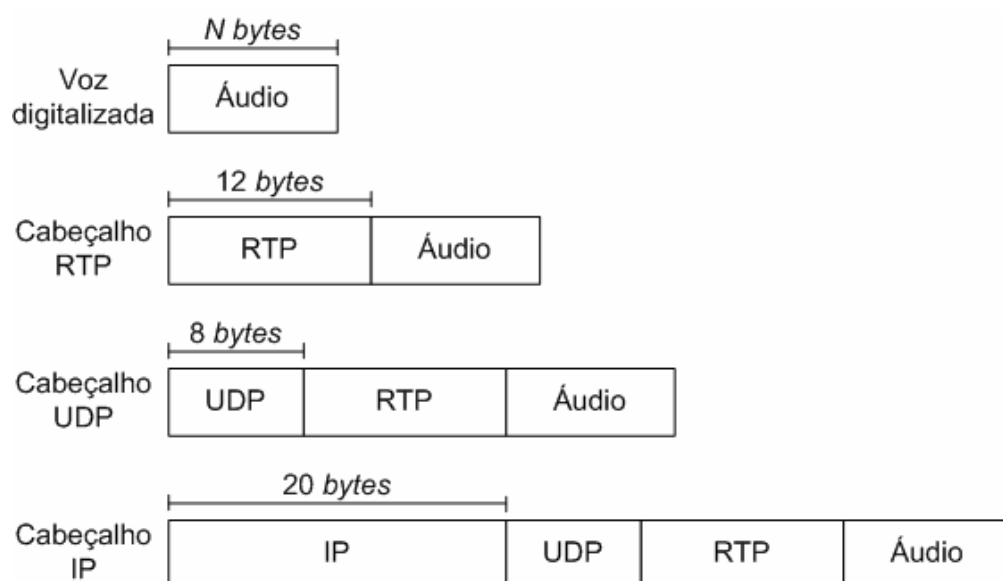


Figura 3.10 – Encapsulamento do Áudio.

A Figura 3.10 representa o processo de encapsulamento do áudio, passando pelos protocolos RTP, UDP e culminando no IP, em que se habilita a ser transmitido nas redes de dados.

Para o trato de chamadas em VoIP, desde o início, manutenção e término, faz-se necessária a utilização de protocolos de sessão, tais como o H.323 e SIP, tratados nas próximas seções.

3.6 H.323

O protocolo H.323 consiste em uma especificação da ITU-T (*ITU's Telecommunication Standardization Sector*) para audioconferência e videoconferência entre sistemas finais na *Internet*. Assim, especifica como o tráfego multimídia é transportado sobre redes de pacotes, com a utilização de padrões existentes, o que o torna complexo, pois, descreve terminais, equipamentos e serviços para comunicação. Abrange ainda o modo como sistemas finais ligados à mesma se comunicam com telefones ligados à PSTN [30] [52].

O H.323 é considerado uma especificação guarda-chuva, pois, é grande e complexo. Consiste em um conjunto de protocolos, integrado verticalmente para conferência multimídia: sinalização, controle de admissão, registro, *codecs* e transporte. O H.323 foi criado para permitir que aplicações multimídia fossem executadas sobre redes de dados não confiáveis, ou seja, sem garantia de qualidade de serviço e, o tráfego de áudio consiste em uma das suas aplicações, entre outras, como compartilhamento de dados e vídeo [22] [30].

Assim, o H.323 tem como peculiaridade dotar-se obrigatoriamente de suporte a áudio, enquanto o suporte a vídeo e transporte de dados lhe são opcionais. Assim, o H.323 cria um canal específico para cada tipo de mídia, em conformidade com a recomendação H.245. Com exceção do canal de áudio, os outros dois canais podem ser usados para diversos tipos de tráfego.

A utilização pelo H.323, no tocante a topologias de rede, é completamente independente, ou seja, utiliza qualquer uma, desde simples a complexas. No entanto,

disso decorre possível perda de escalabilidade. Quanto a sua implementação, ela pode ser tanto em *hardware* quanto em *software*. A Figura 3.11 ilustra a arquitetura do protocolo H.323.

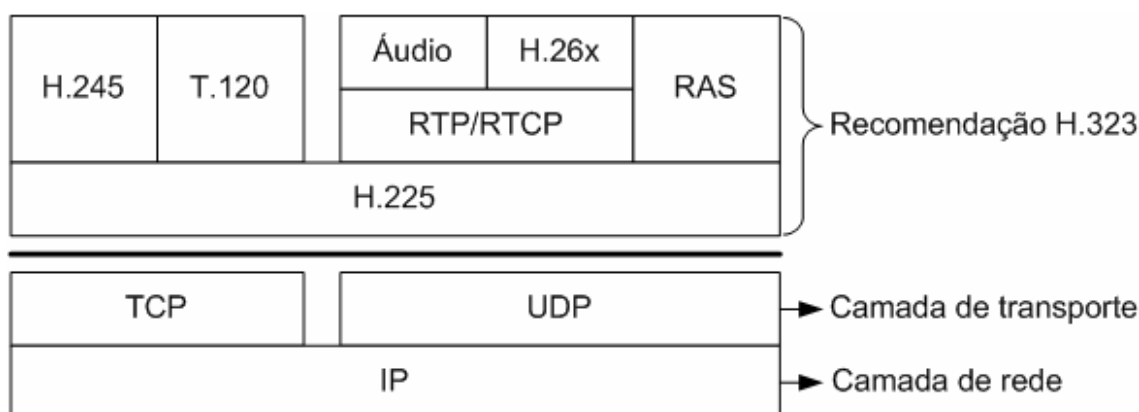


Figura 3.11 – Arquitetura do Protocolo H.323.

São elementos componentes da Figura 3.11:

- H.225: especifica o uso e suporte para mensagens de sinalização Q.931;
- H.245: permite a troca de mensagens de controle fim-a-fim entre entidades;
- T.120: para protocolos de comunicação de dados.
- Áudio: indica o *codec* de áudio utilizado para codificação e decodificação;
- H.26x: indica o *codec* de vídeo utilizado para codificação e decodificação;
- RAS (*Registration, Admission and Status*): sinalização de Registro, Admissão e Status, que fornece o controle de pré-chamada em redes H.323 baseadas em *gatekeepers*.

O registro dos usuários no H.323 é realizado no *gatekeeper* por meio do protocolo RAS. É o responsável por prover serviços de controle pré e durante chamada aos participantes “pontos terminais H.323”, com o decremento de valor presumido de banda disponível a cada admissão. Ainda, baseia-se em domínios administrativos, ou seja, conjunto de *gatekeepers* vizinhos, isso é, servidores contidos numa mesma região administrativa, portadores de diferentes registros de clientes [22] [52].

Para o estabelecimento de uma sessão, previamente à abertura do canal, ocorre a troca de mensagens e detecção das capacidades de mídia e transporte suportadas pelos

terminais envolvidos, o que é feito por meio do protocolo H.245. Os procedimentos de controle de chamadas são baseados na recomendação H.225, que especifica o uso e suporte para mensagens de sinalização Q.931, para conexão, manutenção e desconexão de chamadas de controle confiável. A Figura 3.12 ilustra o estabelecimento de uma sessão entre dois terminais H.323 [52].

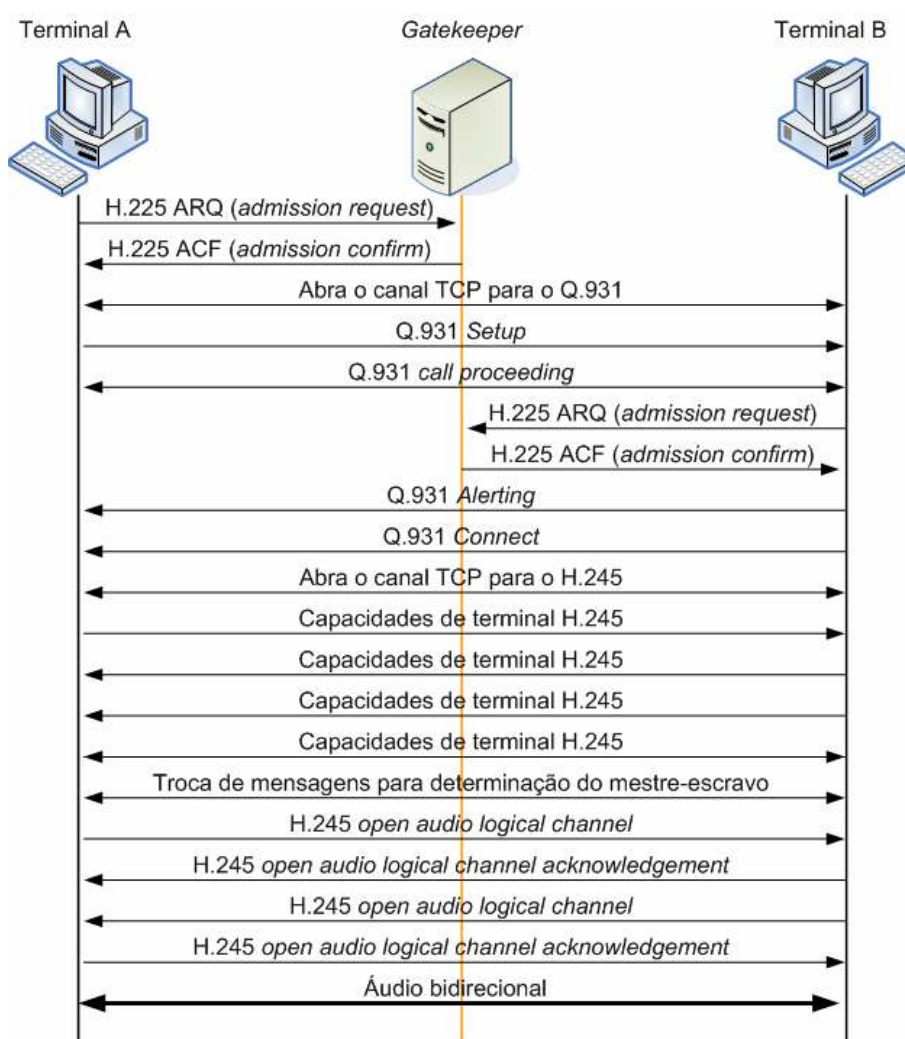


Figura 3.12 – Sinalização H.323 Direta entre Terminais.

A Figura 3.12 é ilustrativa de uma chamada usando a sinalização H.323 direta entre terminais compartilhando o mesmo *gatekeeper*. Assim, a chamada assume extrema complexidade em decorrência da extensa pilha de protocolos que utiliza.

Constata-se na Figura 3.12 serem as mensagens ARQ (*Admission Request*, ou pedido de abertura de sessão) e ACF (*Admission Confirm*, ou a confirmação do pedido)

voltadas exclusivamente aos terminais do protocolo H.323. Tais mensagens em conjunto com as respectivas do *gatekeeper*, LRQ (*Location Request*), LCF (*Location Confirm*) e LRJ (*Location Reject*) constituem o RAS, que possibilita as funcionalidades de Registro, Admissão e Status.

Um terminal registrado em um *gatekeeper* requisita-a para iniciar e ou aceitar chamadas, por meio de mensagens Q.931: *Setup* para estabelecimento de chamada ISDN (*Integrated Services Digital Network*), *Call Proceeding*, equivalente a chamando e, *Connect*, confirma o estabelecimento da chamada.

A fase de inicialização de mídia se dá pelo protocolo H.245, que abre uma porta TCP para negociação dos subconjuntos de mídias suportados bem como a respectiva ordem de preferência, abrindo-se, com isso, um canal, mantido aberto, caso seja estabelecida uma nova sessão de mídia, ou alterada alguma já em curso. São mensagens básicas do H.245:

- *Capability Exchange*: para troca de conjuntos de capacidades de mídia entre terminais;
- *Open Logical Channel*: para abertura de canal de controle do fluxo de mídia;
- *Open Logical Channel Acknowledgement*: para confirmação da abertura do anterior.

Com isso, o fluxo de mídia é transportado via RTP. Como alternativa ao H.323, foi desenvolvido o SIP, tratado na seção seguinte.

3.7 SIP

O SIP (*Session Initiation Protocol*) consiste num protocolo da camada de aplicação usado para estabelecer, modificar e terminar sessões ou chamadas multimídia, que pode ser baseado em texto, permite fácil implementação a linguagens como Java, Pearl e outras, descrito principalmente nas RFCs, 2543 e 3261, a última também conhecida como SIP versão 2. Foi aferido pela IETF (*Internet Engineering Task Force*), pelo que tem se tornado padrão em Soluções IP, e já sendo o mais utilizado [69] [72] [74].

As referidas sessões (*unicast ou multicast*) podem constituir-se em conferências, *e-learning*, VoIP e aplicações similares, a que os participantes podem ser convidados, a ingressarem numa sessão multimídia já em curso, ou iniciar uma nova [25] [26].

O SIP possui arquitetura e paradigma similares aos dos protocolos HTTP (*Hypertext Transfer Protocol*) e SMTP (*Simple Mail Transfer Protocol*), pelo que as requisições geradas pelos clientes são enviadas ao servidor, que as processa para então enviar as respostas aos clientes; incorpora, ainda, o conceito de números de portas fixas para os dispositivos e permite o uso de servidores *Proxy*, em vista à segurança da rede interna [58] [68] [75].

Os serviços do SIP para o estabelecimento e encerramento de sessões multimídia incluem:

- Localização do Usuário: por portar mobilidade o usuário necessita ser localizado previamente ao início de uma comunicação, e ter aferida sua viabilidade para participar da mesma;
- Capacidades do Usuário: determinação das capacidades e parâmetros de mídia a serem utilizados pelos usuários envolvidos na comunicação;
- Disponibilidade do Usuário: após um usuário ser localizado, é necessário saber se estão disponíveis tanto o usuário quanto os recursos;
- Configuração de Chamada: é o processo de definição dos parâmetros que serão utilizados para o estabelecimento da chamada, tanto por parte de quem chama, quanto de quem foi chamado, além de seu respectivo progresso enquanto tocando “*ringing*”;
- Controle da Chamada: é o processo de gerenciamento da chamada, incluindo transferência e encerramento de ligações.

O SIP foi projetado fazendo uso de partes de arquiteturas de dados e controles multimídia, como dos protocolos RSVP (*ReSerVation Protocol*), RTP, RTSP (*Real Time Streaming Transport Protocol*), SDP (*Session Description Protocol*) e SAP (*Session Announcement Protocol*), dos quais, entretanto, independe seu funcionamento [38]. Nas seções seguintes são apresentados maiores detalhamentos do SIP.

3.7.1 SIP URIs

O URI (*Uniform Resource Indicator*) é definido na RFC 3261 como o modo pelo qual os usuários são identificados nas mensagens SIP, constituindo-se numa forma de endereçamento do referido protocolo, cuja formação pode compor-se de várias opções, como o método, o usuário, o número do telefone ou o protocolo que transporte [75]. Elenca-se a seguir alguns exemplos:

- sip:fred@provedor.com.br
- sip:558133332222@provedor.com.br
- sip:fulano@ciclano.com.br;transport=udp

As URIs estão presentes em alguns campos do cabeçalho SIP, entre os quais, *To*, *From*, *Contact* e *Request-URI*, que indicam os destinos, analogamente ao *mailto* utilizado como *hyperlink* em páginas *web*.

3.7.2 Elementos da Arquitetura SIP

Os elementos centrais à arquitetura SIP são os abaixo elencados, que podem entrelaçar-se mutuamente [74] [75] [102]:

- *User Agents* (UA): consiste em qualquer aplicação cliente ou dispositivo que inicie uma conexão SIP, compondo-se de UAC (*User Agent Client*), gerador das requisições, e de UAS (*User Agent Server*), que as responde:
 - UAC: consiste na porção cliente do UA responsável pelo início da comunicação entre cliente e servidor, a qual se inicia por solicitação ou mensagem do tipo *REQUEST*, que desencadeia a transação;
 - UAS: consiste na porção do servidor UA responsável por processar uma mensagem do tipo *REQUEST* enviada pelo UAC;
- *Multimedia Session*: consiste na troca de fluxos de informações entre transmissores e receptores multimídia;

- *Server*: aplicação responsável por receber dos usuários mensagens *REQUEST* e enviar-lhes outras *RESPONSE*, que na prática o constitui num *hardware*, o qual implementa funções de *Proxy Server*, *Redirect Server* e a porção UAS;
- *Proxy Server*: servidor *proxy* que pode atuar tanto como Servidor ou Cliente, como elemento intermediador entre os UAs para interpretação, e, se for o caso, reescrita das mensagens antes de enviá-las. Desta forma, constitui-se o responsável por receber mensagens e encaminhá-las para outros servidores, ou seja, ao receber um *INVITE*, vale-se do *Registrar Server* para saber a localização e status do UA convidado. Possui as funcionalidades de autorização, tradução de endereços, autenticação, aplicação de regras de segurança por meio do controle de acesso à rede e roteamento de chamadas;
- *Registrar* (ou *Registration*) *Server*: é o elemento responsável por manter atualizadas as informações de registro sobre os UAs e ou autenticá-los, bem como compartilhá-los com outros servidores da rede. Sua localização usualmente é no mesmo servidor de que também se vale o *Proxy Server*;
- *Redirect Server*: é o UAS redirecionador das mensagens para outro servidor que contenha informações sobre o respectivo destino, por fornecer ao cliente a lista de endereços possíveis para alcançar a ambos;
- *Location Server*: é o elemento utilizado pelo *Redirect Server* ou *Proxy Server* para identificar as possíveis localizações dos destinos chamados, função geralmente realizada pelo *Registrar Server*.

3.7.3 Requisições SIP

O SIP define dois tipos de mensagens geradas pelo Cliente e Servidor, a saber: *REQUEST* e *RESPONSE*. Uma mensagem do tipo *REQUEST* é enviada de um cliente a um servidor e apresenta o formato indicado pela Figura 3.13 [75] [102].

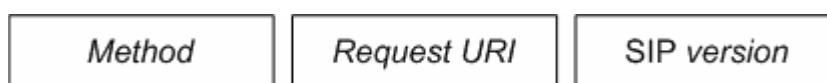


Figura 3.13 – Formato de uma Mensagem do Tipo *REQUEST*.

Tal mensagem pode ter vários métodos (*Method*), cada qual representando uma ação requerida, conforme os exemplos apresentados a seguir [75]:

- *REGISTER*: uma mensagem com este método envia informações sobre identificação e localização do usuário, a fim de registrá-lo em um Servidor SIP;
- *INVITE*: este método é utilizado quando se deseja convidar um novo participante para uma sessão já existente ou para uma nova sessão;
- *BYE*: método utilizado para encerrar a participação numa sessão, liberando-a;
- *CANCEL*: cancela uma requisição pendente, pois não possui efeito numa chamada já estabelecida;
- *OPTIONS*: consulta as funcionalidades suportadas;
- *ACK*: mensagem de confirmação final, a qual é enviada por um usuário que mandou um *INVITE* para avisar do recebimento de uma mensagem *RESPONSE*. Portanto, para toda mensagem com um *ACK* deve existir outra anterior com *INVITE*.

O campo *REQUEST URL* é um SIP URL que compõe a mensagem para indicar quem a originou (*From*) e, quais seus destinos corrente (*REQUEST URL*) e final (*To*). Uma mensagem do tipo *RESPONSE* é enviada após uma mensagem do tipo *REQUEST* ter sido recebida e processada, e apresenta o formato indicado pela Figura 3.14.

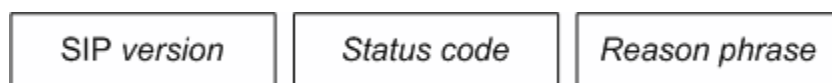


Figura 3.14 – Formato de uma Mensagem do Tipo *RESPONSE*.

O campo *SIP version* da mensagem *RESPONSE* possui a mesma função da mensagem *REQUEST*, portanto, garante a interpretação correta das mensagens SIP. O campo *Status code* possui um número inteiro que identifica o resultado do processamento da mensagem *REQUEST*, à qual a *RESPONSE* se refere. Esse campo apresenta as seguintes informações:

- *INFORMATIONAL*: informa que a mensagem *REQUEST* foi recebida e está sendo processada;

- *SUCCESS*: informa que a mensagem foi recebida, entendida, processada e aceita;
- *CLIENT ERROR*: informa que o servidor não pode processar a mensagem *REQUEST*, por problema interno;
- *GLOBAL FAILURE*: informa que a mensagem *REQUEST* não pode ser processada em nenhum servidor disponível.

3.7.4 Respostas SIP

Os requerimentos do SIP acionam respostas que constam de seis classes, listadas a seguir [52]:

- 1xx = respostas de informações, tal como 200, que significa OK;
- 2xx = respostas de confirmação;
- 3xx = respostas de redirecionamento;
- 4xx = comandos não realizados, erro no cliente;
- 5xx = erros no servidor;
- 6xx = erros globais.

A seção seguinte trata do estabelecimento de uma sessão SIP.

3.7.5 Estabelecimento de uma Sessão SIP

Para o estabelecimento de uma sessão SIP, ou seja, uma chamada, a origem que a inicia gera um *INVITE*, que pode ser encaminhado diretamente ao destino ou a um Módulo Servidor de Sessão, a fim de obter-se o endereço IP do referido destino. Assim que o destino recebe o *INVITE*, ele responde a origem com mensagens indicativas de que a chamada está em progresso. Neste momento, é verificado se o destino, por exemplo, possui algum *codec* em comum à origem, pois, caso não disponha, uma mensagem de erro é retornada a origem.

Assim que a chamada é aceita pelo destino, este encaminha à origem uma mensagem 200 OK, informando que podem iniciar a comunicação, ao que a origem

responde com um *ACK*, em seguida o fluxo de mídia é estabelecido. A partir de então, a chamada pode ser encerrada por qualquer uma das partes, o que ao ocorrer encaminha a outra parte uma mensagem *BYE*, a qual é respondida com um 200 OK [72].

A Figura 3.15 ilustra o estabelecimento de uma chamada utilizando o SIP.

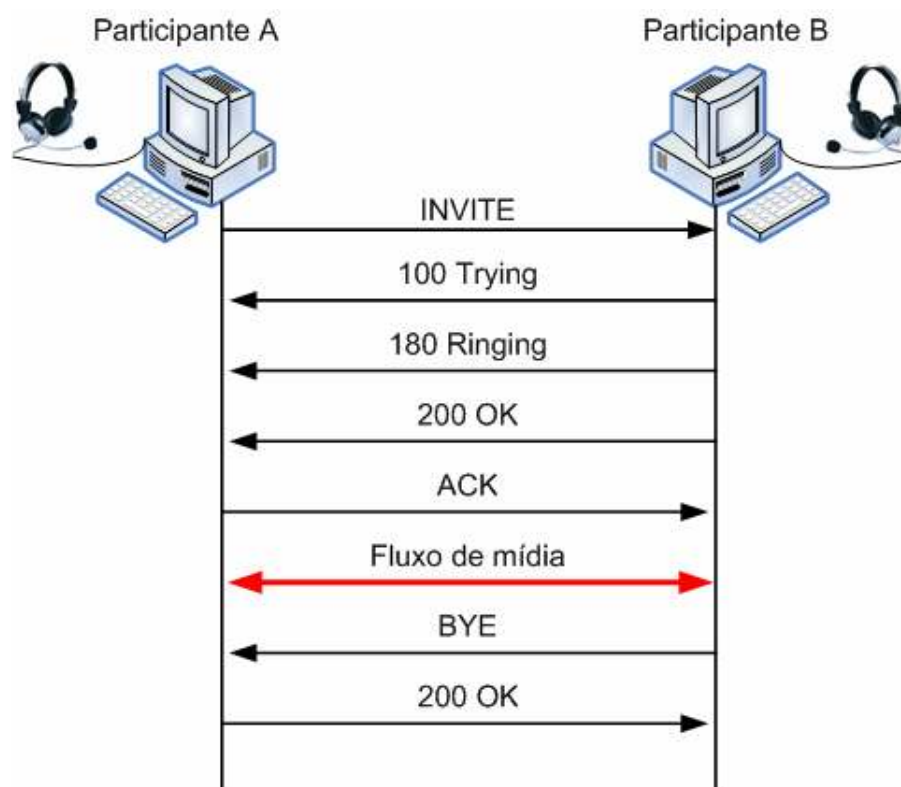


Figura 3.15 – Estabelecimento de uma Chamada Utilizando o SIP.

Constata-se pela análise da Figura 3.15 a troca de mensagens SIP, ilustrativa de uma comunicação via VoIP. A seção seguinte descreve sucintamente o protocolo SDP, o qual leva informações sobre os fluxos de mídia envolvidos na comunicação.

3.8 SDP

O SIP utiliza o protocolo de descrição de sessão denominado SDP (*Session Description Protocol*), especificado na RFC 2327, pois este lhe disponibiliza informações sobre tipos do fluxo de mídia e de conteúdo, endereços, portas, em formato textual. Quando uma chamada é estabelecida via SIP, a mensagem de pedido *INVITE*

contém os parâmetros SDP trazendo as capacidades do originador da chamada e no envio da resposta, os parâmetros SDP são modificados contendo as capacidades do destino da chamada [48].

O SDP provê um formato simples para descrever informação de sessão a seus potenciais participantes, o qual consiste, basicamente, em vários fluxos de mídia, envolvendo a especificação de vários parâmetros relacionados a cada fluxo de mídia e a sessão em geral. Os parâmetros de nível de sessão contêm informações referentes a seu nome, como criador e tempo de atividade, e quanto as referentes ao nível de mídia, abordam tipo, número da porta, protocolo de transporte e formato [12].

Como o SDP provê apenas descrições de sessão e não controla os meios por transportar ou anunciar as sessões aos potenciais participantes, deve ser utilizado em conjunto com outros protocolos, a exemplo do SIP, que o porta integrado à mensagem [12].

Semelhante ao SIP, o SDP constitui-se em um protocolo baseado em texto, que utiliza o conjunto de caracteres ISO 10646 em codificação UTF-8, que habilita o uso de idiomas múltiplos e inclui o EUA-ASCII como um subconjunto [110].

3.8.1 Sintaxe e Campos do SDP

O SDP carrega informações de sessão utilizando-se de várias linhas de texto, cada qual com formato *campo = valor*, em que o campo é equivalente a um caractere e o valor depende do respectivo campo e que, em alguns casos, pode ser fragmentado se, contudo, possibilitar espaço entre campo e o sinal = (igual) ou entre o sinal = e o valor [12].

Os campos de nível de sessão são incluídos previamente aos de nível de mídia. A divisão entre dados de sessão e mídia se dá na primeira ocorrência do campo de descrição de mídia (m=), a partir do que cada ocorrência marca o começo de dados relacionados a outros fluxos na sessão.

O SDP possui campos que devem estar contidos em qualquer descrição de sessão, ou seja, são obrigatórios, bem como outros opcionais, que são descritos a seguir [12] [22].

Obrigatórios:

- v (*Protocol Version*): descreve a versão do protocolo e constitui no delimitador entre o fim da descrição de uma sessão e o início de outra;
- o (*Origin*): descreve a origem da sessão, criador ou identificador da mesma;
- s (*Session Name*): descreve o nome de sessão, que pode ser lido pelos participantes da mesma;
- t (*Timing*): identifica o tempo da sessão, que provê o tempo de início e fim da mesma;
- m (*Media Descriptions*): Existem três linhas “m” cada uma delas identifica o tipo de *stream* de mídia (áudio, vídeo e aplicação de quadro branco), o número da porta para aquele stream, o protocolo e uma lista de cargas.

Opcionais:

- i (*Session Information*): contém informação a respeito da sessão e, pode ser especificado ao nível de sessão como de mídia;
- u (*URI*): contém endereço URI, possibilitador de informações adicionais;
- e (*E-mail Address*): contém o endereço de *e-mail* da pessoa responsável pela sessão, podendo haver várias instâncias deste campo;
- p (*Phone Number*): contém o número de telefone da pessoa responsável pela sessão, podendo haver várias instâncias deste campo;
- c (*Connection Data*): provê informações da conexão, como tipo, rede e endereço e pode ser aplicado aos níveis de sessão ou mídia;
- b (*Bandwidth*): especifica a largura de banda necessária em kbps;
- r (*Repeat Times*): indica a quantidade de vezes em que uma sessão pré-programada será repetida;
- z (*Time Zone Adjustments*): possibilita o ajuste de horários previamente em sessões pré-programadas;
- k (*Encryption Key*): provê a chave para criptografia da sessão. O campo pode ser aplicado em todos os níveis de sessão, mídia, ou ambos;
- a (*Attributes*): campo utilizado para descrever atributos adicionais aplicados à sessão ou a mídias individuais.

A Figura 3.16 ilustra o exemplo da descrição de uma sessão SDP, na qual vários dos campos apresentados são utilizados [28].

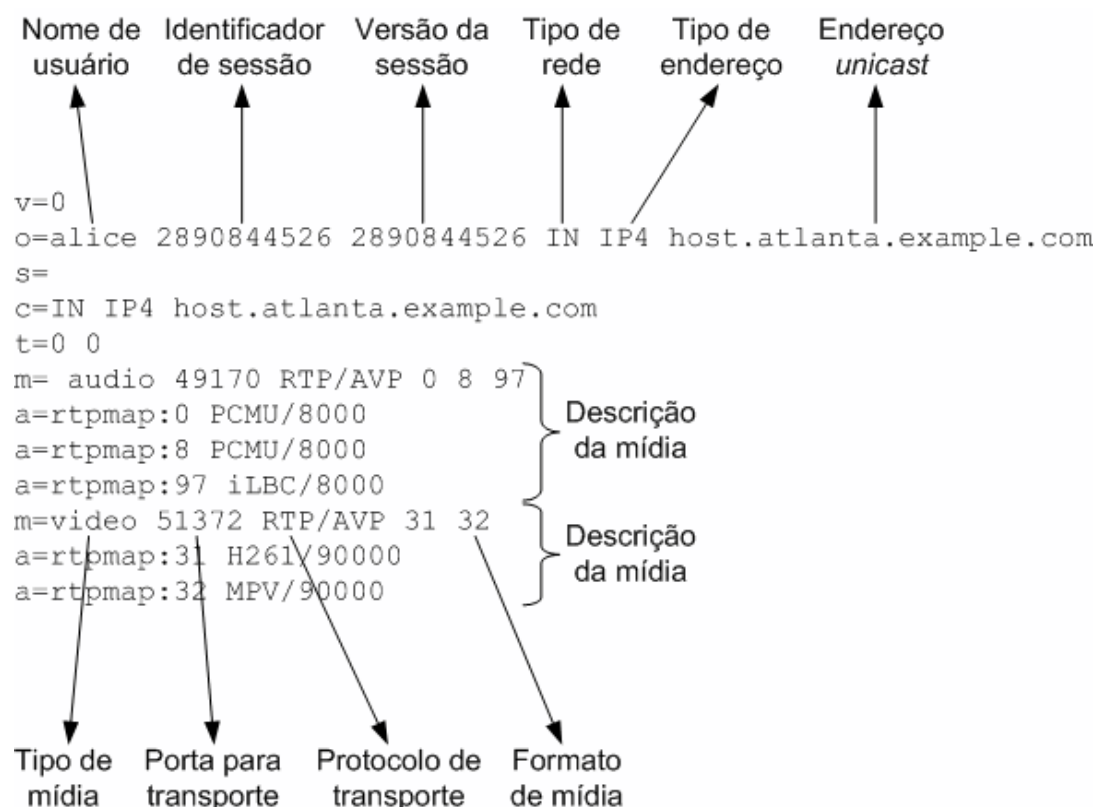


Figura 3.16 – Descrição de uma Sessão SDP.

A Figura 3.16 consiste em uma descrição de sessão SDP, com os campos obrigatórios e alguns opcionais.

Em síntese, o SIP provê o mecanismo para o estabelecimento das sessões multimídia, enquanto o SDP a linguagem para descrevê-las.

3.9 Conclusão

Neste capítulo, são apresentados os principais protocolos empregados para a transmissão de voz em redes de dados, ou seja, o que caracteriza a VoIP. Foram analisadas suas peculiaridades respectivas, com abordagem das restrições e

contribuições de cada qual, daí ser possível aferir a viabilidade da utilização ou não empregabilidade dos mesmos em VoIP.

Em suma, grande parte dos protocolos de sinalização VoIP utiliza RTP/UDP/IP como mecanismo de transporte para o tráfego de áudio. No entanto, os fluxos multimídia por eles transportados necessitam ser codificados, o que, no caso da VoIP, é feito pelos *codecs* de áudio e será objeto de análise do próximo capítulo [52] [75].

Capítulo 4

Codificação da Voz

O *codec*, do inglês *Coder/DECoder*, significa codificador-decodificador, e é referenciado a algoritmo implementado em *software* ou embarcado em *hardware*. Um *codec* possui duas funções básicas: a de codificar e a de decodificar um fluxo de bits. O fluxo de *bits* é resultante da conversão analógica/digital de um sinal de áudio ou de vídeo, de um dispositivo acoplado ao sistema final, que pode ser, por exemplo, um microfone ou uma *webcam*.

Ainda, o *codec* é parte fundamental de uma comunicação que emprega a tecnologia VoIP, pois codifica o fluxo de *bits* proveniente de dispositivo externo, num outro fluxo de *bits* para transmissão numa rede de dados. O mesmo *codec* é utilizado no destino para decodificar, produzindo um fluxo de *bits* o mais fiel possível ao original. Na maioria das vezes, a codificação tem por objetivo a compactação do fluxo de *bits*, promovendo, com isto, a redução da banda de rede necessária a sua transmissão [62] [75]. A Figura 4.1 ilustra o esquema de funcionamento de um *codec* em VoIP.

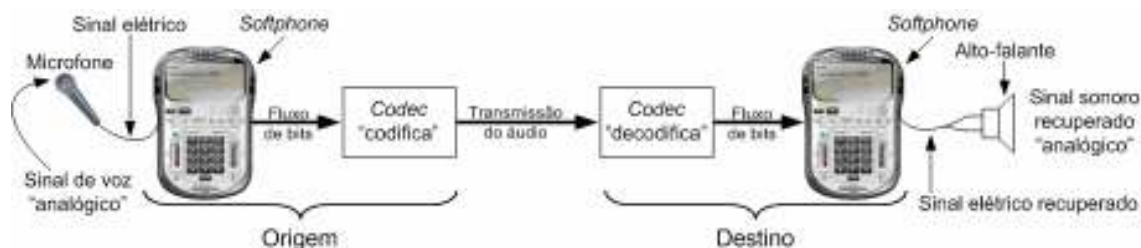


Figura 4.1 – Ilustração do Uso de um *Codec* em VoIP.

A usualidade da conversão digital/analógico é observada em CD *players*, telefones celulares e até consoles de *videogame*. A modalidade *codec* de áudio consiste na parte do sistema VoIP responsável por transformar um fluxo binário de áudio em um novo, que preserve as mesmas características da informação inicial, por meio de uma codificação, como pode ser observado na Figura 4.1.

A codificação tem o intuito de diminuir, ou mesmo de melhor aproveitar a banda de rede utilizada para a comunicação, primando por não afetar de forma significativa a qualidade apresentada na chamada, proveniente da decodificação do respectivo fluxo recebido. De modo análogo, realiza a codificação da voz humana, analógica, transformando-a em uma sequência de *bits*, ou seja, de natureza digital [39] [68] [69].

4.1 Representação dos Sinais A/D e D/A

Um conversor A/D (analógico-digital) é um dispositivo que recebe a entrada de um sinal contínuo analógico e, produz na saída uma sequência de valores discretos ou *amostras* pelo processo de quantização, as quais representam as informações com determinado grau de precisão, pois, são convertidas de analógicas em digitais [12] [35].

Desse modo, as amostras constituem-se como se sucessivas fotografias de um sinal, no caso uma onda sonora, que, por meio de um microfone, é captada analogicamente e convertida pelo A/D em digital, com precisão dependente de sua resolução em tempo e amplitude, ou seja, respectivamente taxa de amostragem e quantização, sendo esta última a quantidade de *bits* utilizados para representá-la. A Figura 4.2 ilustra tal processo [12] [35].

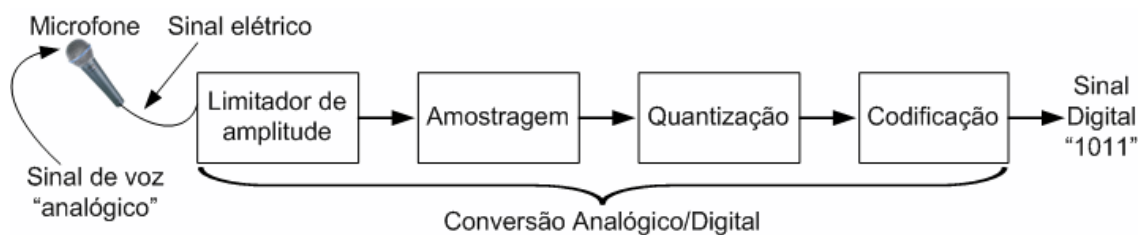


Figura 4.2 – Processo de Conversão Analógico para Digital.

De forma inversa, pode-se recuperar o sinal analógico a partir do digital, por meio do conversor D/A (digital-analógico), a exemplo da Figura 4.3.

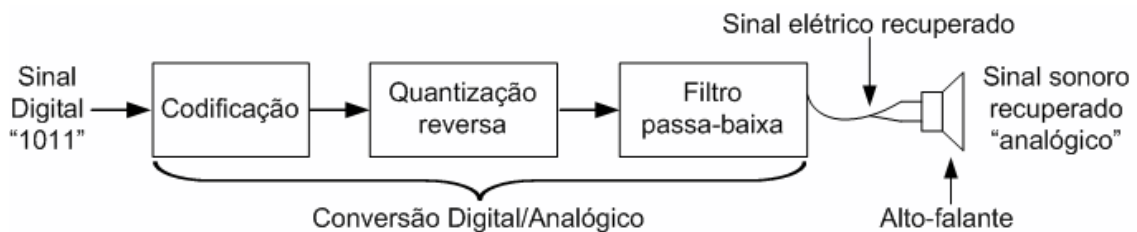


Figura 4.3 – Processo de Conversão Digital para Analógico.

O processo de transformação do sinal analógico em digital, processa-se conforme o ilustrado na Figura 4.4 [90].



Figura 4.4 – Digitalização do Sinal.

A análise da Figura 4.4, possibilita aferir que o processo se dá nas três fases:

- Amostragem: retirada de amostras do sinal original em frequência pré-determinada;
- Quantização: refinamento do sinal amostrado;
- Codificação: transformação do sinal quantizado em binário.

São descritos a seguir os processos de amostragem, quantização e codificação.

4.1.1 Amostragem

Trata-se do processo de medir valores instantâneos de um sinal analógico em intervalos regulares, que são determinados por pulsos de *clock*, cuja frequência é chamada taxa de amostragem, a qual está diretamente relacionada à largura de banda da

freqüência do sinal, ou seja, a sua amplitude a ser transmitida ou armazenada, o que é ilustrada pela Figura 4.5 [12] [90] [95].

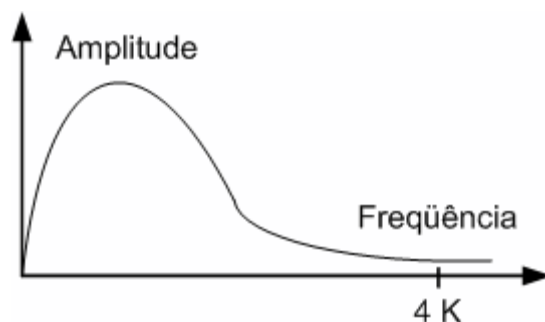


Figura 4.5 – Representação da Faixa de Freqüência.

A Figura 4.5 mostra que a largura de banda ou amplitude de um sinal compreende o intervalo correspondente à diferença entre a maior e a menor freqüência do referido sinal.

A conversão de um sinal analógico em digital envolve a captura de uma série de amostras de fonte analógica, cuja agregação forma o equivalente digital de uma onda sonora. Uma maior taxa de amostragem implica em uma melhor qualidade do som digital, por utilizar mais pontos de referência à replicação do sinal analógico.

Em conformidade com a teoria de *Nyquist*, referente ao processamento de sinais, para representar fielmente um sinal, a taxa de amostragem deve ser, no mínimo, o dobro da freqüência do sinal, ou seja, da freqüência mais alta a ser reproduzida [52].

As freqüências audíveis ao ser humano compreendem, aproximadamente, o intervalo entre 20 a 20000 Hz (*Hertz*). Com isso, torna-se necessária uma taxa mínima de amostragem em torno de 40 KHz (*Kilohertz*) para produzi-la corretamente e integralmente [94].

Para o mesmo processo de conversão analógico-digital, a voltagem (V) deve ser amostrada em intervalos regulares (t), milhares de vezes por segundo, com o arredondamento de cada valor detectado para o inteiro mais próximo na escala de variação, conforme a grandeza da resolução do sinal, para posterior conversão em binários. A Figura 4.6 ilustra o processo de amostragem de um sinal de voz [90].

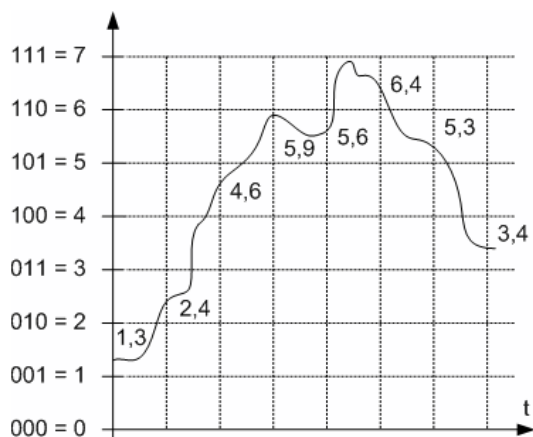


Figura 4.6 – Sinal de Voz Amostrado.

A Figura 4.6 ilustra um sinal de voz amostrado, compreendido entre 300 Hz e 3000 Hz, intervalo que proporciona a recuperação do sinal de voz o mais fidedigno ao original [22]. Ainda, são utilizados 3 *bits* para representar cada amostra. A seção seguinte apresenta a quantização.

4.1.2 Quantização

A quantização consiste no processo de selecionar números para representar a amplitude da tensão de cada amostra, pelo conversor A/D em níveis mais próximos de sinais nos instantes amostrados [48] [95]. A Figura 4.7 ilustra o processo de quantização utilizando 3 e 4 *bits* para o sinal de voz da Figura 4.6.

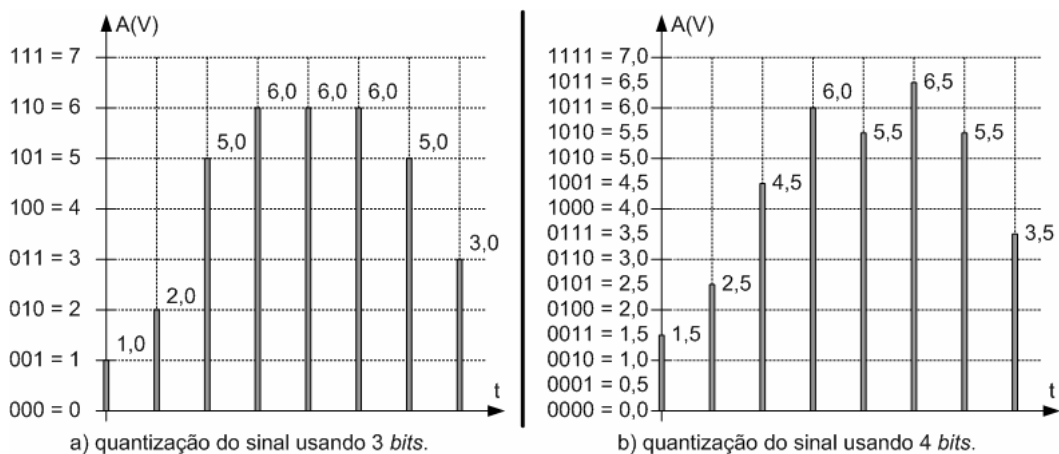


Figura 4.7 – Sinal de Voz Quantizado.

A Figura 4.7 apresenta dois cenários, a) e b), ilustrativos do processo de quantização para o sinal de voz apresentado na Figura 4.6, sendo que na Figura 4.7 a) utiliza-se 3 *bits* para representar os valores de amplitude amostrados. Já na Figura 4.7 b), o processo de quantização é realizado utilizando-se 4 bits para representar os valores. Desse modo, o sinal medido na amostragem é discretizado na quantização, ou seja, convertido num número de *bits* necessários para se representar cada amostra.

Dessa forma, o valor da amostra é arredondado para o código binário mais próximo, como pode ser visto na Figura 4.7. Assim sendo, se aumentado o número de *bits* utilizados no processo de quantização, bem como a taxa de amostragem, há um significativo aumento na qualidade à ser apresentada pelo áudio.

4.1.3 Codificação

Os meios de transmissão, redes de dados, são finitos, o que faz com que a busca por soluções para melhor utilizá-los seja constante alvo de pesquisas no âmbito de codificação. Nesse contexto, para que a comunicação em VoIP ocorra, usualmente utiliza-se o RTP como protocolo de transporte de mídia [12].

Além disso, há a necessidade do uso de codificador de voz, que tem por objetivo representar a fala com um número mínimo de *bits*, mantendo a qualidade perceptível, visando otimizar a transmissão pelo respectivo meio [12] [22]. A Figura 4.8 ilustra o sinal de voz da Figura 4.7 a) codificado.

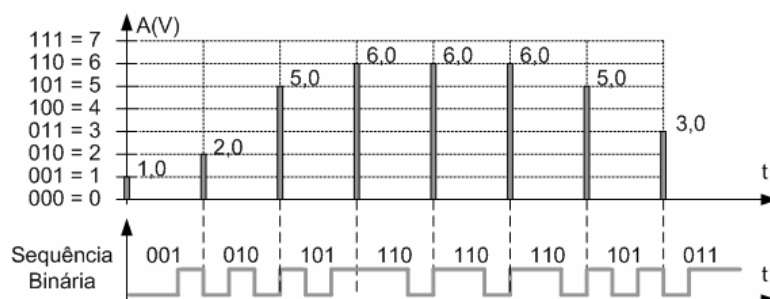


Figura 4.8 – Sinal de Voz Codificado.

A Figura 4.8 ilustra o processo de codificação de um sinal de voz quantizado em três *bits* por amostra, em que o sinal é transformado em binário, levando-se em conta a

sequência em que foi gerado. Para tanto, existem três abordagens aplicáveis à codificação de voz, listadas a seguir [35] [92] [93]:

- Codificação por forma de onda (*Waveform Encoding*), que apresenta excelente qualidade de áudio, no entanto, pouca compressão é aplicada, logo, necessita de grande largura de banda, acima de dezesseis kbps;
- Codificação paramétrica (*Parameter Encoding*), a qual provê elevada taxa de compressão, no entanto, com significativa perda na qualidade;
- Codificação híbrida (*Hybrid Encoding*), a qual combina a qualidade referente aos codificadores de forma de onda com a eficiência da codificação paramétrica.

A Figura 4.9 constitui uma descrição das modalidades referenciadas [35].

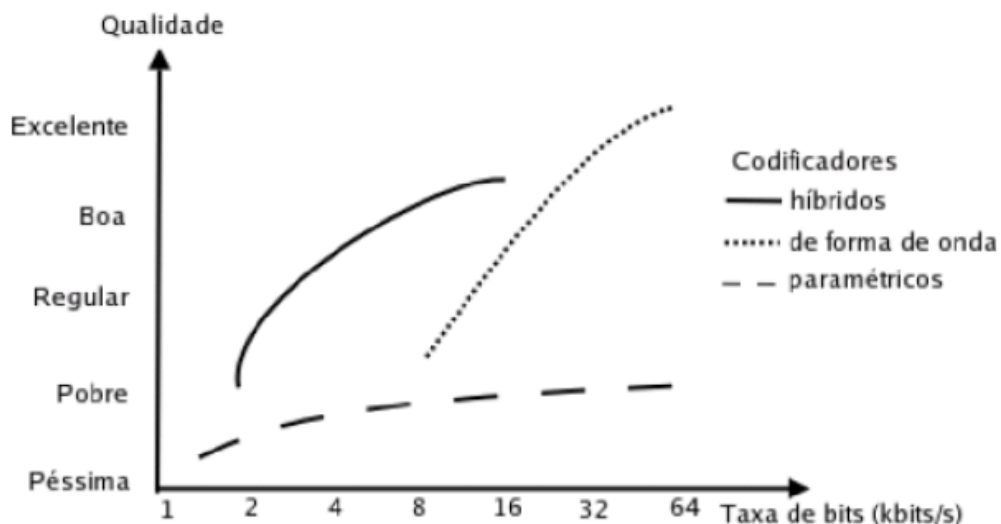


Figura 4.9 – Comparação dos Três Principais Tipos de Codificadores.

A seguir apresenta-se sucintamente os referidos modos de codificação.

4.1.3.1 Codificação por Forma de Onda

As técnicas de codificação por forma de onda (*Waveform Encoding*) mapeiam o sinal original no domínio do tempo com a utilização dos *bits* do sinal digital, constituidoras das amplitudes no tempo.

Segundo [92], conforme o modo de codificação empregado, essas técnicas produzem *bit rates* de altos a moderados, mas, em contrapartida, obtém os melhores índices de qualidade, como o MOS por exemplo, o qual é abordado na seção 4.2. Nas subseções seguintes são abordados os principais codificadores por forma de onda, PCM (*Pulse Code Modulation*), DPCM (*differential PCM*) e ADPCM (*Adaptive Differential PCM*) [96].

4.1.3.1.1 PCM

A mais tradicional entre as técnicas de codificação por forma de onda é conhecida como PCM, cujo uso em telefonia é padronizado pela recomendação G.711 da ITU-T (ITU's *Telecommunication Standards Sector*), que efetivamente se constitui em norma, ou seja, um padrão [93].

Pode-se considerá-la a mais básica para codificar espécies de sinais na forma digital, por atribuir a cada amostra um nível discreto de amplitude. A somatória do número de níveis discretos é dada pela n -ésima potência de dois para os *bits* disponíveis à quantização. Isso significa que para dez níveis distintos de quantização, pode-se dispor de 1024 níveis discretos, com cada uma das amostras representada pela forma binária de sua amplitude [93].

A codificação PCM apresenta a melhor qualidade entre os codificadores utilizados, todavia demanda alto consumo de banda. Contudo, a sensibilidade audível humana para perceber diferenças de volume “amplitude” e maior percepção de sons de baixa intensidade, decresce a medida que a intensidade aumenta. Assim, por exemplo, para se ter a sensação de que o volume dobrou, a potência do sinal, diretamente proporcional à amplitude, deve ser multiplicada por dez, o que justifica a utilização de amplificadores com elevada potência pelos audiófilos [92] [96].

Ainda, segundo [92], a consequência prática é que os erros de quantização são mais perceptíveis para o ouvinte na parte baixa da escala de quantização do que na alta. Na telefonia fixa digital, utiliza-se taxa de amostragem de oito KHz e resolução de oito *bits* pertinente ao ser humano, contudo, o produto de ambos, 64 kbits/s afigura-se relativamente alto para transmissão digital, quer *Internet* quer pela telefonia celular [93].

Esse tipo de codificação ainda é o padrão para todo o tráfego de voz digital pelas estruturas convencionais de comutação e transmissão nas operadoras de telecomunicações [93] [96].

4.1.3.1.2 DPCM e ADPCM

Ao se observar o comportamento dos sinais PCM, constata-se a ausência de variações acentuadas entre duas amostras consecutivas. Comparando os valores binários que codificam uma amostra e sua antecessora, percebe-se que a diferença consiste em um número possível de ser codificado com menos de oito *bits*. Esta técnica é conhecida como DPCM [92] [96].

A codificação DPCM é feita da seguinte forma segundo [92]:

- O sinal de voz é captado e codificado no formato PCM convencional;
- o valor binário de cada amostra PCM é passado para dois circuitos, preditor e diferenciador;
- o circuito preditor cria um *delay* com intervalo de amostragem em torno de 125 μ s, portanto, fazendo com que sua saída sempre contenha o valor binário da amostra anterior;
- o circuito diferenciador compara os valores binários das amostras corrente e anterior, ou seja, na saída do preditor, e calcula a diferença binária entre ambos, sendo a saída do diferenciador o sinal digital a transmitir, como ilustrado pela Figura 4.10.

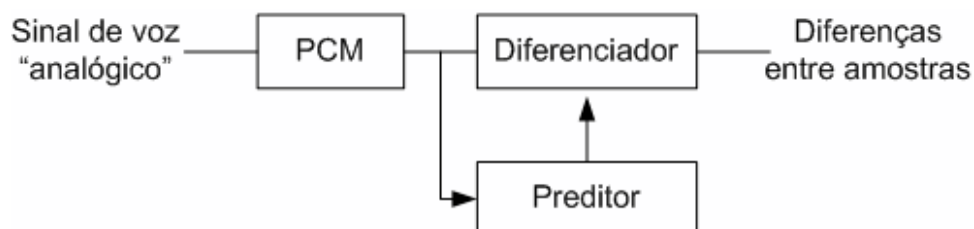


Figura 4.10 – Codificação DPCM.

Já a decodificação segue o processo inverso, como é apresentado pela Figura 4.11.

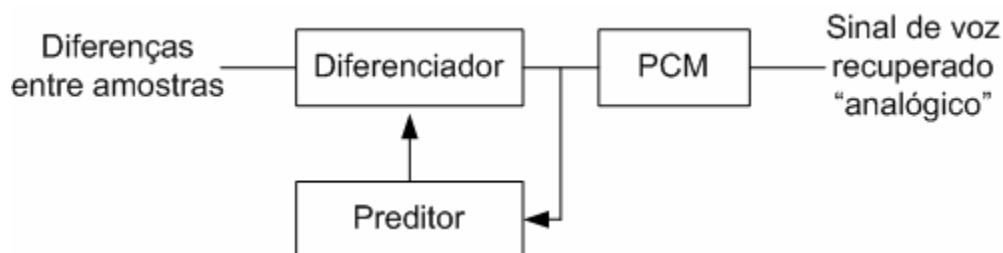


Figura 4.11 – Decodificação DPCM.

Com a manutenção do mesmo esquema básico, apenas com o aprimoramento dos comportamentos dos circuitos de predição e diferenciação, faz-se a transmissão das diferenças calculadas entre cada amostra e seu valor previsto, forma de codificação conhecida como ADPCM, padronizada pela ITU-T nas recomendações G.721, G.726 e G.727, com *bit rates* variáveis entre 16 a 40 kbps [92] [96].

4.1.3.2 Codificação Paramétrica

A codificação paramétrica utiliza-se de determinadas características da fonte de sinal representada. Então, quando for codificar os sinais de voz, deve-se fazer um mapeamento detalhado do trato vocal do ser humano, bem como das propriedades da natureza da voz humana, levando em consideração características como idade, sexo e timbre entre outras [93].

Ao contrário dos codificadores por forma de onda, os paramétricos fazem uso de aspectos particulares, intrínsecos à voz humana, o que os torna específicos para cada tipo de sinal a ser codificado. Dessa forma, estes codificadores produzem apenas informações necessárias para recriar o sistema gerador do sinal. Assim, pode-se conceber o sinal de voz como uma filtragem digital de um sinal aleatório, que pode ser considerada constante se em pequenos intervalos de tempo [93] [96].

Assim, o codificador paramétrico encaminha os coeficientes do filtro, e o reconstitui no receptor, que gera o sinal de voz. Se forem usados intervalos de, por exemplo, 20 ms, um novo modelo de filtro teria que ser gerado e encaminharia os coeficientes a cada 20 ms, ter-se-ia a cada tempo um novo filtro, para que o receptor

pudesse ser atualizado adequadamente. O principal representante dessa modalidade é o LPC (*Linear Predictive Coding*) [93] [96].

Esta codificação é largamente utilizada em aplicações militares, caso em que os *bit rates* baixos importam mais que a qualidade [100].

4.1.3.2.1 Codificação LPC

O LPC é o mais importante dos codificadores da modalidade paramétrica, baseados em uma série de parâmetros inerentes à voz humana. Os codificadores LPC são os mais usuais entre os paramétricos, uma vez que o seu modelo de apenas polos de canal de voz humana funciona de forma bastante eficiente [22].

A voz é um sinal fundamentalmente não estacionário e não periódico cuja análise é muito complexa. Entretanto, se considerado um intervalo de tempo, uma janela, relativamente pequeno (de 10 a 30 ms), o sinal de voz pode ser concebido como estacionário e cada uma das janelas do filtro podem ser atualizadas [93].

A essa análise dá-se o nome de Análise LPC, que deixa de encaminhar as amostras do sinal de voz quantizadas, passando a transmitir os coeficientes por ela determinados, com isto há uma extrema redução de parâmetros com relação aos codificadores por forma de onda, o que diretamente resulta em uma taxa de codificação bem inferior, cerca de até 2,4 kbits/s. O resultado disso é o sinal codificado de áudio ser um tanto “robótico”, característica inerente aos codificadores paramétricos [93] [96] [101].

4.1.3.3 Codificação Híbrida

Essa modalidade de codificadores caracteriza-se pela união de potencialidades das duas anteriores, ou seja, por forma de onda e paramétrica. Em linhas gerais, os codificadores híbridos mantêm a parametrização dos codificadores paramétricos, enquanto geram a excitação pelo formato de onda. Contudo, é possível elevar sua excitação, o que permite redução nas taxas de codificação manutenção da qualidade

bem superior à dos codificadores paramétricos e comparável à dos forma de onda [96] [97].

4.1.3.3.1 CELP

A técnica mais usada nesse tipo de codificador é a CELP (*Code Excited Linear Prediction*), a qual produz boa qualidade de voz a taxas inferiores a 10 kbits/s. Este sistema reúne os mesmos princípios da análise LPC para redução de parâmetros a serem transmitidos. Entretanto, como característica dos codificadores híbridos, a CELP lança mão de um artifício para anular a desvantagem do codificador LPC [96].

A excitação é dada por uma lista de códigos que quantiza os sinais vetorialmente e possibilita o ganho com o controle da sua potência. Em geral utiliza-se índice de dez *bits*, capaz de produzir 1024 entradas com ganho codificado de cinco *bits*, o que reduz a taxa necessária para transmissão [96].

Sendo assim, um número bem maior de excitações com relação ao LPC pode ser utilizado, o que significa que uma gama maior de sinais de voz pode ser reconstituída. Isso torna o sistema bastante mais completo e capaz de gerar um sinal de voz com qualidade bem superior ao gerado pelo LPC. Existe uma série de codificadores baseados na técnica CELP disponíveis no mercado e na literatura, o que comprova a eficácia deste tipo de codificação [93].

4.2 Avaliação de Áudio

A percepção da qualidade de voz por parte do usuário é uma característica subjetiva. Se bem conhecido o emissor e entendidos os padrões de sua fala, mesmo conexões de baixa qualidade podem proporcionar uma comunicação compreensível, diferente de conexões similares com pessoas de sotaques desconhecidos, ou falas rápidas, que tornam a compreensão da conversação muito mais difícil [22].

Outra variável que afeta a percepção da qualidade audível da comunicação é a característica do serviço prestado. Comumente os usuários de telefonia celular ou via

satélite convivem com problemas relativos à qualidade, motivados pela utilidade da respectiva tecnologia.

A Figura 4.12 ilustra uma comunicação VoIP, em que o áudio transmitido e respectivamente recebido pelo destino, corresponde a som, que, por conseguinte, pode compreender voz, do que se depreende que a mesma imprescinde de som e não, necessariamente, de voz.

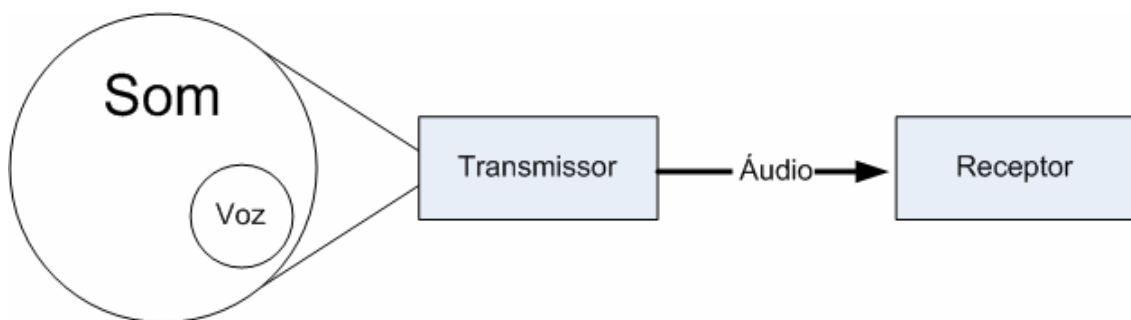


Figura 4.12 – Ilustração de uma Comunicação VoIP.

Da Figura 4.12 decorrem as conceituações [32]:

- Som: tudo o que soa ou ruído;
- voz: som produzido pelo sistema fonador;
- áudio: som transmitido numa faixa de espectro audível.

Dessa forma, os elementos que definem a qualidade de serviço para o nível de usuário da comunicação em VoIP são:

- Qualidade do Som;
- Qualidade da Voz.

Os quais em síntese são apresentados como subitens seguintes.

4.2.1 Qualidade do Som

Numa forma primária de descrição de uma chamada de voz, são atribuídos ao som os atributos clareza, fidelidade, inelegibilidade e não apresentar distorção.

Uma chamada é definida pelos componentes [14]:

- Nível do Sinal (*Loudness*) – o volume da fala não pode ser muito baixo (sussurro) ou muito alto (grito);
- distorção (*Distortion*) – ocorre distorção na conversão da fala analógica para digital na tecnologia PSTN, a qual é perceptível para quem está escuta, e quanto maior, pior a compreensão da chamada, pode até mesmo nem ser reconhecida a voz do falante pelo ouvinte;
- ruído (*Noise*) – existe como fundo nas chamadas de modo estático ou como zumbidos, apesar de poderem apresentar-se imperceptíveis;
- *fading* – o nível do sinal pode sofrer variações de aumento ou diminuição durante a chamada;
- chamada cruzada (*Crosstalk*) – situação em que outra conversação em uma chamada paralela possa ser ouvida na chamada do usuário.

4.2.2 Qualidade da Voz

Os próximos elementos a seguir, mensuradores da qualidade da voz, combinados com os anteriores resultam na qualidade da conversação [14] [33] [62] [75]:

- Eco (*Echo*) – é o som da voz emitida retornando para o emissor e sendo ouvida pelo próprio. Pensa-se o *echo* como um problema de *delay* em função da distância de ida e volta da voz em que pequenos *delays* de eco podem ser imperceptíveis, ao passo que quanto mais longo for o *delay* na ida e volta da voz torna-se mais difícil ser ignorado. Com isso fazem-se necessárias pausas durante a fala, com vistas a evitar interferências na conversação pelo eco [64];
- latência (*Delay* de fim-a-fim) – é o tempo que a voz leva para atingir aos ouvidos do receptor. Na PSTN, o *delay* atinge aproximadamente cerca de 30 ms. Já em VoIP, o objetivo é manter a latência, de uma única via, em até 100 ms ou menos, com tolerância de no máximo 150 ms, a fim de evitar-se pausas durante a fala, desconhecimento do seu término ou sobreposição [49];
- variação do atraso (*jitter*) – é a variação do tempo de atraso da chegada dos pacotes, decorrente da variação ocasionada na transmissão pela rede, em função

dos pacotes nem sempre serem transmitidos pelo mesmo caminho, podendo alguns levar mais tempo que outros. A Figura 4.13 ilustra o *jitter* [33].

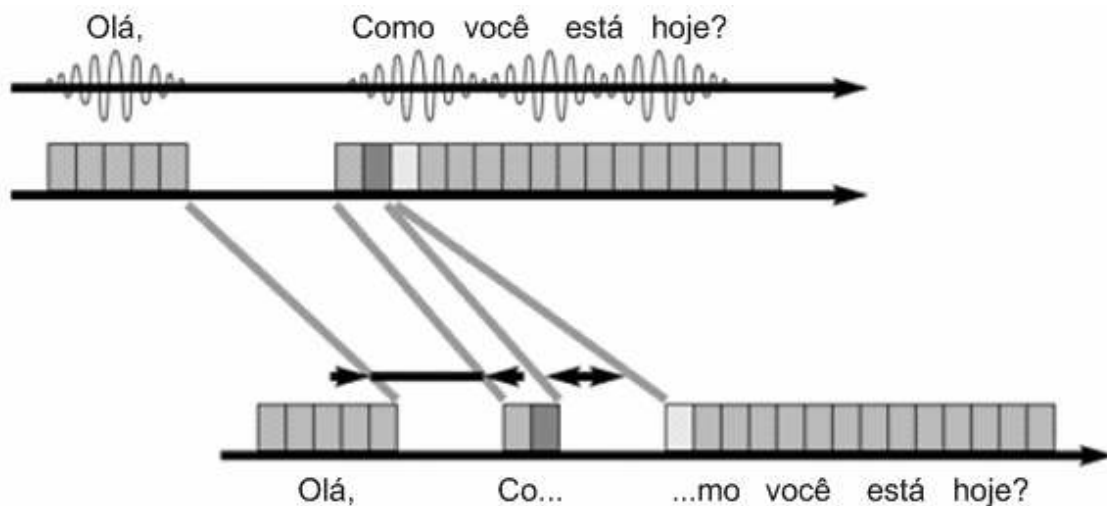


Figura 4.13 – Efeito do *Jitter* na Percepção da Fala.

- supressão de silêncio / VAD (*Voice Activity Detection*) – consiste em uma função utilizada em VoIP para prover a redução do consumo de banda, pois, durante uma comunicação, usualmente, apenas um dos participantes fala, enquanto o outro escuta, forma pela qual o ruído de fundo do ambiente é filtrado e desse modo somente ocorre a transmissão dos pacotes quando alguém estiver falando. A redução de pps (pacotes por segundo) obtida pela utilização de VAD é variável, dependendo dos padrões de conversação dos usuários, num percentual estimativo conservador de 35% a 50% dos mesmos [18] [64];
- perda de pacotes (*Packet Loss*) – As redes de dados não asseguram a entrega de pacotes nem a chegada dos mesmos na ordem de envio, bem como não foram projetadas para transmitir voz, a qual por ser encapsulada em pacotes de dados, pode, em tráfego congestionado, sofrer descartes;
- cancelamento de eco – quanto maior a latência, maior é a necessidade de eliminação do *echo*, que pode ocorrer tanto uni quanto bidirecionalmente, e o seu cancelamento pode não funcionar, ou ainda não ser capaz de produzir compensação efetiva durante a chamada VoIP, quando existir um significativo *jitter*.

Conforme os mencionados parâmetros, o ITU (*International Telecommunication Union*) sintetiza na Tabela 4.1 as medidas pertinentes a VoIP [49] [64].

Tabela 4.1 – Medidas de Qualidade em VoIP.

Parâmetro de rede	Bom	Aceitável	Pobre
<i>Delay</i> (ms)	0 - 150	150 - 300	> 300
<i>Jitter</i> (ms)	0 - 20	20 - 50	> 50
Perda de Pacotes	0 - 0,5%	0,5 - 1,5%	> 1,5%

A observância da adequada combinação desses elementos contribui para a clareza da voz na chamada, que em seu respectivo destino é avaliada por um MOS.

4.2.3 MOS

O MOS (*Mean Opinion Score*) constitui-se num padrão numérico utilizado para mensurar e reportar a qualidade da voz após a sua compressão e ou transmissão. O MOS pontua apenas a qualidade da voz e do som, cujos valores decrescem de cinco pontos (considerado como falar próximo ao ouvido de uma pessoa) ao mínimo de um (qualidade inaceitável) [22] [75].

A Tabela 4.2 apresenta a pontuação utilizada no MOS, bem como sua descrição segundo [109].

Tabela 4.2 – Pontuação de MOS.

Pontuação	Definição	Descrição
5	Excelente	Um sinal de voz perfeito gravado em um local silencioso
4	Bom	Qualidade de uma chamada telefônica de longa distância (PSTN)
3	Razoável	Requer algum esforço na escuta
2	Pobre	Fala de baixa qualidade e difícil de entender
1	Ruim	Fala não clara, quebrada

Em VoIP, um MOS de 4,4 a 4,5 é considerado equivalente à qualidade obtida em uma chamada por PSTN, pelo que satisfaz aos usuários que a utilizam. Enquanto um MOS de 4,0 ainda é aceitável à maioria dos usuários, o mesmo não acontece para valores de 3,5 abaixo.

A maioria das chamadas por celular possuem um MOS entre 3,8 a 4,0, razão pela qual a voz emitida bem como o reconhecimento da mesma podem ser afetados. Um MOS abaixo de 2,6 caracteriza uma péssima chamada, pelo que os usuários necessitam recorrer a outra rede para realizarem chamadas.

O MOS tem por padrão de medição o P.800 da ITU, que define as técnicas pertinentes, cuja última atualização ocorreu em meados de 1990. Para a medição do MOS submetem-se 30 ou mais pessoas entre oito a dez segundos de fala, em condições controladas e lhes solicitam a opinarem sobre as chamadas, conceituando-as desde muito boas a terríveis, o que corresponde à pontuação entre cinco e um. No entanto, com advento da telefonia celular, a indústria passou a preocupar-se com a mensuração da qualidade de voz, de modo mais acurado com o recurso de máquinas [14] [20].

A Tabela 4.3 relaciona o MOS de alguns *codecs* usuais em chamadas VoIP [42] [62].

Tabela 4.3 – Alguns *Codecs* e suas Respectivas Taxas de Transmissão e MOS.

Método de Compressão	Bit Rate (kbit/s)	MOS
G.711 PCM	64	4.1
G.726 ADPCM	32	3.85
G.728 LD-CELP	16	3.61
G.729 CS-ACELP	8	3.92
G.729 x 2 <i>Encodings</i>	8	3.27
G.729 x 3 <i>Encodings</i>	8	2.68
G.729a CS-ACELP	8	3.7
G.723.1 MP-MLQ	6.3	3.9
G.723.1 ACELP	5.3	3.65

Constata-se, pela Tabela 4.3, que um *codec* que apresente baixa taxa de utilização da rede para transmissão aliado a elevado MOS, propicia melhor qualidade na comunicação.

A comunicação no âmbito da VoIP, processa-se naturalmente, quando os participantes utilizam idênticos *codecs*. O mesmo não ocorre quando os *codecs* são distintos, situação que somente torna possível a comunicação por meio da transcodificação, a qual é tratada no tópico seguinte.

4.3 Transcodificação

Entende-se pelo processo de transcodificação a conversão entre diferentes tipos de mídias, com o intuito de prover comunicação, a exemplo de um fluxo de vídeo encapsulado em um formato, que, após transcodificado, é encaminhado ao destino em outra forma, ou uma comunicação telefônica, na qual o originador efetua uma chamada por meio da PSTN, com intuito de atingir um usuário da VoIP, em que se faz necessário, para que se complete um *Media Gateway*, dotado de *hardware* específico, como placas FXO (*Foreign eXchange Office*) ou E1 [66] [68] [69] [75] [79].

Nesta dissertação, o termo transcodificação é utilizado para descrever a operação realizada por um *Media Gateway* com *codecs* de áudio. Ou seja, caso dois ou mais usuários da VoIP tentem estabelecer uma chamada e esta não seja completada devido à inexistência de *codecs* comuns entre os mesmos, conforme ilustra a Figura 4.14.

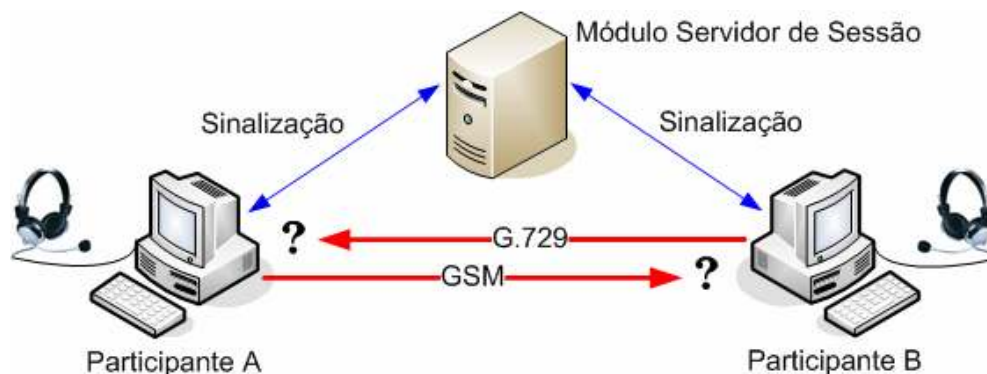


Figura 4.14 – Chamada VoIP não Completada.

Para que, então, a chamada se complete, utiliza-se um *Media Gateway* que, ao receber um fluxo de áudio encapsulado em um formato, o desemcapsula e em seguida o encapsula num formato no qual o destino consiga entendê-lo, conforme o *codec* disponível. Um exemplo prático é ilustrado na Figura 4.15, em que uma chamada VoIP intermediada por um Módulo Servidor de Sessão é estabelecida entre os Participantes A e B, que não dispõem de *codecs* comuns, pelo que entra em cena o *Media Gateway*, que provê a comunicação efetiva entre eles por meio da Transcodificação [68] [79].

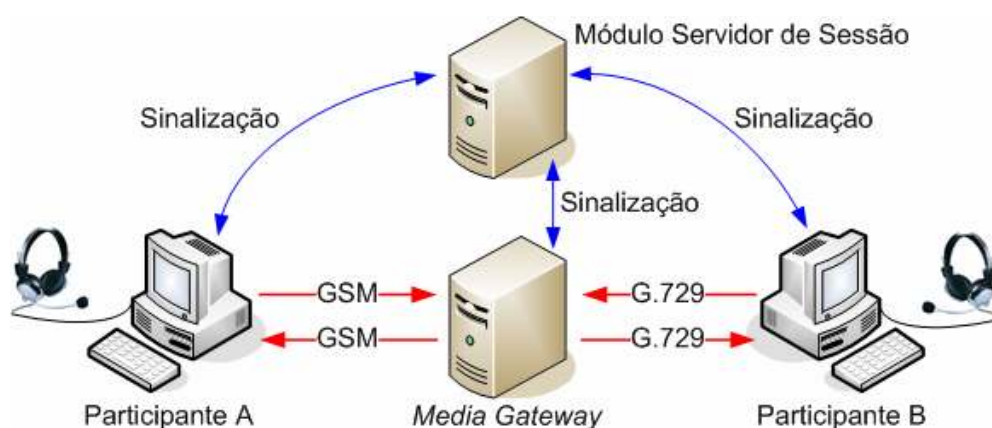


Figura 4.15 – Chamada VoIP Estabelecida por Meio de *Media Gateway*.

O Participante A encaminha seu fluxo de áudio encapsulado pelo *codec* GSM com destino ao Participante B, que, então, por não dispor deste *codec* utiliza-se do *Media Gateway*, que decodifica o referido fluxo e novamente o codifica no formato G.729. Vice-versa ocorre de B para A, possibilitando assim a transcodificação e com isso a comunicação entre ambos, fato que não ocorre pela ausência de *Media Gateway*, como ilustrado na Figura 4.15.

Pelo fato de a Transcodificação demandar considerável uso do processador, um elevado número de chamadas que dela necessitem podem transformar-se num problema crítico ou mesmo torná-la fator limitante se atingido 100% da utilização do recurso, circunstância em que novas chamadas não podem ser completadas, o que é indesejável a qualquer sistema de comunicação [44] [47] [77].

Segundo *benchmark* da TransNexus, cada um GHz (*Gigahertz*) de processamento de um *Media Gateway* pode atender a cerca de 30 chamadas simultâneas com transcodificação do *codec* G.711 μ -law para o G.729. Exemplo prático é o servidor

Dell (*Intel Xeon X3220 Quad core*, 2.40 GHz, 4 GB (*Gigabyte*) RAM (*Random Access Memory*)), que atingiu 95% de utilização do processador, ao transcodificar 288 chamadas simultâneas [47].

O fato de um *Media Gateway* estar com seu processador em alto percentual de uso pode comprometer a qualidade de uma ligação, pois isso pode ocasionar atrasos na transcodificação. Acrescem-se a isso os riscos inerentes à própria VoIP relativos a perdas de pacotes ou *delays* durante a transmissão.

Além disso, a transcodificação demanda pequeno intervalo de tempo, em milissegundos, para que acontecer. As Tabelas 4.4 e 4.5 referem-se aos dois *Media Gateways* utilizados no cenário de desenvolvimento, cujos dados foram obtidos do *software* Asterisk PBX por meio do comando *core show translation* [38] [41].

Tabela 4.4 – Tempos de Transcodificação pelo *Media Gateway* 1 (*asterisk1*).

Codecs de origem	Codecs de destino											
	g723	gsm	ulaw	alaw	g726aal2	slin	lpc10	g729	speex	ilbc	g726	g722
g723	-	-	-	-	-	-	-	-	-	-	-	-
gsm	-	-	2	2	3	1	6	-	-	39	3	-
ulaw	-	4	-	1	3	1	6	-	-	39	3	-
alaw	-	4	1	-	3	1	6	-	-	39	3	-
g726aal2	-	4	2	2	-	1	6	-	-	39	1	-
slin	-	3	1	1	2	-	5	-	-	38	2	-
lpc10	-	5	3	3	4	2	-	-	-	40	4	-
g729	-	-	-	-	-	-	-	-	-	-	-	-
speex	-	-	-	-	-	-	-	-	-	-	-	-
ilbc	-	6	4	4	5	3	8	-	-	-	5	-
g726	-	4	2	2	1	1	6	-	-	39	-	-
g722	-	-	-	-	-	-	-	-	-	-	-	-

Tabela 4.5 – Tempos de Transcodificação pelo *Media Gateway* 2 (*asterisk2*).

Codecs de origem	Codecs de destino											
	g723	gsm	ulaw	alaw	g726aal2	slin	lpc10	g729	speex	ilbc	g726	g722
g723	-	-	-	-	-	-	-	-	-	-	-	-
gsm	-	-	3	3	5	2	9	-	-	-	6	-
ulaw	-	5	-	1	4	1	8	-	-	-	5	-
alaw	-	5	1	-	4	1	8	-	-	-	5	-
g726aal2	-	7	4	4	-	3	10	-	-	-	1	-
slin	-	4	1	1	3	-	7	-	-	-	4	-
lpc10	-	8	5	5	7	4	-	-	-	-	8	-
g729	-	-	-	-	-	-	-	-	-	-	-	-
speex	-	-	-	-	-	-	-	-	-	-	-	-
ilbc	-	-	-	-	-	-	-	-	-	-	-	-
g726	-	7	4	4	1	3	10	-	-	-	-	-
g722	-	-	-	-	-	-	-	-	-	-	-	-

Ao se compararem as Tabelas 4.4 e 4.5, percebe-se a nítida diferença entre os respectivos tempos despendidos por ambos os *Media Gateways*, quando da transcodificação; o dotado de maior capacidade de processamento apresenta considerável redução, ou seja, para efetuá-la requer menor tempo.

As Tabelas 4.4 e 4.5 apresentam-se tabuladas por linhas de entrada e colunas de saída. Delas tomam-se por exemplo os tempos de transcodificação dos *codecs* GSM para G.711 *a-law*, pelo que se constata o emprego de menor tempo pelo *Media Gateway* 2, por dispor de maior capacidade de processamento para realizá-la.

Os tracejados constantes das Tabelas 4.4 e 4.5 ilustram *codecs* indisponíveis dos respectivos *Media Gateways* utilizados no ambiente VoIP cenário de testes, apresentado no capítulo seguinte, que, no entanto, podem ser instalados.

Uma solução prática ao problema de descartes de chamadas VoIP com necessidade de transcodificação consiste na adição de tantos *Media Gateways* quantos forem necessários ao atendimento das chamadas demandadas. O problema constitui-se em como distribuir as chamadas VoIP que necessitam de transcodificação de modo a não sobrecarregar nenhum dos *Media Gateways* durante o processo de encaminhamento das mesmas.

A utilização do método *Round Robin*, mecanismo de balanceamento local utilizado pelos servidores de DNS (*Domain Name Server*) para partilhar e distribuir cargas de recursos na rede, é ineficaz em VoIP por não levar em consideração a carga dos *Media Gateways* [27].

No entanto, ainda, diferentemente de uma página *web*, que é carregada “*download*” com imediata desalocação do recurso, o *Media Gateway*, em uma chamada VoIP não há como prever sua respectiva duração, e a utilização do recurso se dá de modo contínuo, cuja imprevisibilidade de desalocação desfavorece o balanceamento.

Dessa forma, torna-se mais factível o balanceamento das chamadas VoIP carentes de transcodificação entre os *Media Gateways* com base no percentual de uso de seus respectivos processadores. Entretanto, se os *Media Gateways* estiverem com todos os processadores em elevado percentual de uso, as chamadas que chegarem sofrerão descartes a fim de evitar comprometimento das já em andamento [59]. Dessa forma, o

Módulo Balanceador de Chamadas apresenta-se como inovadora e eficiente *solução* ao problema de fato do balanceamento.

4.4 Conclusão

Este capítulo apresenta a codificação da voz, desde sua amostragem, quantização e, por fim, a codificação, para que, a partir de então, ela possa ser transmitida por meio das redes de dados. Enfoque especial foi dado à codificação, ilustrando as três formas, híbrida, paramétrica e por forma de onda, em vista a evidenciar as peculiaridades de cada modalidade.

Apresenta, ainda, a transcodificação, que permite a comunicação entre participantes com *codecs* distintos. No entanto, ela incorre em considerável demanda do recurso processador, o que pode acarretar problemas tanto de escalabilidade quanto de qualidade das chamadas que necessitem de transcodificação. A fim de prover solução eficaz a tal problema, é apresentado no próximo capítulo, o Módulo Balanceador de Chamadas.

Capítulo 5

Balanceamento de Chamadas VoIP a Transcodificar

A transcodificação faz uso de consideráveis recursos computacionais, em especial de processador, para o que requer um cenário adequado, ou seja, com *Media Gateway*. Contudo, um único *Media Gateway* restringe o número de chamadas VoIP a transcodificar, pelo que se faz necessária a adição de outros. Contudo, isso implica decidir a qual deles encaminhar a chamada, sobretudo, obedecendo a um critério equitativo, que possibilite a adequada utilização dos recursos disponibilizados e reduza a probabilidade do descarte de chamadas.

Para a distribuição equitativa das chamadas VoIP que necessitam de transcodificação, ou seja, a realização do balanceamento de chamadas VoIP a transcodificar, há necessidade de monitorar os percentuais de carga disponíveis nos respectivos *Media Gateways*, que compõem o cenário ou ambiente VoIP e a utilização do SNMP (*Simple Network Management Protocol*) torna-o possível.

De posse dos respectivos percentuais, ocorre a demanda por funções capazes de interagir, a fim de realizarem o equitativo balanceamento. Se concebidas em suas individualidades, essas funções constituem sub-módulos específicos, componentes de um possível modelo ou módulo integrador, dotado do mecanismo de balanceamento. Dessa forma, atenderia ao objetivo deste trabalho, que consiste no *balanceamento de*

chamadas VoIP a transcodificar, pelo desenvolvimento e utilização de um Módulo Balanceador de Chamadas.

Este capítulo apresenta a concepção e o desenvolvimento do Módulo Balanceador de Chamadas como solução ao balanceamento de chamadas VoIP a transcodificar, de forma descritiva e funcional, cujas etapas são ilustradas pela Figura 5.1.



Figura 5.1 – Esquema Ilustrativo das Etapas do Trabalho.

Da Figura 5.1 depreende-se que o trabalho pautou-se em síntese:

- 1 – Na concepção e desenvolvimento do Módulo Balanceador de Chamadas;
- 2 – Adaptação do Módulo Balanceador de Chamadas ao Módulo Servidor de Sessão e customização dos *Media Gateways*.

Num ambiente VoIP, composto apenas pelos elementos Módulo Servidor de Sessão e os participantes, constata-se que, caso os últimos não disponham de *codecs* compatíveis, não é possível o estabelecimento de uma chamada entre os mesmos, haja vista a inaptidão do Módulo Servidor de Sessão à transcodificação de chamadas; isso demanda a inserção do *Media Gateway* nesse ambiente, como terceiro elemento, pois ele provê tal funcionalidade.

Assim sendo, faz-se necessária a integração entre o Módulo Servidor de Sessão e o *Media Gateway*, visto que ambos apresentam peculiares funcionalidades, com o intuito de melhor aproveitá-las e possibilitar o estabelecimento de uma chamada entre os participantes com *codecs* diferentes. Enquanto o Módulo Servidor de Sessão é especializado em atender a elevado número de chamadas VoIP não transcodificadas, o *Media Gateway* supre tal necessidade [38].

A união de ambos pode prover transparência aos participantes, quando da necessidade de transcodificação, sem que eles notem o que ocorreu. Ainda, por tal

integração, reiteradamente intentada, a viabilidade técnica obtida restringiu-se a apenas um *Media Gateway*, o que suscitou o problema do tratamento da imprevisibilidade de demanda, que, acima do limite por ele suportado compromete o sistema, no tocante à qualidade das chamadas, bem como pela ocorrência do descarte das que não possa atender.

A referida imprevisibilidade, se tratada apenas com o acréscimo de *Media Gateways*, não se resolve, por não contemplar um mecanismo sequer de distribuição de chamadas, o que resultou na necessidade de desenvolvê-lo. Isso se deu pelo Módulo Balanceador de Chamadas, que provê um balanceamento de chamadas VoIP a transcodificar, ou seja, chamadas VoIP que necessitam de transcodificação serão por ele distribuídas equitativamente entre os *Media Gateways* de modo a evitar a sobrecarga de qualquer deles, além de prover escalabilidade. No entanto, para esse balanceamento é imprescindível o monitoramento dos percentuais disponíveis de carga dos *Media Gateways*.

Para a aferição dos percentuais disponíveis de carga nos *Media Gateways*, faz-se necessário o monitoramento dos principais recursos necessários a transcodificação, entre os mais importantes, destaca-se principalmente o processador, bem como a memória. Assim, fica evidente a dinâmica de funcionamento do Módulo Balanceador de Chamadas, mostrando com isto, que vários outros elementos podem ser monitorados e tomados como referência para a realização do balanceamento de chamadas VoIP a transcodificar [39] [47] [59] [77].

Assim sendo, para que o Módulo Balanceador de Chamadas proceda ao equitativo balanceamento de chamadas VoIP a transcodificar, foi utilizado o protocolo SNMP, para por meio dele monitorar os recursos computacionais dos *Media Gateways*, no caso o processador e a memória, e repassar os respectivos percentuais ao Módulo. O protocolo SNMP é contemplado na próxima seção [39] [55] [59].

5.1 SNMP

O SNMP (*Simple Network Management Protocol*), definido principalmente nas RFCs 3410 e 3584, constitui-se num protocolo de gerência de redes de computadores,

muito utilizado para possibilitar a comunicação entre os seus respectivos elementos. Devem ser disponibilizados objetos por ele gerenciáveis, agrupados em uma base de dados virtual, denominada MIB (*Management Information Base*), que contenha informações dos mesmos, e possa ser customizada e estendida a valores específicos [30] [31].

Na MIB está disposto o conjunto padrão de dados estatísticos e de controle, que descreve o status fornecido pelo agente SNMP dos recursos computacionais monitorados, que nesta pesquisa, constituem em processador e memória dos *Media Gateways*. Constitui-se ainda em uma estrutura de dados organizada na forma de árvore, isso é, um grafo conexo acíclico com um nó especial, denominado raiz (*root*), na qual se encontra a informação mais geral sobre uma rede, como visto na Figura 5.2 [39].

Os objetos dentro de uma MIB SNMP e a sua estrutura interna são definidos usando a ASN.1 (*Abstract Syntax Notation One*). Um identificador de objeto é um identificador único, constituído de uma sequência de inteiros com subidentificadores. A sequência lida da esquerda para a direita define a localização do objeto na árvore da MIB, por exemplo, observando a Figura 5.2, o identificador para o objeto *enterprises* é 1.3.6.1.4.1 [104].

Assim, associado a cada tipo de objeto numa MIB, tem-se um identificador do tipo *OBJECT IDENTIFIER* da sintaxe ASN.1. O identificador serve para nomear o objeto. Como o valor associado com o tipo *OBJECT IDENTIFIER* é hierárquico, a convenção de nomes também serve para identificar a estrutura de tipos de objetos [104].

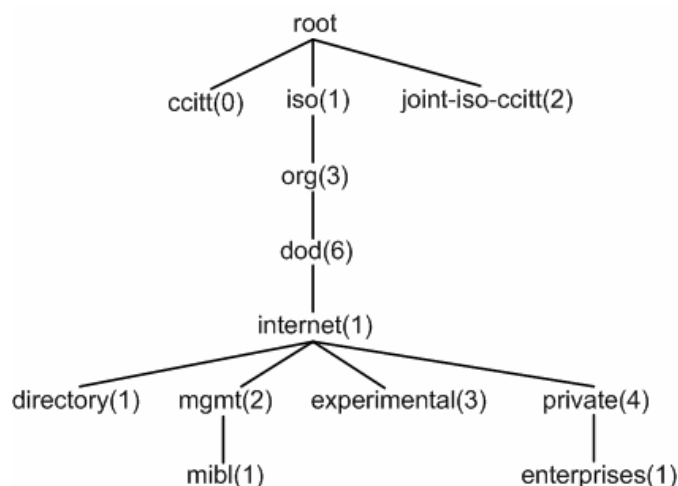


Figura 5.2 – Ilustração de Parte da Árvore da MIB.

Ao analisar a Figura 5.2, observa-se que mais detalhes de uma área específica da árvore são obtidos ao se caminhar pelos seus ramos até atingir as folhas, ou seja, as variáveis de fato, usualmente denominadas objetos. Observa-se ainda, que cada ponto de ramo da árvore possui um nome e um número, entre parênteses, o que permite que os objetos possam ser identificados à partir da raiz pela sequência de nomes ou mesmo números que especificam o trajeto até o objeto, permitindo assim a leitura do objeto [30].

O SNMP apresenta as cinco tipos de mensagens ou primitivas, *Get-Request*, *Get-Response*, *Get-Next-Request*, *Set-Request* e *Trap*, das quais apenas foram utilizadas as:

- *Get-Request*: gerada pelo servidor SNMP, que consulta ao agente, que lhe responde com uma mensagem *Get-Response*;
- *Get-Response*: mensagem gerada pelo agente e enviada ao servidor SNMP, como resposta à *Get-Request*.

Existem vários pacotes de *software* para gerenciamento de rede que fazem uso do protocolo SNMP, mais precisamente de suas primitivas. Dentre estes pacotes, tem-se o Net-SNMP, utilizado para a validação da proposta, que possui alguns comandos que podem ser usados no *shell (prompt)* de sistemas operacionais do tipo Unix. Seus principais comandos são: *snmpget*, *snmpgetnext*, *snmpwalk*, *snmptable*, *snmpdelta*, *snmpset*, *snmpdf*, *snmpnetstat*, *snmpstatus* e *snmptranslate* [24].

A fim de obter os percentuais disponíveis de carga nos *Media Gateways* componentes do ambiente VoIP, monitoraram-se os objetos, ou seja, os recursos computacionais processador e memória, para, a partir deles, calcular o VCB (Valor Calculado para o Balanceamento), a fim de prover eficaz e fidedigno mecanismo de balanceamento de chamadas VoIP a transcodificar. O Cálculo do VCB é apresentado na seção 5.2.1.

Ressalta-se, que o impacto das chamadas de sistema do SNMP não é significativo no tocante ao *overhead*, pois para cada *Media Gateway* monitorado serão disparadas duas chamadas de sistema do tipo *Get-Request*, e por conseguinte respondidas com outras duas do tipo *Get-Response*, ambas correspondentes aos objetos

monitorados, que neste caso são processador e memória. Já quanto ao tamanho do pacote SNMP, é utilizado o padrão, que é de 1500 bytes. Ainda, tais chamadas não são realizadas sistematicamente, portanto, o *overhead* pertinente ao número de chamadas de sistema do SNMP realizadas não impacta na rede, e consequentemente no balanceamento de chamadas VoIP a transcodificar [98] [99] [103]. A seguir são tratados os objetos monitorados.

5.1.1 Processador

Para o monitoramento do percentual de carga disponível no processador, usa-se o objeto `ssCpuIdle(11)`, cujo caminho é `1.3.6.1.4.1.2021.11`. A Figura 5.3 apresenta o objeto `ssCpuIdle(11)`, destacado no ramo *systemStats* da árvore da MIB, obtido pelo comando `snmptranslate -Tp -OS 1.3.6.1.4.1.2021.11`.

```
root@gasterisk1:~# snmptranslate -Tp -OS 1.3.6.1.4.1.2021.11
+--systemStats(11)
|
+-- -R-- Integer32 ssIndex(1)
+-- -R-- String    ssErrorName(2)
|
|   Textual Convention: DisplayString
|   Size: 0..255
+-- -R-- Integer32 ssSwapIn(3)
+-- -R-- Integer32 ssSwapOut(4)
+-- -R-- Integer32 ssIOSent(5)
+-- -R-- Integer32 ssIOReceive(6)
+-- -R-- Integer32 ssSysInterrupts(7)
+-- -R-- Integer32 ssSysContext(8)
+-- -R-- Integer32 ssCpuUser(9)
+-- -R-- Integer32 ssCpuSystem(10)
+-- -R-- Integer32 ssCpuIdle(11) ←
+-- -R-- Counter   ssCpuRawUser(50)
+-- -R-- Counter   ssCpuRawNice(51)
+-- -R-- Counter   ssCpuRawSystem(52)
+-- -R-- Counter   ssCpuRawIdle(53)
+-- -R-- Counter   ssCpuRawWait(54)
+-- -R-- Counter   ssCpuRawKernel(55)
+-- -R-- Counter   ssCpuRawInterrupt(56)
+-- -R-- Counter   ssIORawSent(57)
+-- -R-- Counter   ssIORawReceived(58)
+-- -R-- Counter   ssRawInterrupts(59)
+-- -R-- Counter   ssRawContexts(60)
+-- -R-- Counter   ssCpuRawSoftIRQ(61)
+-- -R-- Counter   ssRawSwapIn(62)
+-- -R-- Counter   ssRawSwapOut(63)
```

Figura 5.3 – Ramo *systemStats* da Árvore da MIB.

A Figura 5.3 apresenta o objeto `ssCpuIdle(11)`, destacado no ramo *systemStats* da árvore da MIB, obtido pelo comando `snmptranslate -Tp -OS 1.3.6.1.4.1.2021.11`, cujo caminho pode ser ilustrado pelo comando `snmptranslate`, apresentado na Figura 5.4.

```
root@asterisk1:~# snmptranslate -Of 1.3.6.1.4.1.2021.11.11
.iso.org.dod.internet.private.enterprises.ucdavis.systemStats.ssCpuIdle
```

Figura 5.4 – Caminho do Objeto `ssCpuIdle(11)` na Árvore da MIB.

A Figura 5.4 apresenta o caminho do objeto monitorado, `ssCpuIdle(11)`, na árvore da MIB, obtido pelo comando `snmptranslate -Of 1.3.6.1.4.1.2021.11.11`.

Já a obtenção do percentual disponível de processador é possível por meio da primitiva *Get-Request* ao objeto monitorado, por meio do comando `snmpget`, como ilustrado pela Figura 5.5.

```
root@asterisk1:~# snmpget -v 2c -c public 200.131.189.85 1.3.6.1.4.1.2021.11.11.0
| cut -d " " -f 4
99
```

Figura 5.5 – Obtenção do Percentual de Carga Disponível no Processador.

Constata-se na Figura 5.5, referente a primitiva *Get-Request*, que o *Media Gateway* 1 “200.131.189.85” monitorado, retornou o valor 99, o que lhe corresponde percentual de 99% de disponibilidade.

5.1.2 Memória

Para o monitoramento do percentual disponível de memória, usa-se o objeto `memAvailReal(6)`, cujo caminho é 1.3.6.1.4.1.2021.4.6. A Figura 5.6 apresenta o objeto `memAvailReal(6)`, destacado no ramo *memory* da árvore da MIB, obtido pelo comando `snmptranslate -Tp -OS 1.3.6.1.4.1.2021.4.6`.

```

root@asterisk1:~# snmptranslate -Tp -OS 1.3.6.1.4.1.2021.4
+--memory(4)
|
+-- -R-- Integer32 memIndex(1)
+-- -R-- String      memErrorName(2)
|      Textual Convention: DisplayString
|      Size: 0..255
+-- -R-- Integer32 memTotalSwap(3)
+-- -R-- Integer32 memAvailSwap(4)
+-- -R-- Integer32 memTotalReal(5)
+-- -R-- Integer32 memAvailReal(6) ←
+-- -R-- Integer32 memTotalSwapTXT(7)
+-- -R-- Integer32 memAvailSwapTXT(8)
+-- -R-- Integer32 memTotalRealTXT(9)
+-- -R-- Integer32 memAvailRealTXT(10)
+-- -R-- Integer32 memTotalFree(11)
+-- -R-- Integer32 memMinimumSwap(12)
+-- -R-- Integer32 memShared(13)
+-- -R-- Integer32 memBuffer(14)
+-- -R-- Integer32 memCached(15)
+-- -R-- Integer32 memUsedSwapTXT(16)
+-- -R-- Integer32 memUsedRealTXT(17)
+-- -R-- EnumVal    memSwapError(100)
|      Textual Convention: UCDErrorFlag
|      Values: noError(0), error(1)
+-- -R-- String      memSwapErrorMsg(101)
|      Textual Convention: DisplayString
|      Size: 0..255

```

Figura 5.6 – Ramo *memory* da Árvore da MIB.

A Figura 5.6 ilustra o ramo *memory* da árvore da MIB, obtido pelo comando *snmptranslate -Tp -OS 1.3.6.1.4.1.2021.4.6*, em que consta o objeto monitorado, *memAvailReal(6)*, nela destacado, cujo caminho pode ser ilustrado pelo comando *snmptranslate*, apresentado na Figura 5.7.

```

root@asterisk1:~# snmptranslate -Of 1.3.6.1.4.1.2021.4.6
.iso.org.dod.internet.private.enterprises.ucdavis.memory.memAvailReal

```

Figura 5.7 – Caminho do Objeto *memAvailReal(6)* na Árvore da MIB.

A Figura 5.7 apresenta o caminho do objeto monitorado, *memAvailReal(6)*, na árvore da MIB, obtido pelo comando *snmptranslate -Of 1.3.6.1.4.1.2021.4.6*.

A partir do monitoramento desse objeto, obtém-se o respectivo total de memória física disponível do *Media Gateway*, para que, então, o Sub-módulo Atualizador, apresentado na seção 5.2, efetue regra de três, a fim de obter o respectivo percentual disponível de memória. A obtenção desse total é possível por meio da primitiva *Get-Request*, por meio do comando *snmpget*, como ilustrado pela Figura 5.8.

```
root@asterisk1:~# snmpget -v 2c -c public 200.131.189.85 1.3.6.1.4.1.2021.4.6.0 | cut -d  
" " -f 4  
334044
```

Figura 5.8 – Obtenção do Total de Memória Física Disponível.

A Figura 5.8 representa o total de memória física disponível em *bytes*. De posse deste total, calcula-se no Sub-módulo Atualizador, o percentual de memória física disponível, o que habilita o Módulo Balanceador de Chamadas a fidedigno balanceamento, conforme objetivado. A seção seguinte aborda sua implementação.

5.2 O Módulo Balanceador de Chamadas

O sistema que trabalha com a VoIP e que proveja balanceamento de chamadas que necessitem de transcodificação, é composto por *Media Gateways*, os quais efetivamente realizam a transcodificação, bem como o Servidor de Chamadas. Assim, tanto o Servidor de Chamadas quanto os *Media Gateways* são disponibilizados na rede de dados, no caso, *Internet* ou *Intranets*.

A estrutura do Servidor de Chamadas é composta por dois grandes módulos: Módulo Servidor de Sessão e Módulo Balanceador de Chamadas. O Módulo Balanceador de Chamadas é composto por quatro Sub-módulos integrados, Verificador, Atualizador, Contador e Indicador de MG (*Media Gateway*), cuja breve descrição é apresentada a seguir:

- Sub-módulo Acionador: aciona o Sub-módulo Verificador;
- Sub-módulo Verificador: verifica quais os *Media Gateways* estão operantes e atualiza a Lista de Recursos de MGs, que é apresentada nesta seção;
- Sub-módulo Atualizador: consulta a Lista de Recursos de MGs e atualiza a Lista de Prioridades;
- Sub-módulo Indicador de MG: encaminha ao Módulo Servidor de Sessão o identificador do *Media Gateway*, da Lista de Prioridades, com menor VCB.

Para que haja o balanceamento de chamadas VoIP a transcodificar, o Módulo Balanceador de Chamadas atua baseado nos percentuais de recursos computacionais disponíveis nos *Media Gateways*, listando-os em ordem de prioridade para o

atendimento das chamadas VoIP a transcodificar. Equaliza sua distribuição entre os *Media Gateways* que compõem o ambiente VoIP no qual é implementado. O referido módulo provê o balanceamento de chamadas VoIP a transcodificar.

A Figura 5.9 apresenta o Módulo Balanceador de Chamadas em respectivo cenário de aplicabilidade.

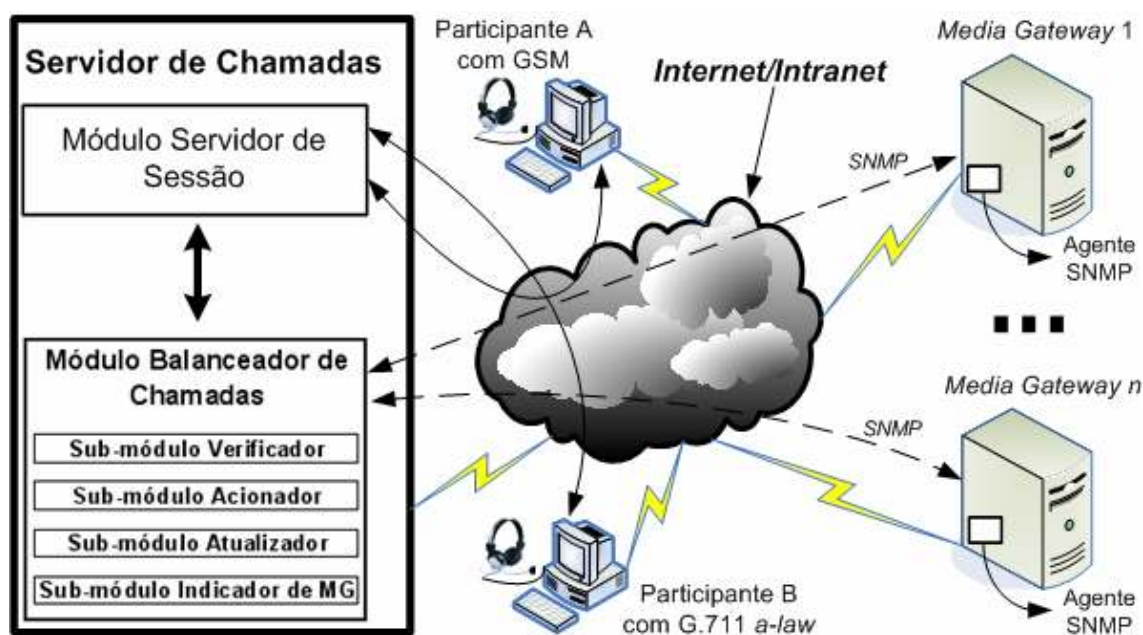


Figura 5.9 – Cenário de Aplicabilidade do Módulo Balanceador de Chamadas.

Na Figura 5.9 pode ser visto o Módulo Balanceador de Chamadas com seus respectivos Sub-módulos integrado aos demais componentes do cenário que possibilitam seu funcionamento. Desta forma, o Módulo Balanceador de Chamadas interage com o Módulo Servidor de Sessão, estando ambos localizados no Servidor de Chamadas. Dele constam, ainda, dois participantes, A e B, com distintos *codecs*, o que quando de uma chamada requer transcodificação, bem como *Media Gateways*, que a provê.

A pluralidade dos *Media Gateways* decorre da operacionalidade conferida ao Módulo Balanceador de Chamadas para o balanceamento em decorrência de proverem a transcodificação necessária.

Para que esses Sub-módulos atuem, faz-se necessário o uso de:

- Lista de MGs: contém a relação de todos os *Media Gateways*, cada qual com a respectiva identificação e endereço. Esta lista é mantida pelo administrador do sistema, a fim de inserir e ou excluir *Media Gateways* do ambiente VoIP. Desta forma, possibilita segurança, pois, pode ser alterada apenas pelo administrador, não permitindo com isso o encaminhamento de chamadas a *Media Gateways* não previamente cadastrados;
- Contador: é usado para estabelecer um número máximo de chamadas encaminhadas ao *Media Gateway* de menor VCB, obtido da Lista de Prioridades, isto enquanto não haja uma atualização desta lista pelo Sub-módulo Atualizador. Uma vez acionado o Sub-módulo Atualizador, o contador é zerado;
- Temporizador: é o responsável por acionar o Sub-módulo Acionador periodicamente;
- Lista de Recursos de MGs: contém os dados referentes aos *Media Gateways* que se encontram operacionais, contendo suas identificações e recursos pertinentes ao balanceamento de chamadas VoIP a transcodificar. Esta lista é atualizada pelo Sub-módulo Verificador e consultada pelo Sub-módulo Atualizador;
- Lista de Prioridades: contém a identificação dos *Media Gateways* ativos, ou seja, aptos a receberem chamadas, com as informações em ordem crescente de VCB, normalizados na escala de 1 a 100. As referentes informações são atualizadas pelo Sub-módulo Atualizador.

Constitui-se o *Media Gateway* ativo aquele que dispõe de um VCB maior que o mínimo necessário para que não degrade a comunicação. Assim, o limiar de VCB máximo aceitável é aquele que não degrada a comunicação VoIP. Portanto, deve haver um limiar máximo aceitável de VCB, conforme calculado a partir da equação (1), apresentada na seção 5.2.1.

Com isso, a abstração do modelo de balanceamento de chamadas remete à observação das funcionalidades agrupadas nos Sub-módulos componentes do Módulo Balanceador de Chamadas:

- Sub-módulo Acionador: é o responsável por acionar o Sub-módulo Verificador, o que ocorre em períodos pré-definidos no Temporizador, ou quando o Sub-módulo Indicador de MG o aciona. A Figura 5.10 o ilustra.

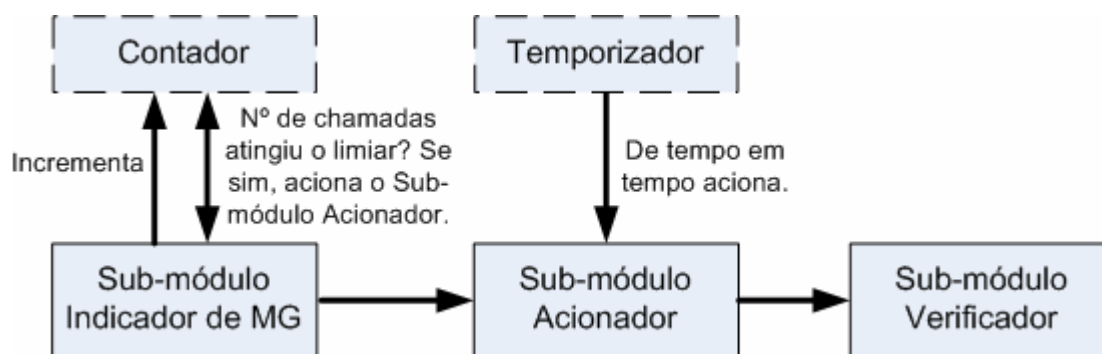


Figura 5.10 – Ilustração do Sub-módulo Acionador.

O mecanismo de funcionamento, ilustrado na Figura 5.10, demonstra que o Sub-módulo Acionador funciona periodicamente, devido ao Temporizador, ou quando o contador atingir o seu limiar, o que ocasiona que o Sub-módulo Indicador de MG o acione e, esse, por conseguinte, ao Sub-módulo Verificador.

- Sub-módulo Verificador: assim que acionado, inicia sua função consultando a Lista de MGs e após listá-los verifica se os mesmos estão operacionais, usando para tanto as chamadas de sistema do SNMP. Satisfeito tal requisito, o mesmo os disponibiliza na Lista de Recursos de MGs, acompanhados, neste caso, das respectivas informações de recursos disponíveis e, aciona o Sub-módulo Atualizador. A Figura 5.11 o ilustra.

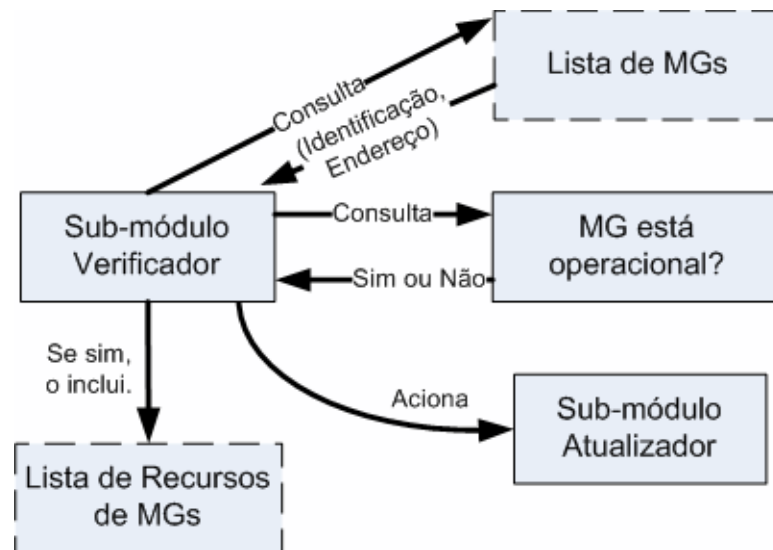


Figura 5.11 – Ilustração do Sub-módulo Verificador.

A Figura 5.11 mostra que o mecanismo de funcionamento do Sub-módulo Verificador que consiste em consultar a Lista de MGs para aferir os *Media Gateways* que estão operacionais e atualizar a Lista de Recursos de MGs. Feito isto, este Sub-módulo aciona o Sub-módulo Atualizador.

- Sub-módulo Atualizador: é responsável por consultar as informações necessárias na Lista de Recursos de MGs, para o cálculo do VCB. Assim, a Lista de Prioridades, composta pelo identificador e o valor do VCB, ambos de um respectivo *Media Gateway*, é atualizada com o valor calculado. A Lista de Prioridades é ordenada crescentemente em relação ao valor do VCB dos *Media Gateways*. A Figura 5.12 o ilustra.

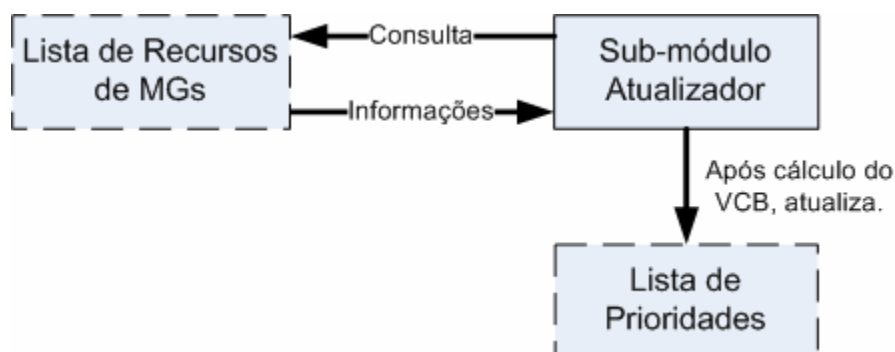


Figura 5.12 – Ilustração do Sub-módulo Atualizador.

A Figura 5.12 ilustra que o funcionamento do Sub-módulo Atualizador consiste essencialmente em calcular o VCB e na atualização da Lista de Prioridades.

- Sub-módulo Indicador de MG: após inquirido pelo Módulo Servidor de Sessão, consulta a Lista de Prioridades e encaminha o identificador do *Media Gateway* apto a atender à chamada a ser transcodificada. Em seguida, incrementa o Contador e, caso o número de chamadas tenha ultrapassado um limiar pré-definido, o mesmo aciona o Sub-módulo Acionador, quando, então, o Contador é reiniciado. A Figura 5.13 o ilustra.

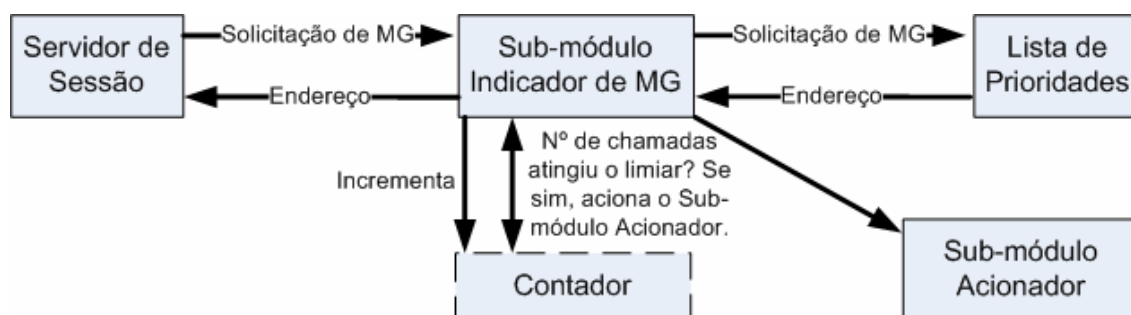


Figura 5.13 – Ilustração do Sub-módulo Indicador de MG.

A Figura 5.13 do Sub-módulo Indicador de MG demonstra que o mesmo retorna ao Módulo Servidor de Sessão o endereço do *Media Gateway* melhor classificado na Lista de Prioridades e, após isso, incrementa o Contador e aciona o Sub-módulo Acionador, possibilitando a continuidade do processo de funcionamento do Módulo Balanceador de Chamadas.

Didaticamente a Figura 5.14 esquematiza o funcionamento do Módulo Balanceador de Chamadas, pela integração dos Sub-módulos que o compõe.

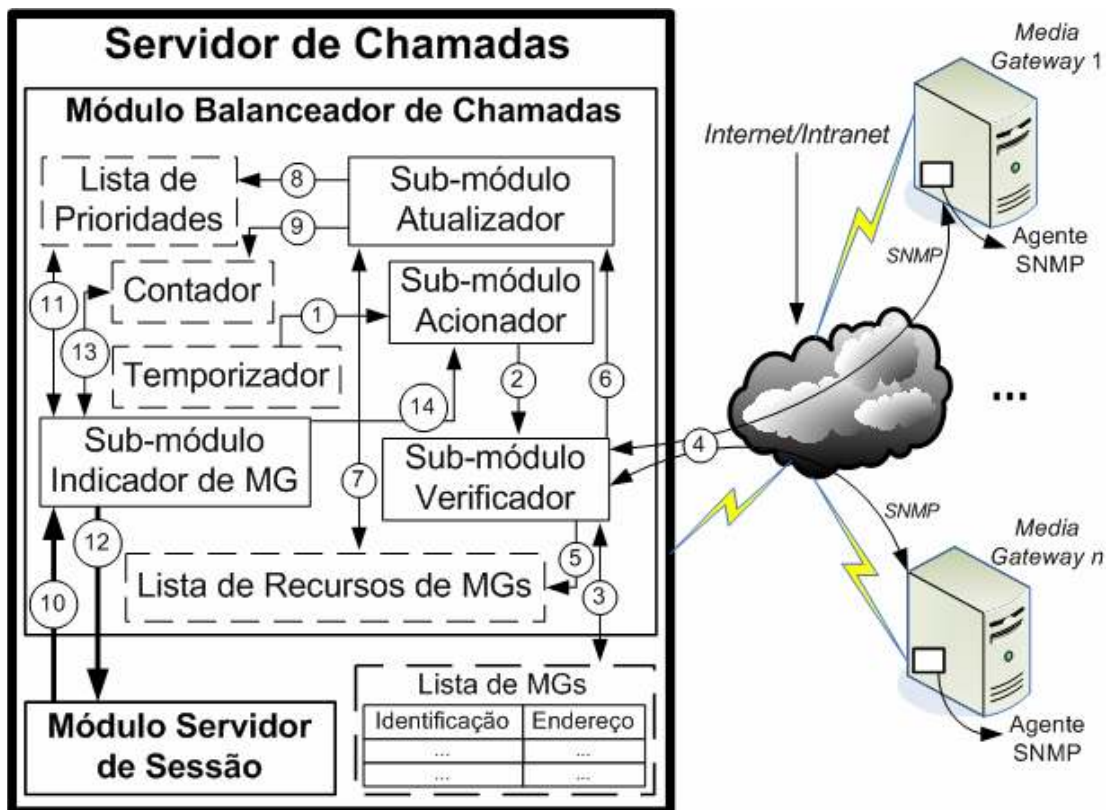


Figura 5.14 – Esquema de Funcionamento do Módulo Balanceador de Chamadas.

O Módulo Balanceador de Chamadas atua de forma equânime de acordo com os parâmetros monitorados, permitindo assim um balanceamento de chamadas adequado [39] [59].

As funcionalidades referentes à composição do Módulo Balanceador de Chamadas descrevem por si só um algoritmo de balanceamento de chamadas cuja interação dos Sub-módulos componentes, ilustrada na Figura 5.14, ocorre da seguinte maneira:

- 1 – O Temporizador periodicamente aciona o Sub-módulo Acionador;
- 2 – O Sub-módulo Acionador aciona o Sub-módulo Verificador;
- 3 – o Sub-módulo Verificador consulta na Lista de MGs os *Media Gateways* que compõem o ambiente VoIP;

4 – feita a consulta, o Sub-módulo Verificador checa quais dos *Media Gateways* entre os consultados se encontram operacionais e, dos que estão obtêm as respectivas informações dos recursos monitorados, o que é possível por meio das chamadas de sistema do SNMP;

5 – assim que são checados quais os *Media Gateways* operacionais, o Sub-módulo Verificador os disponibiliza na Lista de Recursos de MGs, juntamente com as informações dos recursos monitorados;

6 – em seguida, o Sub-módulo Verificador aciona o Sub-módulo Atualizador;

7 – o Sub-módulo Atualizador quando acionado, consulta a Lista de Recursos de MGs, buscando todas as informações necessárias para proceder ao cálculo do VCB;

8 – após a realização do cálculo do VCB pelo Sub-módulo Atualizador, o VCB juntamente ao identificador do *Media Gateway*, são então registrados na Lista de Prioridades e ordenados em forma crescente de VCB;

9 – montada a Lista de Prioridades o Sub-módulo Atualizador zera o Contador;

10 – quando uma chamada que necessita de transcodificação chega ao Módulo Servidor de Sessão, este faz uma requisição ao Módulo Balanceador de Chamadas por meio do Sub-módulo Indicador de MG;

11 – o Sub-módulo Indicador de MG por sua vez busca na Lista de Prioridades o *Media Gateway* que disponha do maior percentual de recursos computacionais disponíveis, indicado pelo menor VCB;

12 – o Sub-módulo Indicador de MG, de posse do identificador do *Media Gateway* obtido, o encaminha ao Servidor Sessão, que o repassa aos usuários da chamada, possibilitando que estes se comuniquem;

13 – em seguida, o Sub-módulo Indicador de MG incrementa o Contador e verifica se o número de chamadas transcodificadas atingiu o limiar pré-definido;

14 – logo após, caso o limiar tenha sido atingido, o Sub-módulo Indicador de MG aciona o Sub-módulo Acionador.

A partir de então se reinicia o ciclo e o processo tem continuidade enquanto o Módulo Balanceador de Chamadas permanecer ativo.

5.2.1 Cálculo do VCB

Para que o funcionamento do Módulo Balanceador de Chamadas se efetive de modo eficaz, faz-se necessária a aferição dos recursos disponíveis nos *Media Gateways*, o que requer uma métrica pertinente que lhe permita desempenhar sua função de distribuição de maneira equitativa as chamadas VoIP a transcodificar [59].

Para tanto, a fim de satisfazer aos requisitos de fidedignidade, utilizou-se o cálculo do VCB (Valor Calculado para o Balanceamento), que possibilita a requisitada eficácia, por normalizar os percentuais de recursos disponíveis monitorados via SNMP em todos os *Media Gateways*. O respectivo cálculo do VCB é realizado pelo Sub-módulo Atualizador. Assim, desencadeou-se a equação:

$$VCB = 100 - \left(\frac{\sum_{i=1}^n Recurso_i * Peso_i}{\sum_{i=1}^n Peso_i} \right) \quad (1)$$

A expressão $\left(\frac{\sum_{i=1}^n Recurso_i * Peso_i}{\sum_{i=1}^n Peso_i} \right)$ representa o percentual *disponível* obtido para cada *Media Gateway* utilizado, em que há n recursos a serem monitorados, apresentando cada qual pesos pré-definidos. Na fórmula 1, o termo 100 é indicativo da percentagem.

Tal fórmula apresenta caráter genérico para o cálculo do VCB, pelo que n assume valor correspondente a quantidade de recursos monitorados. Para a validação da

proposta, como apresentado no Capítulo 6, $n = 2$, pois são considerados apenas dois recursos: processador e memória.

Para a equitativa alocação de *Media Gateways* para o atendimento das chamadas VoIP a transcodificar, deve-se considerar as respectivas informações dos recursos de processador e memória disponibilizados pelos mesmos. Isso incorre na imputação de respectivos pesos aos mesmos, aqui convencionados cinco para processador e um para memória, em conformidade com o resultado de testes “*benchmarks*” em que se constata a acentuada discrepância gráfica entre os respectivos percentuais de recursos no decorrer da transcodificação de chamadas VoIP simultâneas [46] [47] [77].

A fórmula 1 faculta ainda os respectivos cálculos dos VCBs para múltiplos *Media Gateways* que componham o cenário de aplicabilidade do Módulo Balanceador de Chamadas, sem a necessidade de alterações ou possíveis desdobramentos. Assim, de posse do VCB, o Sub-módulo Atualizador entra em ação. Com isso, tem-se um eficaz mecanismo para o balanceamento de chamadas VoIP a transcodificar, pela manutenção da Lista de Prioridades. A partir de então, o Módulo Balanceador de Chamadas provê o equitativo e efetivo balanceamento das chamadas e possibilita a integração entre diferentes equipamentos e ambientes VoIP, que, não possuindo os mesmos *codecs*, são transcodificadas, após o Módulo Balanceador de Chamadas indicar o *Media Gateway* com maior aptidão para realizá-las.

Em vista disso, quando se utiliza um Telefone IP ou mesmo ATA, não há que se preocupar com o *codec* com que o mesmo opere, visto que o Módulo Balanceador de Chamadas ocupa-se de disponibilizar um *Media Gateway* apto a atender chamadas que demandem transcodificação, caso a outra parte não disponha de *codec* correspondente.

Ainda, quando do surgimento de novas tecnologias, em especial no âmbito de *codecs*, apenas faz-se necessária a atualização dos *Media Gateways*, para a viabilização das mesmas, se for necessária a transcodificação [68].

5.3 Implementação do Módulo Balanceador de Chamadas

Esta seção apresenta o cenário em que a pesquisa foi desenvolvida, que se constitui de cinco máquinas, designadas: Participantes A e B, *Media Gateways* 1 e 2 e o Servidor de Chamadas, do qual o Módulo Servidor de Sessão e o Módulo Balanceador de Chamadas fizeram uso. Ainda, para o balanceamento de chamadas VoIP a transcodificar, monitorou-se os percentuais disponíveis de processador e memória.

As características das máquinas utilizadas são:

- Participantes A e B: constituíram usuários da VoIP, com os respectivos endereços IP, 200.131.189.70 e 200.131.189.89, ambos dotados dos *codecs* G.711 (*μ -law* e *a-law*) e GSM. Por ocasião da realização dos testes, para cada chamada, habilitava-se cada qual por vez, enquanto os demais permaneciam desabilitados. Ainda, para utilizarem a VoIP, aos participantes foi disponibilizado o *softphone X-Lite* versão 3.0, recomendável para máquinas com a configuração mínima de processador Intel Pentium III 700 MHz (*Megahertz*) ou equivalente e, 256 MB (*Megabytes*) de memória RAM [56];
- *Media Gateways* (1 e 2): máquina oportunizadora de interconexão entre diferentes tipos de mídia e, em especial à transcodificação, cujo *software* possibilitador de tais funcionalidades foi o Asterisk PBX versão 1.4.19.1, utilizado nos testes comprobatórios da eficácia do Módulo Balanceador de Chamadas, os quais são apresentados no Capítulo 6. Suas especificações são dadas a seguir [41].
 - *Media Gateway* 1: denominado *asterisk1*, com endereço IP 200.131.189.85, dotado de processador Intel(R) Celeron(R) CPU 2.00 GHz, 512 MB de memória, Sistema Operacional *Slackware Linux* 12.1 e Net-SNMP 5.4.1 [24] [60];
 - *Media Gateway* 2: denominado *asterisk2*, com endereço IP 200.131.189.86, dotado de processador Pentium III (*Coppermine*) de 900

MHz, 384 MB de memória, Sistema Operacional *Slackware* Linux 12.1 e Net-SNMP 5.4.1 [24] [60].

- Servidor de Chamadas: máquina à qual se integram o Módulo Balanceador de Chamadas e o Módulo Servidor de Sessão, dotada de processador Intel(R) Pentium(R) 4 CPU 3.40 GHz *dual core*, 1 GB de memória, Sistema Operacional *Slackware* Linux 12.1 e *software* OpenSER 1.3.2-*notls* [45] [60].

A Figura 5.15 apresenta o cenário de implementação do Módulo Balanceador de Chamadas.

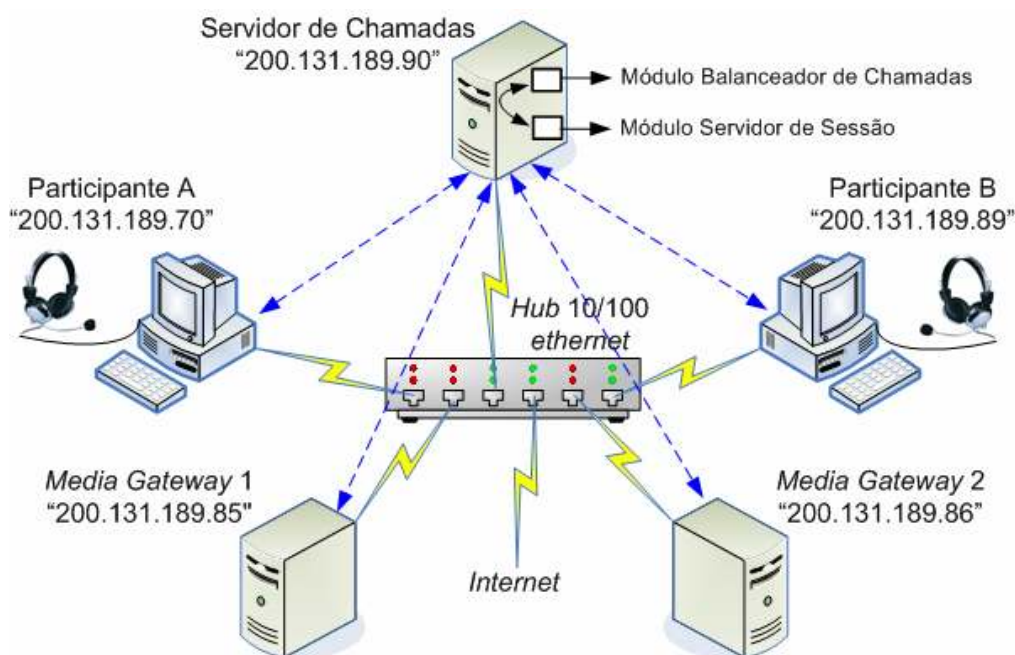


Figura 5.15 – Cenário de Implementação do Módulo Balanceador de Chamadas.

Cada máquina ilustrada na Figura 5.15 do cenário foi configurada a fim de prover a viabilização do Módulo Balanceador de Chamadas, como solução buscada ao balanceamento de chamadas VoIP a transcodificar. As seções seguintes tratam das configurações que se fizeram necessárias às mesmas.

5.3.1 Participantes A e B

Nessas máquinas, dotadas do Sistema Operacional Windows XP, utilizou-se o *softphone X-Lite*, que é *freeware*, de fácil configuração, que é feita de forma interativa por meio visual, como pode ser visto na Figura 5.16.

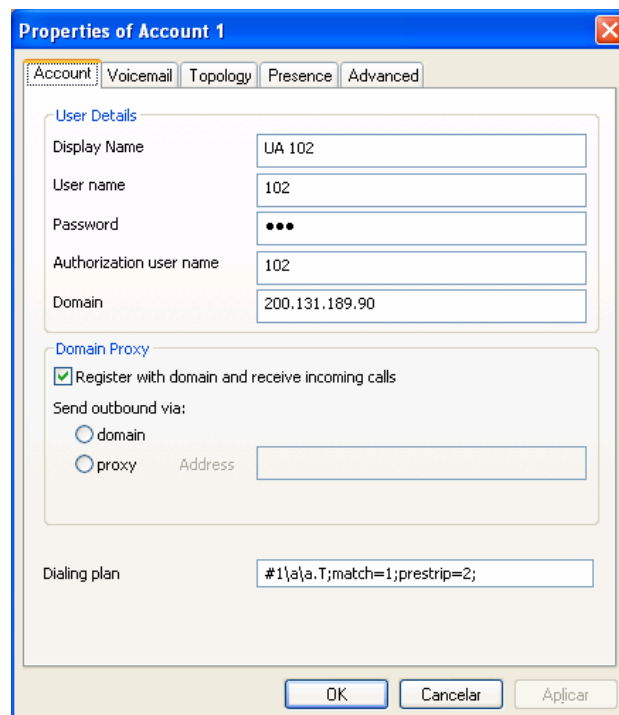


Figura 5.16 – Tela de Configuração de Conta do *X-Lite*.

As principais informações configuráveis no *X-Lite* são descritas a seguir, de acordo com a Figura 5.16:

- *Display Name* – nome que aparecerá para a parte chamada;
- *User name* – nome ou número do usuário;
- *Password* – senha;
- *Authorization user name* – usualmente o mesmo conteúdo do campo *User name*;
- *Domain* – endereço IP do Servidor de Chamadas no qual o Módulo Servidor de Sessão está contido.

Ressalta-se que o *X-Lite* dispõe de versões para Mac, Windows e Linux [56].

5.3.2 Servidor de Chamadas

A viabilização da solução buscada, possibilitadora do balanceamento de chamadas VoIP a transcodificar não inclui apenas o desenvolvimento do Módulo Balanceador de Chamadas, como ainda a adaptação do mesmo ao ambiente VoIP. Para tanto, o mesmo é agregado ao Servidor de Chamadas e, juntamente com o Módulo Servidor de Sessão, atuam de forma a prover o objetivado. A Figura 5.17 ilustra o Servidor de Chamadas.

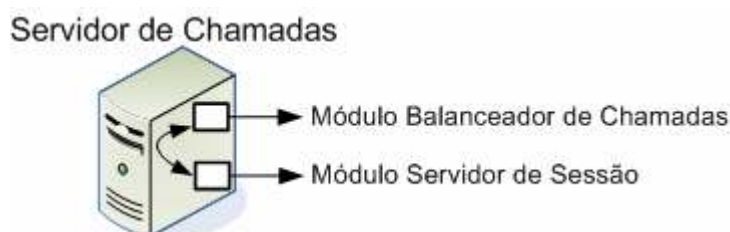


Figura 5.17 – Servidor de Chamadas.

Pela análise da Figura 5.17, depreende-se que no Servidor de Chamadas estão contidos o Módulo Balanceador de Chamadas e o Módulo Servidor de Sessão. A seguir descreve-se o Módulo Servidor de Sessão.

5.3.2.1 Módulo Servidor de Sessão

O Módulo Servidor de Sessão para validação da proposta, é composto pelo *software* OpenSER 1.3.2-notls, que é de código aberto e passível de customização e, graças a isso, pode ter parte do código fonte do arquivo *openser.cfg* adequada para tratar o caso de chamadas VoIP a transcodificar, ou seja, além de acionar o Módulo Balanceador de Chamadas, contém os dados para registro dos participantes nos *Media Gateways* que compõe o ambiente VoIP [45]. É ilustrado na Figura 5.18.

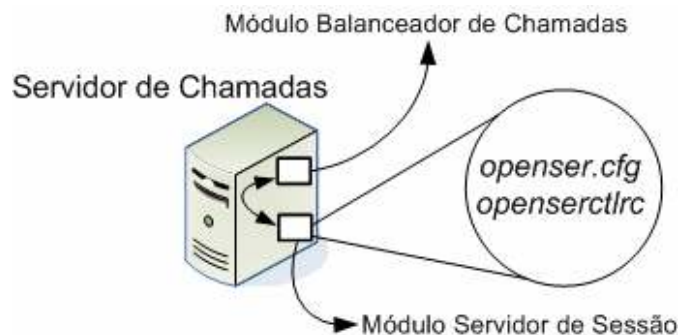


Figura 5.18 – Módulo Servidor de Sessão.

A Figura 5.18 ilustra o Módulo Servidor de Sessão e seus respectivos arquivos utilizados, *openserctrlrc* e *openser.cfg*, dos quais os respectivos códigos fontes se encontram na íntegra nos Anexos A.1 e A.2 respectivamente.

Para o funcionamento do Módulo Balanceador de Chamadas, é imprescindível adaptar no arquivo de configuração do OpenSER, *openser.cfg*, a adição de uma regra de tratamento de erro, que no caso constitui-se na mensagem SIP 488 (*Not Acceptable Here*), inerente à ocorrência de incompatibilidade de *codecs*. Logo, ao se tentar estabelecer uma chamada e ela não ser possível, devido ao fato de os *codecs* não serem compatíveis entre si, o que caracteriza o erro 488, a referida mensagem é retornada do participante destino ao Módulo Servidor de Sessão, que por dispor do Módulo Balanceador de Chamadas a trata. Assim, interpreta-a como requisição de nova chamada, que será estabelecida via *Media Gateway*.

Sendo o Módulo Balanceador de Chamadas acionado, o Módulo Servidor de Sessão, de posse de um *Media Gateway*, repassa-lhe a requisição de chamada em curso, a fim de que ela possa ser completada. Ressalta-se que, para o participante, nada muda, pois tal processo lhe é transparente, como ilustrado na Figura 5.19.

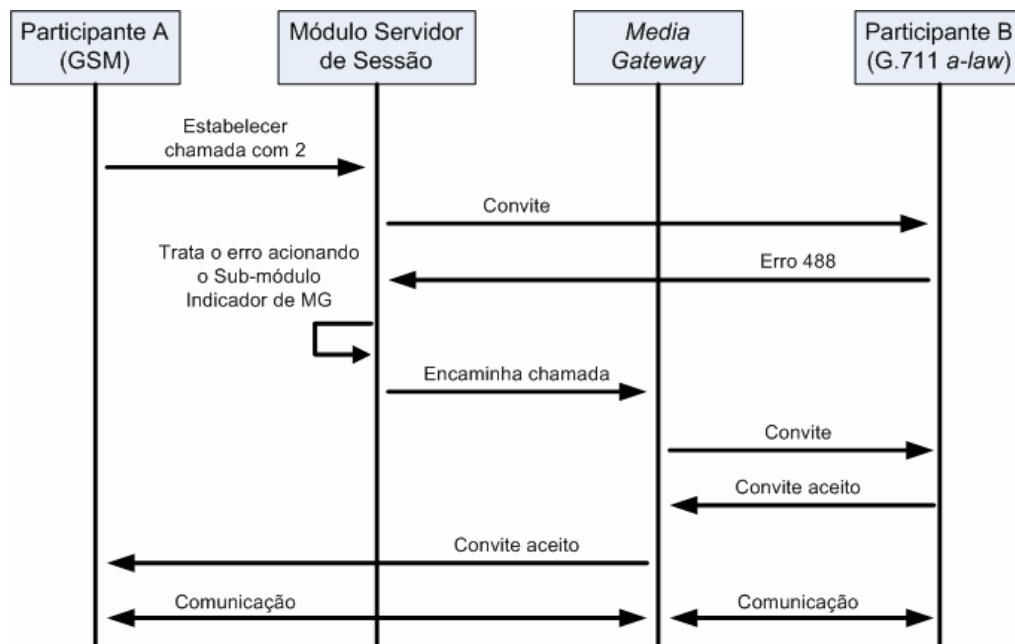


Figura 5.19 – Ilustração do Estabelecimento de Chamada.

Como ilustrado pela Figura 5.19, para que uma chamada entre os Participantes A e B, que é iniciada por A seja estabelecida, inicialmente o Módulo Servidor de Sessão é consultado quanto ao registro “IP” do Participante B. Se nele localizado, possibilita-lhe que a chamada o atinja e seja detectada a compatibilidade, ou não, entre os *codecs* de ambos. Se assim for, a chamada é estabelecida, caso contrário, é retornado ao Módulo Servidor de Sessão um erro 488, inerente à incompatibilidade entre os *codecs*.

Esse fato desencadeia o acionamento do Sub-módulo Indicador de MG, que por sua vez consulta a Lista de Prioridades e informa ao Módulo Servidor de Sessão o IP do *Media Gateway* apto a efetivar a chamada. O Módulo Servidor de Sessão a redireciona então a chamada ao respectivo *Media Gateway*, que por meio da transcodificação, possibilita-a aos participantes.

Ressalte-se ser requisito indispensável para iniciar uma chamada, que os participantes estejam registrados no Módulo Servidor de Sessão, pois, assim, também o serão em todos os *Media Gateways* componentes do ambiente VoIP, independentemente dos *codecs* que utilizem. A Figura 5.20 ilustra a parte do código fonte do arquivo *openser.cfg* que realiza tal função.

```

...
rewritehostport("200.131.189.85");
append_branch();

rewritehostport("200.131.189.86");
t_relay();
...

```

Figura 5.20 – Parte do Código Fonte para Registro dos Participantes nos *Media Gateways*.

A Figura 5.20 ilustra o código para replicação do registro dos participantes nos *Media Gateways* que compõe o ambiente VoIP. Já a Figura 5.21 consiste na parte do código fonte do arquivo *openser.cfg*, responsável por tratar o erro 488. Assim, quando é intentado o estabelecimento de uma chamada entre participantes com *codecs* distintos, é gerado o respectivo erro provindo do participante chamado, ou seja, destino e que, ao retornar a sua origem, é interceptado pelo Módulo Servidor de Sessão, que o trata.

```

...
if (t_check_status("488")) {

    exec_avp("/root/Desktop/modulobalanceador/Sub-móduloIndicadordeMG", "$avp(s:pub_ip)");
    pv_printf("$var(my_ip)", "$avp(s:pub_ip)");

    switch($var(my_ip))
    {
        case "200.131.189.86":
            log("vai para o Asterisk 2\n");
            rewritehostport("200.131.189.86:5060");
            append_branch();
            t_relay();
        break;
        case "200.131.189.85":
            log("vai para o Asterisk 1\n");
            rewritehostport("200.131.189.85:5060");
            append_branch();
            t_relay();
        break;
        default:
            log("-- Nada foi executado no switch --\n");
    }
}
...

```

Figura 5.21 – Parte do Código Fonte para Acionar o Módulo Balanceador de Chamadas.

Como ilustrado na Figura 5.21, o tratamento de erro, realizado no Módulo Servidor de Sessão, refere-se a identificar se o erro é o 488, para, assim, acionar o Submódulo Indicador de MG do Módulo Balanceador de Chamadas, a fim de que disponibilize um *Media Gateway* apto a atender a chamada. Então, o referido Sub-

módulo encaminha ao Módulo Servidor de Sessão, o IP do *Media Gateway* e, a chamada pode, então, ser estabelecida. Isso ocorre de forma transparente ao usuário, sem que tome conhecimento do que houve.

São elencados a seguir o principais comandos do OpenSER:

- *openserctl start* – inicia o serviço;
- *openserctl stop* – para o serviço;
- *openserctl restart* – reinicia o serviço.

A seguir é tratada a implementação do Módulo Balanceador de Chamadas.

5.3.2.2 Implementação do Módulo Balanceador de Chamadas

Um exemplo de possível implementação do Módulo Balanceador de Chamadas deu-se pela utilização da linguagem *Shell Script*, ou seja, comandos *shell* em arquivo texto, cujos códigos fontes utilizados nos respectivos Sub-módulos são a seguir apresentados [57].

5.3.2.2.1 Sub-módulo Acionador

O Sub-módulo Acionador é o responsável por acionar o Sub-módulo Verificador em períodos pré-definidos, ou quando acionado pelo Sub-módulo Indicador de MG. A Figura 5.22 ilustra o seu respectivo código fonte.

```
#!/bin/bash
/root/Desktop/modulobalanceador/Sub-móduloVerificador
```

Figura 5.22 – Código Fonte do Sub-módulo Acionador.

Pelo ilustrado na Figura 5.22 explicita-se a interação entre os Sub-módulos Acionador e Verificador, ou seja, o primeiro aciona o segundo. Já para que o mesmo fosse executado periodicamente, com pré-definição, adicionou-se uma regra no *crontab* do sistema operacional do Servidor de Chamadas, Linux no caso, em que a cada minuto o Sub-módulo Acionador é acionado para realizar sua função.

5.3.2.2.2 Sub-módulo Verificador

Assim que acionado, o Sub-módulo Verificador inicia sua função consultando a Lista de MGs e, após listar os *Media Gateways* juntamente com as informações de *clock* do processador e a capacidade de memória, verifica se os mesmos estão operacionais, usando para tanto chamadas de sistema do SNMP. Se satisfeito tal requisito, o Sub-módulo Verificador disponibiliza os *Media Gateways* e suas respectivas informações na Lista de Recursos de MGs. A Figura 5.23 apresenta o código fonte do mesmo.

```
#!/bin/bash
# Ao que uma nova Lista de Recursos MGs será construída, é necessário apagar a anterior
rm /root/Desktop/modulobalanceador/ListadeRecursosdeMGs
while read -a MG
do

# Verifica se o Media Gateway está operacional, trazendo o % de proc. disponível
x1=`snmpget -v 2c -c public "$MG" 1.3.6.1.4.1.2021.11.11.0 | cut -d " " -f 4`

# Se operacional encaminha-o a Lista de Recursos de MGs, juntamente com as informações
# de clock e memória, bem como seus respectivos percentuais livres
if [ ${x1} -lt 101 ]; then

# traz o total de memória física disponível
x3=`snmpget -v 2c -c public "$MG" 1.3.6.1.4.1.2021.4.6.0 | cut -d " " -f 4`

# traz o total de memória física
x2=$((MG[3]))

# calcula o percentual de memória disponível
PercMem=$((100*(x3/1024)/x2))

echo $MG ${MG[2]} ${MG[3]} ${x1} ${PercMem} >>
/root/Desktop/modulobalanceador/ListadeRecursosdeMGs
fi
done < /root/Desktop/modulobalanceador/ListadeMGs

# Após executar as rotinas acima, aciona o Sub-módulo Atualizador
/root/Desktop/modulobalanceador/Sub-móduloAtualizador
```

Figura 5.23 – Código Fonte do Sub-módulo Verificador.

Pela análise da Figura 5.23, constata-se que após a Lista de Recursos de MGs ser composta, o Sub-módulo Verificador aciona o Sub-módulo Atualizador. Também, observa-se que é consultado na Lista de MGs o total de memória física que o *Media Gateway* dispõe, isto se deve ao fato da informação não ser dinâmica, a exemplo do *clock* do processador, ou seja, caso fossem aferidos constantemente, não mudariam, no entanto, causariam um tráfego desnecessário na rede.

5.3.2.2.3 Sub-módulo Atualizador

Inicialmente no Sub-módulo Atualizador, são definidos os pesos dos respectivos recursos monitorados, bem como o limiar de VCB aceitável. Em seguida, a Lista de Recursos de MGs é consultada, retornando as informações pertinentes de cada um dos *Media Gateways* dela constantes. A seguir, o Sub-módulo Atualizador efetua o cálculo do VCB. A Figura 5.24 apresenta o código fonte do mesmo.

```
#!/bin/bash

# Especifica o peso para cada recurso monitorado
pesoProc=5 #peso atribuído ao processador
pesoMem=1 #peso atribuído a memória

# Especifica o limiar de VCB - Para efeito de experimento considera-se 95
limiardeVCB=95

#*****
# consulta a Lista de Recursos de MGs e obtém os respectivos valores:
# maior clock processador
maiorclock=`cat /root/Desktop/modulobalanceador/ListadeRecursosdeMGs | cut -d ' ' -f 2 |
sort -n -r | head -n 1`

# maior quantidade de memoria fisica
maiorqtdmemem=`cat /root/Desktop/modulobalanceador/ListadeRecursosdeMGs | cut -d ' ' -f 3
| sort -n -r | head -n 1`

while read -a MG
do
    # informa quantos MHz livre o processador do Media Gateway dispõe
    procMHz=$((MG[1]*MG[3]/100))
    # traz o total de memória física
    x2=$((MG[2]))
    # traz o percentual de memória física disponível
    x3=$((MG[4]))
    # informa quantos MB livre de memória o Media Gateway dispõe
    memMB=$((x3*x2/100))
    # antes do cálculo da média, normaliza-se os percentuais com base nos maiores
    # recursos disponíveis
    xProc=$((procMHz*100/maiorclock))
    xMem=$((memMB*100/maiorqtdmemem))

    # Cálculo do VCB
    VCB=$((100-(((xProc*pesoProc)+(xMem*pesoMem))/((pesoProc+pesoMem))))
    # Após o cálculo do VCB, conforme seu valor, é analisado se o MG pode
    # ou não integrar a Lista de Prioridades
    if [ ${VCB} -lt ${limiardeVCB} ]; then
        # Após o cálculo do VCB, este e o respectivo endereço IP do MG ao qual pertecem são
        # armazenados em um arquivo temporário
        echo $VCB $MG >> /root/Desktop/modulobalanceador/tmp
    fi
done < /root/Desktop/modulobalanceador/ListadeRecursosdeMGs
# É então realizada a ordenação com base no VCB, e criada a Lista de Prioridades
cat /root/Desktop/modulobalanceador/tmp | sort -n >
/root/Desktop/modulobalanceador/ListadePrioridades
rm /root/Desktop/modulobalanceador/tmp

# Montada a Lista de Prioridades, reinicia o Contador
echo 0 > /root/Desktop/modulobalanceador/Contador
```

Figura 5.24 – Código Fonte do Sub-módulo Atualizador.

Pela análise da Figura 5.24, para o cálculo do VCB, faz-se necessária a normalização dos percentuais dos recursos monitorados. Pois, percentuais de *Media Gateways* com configurações de *hardware* distintas, somente se equivalem, se normalizados. Para tanto, utiliza-se a maior quantidade de recurso para a obtenção das respectivas proporcionalidades. Deste modo, equalizam-se os percentuais de recursos disponíveis e evitam-se discrepâncias, provendo fidedignidade à funcionalidade, em suma, do Módulo Balanceador de Chamadas.

Para cada *Media Gateway* disponível, é calculado o VCB que, conjuntamente com o respectivo identificador, é armazenado temporariamente em um arquivo. Ao término do cálculo de todos os VCBs, estes são inseridos na Lista de Prioridades juntamente com os respectivos identificadores, em ordem crescente de VCBs. Ressalta-se que a indexação do identificador do *Media Gateways* se deu pelo endereço.

5.3.2.2.4 Sub-módulo Indicador de MG

O Sub-módulo Indicador de MG é acionado pelo Módulo Servidor de Sessão, quando este necessita de um *Media Gateway* que possa atender a uma chamada VoIP a transcodificar. A partir disso, ele consulta a Lista de Prioridades e retorna ao Módulo Servidor de Sessão o endereço IP do *Media Gateway* mais apto a atender à chamada. A Figura 5.25 apresenta o código fonte do mesmo.

```
#!/bin/bash
# Especifica o limiar de chamadas
# Para efeito de experimento considera-se 3
limiar=3

# Este é o comando que retorna ao Módulo Servidor de Sessão o MG com o maior
# percentual de recursos disponíveis, o qual está disposto na Lista de Prioridades
head -n 1 /root/Desktop/modulobalanceador/ListadePrioridades | cut -d " " -f 2

# Após o Sub-módulo Indicador de MG encaminhar ao Módulo Servidor de Sessão o IP do MG
# com o maior percentual de recursos disponíveis, o Sub-módulo lê o Contador
i=`head /root/Desktop/modulobalanceador/Contador`
i=`expr $i + 1`

# Caso o número de chamadas VoIP ultrapasse o limiar, aciona o Sub-módulo Acionador
# caso contrário, incrementa o Contador
if [ ${i} -gt ${limiar} ]; then
    /root/Desktop/modulobalanceador/Sub-móduloAcionador &
else
    echo $i > /root/Desktop/modulobalanceador/Contador
fi
```

Figura 5.25 – Código Fonte do Sub-módulo Indicador de MG.

Pela Figura 5.25, depreende-se que ao que o Módulo Servidor de Sessão é atendido, o Sub-módulo Indicador de MG incrementa o Contador e, caso constate que o mesmo atingiu o limiar previamente definido, aciona o Sub-módulo Acionador e reinicia o Contador.

5.3.3 Media Gateway

O *Media Gateway* consiste em um *software* capaz de proporcionar conferência, Telefonia IP, bem como transcodificação, a exemplo do Asterisk PBX, utilizado para a validação da proposta, conforme Capítulo 6, configurado de forma a obter acesso à base de dados de usuários do Módulo Servidor de Sessão, bem como atender as chamadas VoIP com necessidade de transcodificação, que lhe são redirecionadas pelo Módulo Servidor de Sessão. Para tanto, alguns arquivos de configuração foram customizados, os quais são elencados na Figura 5.26.

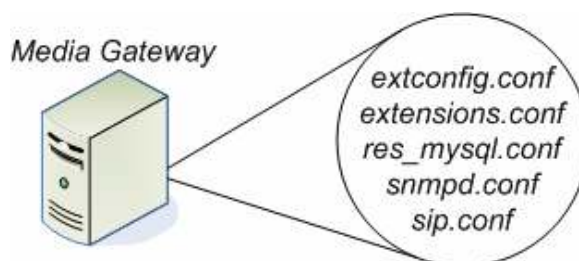


Figura 5.26 – *Media Gateway*.

A Figura 5.26 ilustra um *Media Gateway* cujos arquivos necessários, *extconfig.conf*, *extensions.conf*, *res_mysql.conf*, *snmpd.conf* e *sip.conf* foram customizados, a fim de habilitá-lo ao ambiente VoIP.

No arquivo de configuração *res_mysql.conf* foram adicionados os dados inerentes à conexão com a base de dados de usuários do Módulo Servidor de Sessão, o que é ilustrado pela Figura 5.27.

```

;
; Sample configuration for res_config_mysql.c
;
; The value of dbhost may be either a hostname or an IP address.
; If dbhost is commented out or the string "localhost", a connection
; to the local host is assumed and dbsock is used instead of TCP/IP
; to connect to the server.
;
[general]
dbhost = 200.131.189.90
dbname = asterisk
dbuser = asterisk
dbpass = 7hdwd62hnodb87RR$#
dbport = 3306
dbsock = /tmp/mysql.sock

```

Figura 5.27 – Código Fonte do Arquivo *res_mysql.conf*.

Conforme o exposto na Figura 5.27, para a conexão com a base de dados de usuários do Módulo Servidor de Sessão, foram inseridas no código fonte as seguintes informações: endereço IP, nome da base de dados, nome do usuário, senha, porta de conexão e diretório do arquivo de *socket*.

A alteração do arquivo de configuração *extconfig.conf* fez-se necessária, a fim de explicitar ao *Media Gateway*, o qual não dispõe de uma base de dados de usuários local e, especificar qual *driver* deverá utilizar para acessar à base de dados configurada no arquivo *res_mysql.conf*. A Figura 5.28 ilustra o respectivo código fonte.

```

[settings]
sipusers => mysql,asterisk,sip
sippeers => mysql,asterisk,sip

```

Figura 5.28 – Código Fonte do Arquivo *extconfig.conf*.

A Figura 5.28 informa ao Asterisk PBX que, para o registro de usuários, deve recorrer a uma base de dados “*asterisk*”, do tipo “*mysql*” para usuários do protocolo de sessão “*sip*”.

Já no arquivo *sip.conf* foi necessária apenas a alteração de dois campos:

- *realm* = 200.131.189.90 e
- *rtcachefriends* = yes

Tal alteração foi necessária, a fim de que o arquivo *sip.conf* utilizasse o Módulo Servidor de Sessão para registro dos usuários e, que os mantivesse em memória.

No arquivo *extensions.conf* fez-se necessária a adição de um contexto, como ilustrado na Figura 5.29.

```
[default]
exten => _X.,1,dial(SIP/${EXTEN},60,t)
exten => _X.,n,hangup()
```

Figura 5.29 – Código Fonte do Arquivo *extensions.conf*.

O contexto ilustrado pela Figura 5.29 torna possível o estabelecimento de uma chamada que faça uso de transcodificação, por meio do protocolo de sessão SIP.

Já para o monitoramento do *Media Gateway*, utilizou-se as chamadas de sistema do protocolo SNMP, cujo respectivo arquivo de configuração é apresentado na íntegra como Anexo A.3.

A seguir são apresentados os principais comandos pertinentes ao *Media Gateway*, no caso o Asterisk PBX:

- *asterisk* – inicia o serviço;
- *stop gracefully* – para o serviço;
- *sip show peers* – mostra todos os participantes registrados.

A seção seguinte apresenta três possíveis cenários para a utilização do Módulo Balanceador de Chamadas.

5.4 Cenários de Atuação do Módulo Balanceador de Chamadas

São exemplificativos de possibilidades de atuação do Módulo Balanceador de Chamadas os cenários expostos a seguir, tipificadores das situações: participantes com *codecs* distintos e elevado ou reduzido percentuais disponíveis de processador e memória e, com *codecs* iguais.

A Figura 5.30 ilustra um *suposto* cenário, em que todos os percentuais de disponibilidade dos recursos monitorados nos *Media Gateways* estão normalizados, que é composto por:

- Três *Media Gateways* com percentuais distintos de carga e, por conseguinte, VCBs diferentes, a saber: *Media Gateway* 1 com VCB = 96, *Media Gateway* 2 com VCB = 48 e *Media Gateway* 3 com VCB = 28;

- Contempla, ainda, dois Participantes, A e B, dotados de *codecs* diferentes, respectivamente GSM e G.711 μ -law;
- Um Servidor de Chamadas, ao qual se agrega o Módulo Servidor de Sessão, bem como a respectiva Lista de Prioridades propiciada pelo Módulo Balanceador de Chamadas, na qual afiguram-se apenas os *Media Gateways* 2 e 3, ordenados pelos seus respectivos VCBs.

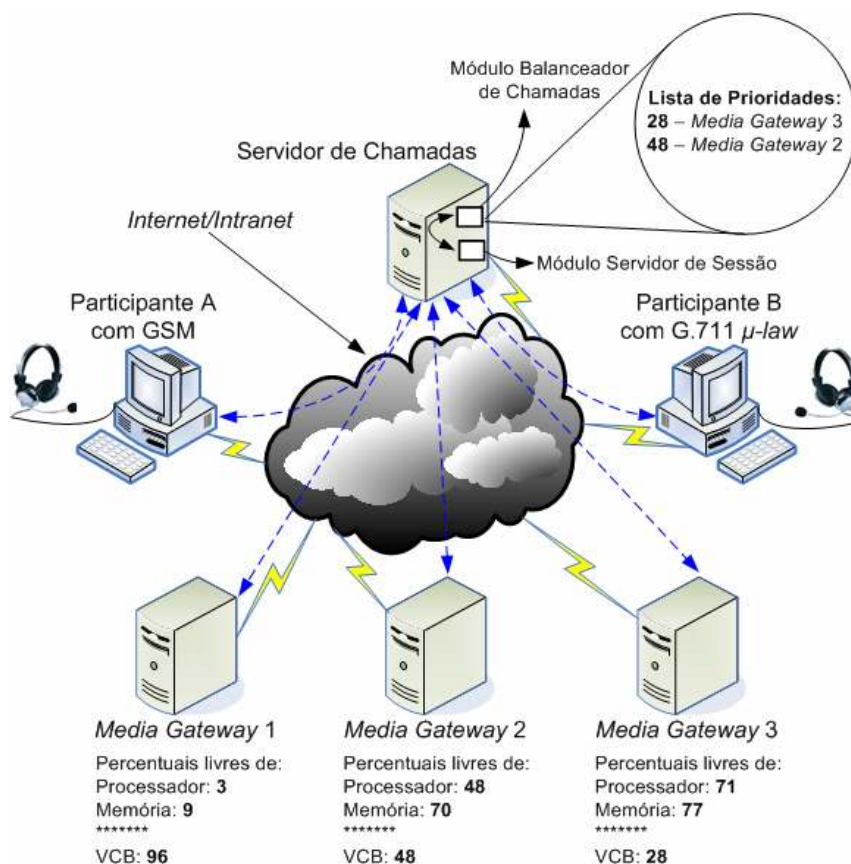


Figura 5.30 – Cenário com *Media Gateway* Inativo.

Depreende-se pela análise da Figura 5.30 que:

- a próxima chamada VoIP a ser transcodificada será atendida pelo *Media Gateway* 3, haja vista seu VCB ser o menor entre todos, pois dispõe de maior percentual de recursos computacionais disponíveis que os demais;
- o *Media Gateway* 1 não se encontra na Lista de MGs Ativos e, conseqüentemente, na Lista de Prioridades, por não perfazer o limiar mínimo de VCB.

A Figura 5.31 ilustra um *suposto* segundo cenário, similar ao da Figura 5.30, em que todos os percentuais de disponibilidade dos recursos monitorados nos *Media Gateways* estão normalizados, que é composto por:

- Três *Media Gateways* com percentuais distintos de carga e, por conseguinte, VCBs diferentes, a saber: *Media Gateway 1* com VCB = 45, *Media Gateway 2* com VCB = 75 e *Media Gateway 3* com VCB = 52;
- Contempla, ainda, dois Participantes, A e B, dotados de *codecs* diferentes, respectivamente GSM e G.711 μ -law;
- Um Servidor de Chamadas, ao qual se agrega o Módulo Servidor de Sessão, bem como a respectiva Lista de Prioridades propiciada pelo Módulo Balanceador de Chamadas, na qual afiguram-se os três *Media Gateways* que compõem o cenário, ordenados pelos seus respectivos VCBs.

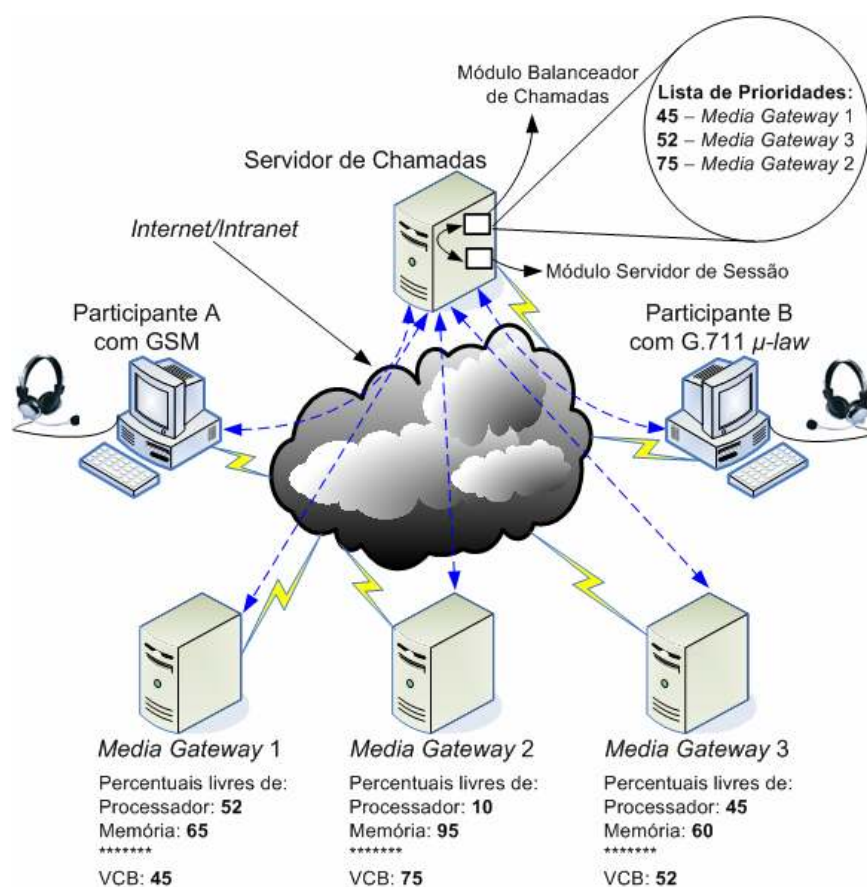


Figura 5.31 – Cenário de *Media Gateway* com Elevado Percentual de Memória Disponível.

Do cenário da Figura 5.31, depreende-se:

- O *Media Gateway* 1, por possuir menor valor de VCB, em razão de apresentar maior percentual disponível de processador, constitui-se entre os demais, o mais apto à transcodificação. Por conseguinte, a próxima chamada VoIP a ser transcodificada será por ele atendida;
- o *Media Gateway* 2, apesar de maior VCB, constitui-se no menos apto à transcodificação, apesar do elevado percentual de memória disponível, mas ínfimo percentual de processador disponível;
- o *Media Gateway* 3 apresenta patamar satisfatório, inferior ao do *Media Gateway* 1 e superior ao do *Media Gateway* 2.

A Figura 5.32 ilustra um *suposto* terceiro cenário, similar ao apresentado na Figura 5.30, em que todos os percentuais de disponibilidade dos recursos monitorados nos *Media Gateways* estão normalizados, que é composto por:

- Três *Media Gateways* com percentuais distintos de carga e, por conseguinte, VCBs diferentes, a saber: *Media Gateway* 1 com VCB = 36, *Media Gateway* 2 com VCB = 50 e *Media Gateway* 3 com VCB = 17;
- Contempla, ainda, dois Participantes, A e B, dotados de *codecs* diferentes, respectivamente GSM e G.711 μ -law;
- Um Servidor de Chamadas, ao qual se agrega o Módulo Servidor de Sessão, bem como a respectiva Lista de Prioridades propiciada pelo Módulo Balanceador de Chamadas, na qual afiguram-se os três *Media Gateways* que compõem o cenário, ordenados pelos seus respectivos VCBs.

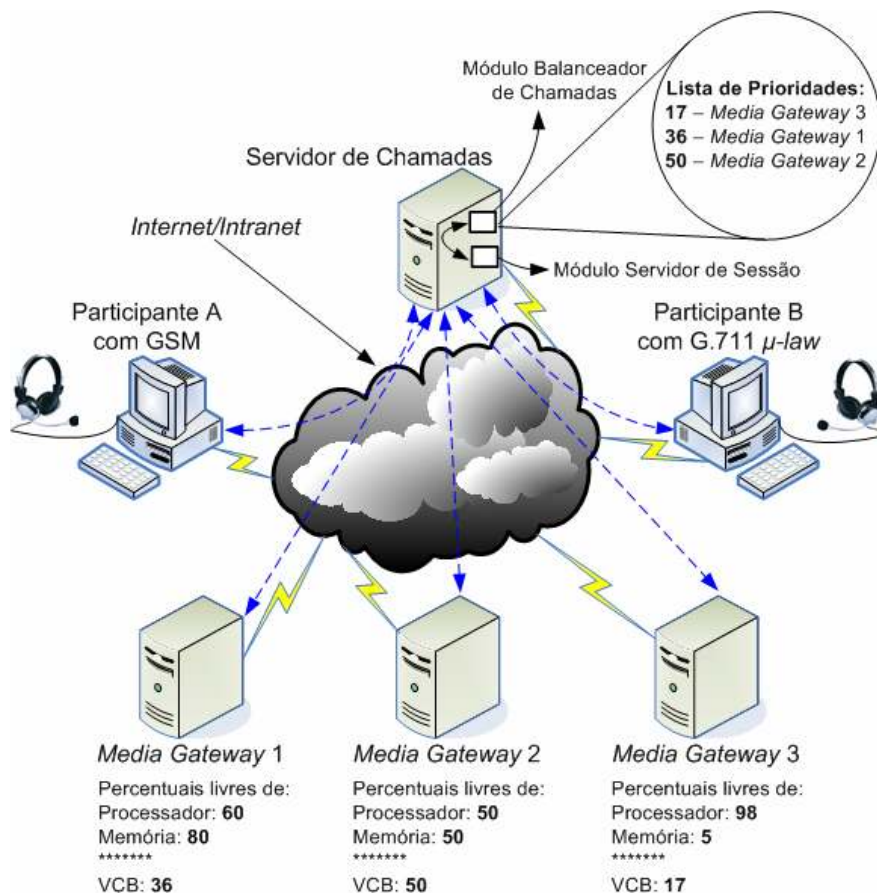


Figura 5.32 – Cenário de *Media Gateway* com Elevado Percentual Disponível de Processador.

Ao analisar o cenário da Figura 5.32, observa-se que:

- Os três *Media Gateways* integram a Lista de Prioridades por portarem VCBs que os habilitam à transcodificação, e nela são por eles ordenados;
- enfatiza-se contudo ser o *Media Gateway* 3, por possuir menor VCB, o que melhor se classifica na Lista de Prioridades, mesmo com ínfimo percentual de memória disponível, mas pelo elevado percentual disponível de processador, o que lhe dota de menor VCB e consequentemente melhor aptidão à transcodificação. Por conguinte, a próxima chamada VoIP a ser transcodificada será por ele atendida.

5.5 Conclusão

O presente capítulo valeu-se das considerações pertinentes quanto ao Módulo Balanceador de Chamadas, desde sua concepção, apresentação dos Sub-módulos e respectiva descrição de suas peculiaridades e em especial quanto à sua implementação. Afigurou-se como de relevância a ponderação de recursos computacionais disponíveis, pelo cálculo do VCB que requereu o desenvolvimento de uma equação provisionadora de fidedignidade ao mesmo.

O capítulo seguinte contempla a validação do Módulo Balanceador de Chamadas, por meio da análise de testes comprobatórios.

Capítulo 6

Validação da Proposta

O presente capítulo consiste na validação do Módulo Balanceador de Chamadas como eficaz solução ao balanceamento de chamadas VoIP a transcodificar. Conforme descrito no capítulo anterior, ele é composto por Sub-módulos com funcionalidades integradoras que lhe oportunizam a aptidão buscada, para resolver o problema do balanceamento das referidas chamadas.

O mecanismo básico do Módulo Balanceador de Chamadas consiste em aferir *Media Gateways* que componham um cenário VoIP e classificá-los por meio do cálculo dos respectivos VCBs. Desta forma, quando uma chamada VoIP necessite ser transcodificada, possibilita-lhe ser encaminhada ao *Media Gateway* que dispuser do maior percentual de recursos computacionais disponíveis, constatado na Lista de Prioridades, composta pelos VCBs em ordem crescente.

Buscou-se comprovar a fidedignidade do Módulo Balanceador de Chamadas pela acurada aferição dos resultados proporcionados pela sua utilização. Para isso, foi necessário, de modo especial, o cálculo dos VCBs, elementos-chave essenciais à equânime distribuição de chamadas, pois normaliza e equaliza os recursos monitorados nos *Media Gateways*, em razão de serem dotados de *hardware* com peculiaridades, quanto às quantidades de memória e processador por exemplo.

Após a implementação do Módulo Balanceador de Chamadas, optou-se, como metodologia para validação, por submetê-lo a testes em cenário real. Os referidos testes foram então realizados no LARC (Laboratório Avançado de Redes de Computadores)

do PPGCC (Programa de Pós-Graduação em Ciência da Computação) da Universidade Federal de Uberlândia. A Figura 6.1 ilustra os elementos componentes de tal cenário, após que são elencados os testes comprovadores da sua eficácia.

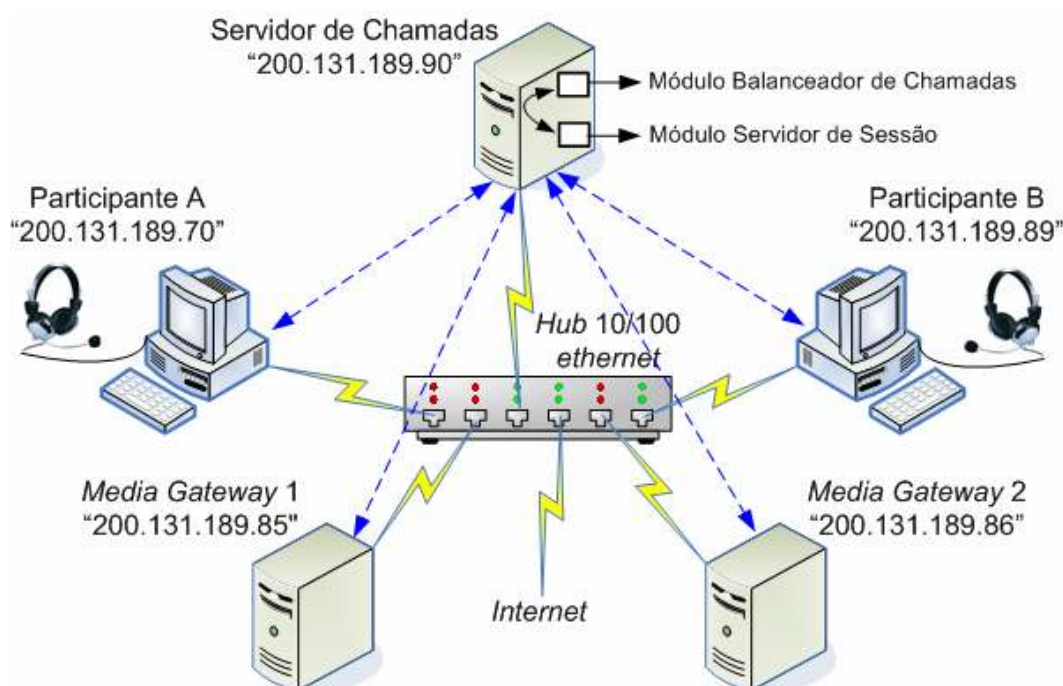


Figura 6.1 – Cenário de Testes do Módulo Balanceador de Chamadas.

Cada máquina ilustrada na Figura 6.1 do cenário foi configurada, a fim de prover a viabilização do Módulo Balanceador de Chamadas, como solução ao balanceamento de chamadas VoIP a transcodificar. As seções seguintes tratam das configurações que se fizeram necessárias às mesmas.

A Figura 6.1 apresenta o cenário utilizado à realização dos testes validadores da proposta, que foi composto por cinco máquinas, designadas: Participantes A e B, *Media Gateways* 1 e 2 e Servidor de Chamadas, integrado pelo Módulo Servidor de Sessão e o Módulo Balanceador de Chamadas.

Caracterizam as máquinas utilizadas:

- Participantes A e B: constituíram usuários da VoIP, com os respectivos endereços IP 200.131.189.70 e 200.131.189.89, ambos dotados dos *codecs* G.711 (*μ-law* e *a-law*) e GSM. Por ocasião da realização dos testes, para cada chamada, habilitava-se cada qual por vez, enquanto os demais permaneciam

desabilitados. Ainda, para utilizarem a VoIP, aos participantes foi disponibilizado o *softphone X-Lite* versão 3.0, recomendável para máquinas com a configuração mínima de processador Intel Pentium III 700 MHz ou equivalente e, 256 MB de memória RAM [56];

- *Media Gateways* (1 e 2): máquina oportunizadora de interconexão entre diferentes tipos de mídia e, em especial à transcodificação, cujo *software* possibilitador de tais funcionalidades é o Asterisk PBX versão 1.4.19.1, utilizado nos testes comprobatórios da eficácia do Módulo Balanceador de Chamadas. Suas especificações são dadas a seguir [41].
 - *Media Gateway* 1: denominado *asterisk1*, com endereço IP 200.131.189.85, dotado de processador Intel(R) Celeron(R) CPU 2.00 GHz, 512 MB de memória, Sistema Operacional *Slackware* Linux 12.1 e Net-SNMP 5.4.1 [24] [60];
 - *Media Gateway* 2: denominado *asterisk2*, com endereço IP 200.131.189.86, dotado de processador Pentium III (*Coppermine*) de 900 MHz, 384 MB de memória, Sistema Operacional *Slackware* Linux 12.1 e Net-SNMP 5.4.1 [24] [60].
- Servidor de Chamadas: máquina à qual se integram o Módulo Balanceador de Chamadas e o Módulo Servidor de Sessão, dotada de processador Intel(R) Pentium(R) 4 CPU 3.40 GHz *dual core*, 1 GB de memória, Sistema Operacional *Slackware* Linux 12.1 e *software* OpenSER 1.3.2-notls [45] [60].

Como medidas de segurança, ressalta-se que as máquinas utilizadas para a validação da proposta foram dedicadas, ou seja, no sentido de utilizadas apenas por uma pessoa e, a rede utilizada era fechada, mesmo com acesso à *Internet* em todos os componentes.

Feitas tais ponderações, conceberam-se e executaram-se testes validadores do Módulo Balanceador de Chamadas, contemplando as variáveis atinentes ele. Descrevem-se a seguir, os quatro testes concebidos e efetuados, com o objetivo de validar o proposto:

- O primeiro “Teste 1”: participantes com *codecs* iguais, com o objetivo de aferir o comportamento do Módulo Balanceador de Chamadas, frente a chamadas sem necessidade de transcodificação e não carecedoras portanto, de balanceamento;
- O segundo “Teste 2”: participantes com *codecs* distintos, com o objetivo de aferir a eficácia do Módulo Balanceador de Chamadas, em face da chamada com necessidade de transcodificação, que, conseqüentemente, faz uso do balanceamento provido por ele;
- O terceiro “Teste 3”: participantes com *codecs* distintos, com o objetivo de aferir a eficácia do Módulo Balanceador de Chamadas, frente à situação em que um dos *Media Gateways* se encontrava inativo, com o objetivo de mostrar o comportamento do Módulo Balanceador de Chamadas frente a ambos;
- O quarto “Teste 4”: participantes com *codecs* distintos, com o objetivo de aferir a eficácia do Módulo Balanceador de Chamadas, frente à situação em que ambos os *Media Gateways* encontravam-se inativos.

A seguir, detalham-se e apresentam-se os resultados dos testes em seus respectivos cenários.

6.1 Teste 1, Participantes com Idênticos *Codecs*

Esse teste foi realizado com participantes de idênticos *codecs*, G.711 *a-law*, no respectivo cenário de testes, apresentado na Figura 6.1. Ilustra-o a Figura 6.2.

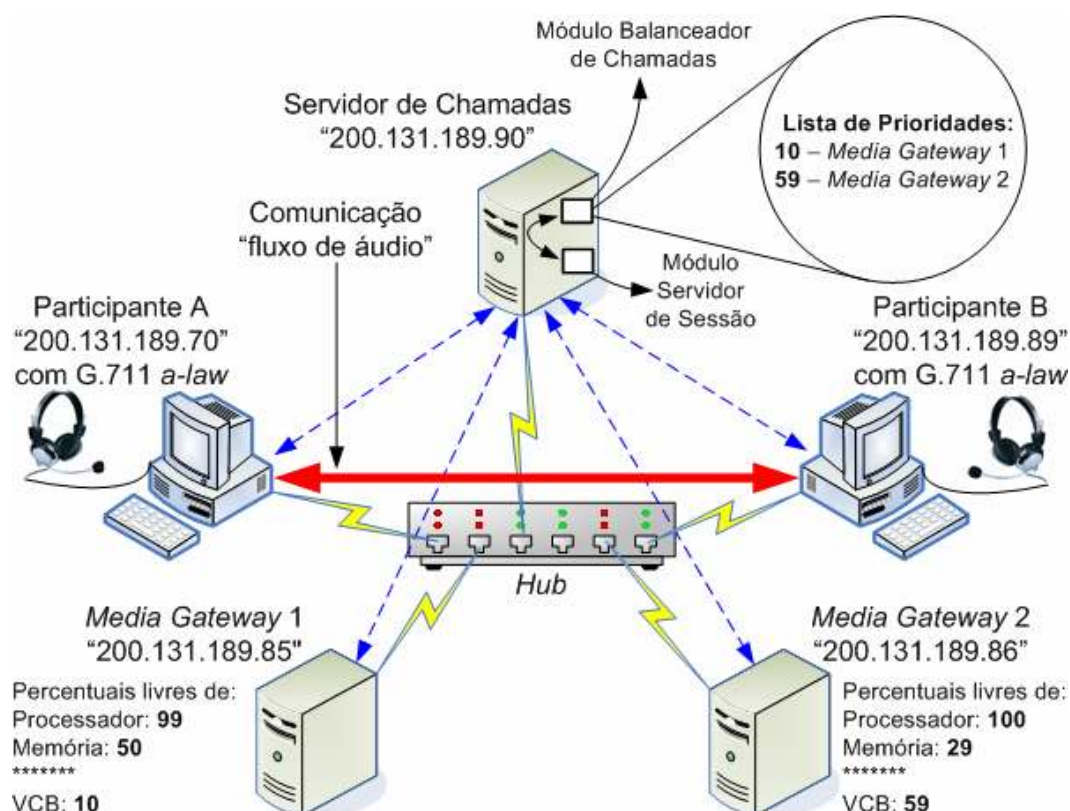


Figura 6.2 – Teste 1, Participantes com Idênticos Codecs.

O objetivo do Teste 1, ilustrado pela Figura 6.2, foi verificar como o Módulo Balanceador de Chamadas se portou no ambiente VoIP, cenário de testes, entre participantes que possuíam idênticos *codecs* e, conseqüentemente, o fluxo de áudio da comunicação era encaminhado de modo (*Peer-to-Peer*). Ressalta-se que figuram na mesma apenas os VCBs correspondentes ao antes da realização da primeira chamada.

Ao todo, foram efetuadas cinco chamadas, tanto originárias do Participante A com destino ao Participante B e vice-versa, as quais se encontram listadas na Tabela 6.1. Ambos os participantes utilizaram-se do *codec* G.711 *a-law*.

Tabela 6.1 – Chamadas do Teste 1, com Participantes de Idênticos Codecs.

Chamada	Origem	Destino	Início	Término
1ª	Participante B	Participante A	11:02	11:03
2ª	Participante A	Participante B	11:04	11:05
3ª	Participante A	Participante B	11:06	11:08
4ª	Participante B	Participante A	11:09	11:10
5ª	Participante A	Participante B	11:11	11:13

Figuram na Tabela 6.1 os participantes das chamadas efetuadas, caracterizados como origem e destino, bem como sua duração, pelo registro dos seus respectivos inícios e términos. O transcurso para as chamadas do Teste 1 é representado pela Figura 6.3, sob a forma de linha de tempo.

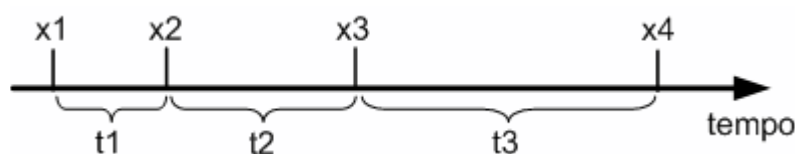


Figura 6.3 – Linha de Tempo para o Teste 1.

A Figura 6.3 representa a síntese das etapas das chamadas VoIP efetuadas no Teste 1, em que x e t correspondem, respectivamente, a momentos e intervalos de tempo, descritos a seguir:

- x1: momento que marca o início da chamada;
- t1: intervalo de tempo decorrido para o início do tocar “ringing”;
- x2: ocorrência do tocar “ringing”;
- t2: intervalo de tempo em que a chamada tocou;
- x3: momento de atendimento da chamada;
- t3: intervalo de tempo de duração da conversação;
- x4: momento de término da chamada.

Observadas as etapas integrantes da linha de tempo da Figura 6.3, representativa das chamadas efetuadas no Teste 1, constatou-se a não figuração nos seus respectivos *logs*, das etapas entre x1 até x3, pelo fato do Módulo Servidor de Sessão apenas informar os instantes de x3 até x4. Constatam-se como peculiares as chamadas VoIP não transcodificadas, a omissão pelo Módulo Servidor de Sessão nos respectivos *logs* dos dados referentes às etapas de x1 até x3. Comprova tal constatação o Anexo B.1 dos *logs* das chamadas realizadas para o Teste 1.

Tabela 6.2 – Momentos e Intervalos de Tempo do Teste 1.

	x1	t1	x2	t2	x3	t3	x4
1ª	-	-	-	-	11:02:04	68s	11:03:12
2ª	-	-	-	-	11:04:18	72s	11:05:30
3ª	-	-	-	-	11:06:23	103s	11:08:06
4ª	-	-	-	-	11:09:13	80s	11:10:33
5ª	-	-	-	-	11:11:21	110s	11:13:11

Nas chamadas VoIP realizadas no Teste 1, com o uso de *codecs* idênticos, não houve necessidade de transcodificação, pelo que se constatou que mesmo o Módulo Balanceador de Chamadas estando ativo no Módulo Servidor de Sessão, conforme a Tabela 6.3 de VCBs, disponibilizados na Lista de Prioridades durante o Teste 1, o mesmo não foi acionado.

Tabela 6.3 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 1.

VCBs do <i>Media Gateway</i> 1							VCBs do <i>Media Gateway</i> 2						
	Antes		Durante		Depois			Antes		Durante		Depois	
	Hora	VCB	Hora	VCB	Hora	VCB		Hora	VCB	Hora	VCB	Hora	VCB
1ª	11:02:02	10	11:03:02	9	11:04:02	10	1ª	11:02:02	59	11:03:02	59	11:04:02	59
2ª	11:04:02	10	11:05:02	10	11:06:02	9	2ª	11:04:02	59	11:05:02	60	11:06:02	59
3ª	11:06:02	9	11:07:02	9	11:09:02	9	3ª	11:06:02	59	11:07:02	59	11:09:02	59
4ª	11:09:02	9	11:10:02	9	11:11:02	10	4ª	11:09:02	59	11:10:02	59	11:11:02	59
5ª	11:11:02	10	11:13:02	9	11:14:02	10	5ª	11:11:02	59	11:13:02	59	11:14:02	59
	Média:	9,6	Média:	9,2	Média:	9,6		Média:	59	Média:	59,2	Média:	59

Pela análise da Tabela 6.3, dos VCBs disponibilizados na Lista de Prioridades, cujos *logs* compõem o Anexo C.1, constatou-se que o Módulo Balanceador de Chamadas estava ativo em todos os momentos “Antes, Durante e Depois” nas cinco chamadas efetuadas. Ainda, detectou-se que as médias para os *Media Gateways* 1 e 2 durante a realização das chamadas sofreram ligeiras alterações em referência aos momentos Antes e Depois. No *Media Gateway* 1 sofreu pequeno decréscimo, ao passo que no *Media Gateway* 2, ao contrário, sofreu pequeno aumento.

Contudo, como não houve necessidade de transcodificação e, consequentemente, de balanceamento, o Módulo Balanceador de Chamadas não foi acionado. Tal teste

validou que o módulo apenas faz-se necessário quando da necessidade de transcodificação.

6.2 Teste 2, Participantes com *Codecs* Distintos

Ao contrário do Teste 1, em que o Módulo Balanceador de Chamadas, mesmo ativo não foi acionado, realizou-se o presente teste, composto pelos Participantes A e B, com respectivos *codecs*, GSM e G.711 *a-law* e dois *Media Gateways* ativos, 1 e 2. Nesse caso, quando da chamada entre ambos, houve a necessidade de transcodificação, que, para ocorrer, careceu de um *Media Gateway*, provido pelo Módulo Balanceador de Chamadas, acionado pelo Módulo Servidor de Sessão ao receber a resposta SIP 488. A Figura 6.4 ilustra o cenário utilizado no respectivo teste.

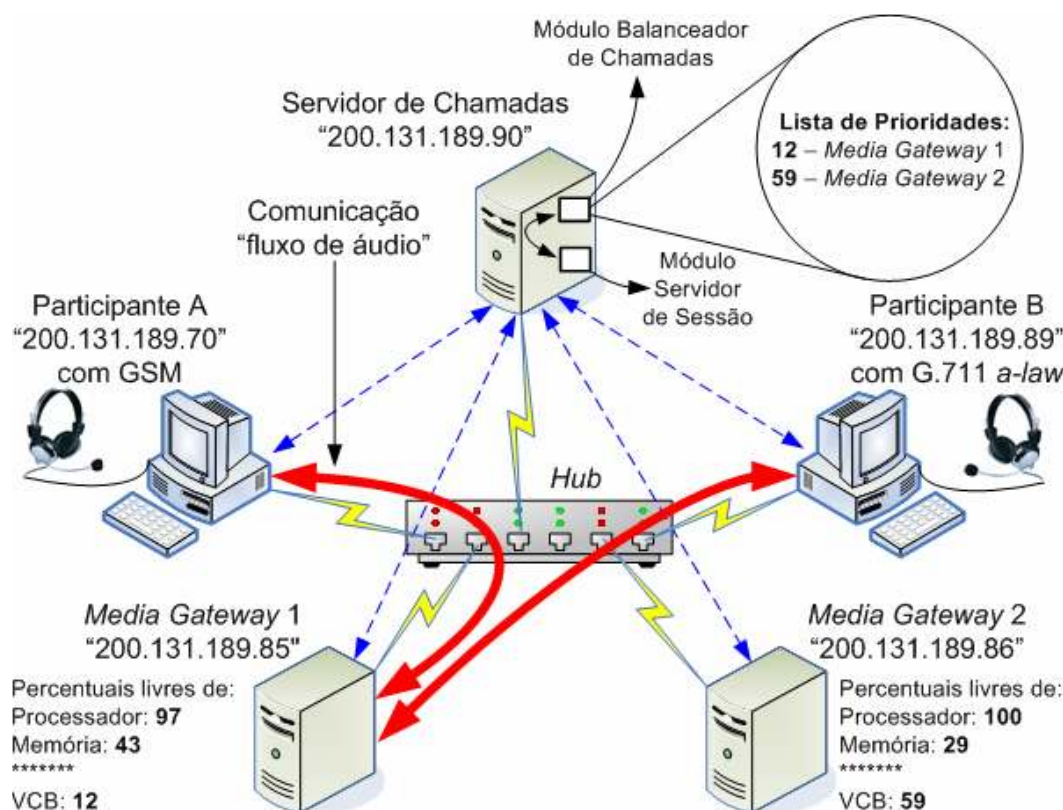


Figura 6.4 – Teste 2, Participantes com *Codecs* Distintos.

O objetivo do Teste 2, ilustrado pela Figura 6.4, foi verificar como o Módulo Balanceador de Chamadas se portou no ambiente VoIP, cenário de testes, com participantes que possuíam *codecs* distintos. Em todas as chamadas realizadas, os fluxos de áudio da comunicação deram-se via *Media Gateway* e não de forma P2P, devido à necessidade de transcodificação, ao contrário do Teste 1.

O tempo requerido ao processamento da transcodificação é relativo aos *codecs* dos participantes e ao *hardware* do *Media Gateway* e, portanto, difere em sentidos contrários, ou seja, quando, ela é direcionada do Participante A para o B, ou, do B para o A, como comprovado nas Tabelas 4.4 e 4.5.

Ressalta-se, que na Figura 6.4 figuram-se apenas os VCBs correspondentes à ocasião “Antes” da realização da primeira chamada, que perfizeram um total de cinco, tanto originárias do Participante A com destino ao Participante B, como vice-versa, que se encontram listadas na Tabela 6.4.

Tabela 6.4 – Chamadas do Teste 2, Participantes com *Codecs* Distintos.

Chamada	Origem	Destino	Início	Término
1ª	Participante A	Participante B	11:24	11:25
2ª	Participante B	Participante A	11:27	11:28
3ª	Participante A	Participante B	11:29	11:30
4ª	Participante B	Participante A	11:32	11:33
5ª	Participante A	Participante B	11:35	11:37

Figuram na Tabela 6.4 os participantes das chamadas efetuadas, caracterizados como origem e destino, bem como a duração das chamadas, pelo registro dos seus respectivos inícios e terminos. O transcurso para as chamadas do Teste 2 é representado pela Figura 6.5, sob a forma de linha de tempo.

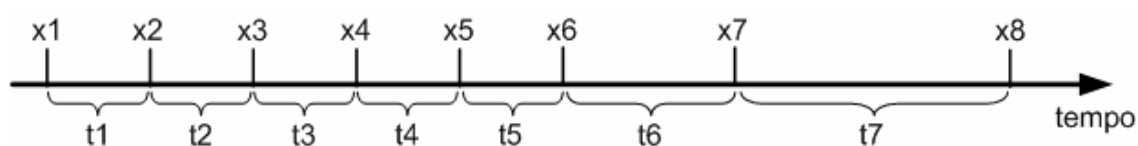


Figura 6.5 – Linha de Tempo para o Teste 2.

A Figura 6.5 representa a síntese das etapas das chamadas VoIP efetuadas no Teste 2, em que x e t correspondem, respectivamente, a momentos e intervalos de tempo, descritos a seguir:

- x_1 : momento que marca o início da chamada;
- t_1 : intervalo de tempo decorrido até a detecção do erro 488;
- x_2 : momento em que o Módulo Servidor de Sessão detecta o erro 488;
- t_2 : intervalo de tempo decorrido para que o Módulo Servidor de Sessão acione o Módulo Balanceador de Chamadas;
- x_3 : momento em que o Módulo Balanceador de Chamadas é acionado;
- t_3 : intervalo de tempo decorrido para que o Sub-módulo Indicador de MG consulte a Lista de Prioridades e retorne o endereço IP do *Media Gateway* nela melhor classificado, ou seja, que possua menor VCB;
- x_4 : momento em que o Módulo Servidor de Sessão recebe o endereço IP pelo Módulo Balanceador de Chamadas;
- t_4 : intervalo de tempo decorrido para que o Módulo Servidor de Sessão encaminhe a chamada ao *Media Gateway* indicado;
- x_5 : momento em que o Módulo Servidor de Sessão efetiva o encaminhamento da chamada ao *Media Gateway* indicado;
- t_5 : intervalo de tempo decorrido até que o *Media Gateway* seja acionado;
- x_6 : momento em que o *Media Gateway* é acionado e inicia-se o tocar “*ringing*”;
- t_6 : intervalo de tempo durante o qual a chamada tocou “*ringing*”;
- x_7 : momento de atendimento da chamada e início da conversação;
- t_7 : intervalo de tempo de duração da conversação;
- x_8 : momento de término da chamada.

A Tabela 6.5 apresenta os dados contidos nos respectivos *logs* das chamadas efetuadas no Teste 2, apresentados no Anexo B.2 e sumarizadas pela linha de tempo da Figura 6.5.

Tabela 6.5 – Momentos, Intervalos de Tempo e *Media Gateway* do Teste 2.

	x1	t1	x2	t2	x3	t3	x4	<i>Media Gateway</i>
1ª	-	-	11:24:10	< 1s	11:24:10	< 1s	11:24:10	200.131.189.85
2ª	-	-	11:27:16	< 1s	11:27:16	< 1s	11:27:16	200.131.189.85
3ª	-	-	11:29:06	< 1s	11:29:06	< 1s	11:29:06	200.131.189.85
4ª	-	-	11:32:08	< 1s	11:32:08	< 1s	11:32:08	200.131.189.85
5ª	-	-	11:35:47	< 1s	11:35:47	< 1s	11:35:47	200.131.189.85

Continuação da Tabela 6.5.

	t4	x5	t5	x6	t6	x7	t7	x8
1ª	< 1s	11:24:10	< 1s	11:24:10	7s	11:24:17	89s	11:25:46
2ª	< 1s	11:27:16	< 1s	11:27:16	15s	11:27:31	37s	11:28:08
3ª	< 1s	11:29:06	< 1s	11:29:06	13s	11:29:19	81s	11:30:40
4ª	< 1s	11:32:08	< 1s	11:32:08	7s	11:32:15	54s	11:33:09
5ª	< 1s	11:35:47	< 1s	11:35:47	4s	11:35:51	99s	11:37:30

Como comprovado nos *logs* constituintes do Anexo B.2, neles não figuram o momento x1 e o intervalo t1. Ou seja, o início do registro da chamada no *log*, deu-se apenas a partir de x2, momento de detecção do erro 488, inerente a chamadas VoIP com participantes de *codecs* distintos.

O tratamento do erro 488 desencadeou o acionamento do Módulo Balanceador de Chamadas, conforme concebido, por meio do Sub-módulo Indicador de MG, que consultou a Lista de Prioridades e retornou ao Módulo Servidor de Sessão o IP do *Media Gateway* nela melhor classificado. Isso possibilitou o re-encaminhamento da chamada a ser transcodificada ao respectivo *Media Gateway*, indicado pelo Módulo Balanceador de Chamadas, evitando-se o descarte da mesma.

A Lista de Prioridades é composta por VCBs e endereços IPs dos *Media Gateways* aos quais correspondem. O *Media Gateway* somente é nela listado, se dotado de recurso computacional mínimo disponível, conforme já anteriormente tratado, ao contrário, se chamadas fossem encaminhadas a *Media Gateways* saturados, a qualidade seria consideravelmente comprometida.

A Tabela 6.6 é ilustrativa da Lista de Prioridades dos dois *Media Gateways* componentes do Teste 2, aferida nas ocasiões “Antes, Durante e Depois” das chamadas realizadas, conforme constante dos *logs* no Anexo C.2.

Tabela 6.6 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 2.

VCBs do <i>Media Gateway</i> 1							VCBs do <i>Media Gateway</i> 2						
	Antes		Durante		Depois			Antes		Durante		Depois	
	Hora	VCB	Hora	VCB	Hora	VCB		Hora	VCB	Hora	VCB	Hora	VCB
1ª	11:24:02	12	11:25:02	13	11:26:02	14		11:24:02	59	11:25:02	59	11:26:02	59
2ª	11:27:02	12	11:28:02	13	11:29:02	12		11:27:02	59	11:28:02	59	11:29:02	59
3ª	11:29:02	12	11:30:02	13	11:31:02	12		11:29:02	59	11:30:02	59	11:31:02	59
4ª	11:32:02	11	11:33:02	13	11:34:02	12		11:32:02	59	11:33:02	59	11:34:02	59
5ª	11:35:02	11	11:37:02	12	11:38:02	11		11:35:02	59	11:37:02	59	11:38:02	59
	Média:	11,6	Média:	12,8	Média:	12,2		Média:	59	Média:	59	Média:	59

Pela análise da Tabela 6.6, comprovou-se que o VCB do *Media Gateway* acionado foi o menor, entre os integrantes da Lista de Prioridades. No presente teste, referente ao *Media Gateway* 1, é relevante ressaltar que a transcodificação, conforme constatada, consome recursos computacionais, pois, invariavelmente, em todas as chamadas realizadas no Teste 2, constatou-se alteração entre o valor do VCB na ocasião “Durante” em relação às demais ocasiões, ou seja, “Antes e Depois”. Constatou-se ainda que o VCB permaneceu, praticamente, inalterado em tal ocasião, nas chamadas provenientes de ambos os sentidos dos participantes, ou seja, quando direcionadas de A para B e de B para A.

Observa-se no *Media Gateway* 2, não utilizado em nenhuma chamada do presente teste, que o VCB permaneceu inalterado nas ocasiões “Antes, Durante e Depois” às chamadas, mantendo fixamente o único valor com que figurou na Lista de Prioridades, conforme a referida Tabela 6.6.

Constatou-se um comportamento peculiar dos VCBs durante a transcodificação, em cada chamada, ocasião em que apresentaram valores superiores aos dos momentos anterior e posterior. Isso foi comprovado pelo incremento de 1,2 unidades na média em “Durante”, em relação à ocasião “Antes” do referido teste.

Concluindo, o presente teste comprovou a eficácia do Módulo Balanceador de Chamadas, pois a necessidade da transcodificação foi informada por meio de um erro, indicativo da respectiva incompatibilidade entre os *codecs*. Então, o Módulo Servidor de Sessão acionou o Módulo Balanceador de Chamadas, que, após consultar à Lista de Prioridades, encaminhou-lhe o endereço do *Media Gateway* nela constante mais apto à

transcodificação. Ou seja, aquele dotado de menor VCB. Deste modo, o fluxo de áudio não mais foi encaminhado de modo P2P e, sim, via *Media Gateway* selecionado pelo Módulo Balanceador de Chamadas.

6.3 Teste 3, com *Media Gateway* Ativo e Inativo

Tal como no Teste 2, os participantes não dispunham de *codecs* comuns. O Participante A utilizou o GSM, enquanto o Participante B o G.711 μ -law. Neste caso, como no referido teste, quando da chamada, houve a necessidade de transcodificação, que para seu processamento careceu do Módulo Balanceador de Chamadas, que proveu o *Media Gateway* mais apto a possibilitá-la.

Reiterando, a necessidade da transcodificação é informada por meio de um erro, indicativo da respectiva incompatibilidade entre os *codecs*, pelo que o Módulo Servidor de Sessão aciona o Módulo Balanceador de Chamadas, que após consultar à Lista de Prioridades, encaminha-lhe o IP do *Media Gateway* mais apto à transcodificação, ou seja, aquele dotado de menor VCB. Deste modo, o fluxo de áudio não mais é transmitido de modo P2P, e, sim, via *Media Gateway* selecionado. A Figura 6.6 ilustra o cenário utilizado no respectivo teste.

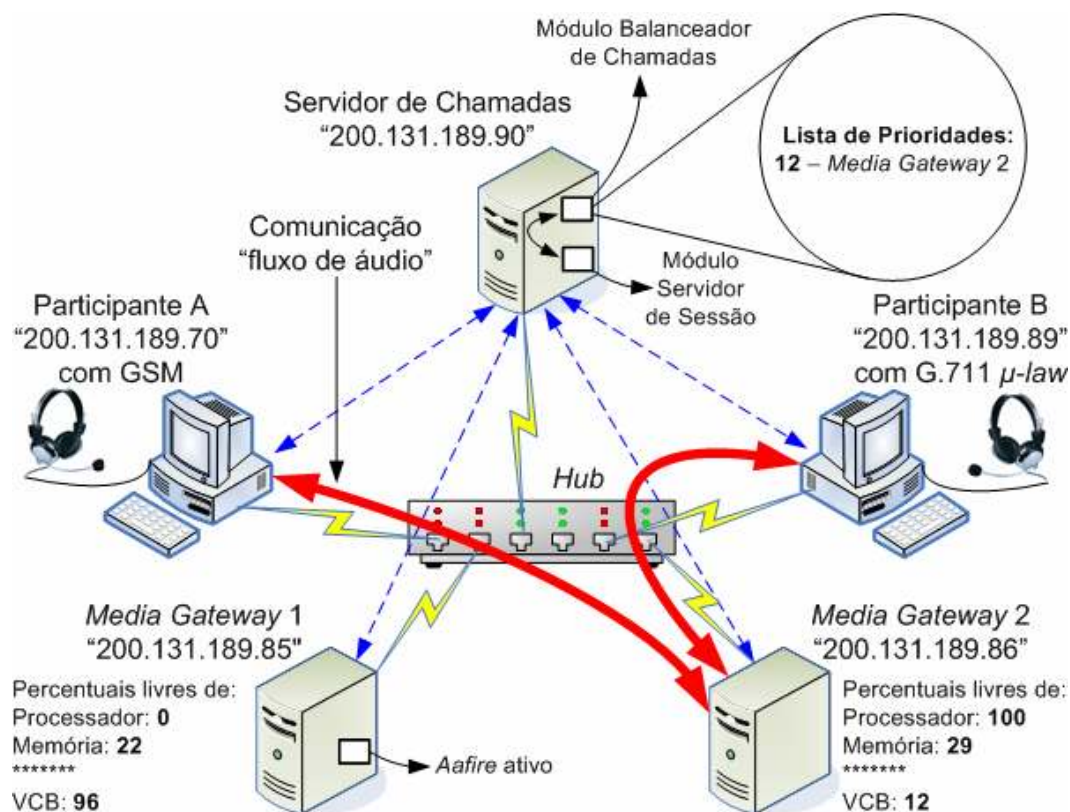


Figura 6.6 – Teste 3, com *Media Gateway* Ativo e Inativo.

O objetivo do Teste 3, ilustrado pela Figura 6.6, foi verificar como o Módulo Balanceador de Chamadas se portou no ambiente VoIP, cenário de testes, com participantes que possuíam *codecs* distintos e, dois *Media Gateways*, sendo um ativo e um inativo propositalmente. Em todas as chamadas realizadas, os fluxos de áudio da comunicação deram-se via *Media Gateway* e não de forma P2P, devido à necessidade de transcodificação, ao contrário do Teste 1.

O que o distinguiu do Teste 2 foi possuir apenas um dos *Media Gateways* ativo, neste caso o *Media Gateway 2*, em razão de o *Media Gateway 1* não dispor do limiar mínimo de VCB, o que, por conseguinte, não lhe possibilitou integrar a Lista de Prioridades. Para a inativação do *Media Gateway 1*, fez-se uso no mesmo do *software aafire*, que lhe exauriu o processador, tornando-o inapto para qualquer outro processamento e, por conseguinte, eliminou-o da Lista de Prioridades pelo cálculo do seu VCB [51].

Assim como no Teste 2, o tempo requerido ao processamento da transcodificação difere em sentidos contrários, ou seja, quando é direcionada do

Participante A para o B, ou, do B para o A, em virtude de ser relativo aos *codecs* dos participantes e ao *hardware* do *Media Gateway*, como comprovado nas Tabelas 4.4 e 4.5.

Na Figura 6.6 consta apenas o VCB correspondente à ocasião “Antes” da realização da primeira chamada, de um total de cinco, tanto originárias do Participante A com destino ao Participante B, como vice-versa, que se encontram listadas na Tabela 6.7.

Tabela 6.7 – Chamadas do Teste 3, com *Media Gateway* Ativo e Inativo.

Chamada	Origem	Destino	Início	Término
1ª	Participante B	Participante A	11:43	11:44
2ª	Participante A	Participante B	11:46	11:47
3ª	Participante B	Participante A	11:49	11:50
4ª	Participante A	Participante B	11:52	11:53
5ª	Participante B	Participante A	11:55	11:57

Figuram na Tabela 6.7 os participantes das chamadas efetuadas, caracterizados como origem e destino, bem como sua duração, pelo registro dos seus respectivos inícios e termos. O transcurso para as chamadas do Teste 3 é representado pela Figura 6.7, sob a forma de linha de tempo.

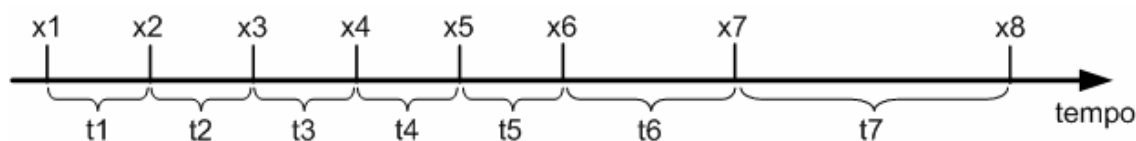


Figura 6.7 – Linha de Tempo para o Teste 3.

A Figura 6.7 representa a síntese das etapas das chamadas VoIP efetuadas no Teste 3, em que x e t correspondem, respectivamente, a momentos e intervalos de tempo, descritos a seguir:

- x1: momento que marca o início da chamada;
- t1: intervalo de tempo decorrido até a detecção do erro 488;
- x2: momento em que o Módulo Servidor de Sessão detecta o erro 488;

- t2: intervalo de tempo decorrido para que o Módulo Servidor de Sessão acione o Módulo Balanceador de Chamadas;
- x3: momento em que o Módulo Balanceador de Chamadas é acionado;
- t3: intervalo de tempo decorrido para que o Sub-módulo Indicador de MG consulte a Lista de Prioridades e retorne o endereço IP do *Media Gateway* nela melhor classificado, ou seja, que possua menor VCB;
- x4: momento em que o Módulo Servidor de Sessão recebe o endereço IP pelo Módulo Balanceador de Chamadas;
- t4: intervalo de tempo decorrido para que o Módulo Servidor de Sessão encaminhe a chamada ao *Media Gateway* indicado;
- x5: momento em que o Módulo Servidor de Sessão efetiva o encaminhamento da chamada ao *Media Gateway* indicado;
- t5: intervalo de tempo decorrido até que o *Media Gateway* seja acionado;
- x6: momento em que o *Media Gateway* é acionado e inicia-se o tocar “*ringing*”;
- t6: intervalo de tempo durante o qual a chamada tocou “*ringing*”;
- x7: momento de atendimento da chamada e início da conversação;
- t7: intervalo de tempo de duração da conversação;
- x8: momento de término da chamada.

A Tabela 6.8 apresenta os dados contidos nos respectivos *logs* das chamadas efetuadas no Teste 3, apresentados no Anexo B.3 e sumarizadas pela linha de tempo da Figura 6.7.

Tabela 6.8 – Momentos, Intervalos de Tempo e *Media Gateway* do Teste 3.

	x1	t1	x2	t2	x3	t3	x4	<i>Media Gateway</i>
1 ^a	-	-	11:43:28	< 1s	11:43:28	< 1s	11:43:28	200.131.189.86
2 ^a	-	-	11:46:06	< 1s	11:46:06	1s	11:46:07	200.131.189.86
3 ^a	-	-	11:49:10	< 1s	11:49:10	< 1s	11:49:10	200.131.189.86
4 ^a	-	-	11:52:09	< 1s	11:52:09	< 1s	11:52:09	200.131.189.86
5 ^a	-	-	11:55:42	< 1s	11:55:42	< 1s	11:55:42	200.131.189.86

Continuação da Tabela 6.8.

	t4	x5	t5	x6	t6	x7	t7	x8
1 ^a	< 1s	11:43:28	< 1s	11:43:28	8s	11:43:36	67s	11:44:43
2 ^a	< 1s	11:46:07	< 1s	11:46:07	4s	11:46:11	63s	11:47:14
3 ^a	< 1s	11:49:10	< 1s	11:49:10	5s	11:49:15	70s	11:50:25
4 ^a	< 1s	11:52:09	< 1s	11:52:09	3s	11:52:12	96s	11:53:48
5 ^a	< 1s	11:55:42	< 1s	11:55:42	5s	11:55:47	78s	11:57:05

Assim como no Teste 2, no Teste 3, é comprovado nos *logs* constituintes do Anexo B.3, que neles não figura o momento x1 e o intervalo t1. Ou seja, o início do registro da chamada no *log* deu-se apenas a partir de x2, momento de detecção do erro 488, inerente a chamadas VoIP com participantes de *codecs* distintos.

O tratamento do erro 488 desencadeou o acionamento do Módulo Balanceador de Chamadas, conforme concebido, por meio do Sub-módulo Indicador de MG, que consultou a Lista de Prioridades e retornou ao Módulo Servidor de Sessão o IP do *Media Gateway* nela melhor classificado. Isso possibilitou o re-encaminhamento da chamada a ser transcodificada ao respectivo *Media Gateway*, indicado pelo Módulo Balanceador de Chamadas, evitando-se o descarte da chamada.

A Lista de Prioridades é composta por VCBs e endereços IPs dos *Media Gateways* aos quais os mesmos correspondem. O *Media Gateway* somente é nela listado se dotado de recurso computacional mínimo disponível, conforme já anteriormente tratado, ao contrário, se chamadas fossem encaminhadas a *Media Gateways* saturados, qualidade das mesmas seria consideravelmente comprometida.

A Tabela 6.9 é ilustrativa da Lista de Prioridades dos dois *Media Gateways* componentes do Teste 3, aferida nas ocasiões “Antes, Durante e Depois” das chamadas realizadas, conforme constante dos *logs* no Anexo C.3.

A diferença entre os Testes 2 e 3, é que apesar de ambos portarem em seus cenários dois *Media Gateways*, no Teste 2, ambos constavam da Lista de Prioridades, pelo que o Módulo Balanceador de Chamadas retornou o melhor classificado, ou seja, de menor VCB. Já no Teste 3, apenas um se encontrava ativo por ocasião do teste, o *Media Gateway 2*, que por portar percentual mínimo de recurso computacional disponível, foi o único a integrar a Lista de Prioridades e, portanto, apto para ser acionado pelo Módulo Balanceador de Chamadas.

O *Media Gateway 1*, embora inativo, foi consultado, a fim de verificar se dispunha de recurso mínimo necessário para figurar na Lista de Prioridades. Sua ausência é comprovada na Tabela 6.9, em que não figura seu VCB.

Ressalta-se, contudo, que, caso o *Media Gateway 1*, inativo, viesse a dispor do requisito mínimo para integrar a Lista de Prioridades, nela constaria e poderia eventualmente ser requisitado pelo Módulo Balanceador de Chamadas. O mesmo pode ocorrer com o *Media Gateway* desligado ao ser ativado.

Tabela 6.9 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 3.

VCBs do <i>Media Gateway 1</i>							VCBs do <i>Media Gateway 2</i>						
	Antes		Durante		Depois			Antes		Durante		Depois	
	Hora	VCB	Hora	VCB	Hora	VCB		Hora	VCB	Hora	VCB	Hora	VCB
1ª	11:43:01	-	11:44:01	-	11:45:01	-		11:43:01	12	11:44:01	14	11:45:01	13
2ª	11:46:01	-	11:47:01	-	11:48:01	-		11:46:01	13	11:47:01	15	11:48:01	13
3ª	11:49:01	-	11:50:01	-	11:51:01	-		11:49:01	12	11:50:01	14	11:51:01	13
4ª	11:52:01	-	11:53:01	-	11:54:01	-		11:52:01	12	11:53:01	15	11:54:01	14
5ª	11:55:01	-	11:56:01	-	11:58:01	-		11:55:01	12	11:56:01	13	11:58:01	13
	Média:	-	Média:	-	Média:	-		Média:	12,2	Média:	14,2	Média:	13,2

Por esse teste, concluiu-se que num cenário com dois *Media Gateways*, um ativo e um inativo, as chamadas foram direcionadas apenas ao ativo, constante da Lista de Prioridades.

O *Media Gateway 2*, dotado de menor capacidade de processamento que o *Media Gateway 1*, único ativo no presente teste, apresentou variação de duas unidades na média de VCBs da ocasião “Durante”, em relação à “Antes”, o que comprova o consumo de recursos computacionais durante a transcodificação, enquanto, o *Media Gateway 1*, no Teste 2, dotado de maior capacidade de processamento no referido teste,

apresentou a diferença de apenas 1,2 unidades, comportamentos atribuídos à normalização provida pelo Módulo Balanceador de Chamadas.

Constatou-se, ainda, no presente teste, que os VCBs da ocasião “Durante” nas chamadas “1, 3 e 5” do Participante B para o A, ficaram abaixo dos valores dos VCBs das chamadas “2 e 4” efetuadas do Participante A para o B.

Em suma, com a adição do Módulo Balanceador de Chamadas, as chamadas que não seriam completadas puderam ser efetuadas e, em especial, de forma transparente ao usuário. Com isso validou-se novamente a proposta objetivada, pois, mais uma vez ficou demonstrada a eficácia do Módulo Balanceador de Chamadas para o balanceamento de chamadas VoIP a transcodificar.

6.4 Teste 4, com *Media Gateways* Inativos

Assim como nos Testes 2 e 3, o presente teste, é composto pelos Participantes A e B, com respectivos *codecs*, G.711 μ -law e G.711 a-law e dois *Media Gateways* inativos, 1 e 2. Quando intentado o estabelecimento das chamadas, elas não foram estabelecidas, devido à ausência de *Media Gateway* apto a transcodificá-la, detectada pelo Módulo Balanceador de Chamadas. A Figura 6.8 ilustra o cenário utilizado no presente teste.

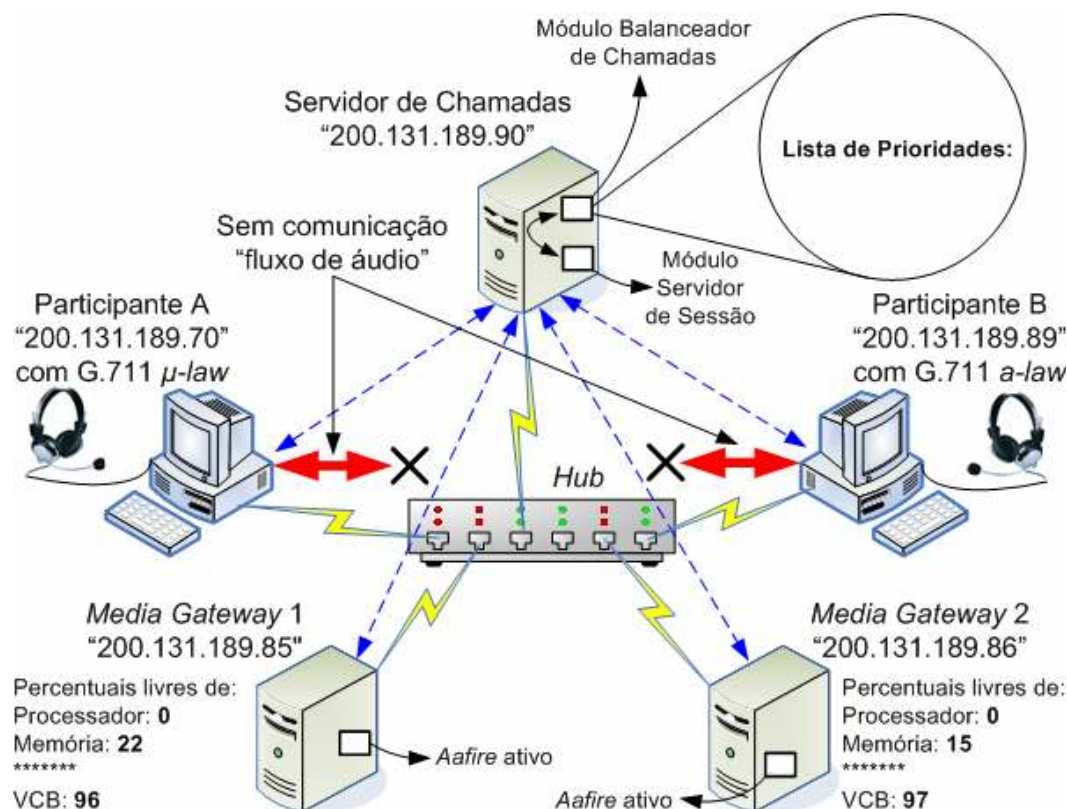


Figura 6.8 – Teste 4, com *Media Gateways* Inativos.

O objetivo do Teste 4, ilustrado pela Figura 6.8, foi verificar como o Módulo Balanceador de Chamadas se portou no ambiente VoIP, cenário de testes, com participantes que possuíam *codecs* distintos e dois *Media Gateways*, ambos inativos propositalmente. Para a inativação dos *Media Gateways* 1 e 2, fez-se uso do *software aafire*, que lhes exauriu o processador, tornando-os inaptos a qualquer outro processamento e, por conseguinte, eliminou-os da Lista de Prioridades [51].

A fim de constatar o objetivado, foram intentadas cinco chamadas. Contudo, nenhuma foi estabelecida, o que é justificado pela ausência de *Media Gateway* apto a completá-la, ou seja, transcodificá-la. A Tabela 6.10 ilustra as chamadas intentadas, tanto originárias do Participante A para o B, ou do B para o A.

Tabela 6.10 – Teste 4, com *Media Gateways* Inativos.

Chamada	Origem	Destino	Início	Término
1 ^a	Participante A	Participante B	12:19	12:19
2 ^a	Participante B	Participante A	12:20	12:20
3 ^a	Participante A	Participante B	12:21	12:21
4 ^a	Participante B	Participante A	12:22	12:22
5 ^a	Participante A	Participante B	12:23	12:23

Figuram na Tabela 6.10 os participantes das chamadas efetuadas, caracterizados como origem e destino, bem como a duração, pelo registro dos seus respectivos inícios e terminos coincidentes. Mesmo não tendo sido possível estabelecer nenhuma chamada em tal cenário, no entanto, o Módulo Balanceador de Chamadas esteve ativo e, quando as chamadas foram intentadas, foi acionado, conforme constante nos *logs* integrantes do Anexo B.4, comprobatórios do respectivo teste.

O fato do Módulo Balanceador de Chamadas, apesar de acionado, não ter retornado nenhum *Media Gateway* apto a estabelecer as chamadas por meio da transcodificação, justifica-se, em virtude da ausência de *Media Gateway* na Lista de Prioridades. O transcurso para as chamadas do Teste 4 é representado pela Figura 6.11, sob forma de linha de tempo.

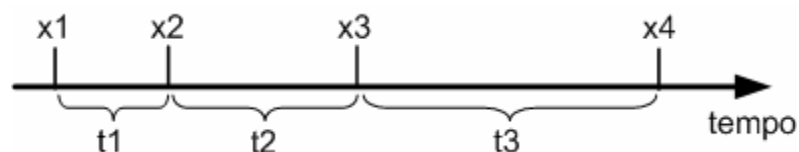


Figura 6.9 – Linha de Tempo para o Teste 4.

A Figura 6.9 representa a síntese das etapas das chamadas VoIP efetuadas no Teste 4, em que x e t correspondem, respectivamente, a momentos e intervalos de tempo, descritos a seguir:

- x1: momento que marca o início da chamada;
- t1: intervalo de tempo decorrido até a detecção do erro 488;
- x2: momento em que o Módulo Servidor de Sessão detecta o erro 488;

- t2: intervalo de tempo decorrido para que o Módulo Servidor de Sessão acione o Módulo Balanceador de Chamadas;
- x3: momento em que o Módulo Balanceador de Chamadas é acionado;
- t3: intervalo de tempo decorrido para que o Sub-módulo Indicador de MG consulte a Lista de Prioridades e retorne o endereço IP do *Media Gateway* nela melhor classificado, ou seja, que possua menor VCB. No entanto, a Lista de Prioridades estava vazia, pelo que não houve retorno;
- x4: momento de término da chamada intentada.

A Tabela 6.11 apresenta os dados contidos nos respectivos *logs* das chamadas efetuadas no Teste 4, apresentados no Anexo B.4 e sumarizadas pela linha de tempo da Figura 6.9.

Tabela 6.11 – Momentos e Intervalos de Tempo do Teste 4.

	x1	t1	x2	t2	x3	t3	x4	<i>Media Gateway</i>
1 ^a	-	-	12:19:37	< 1s	12:19:37	< 1s	12:19:37	-
2 ^a	-	-	12:20:47	< 1s	12:20:47	< 1s	12:20:47	-
3 ^a	-	-	12:21:15	< 1s	12:21:15	< 1s	12:21:15	-
4 ^a	-	-	12:22:31	< 1s	12:22:31	< 1s	12:22:31	-
5 ^a	-	-	12:23:09	< 1s	12:23:09	< 1s	12:23:09	-

Assim como no Teste 2, no Teste 3, é comprovado nos *logs* constituintes do Anexo C.4, que neles não figuram o momento x1 e o intervalo t1. Ou seja, o início do registro da chamada no *log*, deu-se apenas a partir de x2, momento de detecção do erro 488, inerente a chamadas VoIP com participantes de *codecs* distintos.

Constitui-se peculiaridade do Teste 4, a Lista de Prioridades encontrar-se vazia, quando se intentou realizar as chamadas. Tal situação é ilustrada pela Tabela 6.12 e comprovada pelos *logs* contidos no Anexo C.4.

Tabela 6.12 – VCBs Disponibilizados na Lista de Prioridades Durante o Teste 4.

VCBs do <i>Media Gateway</i> 1							VCBs do <i>Media Gateway</i> 2						
	Antes		Durante		Depois			Antes		Durante		Depois	
	Hora	VCB	Hora	VCB	Hora	VCB		Hora	VCB	Hora	VCB	Hora	VCB
1ª	12:19:02	-	-	-	-	-	12:19:02	-	-	-	-	-	-
2ª	12:20:02	-	-	-	-	-	12:20:02	-	-	-	-	-	-
3ª	12:21:02	-	-	-	-	-	12:21:02	-	-	-	-	-	-
4ª	12:22:02	-	-	-	-	-	12:22:02	-	-	-	-	-	-
5ª	12:23:02	-	-	-	-	-	12:23:02	-	-	-	-	-	-
	Média:	-	Média:	-	Média:	-	Média:	-	Média:	-	Média:	-	-

Pela análise da Tabela 6.12, constata-se que não há registro de VCB, o que se faz coerente ao modelo de teste aplicado, que não contou com *Media Gateways* ativos, pois, ambos não possuíam o percentual mínimo de recurso computacional disponível requerido para figurarem na Lista de Prioridades, por utilizarem o *software aafire*. Por conseguinte, nenhum *Media Gateway* foi disponibilizado para o cálculo de VCB e a Lista de Prioridades permaneceu vazia. Contudo, se chamadas fossem encaminhadas a *Media Gateways* saturados, a qualidade seria altamente comprometida.

Apesar de não ter havido a efetivação de nenhuma das chamadas intentadas no presente teste, o Módulo Balanceador de Chamadas mostrou-se igualmente eficaz como nos demais testes implementados, pois aferiu os *Media Gateways* para o cálculo dos seus respectivos VCBs, bem como ao ser acionado pelo Módulo Servidor de Sessão, consultou a Lista de Prioridades, constatando-a vazia, pelo que não retornou nenhum *Media Gateway*.

6.5 Conclusão

O encaminhamento de chamadas a um *Media Gateway* permanece enquanto este dispuser de menor VCB, ou seja, somente quando outro *Media Gateway* obtiver menor VCB, as chamadas a ele serão encaminhadas. Assim, no caso de chamadas simultâneas, o VCB sofrerá alterações e, chamadas que necessitam de transcodificação serão direcionadas ao *Media Gateway* cujo VCB se encontre melhor classificado na Lista de Prioridades.

A inserção do *Media Gateway* na Lista de Prioridades é dinâmica. Desta forma, ao ultrapassar o limiar de VCB, ocorre que ele não constará na Lista de Prioridades, mesmo que esteja em funcionamento, transcodificando chamadas por exemplo. E tão logo atinja o limiar de VCB, voltará a figurar na Lista de Prioridades, o que é detectado pelo Módulo Balanceador de Chamadas; da mesma forma acontece com os demais que porventura venham a compor o cenário.

O *Media Gateway* 2, nos Testes 2 e 3, embora dotado de, praticamente, os mesmos percentuais de recursos computacionais disponíveis, apresenta, no entanto, pela normalização, em cada qual, respectivo VCB na fase prévia às chamadas “Antes”, como apresentado pela Tabela 6.13.

Tabela 6.13 – VCBs do *Media Gateway* 2 nos Testes 2 e 3.

Teste 2		Teste 3	
Antes		Antes	
Hora	VCB	Hora	VCB
11:24:02	59	11:43:01	12
11:27:02	59	11:46:01	13
11:29:02	59	11:49:01	12
11:32:02	59	11:52:01	12
11:35:02	59	11:55:01	12
Média:	59	Média:	12,2

Assim, pela Tabela 6.13, pode-se concluir a equivalência entre ambos os valores de média de VCBs, respectivamente 59 e 12,2, o que, conseqüentemente, demonstra a eficácia do Módulo Balanceador de Chamadas. Ainda, a variação de um respectivo VCB pode ser traduzida como a variabilidade da disponibilidade de recursos computacionais do *Media Gateway*.

Em suma, o Módulo Balanceador de Chamadas teve sua validação comprovada em todos os testes a que foi submetido, com ou sem transcodificação, bem como com *Media Gateways* ativos e inativos [39] [59].

Capítulo 7

Considerações Finais

Atualmente a *Internet* é imprescindível, pois muitos são os serviços proporcionados pela mesma. Assim sendo, várias foram as tecnologias que emergiram para usufruir dela. A VoIP, em especial, é a que tem despertado maior interesse, por propiciar comunicação de voz por meio da *Internet*, ou seja, utiliza as redes de dados para a transmissão de voz e, com isso, em contrapartida à PSTN, provê mobilidade e principalmente considerável redução de custos, o que lhe confere relevância na presente pesquisa. A VoIP tem-se disseminado gradativamente e está presente em aplicações de *Internet* cotidianas, por exemplo, no *MSN* e no *Skype* [4] [17].

Com isso, a VoIP tem sido adotada por muitas instituições e empresas que gradativamente tem substituído os seus equipamentos de PABX por Soluções IP, como PBX VoIP. Desta forma, tem-se a possibilidade de direcionamento das chamadas aos respectivos destinos, quer pelo encaminhamento via rede de dados da própria instituição quer de operadoras VoIP [10].

Com a utilização de Soluções IP, a exemplo da VoIP, a disponibilização do serviço, bem como seu gerenciamento, podem ser realizados pela própria empresa ou instituição que disponibiliza a solução, sem a necessidade de terceirização. Com isto, ganha-se autonomia e funcionalidade, a exemplo da utilização de um Painel de Controle de Administração do sistema via *web*, o que não requer conhecimentos técnicos inerentes à sua manutenção, isto é, inclusão, exclusão, atualização, pelos usuários.

Ainda, pode-se aproveitar o corpo técnico para o desempenho de tarefas que venham a ser solicitadas [53].

Atualmente, num mundo globalizado, as empresas tem buscado novas soluções para redução de seus custos operacionais. Uma das soluções é a econômica e as características intrínsecas que a VoIP proporciona. Isto é, ao invés de usar a comunicação de voz via telefone convencional, por meio da PSTN, pode-se usar a rede de dados, por meio da *Internet* ou das *Intranets*.

Assim, o uso da comunicação de voz via rede de dados se faz com o uso da VoIP, em que, para tal, é necessário o uso de aplicativos do tipo *softphone* ou *hardware* do tipo Telefone IP em cada extremidade do enlace de comunicação. Uma chamada VoIP, quando estabelecida, necessariamente é uma comunicação que faz uso da VoIP, que em cada uma das extremidades faz-se necessária a codificação da voz, transformando-a em pacotes de áudio. Na sequência, estes pacotes são transmitidos a outra extremidade e então decodificados.

A codificação e decodificação é feita por meio de *codecs*. Os *codecs* servem principalmente para diminuir o uso da banda da rede de dados disponível para o serviço. Numa comunicação em que se usa VoIP, há a possibilidade de que não haja compatibilidade entre os *codecs* das extremidades do enlace da comunicação e, com isto, impossibilitando propriamente a comunicação, necessitando então recorrer aos *Media Gateways*, que são responsáveis em fazer a transcodificação.

A necessidade de transcodificação de chamadas VoIP demanda consideráveis recursos de máquina, podendo exauri-la em número elevado. Desta forma, pode ocasionar o descarte de chamadas, bem como a degradação da qualidade das chamadas em curso. Assim, num cenário VoIP, é ideal que se tenha vários *Media Gateways*, e que as chamadas sejam distribuídas entre eles de forma proporcional as suas capacidades de recursos.

Portanto, uma das principais contribuições desta pesquisa é o Módulo Balanceador de Chamadas, responsável por balancear as chamadas VoIP que necessitam de transcodificação, desde que não possuam o *codec* adequado em alguma das extremidades. Assim sendo, o balanceamento de chamadas VoIP a transcodificar é implementado por meio do Módulo Balanceador de Chamadas, composto pelos Submódulos Acionador, Atualizador, Indicador de MG e Verificador.

O Módulo Balanceador de Chamadas monitora os recursos dos *Media Gateways* usando as primitivas do protocolo SNMP. Esta concepção é prática, simples e funcional, pois o Módulo Balanceador de Chamadas e o Módulo Servidor de Sessão se encontram no Servidor de Chamadas, e todo o sistema pode ser implementado usando *software* livre, a exemplo do que foi usado na validação da proposta apresentada nesta dissertação.

Para a validação da proposta, foram implementados e usados um Servidor de Chamadas e dois *Media Gateways*. Os recursos relacionados a memória e processador de máquina foram usados para exemplificar os recursos monitoráveis dos *Media Gateways*. A validação do Módulo Balanceador de Chamadas deu-se pela realização de diferentes testes no ambiente VoIP, abordados no Capítulo 6.

Os testes levaram em consideração participantes possuindo *codecs* idênticos, distintos e dois *Media Gateways*. Tais testes demonstraram a eficácia do Módulo Balanceador de Chamadas, o que vai ao encontro do objetivo proposto nesta dissertação, ou seja, prover o balanceamento de chamadas VoIP a transcodificar.

Um outro ponto positivo do Módulo Balanceador de Chamadas é a viabilidade mercadológica, haja vista o vertiginoso emprego da VoIP, pois há um crescimento na migração das comunicações para Soluções IP, processo este irreversível, constituindo um campo promissor de pesquisas. Com isto, sua flexibilidade peculiar torna o Módulo Balanceador de Chamadas passível de customização a critério das necessidades [59] [76].

Para a VoIP, ainda não há padrões predominantes, isto inclui *hardware*, protocolos e virtualmente cada aspecto do sistema. No final, a VoIP é um grande avanço do sistema de telefonia atual em termos de eficiência, custo e flexibilidade. Como qualquer tecnologia emergente, alguns desafios precisam ser vencidos, mas está claro que será aprimorada a cada dia até que, futuramente, substitua o sistema de telefonia atual. Haja vista isto, é notório que, em curto prazo, o crescimento exponencial da *Internet* e a disseminação da VoIP, tornem imprescindível a utilização do Módulo Balanceador de Chamadas, concebido e desenvolvido nesta dissertação, que é a parte principal da solução ao balanceamento de chamadas VoIP a transcodificar.

Como sugestões para melhoria desta pesquisa, tem-se a execução de testes exaustivos para obtenção de estatísticas, considerando os diversos *codecs* existentes, as

diferentes configurações de *Media Gateways*, atentando-se principalmente as arquiteturas de processadores, memória e barramento de entrada/saída. Outra sugestão para melhorar o Módulo Balanceador de Chamadas é uma melhor definição da fórmula para o cálculo do VCB, considerando a normalização dos diversos recursos possíveis existentes, associado ao MOS dos *codecs* com o consumo de banda de rede.

Como trabalho futuro, é sugerido a implementação do Módulo Balanceador de Chamadas para aplicações multimídia, considerando a interatividade e todos os aspectos relacionados aos *codecs* de áudio e vídeo, bem como, a compactação de dados discretos.

Referências

- [1] AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. **Anatel esclarece uso de VoIP para oferta de serviço de voz**. Brasília, DF, 9 nov. 2005. Disponível em: <<http://www.anatel.gov.br/>>. Acesso em: 10 out. 2008.
- [2] NANINI, L. **IPS in-line**: a solução mais segura para comunicações via VoIP. 12 fev. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 10 out. 2008.
- [3] TANENBAUM, A. S. **Redes de computadores**. 4. ed. Rio de Janeiro: Elsevier, 2003.
- [4] CHIANG, W.-H.; XIAO, W.-C.; CHOU, C.-F. A performance study of VoIP applications: MSN vs. Skype. In: FIRST MULTIMEDIA COMMUNICATIONS WORKSHOP (MULTICOMM), Turkey. **Proceedings...** p. 13-18, June 2006.
- [5] FENÓLIO, J. **Comunicação de Voz** - o 'patinho feio' da TIC?. 14 jun. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 10 out. 2008.
- [6] HALL, J. M. 2007. Beachhead: Waysmall. **Linux Journal**. Specialized Systems Consultants, Inc. Seattle, WA, USA. vol. 2007, n. 155, p. 14, Mar. 2007.
- [7] BARRETO, A. B. **Uma arquitetura de telefonia IP para proteção da mídia fim-a-fim**. 2007. 155 f. Dissertação (Mestrado) – Curso de Engenharia e Computação, Instituto Tecnológico de Aeronáutica, São José dos Campos, 2007.
- [8] NOVAKOSKI, A. **Tecnologia IP**: muito além da redução de custos. 27 fev. 2008. Disponível em: <<http://www.itweb.com.br/>>. Acesso em: 13 out. 2008.

- [9] ALJ, T.; GREGOIRE, J.-C., Admission control and bandwidth management for VLANs. **IEEE WORKSHOP ON HIGH PERFORMANCE SWITCHING AND ROUTING**, Dallas, p. 130-134, May 2001.
- [10] LIRA, P. **Telefonia IP não deve ser utilizada como telefonia convencional**. 19 nov. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 13 out. 2008.
- [11] SANTOS, M. N. **Medidas de qualidade de voz em redes IP**. 2006. 83 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Paraná, Curitiba, 2006.
- [12] SILVA JUNIOR, J. M. **Uma aplicação de voz sobre IP baseada no session initiation protocol**. 2003. 115 f. Dissertação (Mestrado) – Pós-Graduação em Engenharia Elétrica, Universidade Federal de Pernambuco, Recife, 2003.
- [13] KOK, C. W.; BEG, M. S. Simple IP Subnet VLAN Implementation. In: IEEE INTERNATIONAL CONFERENCE ON NETWORKS (ICON). **Proceedings...** IEEE Computer Society, Washington, DC, USA. p. 160-165, 10-12 October 2001.
- [14] MADEIRA, F. **Qualidade de voz e MOS**. 03 dez. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 12 out. 2008.
- [15] ALBERT, A. T. **Uma proposta para a descrição e busca por recursos utilizando metadados XML/RDF em redes peer-to-peer**. 2002. 102 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2002.
- [16] ZAVE, P.; CHEUNG, E. 2006. Compositional control of IP media. In: INTERNATIONAL CONFERENCE ON EMERGING NETWORKING EXPERIMENTS AND TECHNOLOGIES (ACM CoNEXT Conference – Lisboa, Portugal). **Proceedings...** New York, p. 1-12, December 04-07, 2006.
- [17] BONFIGLIO, D.; MELLIA, M.; MEO, M.; ROSSI, D.; TOFANELLI, P. Revealing skype traffic: when randomness plays with you. In: CONFERENCE ON APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATIONS (SIGCOMM'07 – Kyoto, Japan). **Proceedings...** New York, p. 37-48, 27-31 August 2007.

[18] RAMOS, J. R. S. **Série de artigos sobre VoIP (4)**, dimensionamento VoIP (WAN) - Parte 03. 23 mai. 2006. Disponível em: <<http://www.wirelessbrasil.org/>>. Acesso em: 12 nov. 2008.

[19] THERMOS, P.; HADSALL, G. Vulnerabilities in SOHO VoIP gateways. **WORKSHOP OF THE 1ST INTERNATIONAL CONFERENCE ON SECURITY AND PRIVACY FOR EMERGING AREAS IN COMMUNICATION NETWORKS**, Athens, Greece, p. 236-245, 5-9 September 2005.

[20] INTERNATIONAL TELECOMMUNICATION UNION. **ITU-T Recommendation P.800: Methods for subjective determination of transmission quality**. 30 ago. 1996. Disponível em: <<http://www.itu.int/>>. Acesso em: 03 dez. 2008.

[21] VELLO, P. **VoIP derrubando fronteiras**. 01 fev. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 11 out. 2008.

[22] MONTEIRO, T. L. **Solução de telefonia IP em uma rede corporativa**. 2003. 185 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2003.

[23] SALLES, D. **Convergência digital reduz custos nas empresas**. 17 ago. 2007. Disponível em: <<http://www.convergenciadigital.com.br/>>. Acesso em: 13 out. 2008.

[24] Net-SNMP. Disponível em: <<http://www.net-snmp.org/>>. Acesso em: 15 mar. 2008.

[25] SOUSA, J. P.; CARRAPATOSO, E. **Uma arquitetura IPtel baseada no protocolo SIP**. 6ª CONFERÊNCIA SOBRE REDES DE COMPUTADORES (CRC'2003 – Bragança, Portugal). 29-30 set. 2003. Disponível em: <<http://www.fcn.pt/crc2003/>>. Acesso em: 19 out. 2008.

[26] KANG, T-G.; BAE, H-J.; KIM, D-Y.; KIM, D-U. SIP/SDP signaling of media gateway with transcoding function in converged network. **THE 6TH INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGY**. vol. 2, p. 842-845, 2004.

[27] LENZINI, L.; MINGOZZI, E.; STEA, G. Bandwidth and latency analysis of modified deficit round robin scheduling algorithms. In: 1ST INTERNATIONAL

CONFERENCE ON PERFORMANCE EVALUATION METHODOLOGIES AND TOOLS (Pisa, Italy). **Proceedings...** vol. 180. ACM, New York, NY, 41, 11-13 October 2006.

[28] Tech-invite. **ABNF grammar for SDP (RFC 4566)**. 25 nov. 2008. Disponível em: <<http://www.tech-invite.com/Ti-sdp-abnf.html>>. Acesso em: 09 dez. 2008.

[29] OLIVEIRA, A. C. **VoIP: o futuro da comunicação é agora**. 27 abr. 2009. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 29 abr. 2009.

[30] KUROSE, J.; ROSS, K. **Redes de computadores e a internet - uma abordagem top-down**. 3. ed. São Paulo: Pearson Addison Wesley, 2006.

[31] LEINWAND, A.; CONROY, K. F. **Network management: a practical perspective**. 2. ed. Addison Wesley Longman Publishing Co., Inc, 1995.

[32] MICHAELIS. **Moderno dicionário da língua portuguesa**. Disponível em: <<http://michaelis.uol.com.br/>>. Acesso em 01 dez. 2008.

[33] CERUTTI, F. A. **Uma abordagem de plano de controle para QoS dinâmica em fluxos de voz nas redes IP**. 2006. 227 f. Tese (Doutorado) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2006.

[34] SILVEIRA, V. S. **A tecnologia do momento é o VoIP**. 10 dez. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 29 dez. 2008.

[35] BRASIL, E. F. **Adaptação de codificador CELP à transmissão de voz sobre IP**. 2006. 77 f. Trabalho de Conclusão de Curso (Graduação) – Departamento de Eletrônica e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

[36] SKYPE. Disponível em: <<http://ww.skype.com/>>. Acesso em: 01 mar. 2008.

[37] PEREIRA, R. **O que é o PABX virtual?**. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 23 out. 2008.

[38] GONÇALVES, F. E. **Building telephony systems with OpenSER**. 1. ed. Birmingham, UK: Packt Publishing, 2008.

[39] CUNHA, I. T.; BARBAR, J. S.; FAINA, L. F.; MENDES, F. B. C.; ASHIDANI, P. J.; TEIXEIRA, M. A. Balanceamento de chamadas transcodificadas VoIP. In: XL SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2008. **Anais...** João Pessoa – PB, 02-05 set. 2008.

[40] ALBUQUERQUE A. A. **Balanceamento de chamadas E.164 VoIP para a rede de telefonia pública**. 2007. 141 f. Dissertação (Mestrado) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

[41] ASTERISK. **The open source PBX & telephony platform**. Disponível em: <<http://www.asterisk.org/>>. Acesso em: 02 abr. 2008.

[42] CISCO SYSTEM, Inc. **Internetworking technologies handbook**, 4. ed. Cisco Press, 2003.

[43] BRITO, P. N. **O VoIP é simples e qualquer empresa pode usar**. 30 ago. 2007. Disponível em: <<http://www.fndc.com.br/>>. Acesso em 15 out. 2008.

[44] MCGINNIS, M. **How to prepare your network for multiple codecs**. 13 abr. 2007. Disponível em: <<http://telephonyonline.com/>>. Acesso em: 26 jun. 2008.

[45] OpenSER. **The open source SIP server**. Disponível em: <<http://www.openser.org/>>. Acesso em 17 mai. 2008.

[46] TransNexus, Inc. **Performance Benchmark Test for Asterisk B2BUA**. 13 nov. 2007. Disponível em: <<http://www.transnexus.com/>>. Acesso em: 02 abr. 2008.

[47] TransNexus, Inc. **Performance benchmark test for asterisk B2BUA**. 21 out. 2008. Disponível em: <<http://www.transnexus.com/>>. Acesso em: 03 jan. 2009.

[48] NAZARIO, D. L. **Protótipo de um sistema de telefonia IP para LANs baseado no padrão SIP**. 2003. 47 f. Trabalho de Conclusão de Curso (Graduação) – Centro de

Ciências Exatas e Naturais, Curso de Ciências da Computação, Universidade Regional de Blumenau, Blumenau, 2003.

[49] YU, J.; AL-AJARMEH, I. Call admission control and traffic engineering of VoIP. In: SECOND INTERNATIONAL CONFERENCE ON DIGITAL TELECOMMUNICATIONS (ICDT'07). **Proceedings...** IEEE Computer Society, Washington, DC, 1-5 July 2007.

[50] POSTEL, J. **Transmission control protocol** – RFC 793. Set. 1981, atualizada pela RFC 3168. Disponível em: <<http://www.ietf.org/rfc/rfc793.txt>>. Acesso em: 15 mar. 2008.

[51] HESS, J. **aafire(1)** - Linux man page. Disponível em: <<http://linux.die.net/man/1/aafire>>. Acesso em: 03 out. 2008.

[52] DAVIDSON, J. et al. **Fundamentos de VoIP**. 2. ed. Porto Alegre: Bookman, 2008.

[53] BIGIO, D. **Outsourcing de telefonia IP para empresas de médio e grande porte**. 05 jul. 2007. Disponível em: <<http://www.ipnews.com.br/>>. Acesso em: 21 out. 2008.

[54] SCHULZRINNE, H.; CASNER, S.; FREDERICK, R.; JACOBSON, V. **RTP: a transport protocol for real-time applications** – RFC 1889. Jan. 1996. Disponível em: <<http://www.ietf.org/rfc/rfc1889.txt>>. Acesso em 12 out. 2008.

[55] CUNHA, I. T.; BARBAR, J. S. **Esquemas de QoS para balanceamento de chamadas em servidores VoIP**. In: 1º Workshop de dissertações do PPGCC/FACOM/UFU. Uberlândia, 29-30 nov. 2007.

[56] COUNTERPATH CORPORATION. **X-Lite**. Disponível em: <<http://www.counterpath.net>>. Acesso em: 23 out. 2007.

[57] CISNEIROS, H. **Programando em shell-script**. Disponível em: <http://www.devin.com.br/shell_script/>. Acesso em 04 abr. 2008.

[58] ASHIDANI, P. J.; BARBAR, J. S.; CUNHA, I. T.; FAINA, L. F.; SOUZA, M. R. Proposta de framework para autenticação de remetente. In: THE THIRD INTERNATIONAL CONFERENCE OF FORENSIC COMPUTER SCIENCE. **Proceedings...** Rio de Janeiro: ABEAT, vol. 3. p. 112-116, 2008.

[59] CUNHA, I. T.; BARBAR, J. S.; FAINA, L. F.; MENDES, F. B. C.; ASHIDANI, P. J. Balanceamento de chamadas VoIP transcodificadas. In: 7TH INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM. **Proceedings...** Foz do Iguaçu, 2008.

[60] SLACKWARE LINUX. **The slackware linux project**. Disponível em: <<http://www.slackware.com/>>. Acesso em 02 fev. 2008.

[61] SILVA, C. N.; Ramos, L. B. Análise ergonômica de um ambiente de comunicação via WEB. 1999. 167 f. Trabalho de Conclusão de Curso (Graduação) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 1999.

[62] VIANNA, B. A.; MOURA, N. T.; ALBUQUERQUE, C. V. N.; Rebello, V. E. F.; BOERES, C. adaMOS: MOS-adaptive VoIP sources. In: 12TH BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB (WebMedia'06 – Natal, Rio Grande do Norte, Brazil, 19-22 nov. 2006). **Proceedings...** ACM, New York, NY, vol. 192, p. 223-232, 2006.

[63] OLCHIK, A. **Segurança em voz sobre IP**. Disponível em: <<http://www.teleco.com.br/>>. Acesso em 28 nov. 2008.

[64] THORNE, D. J. 2001. VoIP - the access dimension. **BT Technology Journal**. Kluwer Academic Publishers Hingham, MA, USA, vol. 19, n. 2, p. 33-43, Apr. 2001.

[65] SCHWARZ, B. 2004. Asterisk open-source PBX system. **Linux Journal**. Specialized Systems Consultants, Inc. Seattle, WA, USA. vol. 2004, n. 118, p. 6, Feb. 2004.

[66] TURNER, J. 2006. Creating a home PBX using asterisk and digium. **Linux Journal**. Specialized Systems Consultants, Inc. Seattle, WA, USA. vol. 2006, n. 141, p. 1. Jan. 2006.

[67] CERT.br. **Cartilha de segurança para internet**, versão 3.1– São Paulo: Comitê Gestor da Internet no Brasil, 2006. Disponível em: <<http://cartilha.cert.br/>>. Acesso em: 23 dez. 2008.

[68] QADEER, M. A.; IMRAN, A. Asterisk voice exchange: an alternative to conventional EPBX. In: INTERNATIONAL CONFERENCE ON COMPUTER AND ELECTRICAL ENGINEERING (ICCEE 2008). **Proceedings...** IEEE Computer Society Washington, DC, USA, vol., n., p. 652-656, 20-22 Dec. 2008.

[69] CHAVA, K. S.; HOW, J. Integration of open source and enterprise IP PBXs. In: 3RD INTERNATIONAL CONFERENCE ON TESTBEDS AND RESEARCH INFRASTRUCTURE FOR THE DEVELOPMENT OF NETWORKS AND COMMUNITIES (TridentCom 2007). **Proceedings...** vol., n., p. 1-6, 21-23 May 2007.

[70] ALAM, M. Z.; BOSE, S.; RAHMAN, M. M.; AL-MUMIN, M. A. Small office PBX using voice over internet protocol (VOIP). In: THE 9TH INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGY. **Proceedings...** vol.3, n., p. 1618-1622, 12-14 Feb. 2007.

[71] 3CX. **O que significam os termos FXS e FXO?**. Disponível em: <<http://www.3cx.com.br/>>. Acesso em 29 out. 2008.

[72] LINDQVIST, J.; KOMU, M. Cure for spam over internet telephony. In: 4TH IEEE CONSUMER COMMUNICATIONS AND NETWORKING CONFERENCE (CCNC 2007). **Proceedings...** vol., n., p. 896-900, Jan. 2007.

[73] DRYBURGH, L.; HEWETT, J. **Signaling system No. 7 (Ss7/C7)**: protocol, architecture, and applications. Cisco Press, 2003.

[74] FESSI, A.; NIEDERMAYER, H.; KINKELIN, H.; CARLE, G. A cooperative SIP infrastructure for highly reliable telecommunication services. In: 1ST INTERNATIONAL CONFERENCE ON PRINCIPLES, SYSTEMS AND APPLICATIONS OF IP TELECOMMUNICATIONS (IPTComm'07 – New York City, New York, 19-20 July 2007). **Proceedings...** ACM, New York, NY, p. 29-38, 2007.

[75] GOODE, B. Voice over internet protocol (VoIP). In: IEEE. **Proceedings...** vol. 90, n. 9, p. 1495-1517, Sep. 2002.

[76] MARTIN, M. V.; HUNG, P. C. K. Towards a security policy for VoIP applications. In: CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING. **Proceedings...** vol., n., p. 65-68, 1-4 May 2005.

[77] AHMED, M.; MANSOR, A. M. CPU dimensioning on performance of Asterisk VoIP PBX. In: 11TH COMMUNICATIONS AND NETWORKING SIMULATION SYMPOSIUM (CNS'08 – Ottawa, Canada, 14-17 April 2008). **Proceedings...** ACM, New York, NY, p. 139-146, 2008.

[78] NETWORK WORLD. **Conheça as 14 maiores vulnerabilidades do VoIP.** Disponível em: <<http://www.computerworld.com.br/>>. Acesso em: 15 dez. 2008.

[79] MEGGELEN, J. V.; SMITH, J.; MADSEN, L. **Asterisk: the future of telephony.** 2. ed. USA: O'Reilly Media, Inc. 2007.

[80] SPENCER, M.; ALLISON, M.; RHODES, C. **The asterisk handbook version 2.** Digium, Inc. 2003.

[81] GONÇALVES, F. E. A. **Asterisk PBX – guia de configuração.** 3. ed. Florianópolis: V. Office, 2005.

[82] PINHÃO, C. M. B. Estratégias de otimização da interface Abis em redes de telefonia celular GSM. 2006. 119 f.. Dissertação (Mestrado) – Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

[83] YATE. **Yet another telephony engine.** Disponível em: <<http://www.yate.null.ro>>. Acesso em: 03 jan. 2009.

[84] FreeSWITCH. **Communication consolidation.** Disponível em: <<http://www.freeswitch.org/>>. Acesso em: 03 jan. 2009.

[85] SEMS. **The SIP express media server.** Disponível em: <<http://www.iptel.org/sems/>>. Acesso em: 03 jan. 2009.

[86] LINKSYS BY CISCO. Disponível em: <<http://www.linksys.com/>>. Acesso em: 28 nov. 2008.

[87] CISCO SYSTEMS, INC. Disponível em: <<http://www.cisco.com/>>. Acesso em: 11 nov. 2008.

[88] KHASNABISH, B. **Implementing voice over IP**. 1. ed. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[89] DIGIUM. **The asterisk company**. Disponível em: <<http://www.digium.com/>>. Acesso em 12 nov. 2008.

[90] CAMPOS, A. S. **Telefonia: a convergência de voz em dados**. Disponível em: <<http://www.teleco.com.br/>>. Acesso em: 23 dez. 2008.

[91] THERNOS, P.; TAKANEN, A. **Securing VoIP networks: threats, vulnerabilities, and countermeasures**. Addison-Wesley Professional, 2008.

[92] RAMOS, J. R. S. **Métodos de codificação de voz**. 09 jan. 2006. Disponível em: <<http://www.wirelessbrasil.org/>>. Acesso em: 19 dez. 2008.

[93] LAMAS, R. M. L. **Avaliação de codificadores de voz em ambiente VoIP**. 2005. 84 f. Trabalho de Conclusão de Curso (Graduação) – Departamento de Eletrônica e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

[94] TAFNER, M. A., **Reconhecimento de Palavras Faladas Isoladas usando Redes Neurais Artificiais**. 1996. 102 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1996.

[95] RUSCHEL, O. T. **Princípios da comunicação digital**. 1. ed. Porto Alegre: EDIPUCRS, 1996.

[96] LAMARE, R. C. **CELP coder for speech signals of brazilian portuguese**. 1999. 62 f. Trabalho de Conclusão de Curso (Graduação) – Departamento de Eletrônica e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1999.

[97] SILVA, T. F. **Técnicas de codificação de sinais de fala**. 2006. 69 f. Trabalho de Conclusão de Curso (Graduação) – Departamento de Ciência da Computação, Universidade Estadual de Londrina, Londrina, 2006.

[98] MENDONÇA, F. D. **SNMP: estrutura, protocolo e aplicações**. 1998. Disponível em: <<http://www.gta.ufrj.br/>>. Acesso em: 10 jan. 2009.

[99] BREIT, L. **SNMP overhead and performance impact**. 03 fev. 2003. Disponível em: <<http://ietfreport.isoc.org/>>. Acesso em: 10 jan. 2009.

[100] ALENCAR, V. F. S. Atributos e domínios de interpolação eficientes em reconhecimento de voz distribuído. 2005. 114 f. Dissertação (Mestrado) – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

[101] ANTAS, R. B. Z.; FERNANDES, N. C.; TAVEIRA D. M.; CAMPISTA, M. E. M.; COSTA, L. H. M. K.; DUARTE, O. C. M. B. Análise da qualidade de voz em uma rede ad hoc comunitária. In: XI WORKSHOP DE GERÊNCIA E OPERAÇÃO DE REDES E SERVIÇOS (WGRS'2006, 24º SBRC). **Proceedings...** Curitiba, 2006.

[102] SANTOS FILHO, F. H. C. **Implementação de um sistema de iniciação de sessão multimídia para a plataforma Linux**. 2005. 101 f. Dissertação (Mestrado) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2005.

[103] SCHUDEL, G.; SMITH, D. **Router security strategies: securing IP network traffic planes (networking technology: security)**. 1. ed. Cisco Press, 2007.

[104] STALLINGS, W. **SNMP, SNMPv2, and CMIP: the practical guide to network management standards**. 1. ed. USA: Addison-Wesley Publishing Company, Inc., 1993.

[105] TOREZAN, E. L. D. **Implementação de funções avançadas em um gateway VoIP**. 2006. 56 f. Trabalho de Conclusão de Curso (Graduação) – Faculdade de Engenharia de Telecomunicações, Universidade Regional de Blumenau, Blumenau, 2006.

[106] GITEL TELECOMUNICAÇÕES. **PBX e PABX**. Disponível em: <<http://www.gitel.com.br/>>. Acesso em: 12 out. 2008.

- [107] CARNEIRO, L. F.; JÚNIOR, N. A. **Roteadores e segurança em redes**. 22 dez. 1999. Disponível em: <<http://mesonpi.cat.cbpf.br/naj/>>. Acesso em 17 dez. 2008.
- [108] PICCININ, F. D. Q. **Arquitetura OSI**. 19 jan. 1999. Disponível em: <<http://www-usr.inf.ufsm.br/~piccinin/>>. Acesso em 04 dez. 2008.
- [109] PINHEIRO, B. O. **Voz sobre IP utilizando asterisk**. 2005. 80 f. Trabalho de Conclusão de Curso (Especialização) – Departamento de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2005.
- [110] HANDLEY, M.; JACOBSON, V.; PERKINS, C. **SDP: session description protocol – RFC 4566**. Jul. 2006. Disponível em: <<http://www.ietf.org/rfc/rfc4566.txt>>. Acesso em 29 out. 2008.

Anexo A - Código Fonte dos Arquivos de Configuração

A.1 - *openserctlrc*

Nesta seção está disponível o código-fonte do arquivo de configuração *openserctlrc*, utilizado no Módulo Servidor de Sessão.

```
# $Id: openserctlrc 4133 2008-05-08 10:39:51Z miconda $
#
# The OpenSER configuration file for the control tools.
#
# Here you can set variables used in the openserctl and openserdbctl
# scripts. Per default all variables here are commented out, the
# control tools
# will use their internal default values.

## your SIP domain
SIP_DOMAIN=200.131.189.90

## database type: MYSQL, PGSQL, DB_BERKELEY, or DBTEXT, by default
## none is loaded
# If you want to setup a database with openserdbctl, you must at least
# specify
# this parameter.
DBENGINE=MYSQL

## database host
DBHOST=localhost

## database name
DBNAME=openser

# database path used by dbtext or db_berkeley
# DB_PATH="/usr/local/etc/openser/dbtext"

## database read/write user
```

```

DBRWUSER=openser

## password for database read/write user
DBRWPW="openserrw"

## database read only user
DBROUSER=openserro

## password for database read only user
DBROPW=openserro

## database super user
DBROOTUSER="root"

# user name column
# USERCOL="username"

# SQL definitions
# If you change this definitions here, then you must change them
# in db/schema/entities.xml too.
# FIXME

# FOREVER="2020-05-28 21:32:15"
# DEFAULT_ALIASES_EXPIRES=$FOREVER
# DEFAULT_Q="1.0"
# DEFAULT_CALLID="Default-Call-ID"
# DEFAULT_CSEQ="13"
# DEFAULT_LOCATION_EXPIRES=$FOREVER

# Program to calculate a message-digest fingerprint
# MD5="md5sum"

# awk tool
# AWK="awk"

# grep tool
# GREP="egrep"

# sed tool
# SED="sed"

# Describe what additional tables to install. Valid values for the
variables
# below are yes/no/ask. With ask (default) it will interactively ask
the user
# for an answer, while yes/no allow for automated, unassisted
installs.
#

# If to install tables for the modules in the EXTRA_MODULES variable.
# INSTALL_EXTRA_TABLES=ask

# If to install presence related tables.
# INSTALL_PRESENCE_TABLES=ask

# If to install SERWEB related tables.

```

```

# When you don't know what purpose this tables have, don't change
this.
# If you choose to install this tables, you must also activate the
HAS_SERWEB
# variable, otherwise some setup script actions will not work.
# INSTALL_SERWEB_TABLES=ask

# Define what module tables should be installed.
# If you use the postgres database and want to change the installed
tables, then you
# must also adjust the STANDARD_TABLES or EXTRA_TABLES variable
accordingly in the
# openserdbctl.base script.

# openser standard modules
# STANDARD_MODULES="standard acc lcr domain group permissions
registrar usrloc msilo
#                  alias_db uri_db speeddial avpops auth_db pdt
dialog dispatcher"

# openser extra modules
# EXTRA_MODULES="imc cpl siptrace domainpolicy carrierroute"

## type of aliases used: DB - database aliases; UL - usrloc aliases
## - default: none
ALIASES_TYPE="DB"

## control engine: FIFO or UNIXSOCK
## - default FIFO
CTLENGINE="FIFO"

## path to FIFO file
#OSER_FIFO="FIFO" comentei este aqui
OSER_FIFO="/tmp/openser_fifo"

## check ACL names; default on (1); off (0)
VERIFY_ACL=1

## ACL names - if VERIFY_ACL is set, only the ACL names from below
list
## are accepted
ACL_GROUPS="local ld int voicemail free-pstn"

## presence of serweb tables - default "no"
# HAS_SERWEB="yes"

## verbose - debug purposes - default '0'
VERBOSE=1

## do (1) or don't (0) store plaintext passwords
## in the subscriber table - default '1'
# STORE_PLAINTEXT_PW=0

## OPENSER START Options
## PID file path - default is: /var/run/openser.pid
# PID_FILE=/var/run/openser.pid

```

```
## Extra start options - default is: not set
# example: start openser with 64MB share memory: STARTOPTIONS="-m 64"
# STARTOPTIONS=
```

A.2 - *openser.cfg*

Neste tópico está disponível o código-fonte do arquivo de configuração *openser.cfg*, utilizado no Módulo Servidor de Sessão.

```
# http://www.kamailio.net/dokuwiki/doku.php/pseudovariables:1.3.x
#
# $Id: openser.cfg 4083 2008-04-24 19:26:11Z miconda $
#
# OpenSER basic configuration script
#   by Anca Vamanu <anca@voice-system.ro>
#
# Please refer to the Core CookBook at
http://www.openser.org/dokuwiki/doku.php
# for a explanation of possible statements, functions and parameters.
#

##### Global Parameters #####

#debug=3 comentei aqui
debug=9
log_stderr=no
log_facility=LOG_LOCAL0

fork=yes
children=4

/* uncomment the following lines to enable debugging */
#debug=6
#fork=no
#log_stderr=yes

/* uncomment the next line to disable TCP (default on) */
#disable_tcp=yes

/* uncomment the next line to enable the auto temporary blacklisting
of
   not available destinations (default disabled) */
#disable_dns_blacklist=no

/* uncomment the next line to enable IPv6 lookup after IPv4 dns
   lookup failures (default disabled) */
#dns_try_ipv6=yes

/* uncomment the next line to disable the auto discovery of local
aliases
   based on revers DNS on IPs (default on) */
```

```

#auto_aliases=no

/* uncomment the following lines to enable TLS support (default off)
*/
#disable_tls = no
#listen = tls:your_IP:5061
#tls_verify_server = 1
#tls_verify_client = 1
#tls_require_client_certificate = 0
#tls_method = TLSv1
#tls_certificate = "/usr/local/etc/openssl/tls/user/user-cert.pem"
#tls_private_key = "/usr/local/etc/openssl/tls/user/user-privkey.pem"
#tls_ca_list = "/usr/local/etc/openssl/tls/user/user-calist.pem"

port=5060

/* uncomment and configure the following line if you want openssl to
bind on a specific interface/port/proto (default bind on all
available) */
#listen=udp:192.168.1.2:5060

##### Modules Section #####

#set module path
mpath="/usr/local/lib/openssl/modules/"

/* uncomment next line for MySQL DB support */
loadmodule "mysql.so"
loadmodule "sl.so"
loadmodule "tm.so"
loadmodule "rr.so"
loadmodule "maxfwd.so"
loadmodule "usrloc.so"
loadmodule "registrar.so"
loadmodule "textops.so"
loadmodule "mi_fifo.so"
loadmodule "uri_db.so"
loadmodule "uri.so"
loadmodule "xlog.so"
loadmodule "acc.so"

#módulo habilitado para executar funções externas
loadmodule "exec.so"

/* uncomment next lines for MySQL based authentication support
NOTE: a DB (like mysql) module must be also loaded */
loadmodule "auth.so"
loadmodule "auth_db.so"
/* uncomment next line for aliases support
NOTE: a DB (like mysql) module must be also loaded */
#loadmodule "alias_db.so"
/* uncomment next line for multi-domain support
NOTE: a DB (like mysql) module must be also loaded
NOTE: be sure and enable multi-domain support in all used modules
(see "multi-module params" section) */
#loadmodule "domain.so"
/* uncomment the next two lines for presence server support
NOTE: a DB (like mysql) module must be also loaded */

```

```

#loadmodule "presence.so"
#loadmodule "presence_xml.so"

# ----- setting module-specific parameters -----

# ----- mi_fifo params -----
modparam("mi_fifo", "fifo_name", "/tmp/openser_fifo")
#modparam("mi_fifo", "fifo_name", "FIFO")

# ----- rr params -----
# add value to ;lr param to cope with most of the UAs
modparam("rr", "enable_full_lr", 1)
# do not append from tag to the RR (no need for this script)
modparam("rr", "append_fromtag", 0)

# ----- rr params -----
modparam("registrar", "method_filtering", 1)
/* uncomment the next line to disable parallel forking via location */
# modparam("registrar", "append_branches", 0)
/* uncomment the next line not to allow more than 10 contacts per AOR
*/
#modparam("registrar", "max_contacts", 10)

# ----- uri_db params -----
/* by default we disable the DB support in the module as we do not
need it
    in this configuration */
modparam("uri_db", "use_uri_table", 0)
modparam("uri_db", "db_url", "")

# ----- acc params -----
/* what sepcial events should be accounted ? */
modparam("acc", "early_media", 1)
modparam("acc", "report_ack", 1)
modparam("acc", "report_cancels", 1)
/* by default ww do not adjust the direct of the sequential requests.
    if you enable this parameter, be sure the enable "append_fromtag"
    in "rr" module */
modparam("acc", "detect_direction", 0)
/* account triggers (flags) */
modparam("acc", "failed_transaction_flag", 3)
modparam("acc", "log_flag", 1)
modparam("acc", "log_missed_flag", 2)
/* uncomment the following lines to enable DB accounting also */
modparam("acc", "db_flag", 1)
modparam("acc", "db_missed_flag", 2)

# ----- usrloc params -----
#modparam("usrloc", "db_mode", 0)
/* uncomment the following lines if you want to enable DB persistency
    for location entries */
modparam("usrloc", "db_mode", 2)
#modparam("usrloc", "db_url",
#      "mysql://openser:openserrw@localhost/openser")

# ----- auth_db params -----

```

```

/* uncomment the following lines if you want to enable the DB based
   authentication */
modparam("auth_db", "calculate_ha1", yes)
modparam("auth_db", "password_column", "password")
#modparam("auth_db", "db_url",
#      "mysql://openser:openserrw@localhost/openser")
#modparam("auth_db", "load_credentials", "")

# ----- alias_db params -----
/* uncomment the following lines if you want to enable the DB based
   aliases */
#modparam("alias_db", "db_url",
#      "mysql://openser:openserrw@localhost/openser")

# ----- domain params -----
/* uncomment the following lines to enable multi-domain detection
   support */
#modparam("domain", "db_url",
#      "mysql://openser:openserrw@localhost/openser")
#modparam("domain", "db_mode", 1) # Use caching

# ----- multi-module params -----
/* uncomment the following line if you want to enable multi-domain
   support
   in the modules (dafault off) */
#modparam("alias_db|auth_db|usrloc|uri_db", "use_domain", 1)

# ----- presence params -----
/* uncomment the following lines if you want to enable presence */
#modparam("presence|presence_xml", "db_url",
#      "mysql://openser:openserrw@localhost/openser")
#modparam("presence_xml", "force_active", 1)
#modparam("presence", "server_address", "sip:192.168.1.2:5060")

# ----- exec parametros ----- Italo
modparam("exec", "setvars", 1)

##### Routing Logic #####

# main request routing logic

route{
    # xlog("L_INFO","funciona ?");
    if (!mf_process_maxfwd_header("10")) {
        sl_send_reply("483","Too Many Hops");
        exit;
    }

    if (has_totag()) {
        # sequential request withing a dialog should
        # take the path determined by record-routing
        if (loose_route()) {
            if (is_method("BYE")) {
                setflag(1); # do accounting ...
                setflag(3); # ... even if the transaction fails
            }
            route(1);
        }
    }
}

```

```

        } else {
            /* uncomment the following lines if you want to
enable presence */
            ##if (is_method("SUBSCRIBE") && $rd ==
"your.server.ip.address") {
                ##      # in-dialog subscribe requests
                ##      route(2);
                ##      exit;
                ##}
                if ( is_method("ACK") ) {
                    if ( t_check_trans() ) {
                        # non loose-route, but stateful ACK; must
be an ACK after a 487 or e.g. 404 from upstream server
                        t_relay();
                        exit;
                    } else {
                        # ACK without matching transaction ...
ignore and discard.\n");
                        exit;
                    }
                }
                sl_send_reply("404","Not here");
            }
            exit;
        }

#initial requests

# CANCEL processing
if (is_method("CANCEL"))
{
    if (t_check_trans())
        t_relay();
    exit;
}

t_check_trans();

# authenticate if from local subscriber (uncomment to enable
auth)
##if (!(method=="REGISTER") && from_uri==myself)
##{
##    if (!proxy_authorize("", "subscriber")) {
##        proxy_challenge("", "0");
##        exit;
##    }
##    if (!check_from()) {
##        sl_send_reply("403","Forbidden auth ID");
##        exit;
##    }
##}
##
##    consume_credentials();
##    # caller authenticated
##}

# record routing
if (!is_method("REGISTER|MESSAGE"))
    record_route();

```

```

# account only INVITEs
if (is_method("INVITE")) {
    setflag(1); # do accounting
}
if (!uri==myself)
/* replace with following line if multi-domain support is used
*/
##if (!is_uri_host_local())
{
    append_hf("P-hint: outbound\r\n");
    # if you have some interdomain connections via TLS
    ##if($rd=="tls_domain1.net") {
    ##    t_relay("tls:domain1.net");
    ##    exit;
    ##} else if($rd=="tls_domain2.net") {
    ##    t_relay("tls:domain2.net");
    ##    exit;
    ##}
    route(1);
}

# requests for my domain

/* uncomment this if you want to enable presence server
and comment the next 'if' block
NOTE: uncomment also the definition of route[2] from below
*/
##if( is_method("PUBLISH|SUBSCRIBE"))
##    route(2);

if (is_method("PUBLISH"))
{
    sl_send_reply("503", "Service Unavailable");
    exit;
}

if (is_method("REGISTER"))
{
    # authenticate the REGISTER requests (uncomment to enable
auth)
    if (!www_authorize("200.131.189.90", "subscriber"))
    {
        www_challenge("200.131.189.90", "0");

        exit;
    }
    ##
    ##if (!check_to())
    ##{
    ##    sl_send_reply("403","Forbidden auth ID");
    ##    exit;
    ##}

    if (!save("location"))
        sl_reply_error();

    rewritehostport("200.131.189.85");
}

```

```

        append_branch();

        rewritehostport("200.131.189.86");
        t_relay();

        exit;
    }

    if ($rU==NULL) {
        # request with no Username in RURI
        sl_send_reply("484","Address Incomplete");
        exit;
    }

    # apply DB based aliases (uncomment to enable)
    ##alias_db_lookup("dbaliases");

    if (!lookup("location")) {
        switch ($retcode) {
            case -1:
            case -3:
                t_newtran();
                t_reply("404", "Not Found");
                exit;
            case -2:
                sl_send_reply("405", "Method Not Allowed");
                exit;
        }
    }

    # when routing via usrloc, log the missed calls also
    setflag(2);

    route(1);
}

route[1] {
    xlog("L_INFO","passamos pelo route[1]");
    # for INVITEs enable some additional helper routes
    if (is_method("INVITE")) {
        t_on_branch("2");
        t_on_reply("2");
        t_on_failure("1");
    }

    t_on_failure("1");

    if (!t_relay()) {
        sl_reply_error();
    };
    exit;
}

# Presence route
/* uncomment the whole following route for enabling presence
   NOTE: do not forget to enable the call of this route from the main
       route */
##route[2]

```

```

##{
##    if (!t_newtran())
##    {
##        sl_reply_error();
##        exit;
##    };
##
##    if(is_method("PUBLISH"))
##    {
##        handle_publish();
##        t_release();
##    }
##    else
##    if( is_method("SUBSCRIBE"))
##    {
##        handle_subscribe();
##        t_release();
##    }
##
##    exit;
##}

branch_route[2] {
    xlog("new branch at $ru\n");
}

onreply_route[2] {
    xlog("incoming reply\n");
}

failure_route[1] {
    xlog("L_INFO","entramos no failure route - itc");

    if (t_was_cancelled()) {
        exit;
    }

    # uncomment the following lines if you want to block client
    # redirect based on 3xx replies.
    ##if (t_check_status("3[0-9][0-9]")) {
    ##t_reply("404","Not found");
    ##    exit;
    ##}

    # uncomment the following lines if you want to redirect the
    failed
    # calls to a different new destination
    ##if (t_check_status("486|408")) {
    ##    sethostport("192.168.2.100:5060");
    ##    append_branch();
    ##    # do not set the missed call flag again
    ##    t_relay();
    ##}
    if (t_check_status("488")) {

```

```

        exec_avp("/root/Desktop/modulobalanceador/Sub-
móduloIndicadordeMG", "$avp(s:pub_ip)");
        pv_printf("$var(my_ip)", "$avp(s:pub_ip)");

        switch($var(my_ip))
        {
            case "200.131.189.86":
                log("vai para o Asterisk 2\n");
                rewritehostport("200.131.189.86:5060");
                append_branch();
                t_relay();
            break;
            case "200.131.189.85":
                log("vai para o Asterisk 1\n");
                rewritehostport("200.131.189.85:5060");
                append_branch();
                t_relay();
            break;
            default:
                log("-- Nada foi executado no switch --\n");
        }
    }
}

```

A.3 - *snmpd.conf*

Esta seção apresenta o código-fonte do arquivo de configuração *snmpd.conf*, utilizado nos *Media Gateways*.

```

#####
####
#
# snmpd.conf
#
#   - created by the snmpconf configuration program
#

#####
####
# SECTION: System Information Setup
#
#   This section defines some of the information reported in
#   the "system" mib group in the mibII tree.

# syslocation: The [typically physical] location of the system.
#   Note that setting this value here means that when trying to
#   perform an snmp SET operation to the sysLocation.0 variable will
make
#   the agent return the "notWritable" error code. IE, including
#   this token in the snmpd.conf file will disable write access to
#   the variable.

```

```

# arguments: location_string

syslocation LARC

# syscontact: The contact information for the administrator
# Note that setting this value here means that when trying to
# perform an snmp SET operation to the sysContact.0 variable will
# make
# the agent return the "notWritable" error code. IE, including
# this token in the snmpd.conf file will disable write access to
# the variable.
# arguments: contact_string

syscontact italo.tiago

# syssservices: The proper value for the sysServices object.
# arguments: syssservices_number

syssservices 0

#####
#####
# SECTION: Access Control Setup
#
# This section defines who is allowed to talk to your running
# snmp agent.

# rocommunity: a SNMPv1/SNMPv2c read-only access community name
# arguments: community [default|hostname|network/bits] [oid]
rocommunity public

```


Anexo B - Logs das Chamadas Realizadas nos Testes

B.1 - Teste 1, Participantes com Idênticos *Codecs*

São dispostos nesta seção os *logs* das chamadas realizadas no “Teste 1, participantes com idênticos *codecs*”.

1ª chamada: originária do Participante B para o Participante A, com início às 11:02 e término às 11:03.

```
Apr 24 11:02:04 teta /usr/local/sbin/openser[4765]: ACC: transaction answered:
timestamp=1240581724;method=INVITE;from_tag=f9713e05;to_tag=d67ed207;call_id=MGRhMzUwN2VhOWI4YThkMzd1OTgzODNmOWFkNWl0MzY.;code=200;reason=OK
Apr 24 11:02:04 teta /usr/local/sbin/openser[4771]: ACC: request acknowledged:
timestamp=1240581724;method=ACK;from_tag=f9713e05;to_tag=d67ed207;call_id=MGRhMzUwN2VhOWI4YThkMzd1OTgzODNmOWFkNWl0MzY.;code=200;reason=OK
Apr 24 11:03:12 teta /usr/local/sbin/openser[4771]: ACC: transaction answered:
timestamp=1240581792;method=BYE;from_tag=d67ed207;to_tag=f9713e05;call_id=MGRhMzUwN2VhOWI4YThkMzd1OTgzODNmOWFkNWl0MzY.;code=200;reason=OK
```

2ª chamada: originária do Participante A para o Participante B, com início às 11:04 e término às 11:05.

```
Apr 24 11:04:18 teta /usr/local/sbin/openser[4766]: ACC: transaction answered:
timestamp=1240581858;method=INVITE;from_tag=ad247e5e;to_tag=dc256f5e;call_id=ZWZmYWlWYjMlZDgzZmI5NDM3ZDJkNjc4NzA5NjE1YzM.;code=200;reason=OK
Apr 24 11:04:18 teta /usr/local/sbin/openser[4771]: ACC: request acknowledged:
timestamp=1240581858;method=ACK;from_tag=ad247e5e;to_tag=dc256f5e;call_id=ZWZmYWlWYjMlZDgzZmI5NDM3ZDJkNjc4NzA5NjE1YzM.;code=200;reason=OK
```

Apr 24 11:05:30 teta /usr/local/sbin/openser[4774]: ACC: transaction answered:
timestamp=1240581930;method=BYE;from_tag=ad247e5e;to_tag=dc256f5e;call_id=ZWZmYWIwYjMlZDgzZmI5NDM3ZDJkNjc4NzA5NjE1YzM.;code=200;reason=OK

3ª chamada: originária do Participante A para o Participante B, com início às 11:06 e término às 11:08.

Apr 24 11:06:23 teta /usr/local/sbin/openser[4765]: ACC: transaction answered:
timestamp=1240581983;method=INVITE;from_tag=ee4edd02;to_tag=2148f536;call_id=YWVjYmQ5NjE0ZTM5MTU1ZTVmY2NjYjQ5YzBmNWE5Njc.;code=200;reason=OK
Apr 24 11:06:23 teta /usr/local/sbin/openser[4766]: ACC: request acknowledged:
timestamp=1240581983;method=ACK;from_tag=ee4edd02;to_tag=2148f536;call_id=YWVjYmQ5NjE0ZTM5MTU1ZTVmY2NjYjQ5YzBmNWE5Njc.;code=200;reason=OK
Apr 24 11:08:06 teta /usr/local/sbin/openser[4771]: ACC: transaction answered:
timestamp=1240582086;method=BYE;from_tag=ee4edd02;to_tag=2148f536;call_id=YWVjYmQ5NjE0ZTM5MTU1ZTVmY2NjYjQ5YzBmNWE5Njc.;code=200;reason=OK

4ª chamada: originária do Participante B para o Participante A, com início às 11:09 e término às 11:10.

Apr 24 11:09:13 teta /usr/local/sbin/openser[4765]: ACC: transaction answered:
timestamp=1240582153;method=INVITE;from_tag=f71ale3d;to_tag=3e0ef04d;call_id=N2YyMGM4ODMyYWU1NDEyNWEyMTcxZGMxZWZjNmQ0MmI.;code=200;reason=OK
Apr 24 11:09:13 teta /usr/local/sbin/openser[4766]: ACC: request acknowledged:
timestamp=1240582153;method=ACK;from_tag=f71ale3d;to_tag=3e0ef04d;call_id=N2YyMGM4ODMyYWU1NDEyNWEyMTcxZGMxZWZjNmQ0MmI.;code=200;reason=OK
Apr 24 11:10:33 teta /usr/local/sbin/openser[4774]: ACC: transaction answered:
timestamp=1240582233;method=BYE;from_tag=3e0ef04d;to_tag=f71ale3d;call_id=N2YyMGM4ODMyYWU1NDEyNWEyMTcxZGMxZWZjNmQ0MmI.;code=200;reason=OK

5ª chamada: originária do Participante A para o Participante B, com início às 11:11 e término às 11:13.

Apr 24 11:11:21 teta /usr/local/sbin/openser[4774]: ACC: transaction answered:
timestamp=1240582281;method=INVITE;from_tag=cb2c4075;to_tag=8256f91e;call_id=ZTVlOWM3MDI1NWZhZDk3NWM3OThiOGJlMGNlYzIyNTI.;code=200;reason=OK
Apr 24 11:11:21 teta /usr/local/sbin/openser[4771]: ACC: request acknowledged:
timestamp=1240582281;method=ACK;from_tag=cb2c4075;to_tag=8256f91e;call_id=ZTVlOWM3MDI1NWZhZDk3NWM3OThiOGJlMGNlYzIyNTI.;code=200;reason=OK
Apr 24 11:13:11 teta /usr/local/sbin/openser[4771]: ACC: transaction answered:

```
timestamp=1240582391;method=BYE;from_tag=cb2c4075;to_tag=8256f91e;call_id=ZTVlOWM3MDI1NWFhZDk3NWM3OThiOGJlMGNlYzIyNTI.;code=200;reason=OK
```

B.2 - Teste 2, Participantes com *Codecs* Distintos

A seguir estão disponíveis os *logs* das chamadas realizadas no “Teste 2, participantes com *codecs* distintos”.

1ª chamada: originária do Participante A para o Participante B, com início às 11:24 e término às 11:25.

```
Apr 24 11:24:10 teta /usr/local/sbin/openser[4765]: ACC: call missed:
timestamp=1240583050;method=INVITE;from_tag=9b491f21;to_tag=800f800b;call_id=Mjc0MWM5YjQ3NmYyNDc3MzQ2YmM3ZGE2NDE3NmJjZjc.;code=488;reason=Not Acceptable Here
Apr 24 11:24:10 teta /usr/local/sbin/openser[4765]: Chama o MBC
Apr 24 11:24:10 teta /usr/local/sbin/openser[4765]: MBC retornou o endereço
Apr 24 11:24:10 teta /usr/local/sbin/openser[4765]: encaminha a chamada para o MG: 200.131.189.85
Apr 24 11:24:10 teta /usr/local/sbin/openser[4765]: MG já foi acionado
Apr 24 11:24:17 teta /usr/local/sbin/openser[4766]: ACC: transaction answered:
timestamp=1240583057;method=INVITE;from_tag=9b491f21;to_tag=as6f25600a;call_id=Mjc0MWM5YjQ3NmYyNDc3MzQ2YmM3ZGE2NDE3NmJjZjc.;code=200;reason=OK
Apr 24 11:24:17 teta /usr/local/sbin/openser[4771]: ACC: request acknowledged:
timestamp=1240583057;method=ACK;from_tag=9b491f21;to_tag=as6f25600a;call_id=Mjc0MWM5YjQ3NmYyNDc3MzQ2YmM3ZGE2NDE3NmJjZjc.;code=200;reason=OK
Apr 24 11:25:46 teta /usr/local/sbin/openser[4774]: ACC: transaction answered:
timestamp=1240583146;method=BYE;from_tag=9b491f21;to_tag=as6f25600a;call_id=Mjc0MWM5YjQ3NmYyNDc3MzQ2YmM3ZGE2NDE3NmJjZjc.;code=200;reason=OK
```

2ª chamada: originária do Participante B para o Participante A, com início às 11:27 e término às 11:28.

```
Apr 24 11:27:16 teta /usr/local/sbin/openser[4765]: ACC: call missed:
timestamp=1240583236;method=INVITE;from_tag=c8654846;to_tag=225b0e20;call_id=NmMzNzM5NDA2ZWUyNWRkZDQlMmJkMzEwNzRiZjAzMDY.;code=488;reason=Not Acceptable Here
Apr 24 11:27:16 teta /usr/local/sbin/openser[4765]: Chama o MBC
Apr 24 11:27:16 teta /usr/local/sbin/openser[4765]: MBC retornou o endereço
Apr 24 11:27:16 teta /usr/local/sbin/openser[4765]: encaminha a chamada para o MG: 200.131.189.85
```

Apr 24 11:27:16 teta /usr/local/sbin/openser[4765]: MG já foi acionado
 Apr 24 11:27:31 teta /usr/local/sbin/openser[4771]: ACC: transaction answered:
 timestamp=1240583251;method=INVITE;from_tag=c8654846;to_tag=as5c0261d5;call_id=NmMzNzM5NDA2ZWUyNWRkZDQ1MmJkMzEwNzRiZjAzMDY.;code=200;reason=OK
 Apr 24 11:27:31 teta /usr/local/sbin/openser[4765]: ACC: request acknowledged:
 timestamp=1240583251;method=ACK;from_tag=c8654846;to_tag=as5c0261d5;call_id=NmMzNzM5NDA2ZWUyNWRkZDQ1MmJkMzEwNzRiZjAzMDY.;code=200;reason=OK
 Apr 24 11:28:08 teta /usr/local/sbin/openser[4774]: ACC: transaction answered:
 timestamp=1240583288;method=BYE;from_tag=c8654846;to_tag=as5c0261d5;call_id=NmMzNzM5NDA2ZWUyNWRkZDQ1MmJkMzEwNzRiZjAzMDY.;code=200;reason=OK

3ª chamada: originária do Participante A para o Participante B, com início às 11:29 e término às 11:30.

Apr 24 11:29:06 teta /usr/local/sbin/openser[4771]: ACC: call missed:
 timestamp=1240583346;method=INVITE;from_tag=b2108d03;to_tag=2e674821;call_id=MzgXNjA0ZTg2Y2Y0MWI2YjNmOTM2MzI2MDM1YzE5N2E.;code=488;reason=Not Acceptable Here
 Apr 24 11:29:06 teta /usr/local/sbin/openser[4771]: Chama o MBC
 Apr 24 11:29:06 teta /usr/local/sbin/openser[4771]: MBC retornou o endereço
 Apr 24 11:29:06 teta /usr/local/sbin/openser[4771]: encaminha a chamada para o MG: 200.131.189.85
 Apr 24 11:29:06 teta /usr/local/sbin/openser[4771]: MG já foi acionado
 Apr 24 11:29:19 teta /usr/local/sbin/openser[4766]: ACC: transaction answered:
 timestamp=1240583359;method=INVITE;from_tag=b2108d03;to_tag=as0a358cf8;call_id=MzgXNjA0ZTg2Y2Y0MWI2YjNmOTM2MzI2MDM1YzE5N2E.;code=200;reason=OK
 Apr 24 11:29:19 teta /usr/local/sbin/openser[4765]: ACC: request acknowledged:
 timestamp=1240583359;method=ACK;from_tag=b2108d03;to_tag=as0a358cf8;call_id=MzgXNjA0ZTg2Y2Y0MWI2YjNmOTM2MzI2MDM1YzE5N2E.;code=200;reason=OK
 Apr 24 11:30:40 teta /usr/local/sbin/openser[4766]: ACC: transaction answered:
 timestamp=1240583440;method=BYE;from_tag=b2108d03;to_tag=as0a358cf8;call_id=MzgXNjA0ZTg2Y2Y0MWI2YjNmOTM2MzI2MDM1YzE5N2E.;code=200;reason=OK

4ª chamada: originária do Participante B para o Participante A, com início às 11:32 e término às 11:33.

Apr 24 11:32:08 teta /usr/local/sbin/openser[4771]: ACC: call missed:
 timestamp=1240583528;method=INVITE;from_tag=bc7ee832;to_tag=f1749b2b;call_id=YmIyYThlNjcyZTYxODNkYThmNzRjNGI1NTMwNWl5NTE.;code=488;reason=Not Acceptable Here
 Apr 24 11:32:08 teta /usr/local/sbin/openser[4771]: Chama o MBC
 Apr 24 11:32:08 teta /usr/local/sbin/openser[4771]: MBC retornou o endereço

```

Apr 24 11:32:08 teta /usr/local/sbin/openser[4771]: encaminha a
chamada para o MG: 200.131.189.85
Apr 24 11:32:08 teta /usr/local/sbin/openser[4771]: MG já foi acionado
Apr 24 11:32:15 teta /usr/local/sbin/openser[4765]: ACC: transaction
answered:
timestamp=1240583535;method=INVITE;from_tag=bc7ee832;to_tag=as34ff2089;
call_id=YmIyYThlNjcyZTYxODNkYThmNzRjNGI1NTMwNWl5NTE.;code=200;reason=
OK
Apr 24 11:32:15 teta /usr/local/sbin/openser[4766]: ACC: request
acknowledged:
timestamp=1240583535;method=ACK;from_tag=bc7ee832;to_tag=as34ff2089;ca
ll_id=YmIyYThlNjcyZTYxODNkYThmNzRjNGI1NTMwNWl5NTE.;code=200;reason=OK
Apr 24 11:33:09 teta /usr/local/sbin/openser[4766]: ACC: transaction
answered:
timestamp=1240583589;method=BYE;from_tag=as34ff2089;to_tag=bc7ee832;ca
ll_id=YmIyYThlNjcyZTYxODNkYThmNzRjNGI1NTMwNWl5NTE.;code=200;reason=OK

```

5ª chamada: originária do Participante A para o Participante B, com início às 11:35 e término às 11:37.

```

Apr 24 11:35:47 teta /usr/local/sbin/openser[4771]: ACC: call missed:
timestamp=1240583747;method=INVITE;from_tag=474b8971;to_tag=7e2e3e23;c
all_id=N2VmMzFjMzNlODE5OWU3ODI3ZmFhNGNhN2U1MDU0ZjQ.;code=488;reason=No
t Acceptable Here
Apr 24 11:35:47 teta /usr/local/sbin/openser[4771]: Chama o MBC
Apr 24 11:35:47 teta /usr/local/sbin/openser[4771]: MBC retornou o
endereço
Apr 24 11:35:47 teta /usr/local/sbin/openser[4771]: encaminha a
chamada para o MG: 200.131.189.85
Apr 24 11:35:47 teta /usr/local/sbin/openser[4771]: MG já foi acionado
Apr 24 11:35:51 teta /usr/local/sbin/openser[4766]: ACC: transaction
answered:
timestamp=1240583751;method=INVITE;from_tag=474b8971;to_tag=as7c003a70;
call_id=N2VmMzFjMzNlODE5OWU3ODI3ZmFhNGNhN2U1MDU0ZjQ.;code=200;reason=
OK
Apr 24 11:35:51 teta /usr/local/sbin/openser[4771]: ACC: request
acknowledged:
timestamp=1240583751;method=ACK;from_tag=474b8971;to_tag=as7c003a70;ca
ll_id=N2VmMzFjMzNlODE5OWU3ODI3ZmFhNGNhN2U1MDU0ZjQ.;code=200;reason=OK
Apr 24 11:37:30 teta /usr/local/sbin/openser[4771]: ACC: transaction
answered:
timestamp=1240583822;method=BYE;from_tag=474b8971;to_tag=as7c003a70;ca
ll_id=N2VmMzFjMzNlODE5OWU3ODI3ZmFhNGNhN2U1MDU0ZjQ.;code=200;reason=OK

```

B.3 - Teste 3, com *Media Gateway* Ativo e Inativo

São dispostos a seguir os *logs* das chamadas realizadas no “Teste 3, com *Media Gateway* ativo e inativo”.

**1ª chamada: originária do Participante B para o Participante A,
com início às 11:43 e término às 11:44.**

```
Apr 24 11:43:28 teta /usr/local/sbin/openser[4774]: ACC: call missed:
timestamp=1240584208;method=INVITE;from_tag=b7253973;to_tag=e879267e;c
all_id=YTJlOTgzMTI3ZjdjYzdLOGNiNjhZmMlMDI2ZTg1ZGE.;code=488;reason=No
t Acceptable Here
Apr 24 11:43:28 teta /usr/local/sbin/openser[4774]: Chama o MBC
Apr 24 11:43:28 teta /usr/local/sbin/openser[4774]: MBC retornou o
endereço
Apr 24 11:43:28 teta /usr/local/sbin/openser[4774]: encaminha a
chamada para o MG: 200.131.189.86
Apr 24 11:43:28 teta /usr/local/sbin/openser[4774]: MG já foi acionado
Apr 24 11:43:36 teta /usr/local/sbin/openser[4766]: ACC: transaction
answered:
timestamp=1240584216;method=INVITE;from_tag=b7253973;to_tag=as6b8cc1da
;call_id=YTJlOTgzMTI3ZjdjYzdLOGNiNjhZmMlMDI2ZTg1ZGE.;code=200;reason=
OK
Apr 24 11:43:36 teta /usr/local/sbin/openser[4774]: ACC: request
acknowledged:
timestamp=1240584217;method=ACK;from_tag=b7253973;to_tag=as6b8cc1da;ca
ll_id=YTJlOTgzMTI3ZjdjYzdLOGNiNjhZmMlMDI2ZTg1ZGE.;code=200;reason=OK
Apr 24 11:44:43 teta /usr/local/sbin/openser[4774]: ACC: transaction
answered:
timestamp=1240584283;method=BYE;from_tag=as6b8cc1da;to_tag=b7253973;ca
ll_id=YTJlOTgzMTI3ZjdjYzdLOGNiNjhZmMlMDI2ZTg1ZGE.;code=200;reason=OK
```

**2ª chamada: originária do Participante A para o Participante B,
com início às 11:46 e término às 11:47.**

```
Apr 24 11:46:06 teta /usr/local/sbin/openser[4774]: ACC: call missed:
timestamp=1240584366;method=INVITE;from_tag=6606c165;to_tag=e4704e05;c
all_id=Y2U0ZjNhNzg0NjVhNTliMWFjNTIzNTVlY2MxMWE1Yjg.;code=488;reason=No
t Acceptable Here
Apr 24 11:46:06 teta /usr/local/sbin/openser[4774]: Chama o MBC
Apr 24 11:46:07 teta /usr/local/sbin/openser[4774]: MBC retornou o
endereço
Apr 24 11:46:07 teta /usr/local/sbin/openser[4774]: encaminha a
chamada para o MG: 200.131.189.86
Apr 24 11:46:07 teta /usr/local/sbin/openser[4774]: MG já foi acionado
Apr 24 11:46:11 teta /usr/local/sbin/openser[4771]: ACC: transaction
answered:
timestamp=1240584371;method=INVITE;from_tag=6606c165;to_tag=as600b8b6a
;call_id=Y2U0ZjNhNzg0NjVhNTliMWFjNTIzNTVlY2MxMWE1Yjg.;code=200;reason=
OK
Apr 24 11:46:11 teta /usr/local/sbin/openser[4765]: ACC: request
acknowledged:
timestamp=1240584371;method=ACK;from_tag=6606c165;to_tag=as600b8b6a;ca
ll_id=Y2U0ZjNhNzg0NjVhNTliMWFjNTIzNTVlY2MxMWE1Yjg.;code=200;reason=OK
Apr 24 11:47:14 teta /usr/local/sbin/openser[4771]: ACC: transaction
answered:
timestamp=1240584434;method=BYE;from_tag=6606c165;to_tag=as600b8b6a;ca
ll_id=Y2U0ZjNhNzg0NjVhNTliMWFjNTIzNTVlY2MxMWE1Yjg.;code=200;reason=OK
```

**3ª chamada: originária do Participante B para o Participante A,
com início às 11:49 e término às 11:50.**

Apr 24 11:49:10 teta /usr/local/sbin/openser[4774]: ACC: call missed: timestamp=1240584550;method=INVITE;from_tag=2345761b;to_tag=5d2a226f;call_id=MWizMDg1MWM2MDI0MWZiYzNkM2UzYjMwNGRkZmJlNmM.;code=488;reason=Not Acceptable Here
Apr 24 11:49:10 teta /usr/local/sbin/openser[4774]: Chama o MBC
Apr 24 11:49:10 teta /usr/local/sbin/openser[4774]: MBC retornou o endereço
Apr 24 11:49:10 teta /usr/local/sbin/openser[4774]: encaminha a chamada para o MG: 200.131.189.86
Apr 24 11:49:10 teta /usr/local/sbin/openser[4774]: MG já foi acionado
Apr 24 11:49:15 teta /usr/local/sbin/openser[4765]: ACC: transaction answered: timestamp=1240584555;method=INVITE;from_tag=2345761b;to_tag=as5c4549d6;call_id=MWizMDg1MWM2MDI0MWZiYzNkM2UzYjMwNGRkZmJlNmM.;code=200;reason=OK
Apr 24 11:49:15 teta /usr/local/sbin/openser[4774]: ACC: request acknowledged: timestamp=1240584555;method=ACK;from_tag=2345761b;to_tag=as5c4549d6;call_id=MWizMDg1MWM2MDI0MWZiYzNkM2UzYjMwNGRkZmJlNmM.;code=200;reason=OK
Apr 24 11:50:25 teta /usr/local/sbin/openser[4774]: ACC: transaction answered: timestamp=1240584625;method=BYE;from_tag=as5c4549d6;to_tag=2345761b;call_id=MWizMDg1MWM2MDI0MWZiYzNkM2UzYjMwNGRkZmJlNmM.;code=200;reason=OK

**4ª chamada: originária do Participante A para o Participante B,
com início às 11:52 e término às 11:53.**

Apr 24 11:52:09 teta /usr/local/sbin/openser[4765]: ACC: call missed: timestamp=1240584729;method=INVITE;from_tag=a85fbe21;to_tag=6527060f;call_id=Yjc0YTQzMjc5ZTI5ZTlkNjNlM2YyZDdhNGVmZmYwMTE.;code=488;reason=Not Acceptable Here
Apr 24 11:52:09 teta /usr/local/sbin/openser[4765]: Chama o MBC
Apr 24 11:52:09 teta /usr/local/sbin/openser[4765]: MBC retornou o endereço
Apr 24 11:52:09 teta /usr/local/sbin/openser[4765]: encaminha a chamada para o MG: 200.131.189.86
Apr 24 11:52:09 teta /usr/local/sbin/openser[4765]: MG já foi acionado
Apr 24 11:52:12 teta /usr/local/sbin/openser[4766]: ACC: transaction answered: timestamp=1240584732;method=INVITE;from_tag=a85fbe21;to_tag=as760d85c5;call_id=Yjc0YTQzMjc5ZTI5ZTlkNjNlM2YyZDdhNGVmZmYwMTE.;code=200;reason=OK
Apr 24 11:52:12 teta /usr/local/sbin/openser[4765]: ACC: request acknowledged: timestamp=1240584733;method=ACK;from_tag=a85fbe21;to_tag=as760d85c5;call_id=Yjc0YTQzMjc5ZTI5ZTlkNjNlM2YyZDdhNGVmZmYwMTE.;code=200;reason=OK
Apr 24 11:53:48 teta /usr/local/sbin/openser[4766]: ACC: transaction answered: timestamp=1240584828;method=BYE;from_tag=a85fbe21;to_tag=as760d85c5;call_id=Yjc0YTQzMjc5ZTI5ZTlkNjNlM2YyZDdhNGVmZmYwMTE.;code=200;reason=OK

**5ª chamada: originária do Participante B para o Participante A,
com início às 11:55 e término às 11:57.**

```
Apr 24 11:55:42 teta /usr/local/sbin/openser[4771]: ACC: call missed:
timestamp=1240584942;method=INVITE;from_tag=07380061;to_tag=8c4cce68;c
all_id=YjhgNWNmNmIzMzI2NTFkNWZhODgzZWNiOTYwNzNjNjU.;code=488;reason=No
t Acceptable Here
Apr 24 11:55:42 teta /usr/local/sbin/openser[4771]: Chama o MBC
Apr 24 11:55:42 teta /usr/local/sbin/openser[4771]: MBC retornou o
endereço
Apr 24 11:55:42 teta /usr/local/sbin/openser[4771]: encaminha a
chamada para o MG: 200.131.189.86
Apr 24 11:55:42 teta /usr/local/sbin/openser[4771]: MG já foi acionado
Apr 24 11:55:47 teta /usr/local/sbin/openser[4766]: ACC: transaction
answered:
timestamp=1240584947;method=INVITE;from_tag=07380061;to_tag=as4df6a39d
;call_id=YjhgNWNmNmIzMzI2NTFkNWZhODgzZWNiOTYwNzNjNjU.;code=200;reason=
OK
Apr 24 11:55:47 teta /usr/local/sbin/openser[4771]: ACC: request
acknowledged:
timestamp=1240584947;method=ACK;from_tag=07380061;to_tag=as4df6a39d;ca
ll_id=YjhgNWNmNmIzMzI2NTFkNWZhODgzZWNiOTYwNzNjNjU.;code=200;reason=OK
Apr 24 11:57:05 teta /usr/local/sbin/openser[4765]: ACC: transaction
answered:
timestamp=1240585025;method=BYE;from_tag=as4df6a39d;to_tag=07380061;ca
ll_id=YjhgNWNmNmIzMzI2NTFkNWZhODgzZWNiOTYwNzNjNjU.;code=200;reason=OK
```

B.4 - Teste 4, com *Media Gateways* Inativos

A seguir tem-se os *logs* das chamadas realizadas no “Teste 4, com *Media Gateways* inativos”.

**1ª chamada: originária do Participante A para o Participante B,
com início às 12:19 e término às 12:19.**

```
Apr 24 12:19:37 teta /usr/local/sbin/openser[3339]: ACC: call missed:
timestamp=1240586377;method=INVITE;from_tag=85210d21;to_tag=4a64b126;c
all_id=ZTVjZThjOTdmYzk3YTE0MGNiMWU2MTZiM2U5MjAyY2U.;code=488;reason=No
t Acceptable Here
Apr 24 12:19:37 teta /usr/local/sbin/openser[3339]: Chama o MBC
Apr 24 12:19:37 teta /usr/local/sbin/openser[3339]: Nenhum MG na Lista
de Prioridades
```

**2ª chamada: originária do Participante B para o Participante A,
com início às 12:20 e término às 12:20.**

Apr 24 12:20:47 teta /usr/local/sbin/openser[3341]: ACC: call missed: timestamp=1240586447;method=INVITE;from_tag=c95eb727;to_tag=b16c9535;call_id=OGU0MDNhMjMyOTljMDhmMDM5ZWE5ZTI2N2RlYTRkNDc.;code=488;reason=Not Acceptable Here
Apr 24 12:20:47 teta /usr/local/sbin/openser[3341]: Chama o MBC
Apr 24 12:20:47 teta /usr/local/sbin/openser[3341]: Nenhum MG na Lista de Prioridades

3ª chamada: originária do Participante A para o Participante B, com início às 12:21 e término às 12:21.

Apr 24 12:21:15 teta /usr/local/sbin/openser[3339]: ACC: call missed: timestamp=1240586475;method=INVITE;from_tag=3308504d;to_tag=0d79b048;call_id=NTI2ZGQ1MTczMGEyMDMzMjc2OGY5MjVmNGRiZDZkYzg.;code=488;reason=Not Acceptable Here
Apr 24 12:21:15 teta /usr/local/sbin/openser[3339]: Chama o MBC
Apr 24 12:21:15 teta /usr/local/sbin/openser[3339]: Nenhum MG na Lista de Prioridades

4ª chamada: originária do Participante B para o Participante A, com início às 12:22 e término às 12:22.

Apr 24 12:22:31 teta /usr/local/sbin/openser[3345]: ACC: call missed: timestamp=1240586551;method=INVITE;from_tag=b8440910;to_tag=6868250b;call_id=ZjM0NGE3NDRjNzJhZGVlZWViNTkxN2ZmNTViYjUwOTE.;code=488;reason=Not Acceptable Here
Apr 24 12:22:31 teta /usr/local/sbin/openser[3345]: Chama o MBC
Apr 24 12:22:31 teta /usr/local/sbin/openser[3345]: Nenhum MG na Lista de Prioridades

5ª chamada: originária do Participante A para o Participante B, com início às 12:23 e término às 12:23.

Apr 24 12:23:09 teta /usr/local/sbin/openser[3339]: ACC: call missed: timestamp=1240586589;method=INVITE;from_tag=7649d817;to_tag=2052bf13;call_id=NTE4MGFhNGZmYmZjOTU1ZTYzMGI3YjQzMjVlMmQyNDA.;code=488;reason=Not Acceptable Here
Apr 24 12:23:09 teta /usr/local/sbin/openser[3339]: Chama o MBC
Apr 24 12:23:09 teta /usr/local/sbin/openser[3339]: Nenhum MG na Lista de Prioridades

Anexo C - *Logs* dos VCBs nos Testes

C.1 - Teste 1, Participantes com Idênticos *Codecs*

A seguir estão dispostos os *logs* dos VCBs apresentados na Tabela 6.3, referentes ao “Teste 1, participantes com idênticos *codecs*”. Pela análise dos *logs*, é implícito o horário de realização de cada consulta à Lista de Prioridades, o VCB e o IP do *Media Gateway* a que ele pertence. O endereço IP 200.131.189.85 corresponde ao *Media Gateway* 1 e endereço IP 200.131.189.86 ao *Media Gateway* 2.

Teste1-11_01_02
10 200.131.189.85
59 200.131.189.86

Teste1-11_08_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_02_02
10 200.131.189.85
59 200.131.189.86

Teste1-11_09_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_03_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_10_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_04_02
10 200.131.189.85
59 200.131.189.86

Teste1-11_11_02
10 200.131.189.85
59 200.131.189.86

Teste1-11_05_02
10 200.131.189.85
60 200.131.189.86

Teste1-11_12_02
10 200.131.189.85
60 200.131.189.86

Teste1-11_06_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_13_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_07_02
9 200.131.189.85
59 200.131.189.86

Teste1-11_14_02
10 200.131.189.85
59 200.131.189.86

C.2 - Teste 2, Participantes com *Codecs* Distintos

Nesta seção tem-se os *logs* dos VCBs apresentados na Tabela 6.6, referentes ao “Teste 2, participantes com *codecs* distintos”. Pela análise dos *logs*, é implícito o horário de realização de cada consulta à Lista de Prioridades, o VCB e o IP do *Media Gateway* a que ele pertence, sendo que o endereço IP 200.131.189.85 corresponde ao *Media Gateway* 1 e endereço IP 200.131.189.86 ao *Media Gateway* 2.

Teste2-11_23_02
15 200.131.189.85
60 200.131.189.86

Teste2-11_31_02
12 200.131.189.85
59 200.131.189.86

Teste2-11_24_02
12 200.131.189.85
59 200.131.189.86

Teste2-11_32_02
11 200.131.189.85
59 200.131.189.86

Teste2-11_25_02
13 200.131.189.85
59 200.131.189.86

Teste2-11_33_02
13 200.131.189.85
59 200.131.189.86

Teste2-11_26_02
14 200.131.189.85
59 200.131.189.86

Teste2-11_34_02
12 200.131.189.85
59 200.131.189.86

Teste2-11_27_02
12 200.131.189.85
59 200.131.189.86

Teste2-11_35_02
11 200.131.189.85
59 200.131.189.86

Teste2-11_27_16
12 200.131.189.85
59 200.131.189.86

Teste2-11_36_02
11 200.131.189.85
59 200.131.189.86

Teste2-11_28_02
13 200.131.189.85
59 200.131.189.86

Teste2-11_37_02
12 200.131.189.85
59 200.131.189.86

Teste2-11_29_02
12 200.131.189.85
59 200.131.189.86

Teste2-11_38_02
11 200.131.189.85
59 200.131.189.86

Teste2-11_30_02
13 200.131.189.85
59 200.131.189.86

C.3 - Teste 3, com *Media Gateway* Ativo e Inativo

São dispostos a seguir os *logs* dos VCBs apresentados na Tabela 6.9, referentes ao “Teste 3, com *Media Gateway* ativo e inativo”. Pela análise dos *logs*, é implícito o horário de realização de cada consulta à Lista de Prioridades, o VCB e o IP do *Media Gateway* a que ele pertence, em que o endereço IP 200.131.189.85 corresponde ao *Media Gateway* 1 e, o endereço IP 200.131.189.86 ao *Media Gateway* 2.

Teste3-11_42_01
25 200.131.189.85
59 200.131.189.86

Teste3-11_43_01
12 200.131.189.86

Teste3-11_44_01
14 200.131.189.86

Teste3-11_45_01
13 200.131.189.86

Teste3-11_46_01
13 200.131.189.86

Teste3-11_47_01
15 200.131.189.86

Teste3-11_48_01
13 200.131.189.86

Teste3-11_49_01
12 200.131.189.86

Teste3-11_50_01
14 200.131.189.86

Teste3-11_51_01
13 200.131.189.86

Teste3-11_52_01
12 200.131.189.86

Teste3-11_53_01
15 200.131.189.86

Teste3-11_54_01
14 200.131.189.86

Teste3-11_55_01
12 200.131.189.86

Teste3-11_56_01
13 200.131.189.86

Teste3-11_57_01
15 200.131.189.86

Teste3-11_58_01
13 200.131.189.86

C.4 - Teste 4, com *Media Gateways* Inativos

Nesta seção tem-se os *logs* dos VCBs apresentados na Tabela 6.12, referentes ao “Teste 4, com *Media Gateways* inativos”. Pela análise dos *logs*, é implícito o horário de realização de cada consulta à Lista de Prioridades e, que não há nenhum VCB, bem como endereço IP, de nenhum dos *Media Gateways* utilizados nos testes.

Teste4-12:19:02

Teste4-12:20:02

Teste4-12:21:02

Teste4-12:22:02

Teste4-12:23:02