



**UNIVERSIDADE ESTADUAL DO CEARÁ
CENTRO DE CIÊNCIAS E TECNOLOGIA
MESTRADO PROFISSIONAL EM COMPUTAÇÃO APLICADA**

FRANCILDO FELIX DE MOURA

**FERRAMENTA PARA INTEGRAÇÃO DE CANAIS DE COMUNICAÇÃO EM UM
CONTACT CENTER**

FORTALEZA-CEARÁ

2016

FRANCILDO FELIX DE MOURA

FERRAMENTA PARA INTEGRAÇÃO DE CANAIS DE COMUNICAÇÃO EM UM
CONTACT CENTER

Dissertação apresentada ao Curso de Mestrado Profissional em Computação Aplicada do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para à obtenção do título de Mestre em Computação Aplicada. Área de Concentração: Redes.

Orientador: Prof. Dr. Anilton Salles Garcia.

FORTALEZA- CEARÁ

2016

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Moura, Francildo Felix de.

Ferramenta para integração de canais de comunicação em um contact center [recurso eletrônico] / Francildo Felix de Moura. - 2016.

1 CD-ROM: il.; 4 ⅓ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 105 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado profissional) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Profissional em Computação Aplicada, Fortaleza, 2016.

Área de concentração: Redes.

Orientação: Prof. Dr. Anilton Salles Garcia.

1. RTPC (Rede de telefonia pública comutada). 2. Contact Center. 3. Asterisk. 4. CTI. 5. URA. I. Título.

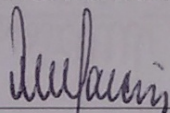
Francildo Felix De Moura

FERRAMENTA PARA INTEGRAÇÃO DE CANAIS DE COMUNICAÇÃO EM UM CONTACT CENTER.

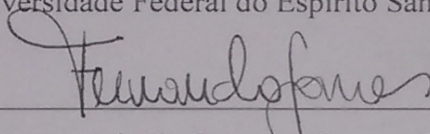
Dissertação apresentada ao Curso de Mestrado Profissional em Computação Aplicada do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para obtenção do Título de Mestre em Computação.
Área de Concentração: Redes de Computadores.

Aprovado em: 16/12/2016

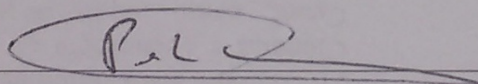
BANCA EXAMINADORA



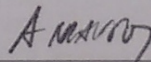
Prof. Dr. Anilton Salles Garcia, DSc. Orientador
Universidade Federal do Espírito Santo – UFES



Prof. Dr. Fernando Antônio de Carvalho Gomes, DSc. – Membro Externo
Universidade Federal do Ceará – UFC



Prof. Dr. Pedro Klecius Farias Cardoso, DSc. – Membro Interno
Instituto Federal de Educação, Ciências e Tecnologia Ceará – IFCE



Prof. Dr. Antonio Mauro Barbosa de Oliveira, PD. DSc. – Membro Interno
Instituto Federal de Educação, Ciências e Tecnologia Ceará – IFCE

RESUMO

Este trabalho valida a implementação de uma ferramenta para integração de canais de comunicação em um cenário de um *contact center*. O caráter diferenciador desta ferramenta apresenta uma forte aderência à arquitetura *open source* que está presente em todas as fases de desenvolvimento, como também colabora como modelo para geração de novos serviços para empresas e instituições que se utilizem de tal estrutura. A ferramenta foi desenvolvida com a motivação principal de facilitar o acesso das melhores práticas de contato com o cliente, utilizadas nas grandes corporações do ramo de *contact center*. Para validar a proposta de integração da ferramenta com o cenário *VoIP* (Voice over IP) atual, todas as sub-rotinas desenvolvidas foram implementadas de forma integrada às principais bibliotecas e plataformas de Telefonia *IP* de formato aberto. A plataforma *Asterisk* foi utilizada em conjunto com as seguintes *softwares*: *freePBX*, *SugarCRM*, *FOP* (Flash Operator Panel): painel de gerenciamento das ligações, *Asterisk-Stat*: que gera os relatórios das ligações efetuadas pelo *CDR* (Call detailed records), *MySQL*, *Apache*, *CentOS*, *PHP* (Hypertext Preprocessor) e outras bibliotecas. A forma de execução dos testes foi dividida em dois cenários: testes em ambiente de laboratório e testes em ambientes corporativos. Nos testes de laboratório foi validada a integração da ferramenta com a central *PABX* (Private Automatic Branch Exchange), utilizando uma *API* (Application Programming Interface) padrão para estabelecer o funcionamento de um módulo *CTI* (Computer telephony integration) onde se garante o funcionamento correto da integração do ambiente computacional com a telefonia. Nos testes executados em ambiente corporativo foi operacionalizada a ferramenta desenvolvida, o atendimento de chamadas originadas da *RTPC* (Rede pública de telefonia comutada) para o *PBX* (Private Branch Exchange) assim como o tratamento da chamada, passando pela *URA* (Unidade de Resposta Audível) até finalizar o atendimento pelo o operador. A integração das duas estruturas de telefonia (interna e externa), assim como a alta disponibilidade das informações a respeito do cliente, para o operador do *Contact Center*, validou a proposta. Conclui-se então que a utilização dessas plataformas e bibliotecas de formato aberto, viabiliza o

real desenvolvimento desta ferramenta que permite o acesso a uma estrutura de telecomunicações de baixo custo.

Palavras-chave: RTPC. Contact Center. PBX. CTI. Asterisk. SS#7. SugarCRM. URA.

ABSTRACT

This work validates the implementation of a tool for integration of communication channels in a contact center scenario. The differentiating nature of this tool presents a strong adherence to the open source architecture that is present in all phases of development, but also collaborates as a model for generating new services for companies and institutions that use such a structure. The tool was developed with the main motivation to facilitate the access of the best practices of contact with the client, used in the big corporations of the contact center branch. To validate the tool integration proposal with the current VoIP (Voice over IP) scenario, all the developed subroutines have been implemented in an integrated way to the main libraries and platforms of open IP Telephony. The Asterisk platform was used in conjunction with the following softwares: freePBX, SugarCRM, Flash Operator Panel (FOP): links management panel, Asterisk-Stat: which generates reports of calls made by CDR, MySQL, Apache, CentOS, PHP (Hypertext Preprocessor) and other libraries. The way the tests were run was divided into two scenarios: lab environment testing and testing in corporate environments. In the laboratory tests, the integration of the tool with the PBX (Private Automatic Branch Exchange) was validated using a standard API (Application Programming Interface) to establish the operation of a CTI (Computer telephony integration) module where the correct functioning of the Integration of the computing environment with telephony. In the tests performed in a corporate environment, the developed tool was used to handle calls originating from the public switched telephone network (PSTN) to the Private Branch Exchange (PBX), as well as handling the call, through the Audible Response Unit (URA) Until the end of the service by the operator. The integration of the two telephony structures (internal and external), as well as the high availability of information about the customer, to the contact center operator,

[validated the proposal. It is concluded that the use of these platforms and open-format libraries, enables the real development of this tool that allows access to a low-cost telecommunications structure.](#)

Keywords: RTPC. Contact Center. PBX. CTI. Asterisk. SS#7. SugarCRM. URA.

LISTA DE FIGURAS

Figura 1 -	Arquitetura convencional do ambiente de <i>Call Center</i>	19
Figura 2 -	Estrutura básica do protocolo <i>RTP</i>	26
Figura 3 -	Comunicação de voz em uma rede comutada a pacotes.....	28
Figura 4 -	Arquitetura do Asterisk e interligação entre os diferentes componentes.....	34
Figura 5 -	Comunicação básica com o protocolo <i>IAX</i>	36
Figura 6 -	Estrutura de um <i>frame</i> completo <i>IAX proxy</i>	37
Figura 7 -	Estrutura de um mini <i>frame</i> completo <i>IAX proxy</i>	39
Figura 8 -	Convergência dos canais de comunicação com o cliente.....	40
Figura 9 -	Integração de canais de comunicação.....	41
Figura 10 -	Ranking de custo do valor da licença mensal de cada software.....	42
Figura 11 -	Topologia do Gateway em ambientes corporativo.....	44
Figura 12 -	Estrutura de rede do <i>Call Center</i> que foi monitorado.....	45
Figura 13 -	Funcionamento do <i>VoIPMonitor</i>	46
Figura 14 -	Figura 14: Arquitetura do Asterisk.....	47
Figura 15 -	Distribuição de unidades ou filiais de um <i>Call Center</i>	49
Figura 16 -	Visão geral da ferramenta de integração.....	51
Figura 17 -	Estrutura básica de funcionamento da Ferramenta.....	52
Figura 18 -	Estrutura dos componentes da Ferramenta de integração de canais.....	54
Figura 19 -	ntegração <i>Interface</i> de convergência.....	55

Figura 20 -	O Asterisk e as <i>interfaces FXS e FXO</i>	56
Figura 21 -	Rede com <i>interface NAT</i>	57
Figura 22 -	Configuração do ambiente.....	59
Figura 23 -	Exemplo de registro de um ramal no arquivo de configuração: <i>sip.conf</i>	61
Figura 24 -	Exemplo de registro de contexto de chamadas em <i>extensions.conf</i>	62
Figura 25 -	Transferência de chamadas em <i>features.conf</i>	62
Figura 26 -	Transferência de chamadas em <i>extensions.conf</i>	63
Figura 27 -	Transferência de chamadas em <i>sip.conf</i>	63
Figura 28 -	Grupos de sinalização múltipla configurado no arquivos <i>queues.conf</i>	64
Figura 29 -	Sinalização múltipla de chamadas configurado no arquivo <i>extensions.conf</i>	64
Figura 30 -	Grupo de monitoramento ou captura de chamadas em <i>sip.conf</i>	65
Figura 31 -	Salas de conferência em <i>meetme.conf</i>	65
Figura 32 -	Configuração das salas de conferência no arquivo <i>extensions.conf</i>	67
Figura 33 -	Configuração das placas <i>FXS e FXO</i> em <i>zaptel.conf</i>	67
Figura 34 -	Configuração dos canais no arquivo <i>zapata.conf</i>	68
Figura 35 -	Configuração dos canais no arquivo <i>zapata.conf</i> na sessão RTPC.....	69
Figura 36 -	Configuração da central <i>PBX</i> no arquivo <i>iax.conf</i>	69
Figura 37 -	Ramal da central <i>PBX</i> configurado em <i>extensions.conf</i>	69
Figura 38 -	Tela de login do sistema. Módulo operador.....	70
Figura 39 -	Tela de atendimento do módulo agente (operador).....	71
Figura 40 -	Tela de <i>CallBack</i> comum a todos os agentes.....	72
Figura 41 -	Tela de monitoria do sistema. Módulo de monitoria de <i>DAC</i>	73
Figura 42 -	Tela de monitoria. Ouvindo as ligações 01 e 02.....	74
Figura 43 -	Tela de login no módulo de relatórios.....	74

Figura 44 - Tela de Relatório de <i>login/pausa</i> por agente.....	75
Figura 45 - Teste de ramal de eco.....	77
Figura 46 - Teste executado entre os ramais <i>VoIP</i> 2011 e 2012.....	78
Figura 47 - Resumo do cenário para os teste de múltiplas chamadas simultâneas.....	80
Figura 48 - Fluxo de comunicação de mensagens <i>RTP</i> e <i>SIP</i>.....	82
Figura 49 - Teste de estabelecimento de múltiplas chamadas simultâneas.....	87
Figura 50 - Resumo do cenário de rede para o teste de chamadas em conferência.....	89
Figura 51 - Fluxo de mensagens <i>RTP</i> e <i>SIP</i> para o teste de carga nas chamadas em conferência.....	91
Figura 52 - Teste de conferência com todos os ramais.....	95
Figura 53 - Teste de conferência com duas salas e dois grupo de ramais.....	96
Figura 54 - Transferência de chamada, captura e sinalização de grupo de ramais.....	97
Figura 55 - Integração com a rede RTPC (primeira etapa).....	98
Figura 56 - Integração com a rede RTPC (segunda etapa).....	99
Figura 57 - Estabelecimento das chamadas entre as duas centrais.....	99
Figura 58 - Topologia de uma rede implementando <i>NAT</i>.....	100

LISTA DE TABELAS

Tabela 1 -	Ranking de custo do valor da licença mensal de cada software	42
Tabela 2 -	Plano de numeração.....	58
Tabela 3 -	Teste de chamada entre dois terminais.....	77
Tabela 4 -	Resultados obtidos para o teste de carga para cem chamadas simultâneas.....	85
Tabela 5 -	Estatísticas de rede para o teste de chamadas simultâneas.....	85
Tabela 6 -	Estatísticas sobre fluxos de áudio de voz <i>RTP</i> para o teste de chamadas simultâneas.....	86
Tabela 7 -	Estatísticas de rede obtidas pela informação do comando <i>ifconfig</i> e resumo da aplicação <i>SIPp</i>	93
Tabela 8 -	Estatísticas sobre fluxos de áudio de voz <i>RTP</i> , coletadas através do aplicativo <i>Wireshark</i>	94
Tabela 9 -	Porcentagem média de utilização de <i>CPU</i> e memória <i>RAM</i> , na máquina onde está instalado o servidor <i>PBX Asterisk</i>	94
Tabela 10 -	Teste de Disponibilidade de banda em relação ao fluxo de chamadas <i>VoIP</i>	101

LISTA DE ABREVIATURAS E SIGLAS

AGI	<i>Asterisk Gateway Interface</i>
ANS	<i>Advanced Networking and Services</i>
API	<i>Application Programming Interface</i>
ARPA	<i>Advanced Research Projects Agency</i>
ATA	<i>Adaptadores para Telefonos Analógicos</i>
CRM	<i>Customer Relationship Management</i>
FDDI	<i>Fiber Distributed Data Interface</i>
FXO	<i>Foreign eXchange Office</i>
FXS	<i>Foreign eXchange Station</i>
GPRS	<i>General Packet Radio Service</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IAX	<i>Inter-Asterisk eXchange</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
ITU	<i>International Telecommunication Union</i>
IVR	<i>Interactive Voice Response</i>
LAN	<i>Local Area Network</i>
MAN	<i>Metropolitan Area Network</i>
MCU	<i>Multipoint Control Unit</i>
MTU	<i>Maximum Transmission Unit</i>
NAT	<i>Network Address Translation</i>
NSF	<i>National Science Foundation</i>
OSI	<i>Open Systems Interconnection</i>
P2P	<i>Peer to Peer</i>
POTS	<i>Plain Old Telephone Service</i>

QoS	<i>Quality of Service</i>
RFC	<i>Request for Comments</i>
RSVP	<i>Resource Reservation Protocol</i>
RTCP	<i>Realtime Transport Control Protocol</i>
RTP	<i>Real-Time Transport Protocol</i>
RTPC	Rede de Telefonia Pública Comutada
RTSP	<i>Real-Time Streaming Protocol</i>
SAP	<i>Session Announcement Protocol</i>
SDP	<i>Session Description Protocol</i>
SIP	<i>Session Initiation Protocol</i>
STFC	Serviço Telefônico Fixo Comutado
STUN	<i>Session Traversal Utilities for NAT</i>
TCP	<i>Transmission Control Protocol</i>
TSAP	<i>Transport Service Access Point</i>
UDP	<i>User Datagram Protocol</i>
URA	Unidade de Resposta Audível
VoIP	<i>Voice over IP</i>
VPN	<i>Virtual Private Network</i>
WAN	<i>Wide Area Network</i>



SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	CONTEXTUALIZAÇÃO DO TRABALHO.....	14
1.2	MOTIVAÇÃO DO TRABALHO.....	15
1.3	JUSTIFICATIVA DO TRABALHO.....	15
1.4	OBJETIVOS GERAIS.....	16
1.5	OBJETIVOS ESPECÍFICOS.....	16
1.6	METODOLOGIA DE DESENVOLVIMENTO DO TRABALHO.....	16
1.7	PRINCIPAIS RESULTADOS.....	17
1.8	PRINCIPAIS CONTRIBUIÇÕES.....	17
1.9	ORGANIZAÇÃO DO TEXTO.....	17
2	FUNDAMENTAÇÃO TEÓRICA.....	19
2.1	CARACTERIZAÇÃO DOS <i>CONTACT CENTERS</i>	19
2.1.1	Perspectiva histórica do <i>Call Center</i> ao <i>Contact Center</i>.....	20
2.2	<i>VOIP</i> E TELEFONIA <i>IP</i>	23
2.2.2	Características do <i>VoIP</i>.....	23
2.2.2	Vantagens e desvantagens da telefonia <i>IP</i>	24
2.3	FUNÇÕES <i>VOIP</i>	24
2.3.1	Sinalização	25
2.3.2	Transmissão da voz	25
2.4	<i>CODEC</i>	27
2.5	<i>VoIP</i> EM UM CENÁRIO DE CONVERGÊNCIA TECNOLÓGICA.....	27
2.5.1	Características de uma chamada <i>VOIP</i>.....	28
2.5.2	Jitter.....	31
2.6	<i>ASTERISK</i>	31
2.6.1	Arquitetura e funcionamento.....	33
2.6.2	Protocolo <i>IAX</i>.....	35
3	PROBLEMÁTICA.....	40
3.1	DESCRIÇÃO DO PROBLEMA.....	40
3.2	TRABALHOS RELACIONADOS.....	43

4	DESENVOLVIMENTO.....	51
4.1	ARQUITETURA E PROJETO.....	51
4.2	COMPONENTES.....	53
4.3	PLANO DE NUMERAÇÃO.....	58
4.4	CONFIGURAÇÃO DO AMBIENTE.....	59
4.4.1	Registro dos Ramais SIP e Configuração das Chamadas.....	60
4.4.2	Configuração da Transferência De Chamadas.....	62
4.4.3	Configuração da Sinalização Múltipla e da Captura Chamada.....	63
4.4.4	Configuração das Salas de Conferência.....	65
4.4.5	Configuração da Integração com a RTPC.....	67
4.4.6	Configuração da Comunicação Entre Centrais Asterisk.....	69
4.5	TELAS DO SISTEMA.....	70
5	TESTES.....	76
5.1	ESTABELECIMENTO DE UMA CHAMADA ENTRE DOIS TERMINAIS.....	77
5.2	ESTABELECIMENTO DE MÚLTIPLAS CHAMADAS SIMULTÂNEAS.....	78
5.3	CONFERÊNCIA.....	87
5.4	TRANSFERÊNCIA DE CHAMADA, CAPTURA.....	96
5.5	INTEGRAÇÃO COM A RTPC.....	97
5.6	ESTABELECIMENTO DE CHAMADAS ENTRE AS DUAS CENTRAIS.....	99
6	CONCLUSÃO.....	102
	REFERÊNCIAS.....	104

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO TRABALHO

Desde o período inicial do surgimento da telefonia como canal de comunicação formal, pode-se afirmar que, de modo geral, a telefonia não mudou seu comportamento operacional de forma relevante. Diversas novas tecnologias surgiram ao longo do tempo para agregar facilidades a uma nova era dos serviços de telefonia. No entanto, o funcionamento básico, assim como as características principais, ainda se mantém. Na verdade, nesse período de amadurecimento tecnológico, apenas foram agregados elementos acessórios que, no final das contas, deram origem à motivação para criação de melhorias que impactaram positivamente na usabilidade e na acessibilidade dos recursos relacionados.

Em meados de 1996 quase todo estudante já possuía um endereço de *e-mail* que possivelmente já tinha a sua importância como ferramenta de trabalho ou estudo. Porém, para a maioria das pessoas a internet ainda era algo abstrato e imensurável. Nesse período era muito difícil para essa tecnologia (*internet*) alcançar uma visibilidade relevante dentro do domínio mercadológico da época, haja visto que não existia, de fato, uma real importância para essa nova ferramenta que surgira a tão pouco tempo. Diante desse cenário ela nunca sairia do domínio físico e lógico das universidades e dos laboratórios de pesquisa. E foi inserido nesse contexto que surgiram os primeiros protótipos do que se pode sinalizar como o início da telefonia através da internet ou propriamente Voz sobre *IP* (*VoIP*). De forma sintetizada pode ser vista como uma rotina formal que tem como objetivo controlar uma segmentação de pacotes de áudio e vídeo, divididos em pacotes menores, para que, dessa forma, possam ser transmitidos ou enviados por uma rede *IP* de forma aleatória, e reorganizados corretamente no lado destino, para que possa ser concretizada uma comunicação.

1.2 MOTIVAÇÃO DO TRABALHO

O desenvolvimento deste trabalho tem como propósito apresentar ao mercado das empresas de médio e pequeno porte, uma gama de recursos tecnológicos baseados na tecnologia de voz sobre *IP (VoIP)*, que hoje só estão disponíveis para as grandes corporações.

Observa-se que com a convergência eminente das tecnologias utilizadas na telefonia convencional (comutação de circuitos) e na telefonia *VoIP* (comutação de pacotes) ocorreram melhorias na qualidade do serviços de telefonia de uma forma geral. Pode-se afirmar que essa tendência, em conjunto com uma boa utilização das facilidades e funcionalidades já existentes nos ambientes de *Contact Centers* ou cenários relacionados (como, por exemplo: um setor de agendamento de consultas de uma clínica médica), resultam em uma melhoria visível nos resultados de produção desse tipo de estrutura.

A ideia principal é a integração dos principais canais de comunicação com o cliente (redes sociais, URA, SMS, recursos de *PBX IP* etc.) em um único canal de utilização, tendo como lastro principal a tecnologia *VoIP*, desenvolvida com base em ferramentas livres. Desse modo, a solução resultante deste trabalho se torna viável financeiramente para uma grande parcela do público corporativo (médio e pequeno porte) que até então estavam sem uma opção com esse perfil, por conta do fator financeiro.

1.3 JUSTIFICATIVA DO TRABALHO

As tecnologias *VoIP* e suas facilidades, implementadas com ferramentas gratuitas, surge neste momento como divisor de águas, em um mercado que até então só tinha como parâmetro de tendência tecnológica a telefonia convencional (comutação de circuitos), outrora ofertada somente pelas grandes operadoras de telefonia e agora disponíveis para todos os nichos de mercado. Baseado nesse contexto observa-se uma lacuna no mercado de soluções (*softwares*) computacionais direcionados especificamente para ambientes de *Call Center* e que sejam baseadas em ferramentas

livres, viabilizando o acesso financeiro dos micros e pequenos empresários a esse tipo de recurso tecnológico.

1.4 OBJETIVO GERAL

Este trabalho tem como objetivo geral desenvolver uma ferramenta de integração de canais de comunicação com o cliente (redes sociais, mensagens de texto, CRM, URA, SMS, recursos de PBX IP etc.) utilizando prioritariamente softwares e ferramenta livres.

1.5 OBJETIVOS ESPECÍFICOS

- Projetar e configurar um sistema de telefonia IP baseado no PBX Asterisk;
- Verificar a funcionalidade e aplicabilidade do sistema;
- Desenvolver uma ferramenta de integração de baixo custo financeiro;
- Testar o sistema e apresentar os resultados.

1.6 METODOLOGIA DE DESENVOLVIMENTO DO TRABALHO

Este trabalho foi desenvolvido com base em consultas em normas, livros, padrões, artigos e documentações de referência de programas que serviram como base teórica para o desenvolvimento dessa ferramenta conforme os objetivos referenciados na seção 1.5.

A metodologia utilizada para o desenvolvimento do trabalho segue a seguinte ordem:

- Realizar pesquisa no mercado para identificar principais fornecedores de centrais telefônicas com interface VoIP e sistema de gerenciamento;
- Desenvolver uma interface de compatibilidade com as APIs (Application Programming Interface) para os principais fornecedores de infraestrutura de Telecom;

- Utilizar software *SiPp* para simular o comportamento da ferramenta em cenários diversos, levando em consideração os ambientes de redes heterogêneos utilizados em *LAN's* e *WLAN's*;
- Desenvolver e implementar a ferramenta em um ambiente controlado (laboratório) e em um ambiente corporativo real;
- Realizar os testes de validação em ambiente de laboratórios;
- Realizar os testes em ambientes reais de produção.

1.7 PRINCIPAIS RESULTADOS

Os principais resultados esperados são:

- Que esta ferramenta, baseada em software livre, disponibilize para as empresas de médio e pequeno porte, o acesso a recursos tecnológicos antes não acessíveis, que utilizam rotinas profissionais de um *Contact Center*;
- Viabilizar um grande poder de acoplamento aos sistemas de telefonia convencionais outrora somente acessíveis por grandes corporações;
- Desenvolver uma ferramenta de baixo custo, para que, desse modo, ela tenha penetrabilidade no mercado e permita o acesso das empresas de menor porte a esse recurso.

1.8 PRINCIPAIS CONTRIBUIÇÕES

Disponibilizar para o mercado uma ferramenta de integração dos principais canais de comunicação com o cliente (redes sociais, mensagens de texto, *CRM*, *URA*, *SMS*, recursos de *PBX IP* etc.) que seja competitiva e de baixo custo, baseada em softwares livres.

1.9 ORGANIZAÇÃO DO TEXTO

A seguir é apresentada uma descrição de como este trabalho está

estruturado:

Capítulo 2 : Introduz os conceitos de *Contact Centers* e de redes de transmissão de dados; explica de forma sucinta as estruturas e conceitos da telefonia *IP*, assim como, transmissão de dados multimídia em tempo real sobre redes de computadores; descreve também o funcionamento do *PBX Asterisk* e dos sistemas de telefonia convencionais e seus serviços básicos;

Capítulo 3: Apresenta a problemática, aqui identificada como a ausência no mercado de *Call Centers* de uma ferramenta de integração dos principais canais de comunicação com o cliente e que seja acessível financeiramente para as micros e pequenas empresas. A seguir são apresentados alguns trabalhos relacionados que abordam temáticas similares a problemática sugerida neste trabalho.

Capítulo 4: É apresentado o projeto e a configuração da ferramenta proposta, juntamente com a documentação do processo e uma visão básica das principais telas do sistema;

Capítulo 5: Apresenta os testes realizados e os resultados obtidos;

Capítulo 6: É feita uma análise dos resultados obtidos e das dificuldades encontradas, são feitas as considerações finais juntamente com algumas perspectivas sobre o tema e algumas sugestões para trabalhos futuros.

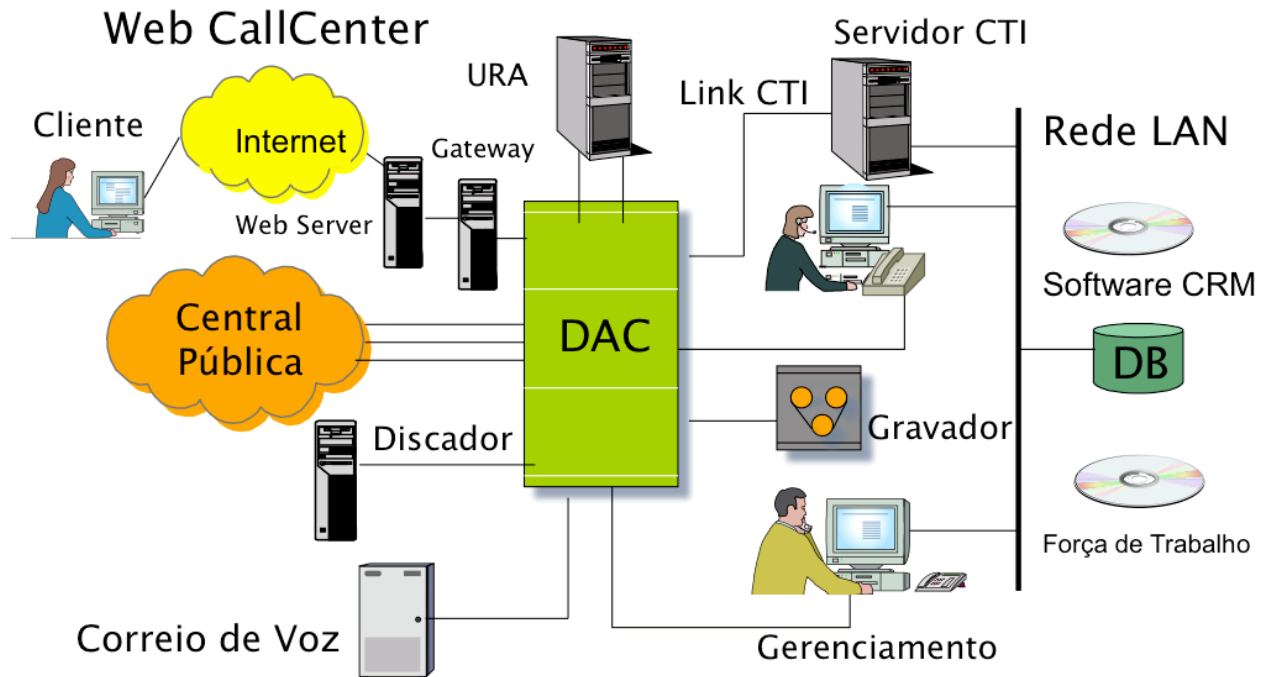
2 FUNDAMENTAÇÃO TEÓRICA

2.1 CARACTERIZAÇÃO DOS CONTACT CENTERS

O cotidiano de um *Contact Center* é sempre multidisciplinar e complexo. Devido a essa condição é de suma importância ter uma visão de suas origens, do seu presente e de suas tendências futuras.

Este capítulo descreve uma perspectiva histórica dos *Contact Centers*, até o seu momento atual, conceitos fundamentais da área e aspectos em aberto para futuras análises e investigações. A Figura 1 ilustra uma visão geral da arquitetura de um *Call Center*.

Figura 1 - Arquitetura convencional do ambiente de *Call Center*



Fonte: Adaptado de Tanenbaum (2011, p.325).

2.1.1 Perspectiva histórica do *Call Center* ao *Contact Center*

Pode-se afirmar que a ideia do serviço de *Call Center* surgiu logo após as empresas inserirem os telefones nas estruturas organizacionais. Nesse momento as secretárias passam a ter o telefone como principal aliado.

O conceito de *Call Center* formalmente só apareceu nos anos 80 (GABALLA *et al.*, 1979; Cardoso, 2000; Hawkins *et al.*, 2001) no entanto, os usuários sempre tinham a possibilidade de telefonar diretamente para a empresa (organização) e falar com um representante, que hoje são chamados formalmente de operadores. Nesse momento a única tecnologia disponível era o telefone. O operador só tinha duas possibilidades: ou conseguia de imediato responder as perguntas ou solicitava o nome e o número de telefone do cliente para lhe retornar a ligação posteriormente. Muitas vezes, para se ter as respostas para os questionamentos dos usuários, o operador precisava fazer uma pesquisa manual em registros, fichas e outros formulários impressos. Tratava-se de uma atividade que consumia muito tempo (HAWKINS *et al.*, 2001).

Em meados dos anos 70 surge o advento do computador que possibilitou às empresas iniciarem uma melhoria em seus serviços telefônicos, que, já na época, serviam como base de apoio aos clientes. Nesse cenário, com a utilização dos computadores, os operadores conseguiam ter acesso mais rápido às informações, mesmo que de forma superficial. Esse breve acesso ocorria enquanto falavam com os clientes. Esse avanço tecnológico possibilitou, por exemplo, uma redução no tempo de pesquisa e da ligação de retorno para o cliente.

Nesse momento, os *Call Centers* inserem, em seu dia a dia, os primeiros equipamentos de comutação. No entanto, os primeiros (*PBX*) eram ainda muito limitados na sua capacidade de suportar múltiplas chamadas e de efetuar a sua distribuição (KOOLE *et al.*, 2002). O *PBX*, simplesmente, disponibilizava uma relação de um para um entre uma ligação que chegava, oriunda de um cliente, e um operador.

A chegada do PC (computador pessoal), já nos anos 80, viabilizou a migração do controle de algumas funcionalidades telefônicas para os computadores. Com a evolução dos computadores foi possível ampliar a capacidade de

processamento dos comutadores, possibilitando, dessa forma, um relevante crescimento na capacidade de gerenciar volumes maiores de ligações telefônicas e de as distribuir para os operadores do *Call Center* que estivessem disponíveis no momento.

Com a evolução das estruturas digitais de telefonia pública, em paralelo ao desenvolvimento do cenário computacional da época, surgiu a possibilidade de tratar as demandas dos clientes de forma imediata. Torna possível, por exemplo, verificar um inventário, processar uma reclamação e até mesmo aceitar uma encomenda. Esse foi o momento em que as empresas conseguiram disponibilizar um serviço de atendimento quase completo pelo telefone aos seus clientes. Essa capacidade de fornecer um serviço completo através de uma chamada telefônica teve um impacto significativo em todo o mundo na forma de organizar e realizar negócios (HAWKINS *et al.*, 2001).

Em meados da década de 1980 tem-se o início da utilização da tecnologia *CTI (Computer Telephony Integration)* para dar suporte às diversas interações telefônicas. Essa tecnologia permite associar uma chamada ao contexto relacionado ao cliente, incluindo os dados pessoais, o serviço solicitado e as transações possivelmente efetuadas durante a execução da chamada. Foi desenvolvido, em conjunto com *CTI*, a tecnologia *IVR (Interactive Voice Response)*, que torna possível à aplicação interagir com o cliente, de tal forma que ele possa escolher se quer ser atendido por uma unidade de resposta audível ou por um operador.

Com a utilização de forma generalizada de canais que possibilitam uma interação além do alcance da telefonia, como o e-mail, fax, Web, o conceito de *Call Center* (central de atendimento) foi se tornando bem mais amplo, passando a ser denominado como *Contact Center* (centro de contatos).

Com isso, a definição de fila de espera de ligações telefônicas, relativo a todas as chamadas que se encontram em espera até serem atendidas, evoluiu para uma visão mais elaborada de fila única de processamento. Nesse novo cenário de fila é possível processar todas as formas de interações do cliente com a empresa, como chamadas telefônicas comuns, *e-mails*, acessos via *Web*, fax, *sms*, entre outras.

O domínio de um *Contact Center* abrange uma vasta gama de recursos que viabilizam o fornecimento de serviços via um ou vários canais. Esse conjunto de

recursos é constituído tipicamente pelos operadores que interagem com os utilizadores da organização e as tecnologias de suporte às comunicações e operações (PICHITLAMKEN *et al.*, 2003).

A necessidade de gestão de interações provenientes de diversos canais, a sua integração com os sistemas já existentes nas instituições e a evolução constante resultante da ligação estratégia-tecnologia-logística-RH, trouxe por sua vez um conjunto de problemas de complexa resolução [...] (Adria e Chowdhury, 2004).

Em geral, o conceito de *Call Center* se traduz em um crescimento significativo na complexidade dos processos, na gestão do RH e nos recursos tecnológicos envolvidos. Pode-se afirmar que os *Call Centers surgiram como um fator de potencialização da competitividade do cenário empresarial*. Nesse contexto pode-se destacar alguns aspectos tais como:

- Aumento da produtividade;
- Aumento da eficiência e da qualidade do serviço;
- Uso dos canais preferidos dos utilizadores de um modo eficiente (telefone, *e-mail*, *Internet*, *sms*, aplicativos de mensagens);
- Uso do telefone de forma intensiva, nas suas vertentes fixa e móvel;
- A retenção de utilizadores e aquisição de novos através de abordagens via *CRM*;
- Realização de processos de fidelização de utilizadores, através da melhoria da qualidade dos serviços prestados;
- Realização de processos de gestão remota de negócio.

Os *Contact Centers* tem como principal característica suportar diversos tipos de serviços como, por exemplo, o suporte técnico a clientes, sanar dúvidas, apoio comercial, registro de pedidos e de encomendas, *home-banking*, apoio médico, serviço de emergência, entre outros. Esses serviços, nos quais predomina o contato do exterior para a organização, através do *Contact Center*, denominam-se de serviços de *inbound* (Koole *et al.*, 2002).

Com a aglutinação dos recursos para contato com o cliente em um *Contact*

Center, fica fácil a execução de campanhas de venda, promoção de produtos, pesquisas, pedido a fornecedores, cobrança, entre outros. As pesquisas podem ser, por exemplo, de satisfação dos clientes, de prospecção de mercado, sondagem de intenção de voto ou até mesmo estudos sócio econômicos. Todos esses serviços, em que predomina o contato da organização com o exterior, denominam-se de serviços ou campanhas de outbound (PICHITLAMKEN *et al.*, 2003).

2.2 VOIP E TELEFONIA IP

2.2.1 Características do VoIP

A telefonia tradicional é muito importante na caracterização dos parâmetros básicos da telefonia *VoIP*. Muitas das características necessárias para a efetivação de uma chamada de telefonia por *IP* tem como referência os conceitos utilizados nas chamadas realizadas por comutação de circuitos. Como demonstrado a seguir, a comunicação de Voz por *IP* possui diversas características e requisitos de maneira análoga a telefonia tradicional, como *Jitter* (variação estatística do atraso na entrega de dados em uma rede) e a qualidade de serviço oferecida (*QoS – Quality of Service*). Porém, existem várias diferenças que tornaram o *VoIP* uma alternativa mais econômica e prática para os usuários. Essas mesmas diferenças geram novos problemas a serem resolvidos.

“O *VoIP (Voice over Internet Protocol)* realiza a comunicação telefônica por uma rede de dados “ (KUROSE, 2010, p.25). Posteriormente, foram definidas as técnicas de empacotamento, transmissão de amostras de voz e método de sinalização na transmissão de voz sobre um protocolo de Internet. O *VoIP* e os protocolos a ele relacionados tiveram um grande desenvolvimento na última década devido a sua vantagem econômica em relação a telefonia tradicional (RTPC). Por ser uma tecnologia de comutação por pacotes, o seu custo é menor que a telefonia tradicional na qual se utiliza comutação por circuitos. A migração da telefonia comutada para *VoIP* se tornou interessante para concessionárias de telefonia, tanto locais quanto de longa distância, operadoras de telefonia, provedores de serviços Internet e serviços de comunicação.

“Além da vantagem econômica, o *VoIP* possibilita várias aplicações relacionadas a utilização da Voz” (HERSENT, GUIDE, PETIT, 2002, p.66) como, por exemplo, a comunicação entre usuários utilizando roteamento de ligações, caixa de mensagens, *UMS (Unified Message System)*, videoconferência e base de dados de contatos.

2.2.2 Vantagens e desvantagens da telefonia *IP*

A telefonia *IP* é sem dúvida uma excelente opção para os ambientes corporativos haja visto que possibilita a utilização de apenas uma rede física, dispensando, dessa forma, investimentos em estruturas dedicadas somente a telefonia comutada por circuito. Além disso, há também o aspecto gerencial, pois nesse cenário tem-se apenas um ponto de monitoramento dos ativos da rede, assim como facilidade em expansões futuras.

As redes que utilizam comutação de pacotes apresentam uma grande tolerância a falhas em seus nós: é, portanto, mais eficaz e eficiente por não necessitar de alocação exclusiva de recurso.

No entanto, é necessário também conhecer as desvantagens dessa tecnologia. Por utilizar uma rede que depende de vários equipamentos ativos (alimentação elétrica), em um caso de ausência da energia elétrica a rede para de funcionar, o que não acontece com a RTPC. Em caso de emergência a localização da chamada muitas vezes não é precisa devido ao processo de roteamento dos pacotes de dados. Para evitar problemas de qualidade da voz e das chamadas faz-se necessário a implementação de um controle de *QoS (Quality of Service)*, para que se possa garantir o mínimo de qualidade nas chamadas.

2.3 FUNÇÕES *VOIP*

Para entender o funcionamento da ferramenta proposta neste trabalho, faz-se necessário conhecer o funcionamento do tráfego de voz por *IP*. Para possibilitar o tráfego de telefonia *VoIP*, diversas características foram migradas do modelo da telefonia tradicional, por se tratar de um modelo conhecido e confiável. Porém, foram

necessárias algumas adaptações para essa nova tecnologia.

Uma das principais diferenças entre esses modelos é a forma de comutação. Enquanto a telefonia tradicional utiliza a comutação por circuitos, o *VoIP* faz uso da comutação por pacotes. Por isso, a telefonia *VoIP* abrange diversos protocolos Internet para realizar as mesmas funções exercidas na telefonia tradicional, como a sinalização (*SIP - Session Initiation Protocol*) e a transmissão de voz (*RTP – Real Time Transport Protocol*). Esses dois protocolos são essenciais para entender o tráfego de voz por *IP*.

2.3.1 Sinalização

“Sinalização é a troca de informações para estabelecer, monitorar e finalizar conexões entre usuários” (HERSENT, GUIDE, PETIT, 2002, p.67). Existe a necessidade da sinalização para otimizar e gerenciar o sistema de comunicação.

As redes tradicionais de telefonia evoluíram substancialmente nessa área. Hoje a RTPC utiliza o *SS7 (Signaling System7)* para sinalização. O *SS7* usa um canal diferenciado, fora da banda, para sinalização, ou seja, ele possui um canal dedicado para troca de informações sobre o sistema.

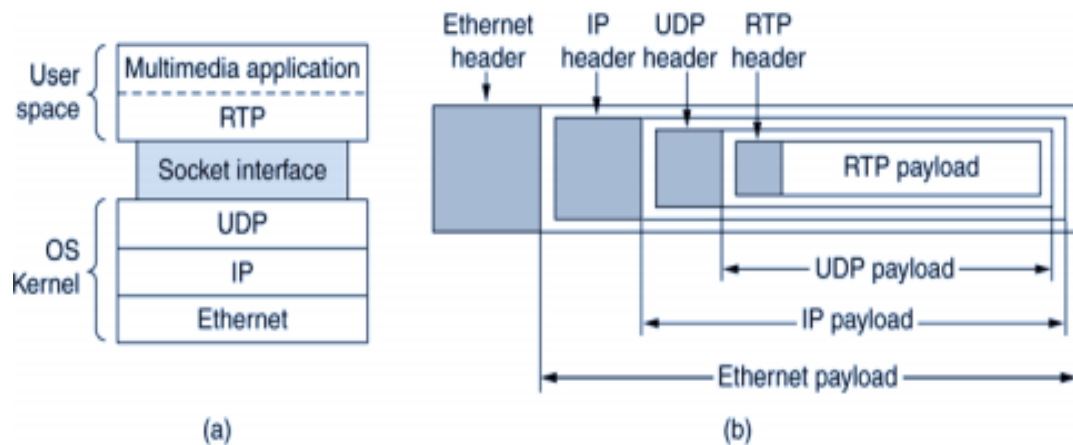
O *VoIP* apresenta diversas opções para sinalização como *H.323*, *SIP (Session Initiation Protocol)*, *H.248*, *MGCP (Media Gateway Control Protocol)* e o *SCCP (Skinny Client Control Protocol)*. Alguns gateways *VoIP* conseguem iniciar o *SS7* com redes RTPC. Os protocolos *MGCP* e *SCCP* são conhecidos como protocolos de interação entre cliente e servidor, onde existe uma sinalização de controle de chamadas baseada na troca de notificações nos *endpoints* e *gateways*. Por outro lado, os protocolos *SIP* e *H.323* possuem um mecanismo otimizado para iniciar e finalizar ligações e interpretar mensagens de controle.

2.3.2 Transmissão da voz

“O protocolo responsável pela transmissão de voz ponto-a-ponto é o *RTP (Real-Time Transport Protocol)* conforme a Figura 2” (HERSENT, GUIDE, PETIT, 2002, p.67). Esse é o protocolo que transmite o fluxo de dados em tempo real, para áudio e

vídeo, incluindo serviços como identificação do tipo de carga transportada para designar a sua aplicação, selo de tempo para sincronização dos pacotes e monitoramento de entrega, devido ao número de sequência. Além disso, possibilita a transmissão *Unicast*, na qual tem-se somente um receptor dos pacotes criado, como também *Multicast* na qual a mensagem é entregue a diversos usuários. Para garantir a efetividade desses serviços existe o protocolo de apoio *RTCP* (*Real-time Transport Control Protocol*) que regulamenta a entrega de dados, identifica os pacotes e os controla. Porém, o RTCP não garante qualidade de serviço e nem reserva de recursos. Ambos os protocolos utilizam o protocolo *UDP*, de forma a serem independentes da averiguação de recebimento de pacotes, o que torna os protocolos mais rápidos e dinâmicos.

Figura 2 - Estrutura básica do protocolo RTP



.Fonte: Hersent, Guide, Petit (2002, p. 67).

O protocolo *RTP*, conforme Figura 2, possui em seu cabeçalho as informações necessárias para o controle do fluxo de voz e determinação do espaçamento entre pacotes durante o fluxo de mensagens. Para exercer essas tarefas o cabeçalho contém *SSRC*, que faz a identificação da fonte, enquanto o *RTCP* faz a sincronização e o *timestamp*, que representa o tempo no qual o pacote foi criado e o cálculo do *jitter* (desvio de tempo médio entre pacotes) do sistema. Essas informações são essenciais para que o tráfego de voz seja transmitido e processado de maneira

ordenada e correta, senão pode haver troca de ordem das palavras ou uma demora acima do normal entre as palavras.

2.4 CODEC (CODIFICADOR / DECODIFICADOR)

A voz humana pode ser interpretada fisicamente como um sinal analógico que possui amplitude e frequência. No sistema *VoIP*, este é o sinal de entrada e o sinal de saída. “O elemento responsável por realizar o empacotamento da voz para transmissão via *IP* é o *codec*” (TANENBAUM, 2011, p.41).

Os *codecs* são responsáveis pela codificação e decodificação entre os dispositivos analógicos e os digitais. Eles convertem os sinais analógicos de voz para canais digitais de 64 *Kbps*. Existem diversos tipos de *codecs*, possuindo cada um características específicas como método de codificação, taxa de compressão e atraso. “A rede RTCP tem disseminado o uso do *codec* para tráfego de voz” (HERSENT, GUIDE, PETIT, 2002, p.70). O valor de 64 *Kbps* é a largura de banda reservada em canais de voz. Na tecnologia *VoIP*, se consegue chegar a valores maiores de compressão para o tráfego de voz, possibilitando assim maior eficiência na transmissão. “O *codec G.729* que é o mais utilizado atualmente para o tráfego de voz por *IP* chega a uma taxa de empacotamento de dados de 16 *Kbps*” (HERSENT, GUIDE, PETIT, 2002, p.70).

Os *codecs* possuem taxas de bits diferentes, ou seja, pode-se ter uma variação no tamanho dos pacotes. As taxas de bits podem ser divididas em três tipos:

- *CBR (Constant Bit Rate)*: A taxa de transmissão de bits será constante ao longo da chamada.
- *VBR (Variable Bit Rate)*: A taxa de bits varia durante a chamada, possibilitando assim que o canal seja otimizado.
- *AVR (Available Bit Rate)*: A taxa de *bits* também é variável como no *VBR*, mas existe uma taxa média mínima pré-definida para a transmissão.

2.5 VOIP EM UM CENÁRIO DE CONVERGÊNCIA TECNOLÓGICA

Voz sobre IP acontece quando se transporta o sinal de voz digitalizado sobre

o protocolo IP. A Figura 3 apresenta um cenário típico de aplicação das tecnologias VoIP, onde observa-se computadores utilizando softphones (software de emulação de um aparelho telefônico convencional), telefones IP e toda uma estrutura de rede que envolve um ambiente extremamente heterogêneo, quando visto pela ótica das tecnologias que interligam as redes de comutação de pacotes. Desse modo, estando todos conectados a um ITSP (Internet telephony service provider) através da internet, pode-se realizar chamadas de um sistema final VoIP para uma linha RTPC, e vice-versa. Este sistema de comunicação está exemplificado na Figura 3.

Figura 3 - Comunicação de voz em uma rede comutada a pacotes



Fonte: Adaptado de Tanenbaum (2011, p.81).

2.5.1 Características de uma chamada VoIP

Como dito anteriormente, *VoIP* é uma tecnologia que pode ser usada para se fazer chamadas telefônicas sobre a rede de Internet. Como é uma tecnologia para transporte de voz, ela se assemelha muito à telefonia fixa tradicional, como se pode destacar nos pontos a seguir:

- principal sinal transmitido é a voz, como no STFC (Sistema de Telefonia Fixa Comutada);
- O *VoIP* estabelece uma comunicação entre dois pontos fixos durante uma chamada em uma rede cabeada. Em uma rede sem fio residencial, o usuário poderá ter uma mobilidade, mas será restrita ao seu imóvel, como o STFC também permite;

- Os *codecs* de voz trabalham a uma taxa máxima de 64 *kbps* (*G.711*);
- Um usuário *VoIP* pode ser nômade, isto é, um telefone *IP* pode estar plugado a qualquer tomada de um ponto de acesso à Internet ou ao serviço de dados do provedor *VoIP*;
- Nas chamadas *VoIP*, não existe o conceito de Longa Distância, todas as chamadas são estabelecidas entre os usuários da rede sem levar em conta a sua localização geográfica.

Para possibilitar a transferência da voz, ou áudio, é necessário compactar e ou descompactar esses fluxos de áudio, o que consiste em transformá-los do formato analógico para digital e vice-versa, visando com isso que uma menor banda do enlace de rede seja utilizada.

Os servidores *VoIP* (utilizando os *codecs*) são os responsáveis pelo encaminhamento das chamadas, bem como do seu recebimento, além de atuarem como *middlewares*, ou seja, intermediadores, pois podem realizar a comutação de protocolos e *codecs*, e aplicar os *codecs* a voz. Por exemplo, em um cenário *VoIP*, pode acontecer de um usuário A estabelecer comunicação com um usuário B. No entanto, o usuário A pode estar utilizando o codec *G.723*, ao passo que o usuário B está utilizando o codec *G.729* e essa comunicação ocorre sem problema algum, pois fica a cargo do servidor *VoIP* a comutação entre esses dois algoritmos de forma transparente ao usuário.

No entanto, a figura do servidor é desnecessária quanto a função de *middleware*, quando for utilizado o protocolo de sessão *SIP* (*Session Initiation Protocol*) e tendo os usuários A e B os mesmos *codecs* em comum, pois dessa forma o fluxo *RTP* segue diretamente entre A e B. Desse modo, o fluxo *RTP* é considerado um protocolo ponto-a-ponto.

Em grande parte dos cenários que usam a *VoIP*, a integração com a Internet ou mesmo outras *LANs* e *MANs* é indispensável. Em decorrência, surge o problema do atraso no recebimento da voz, sendo este um grande vilão que afeta em muito a qualidade de uma ligação que utiliza o *VoIP*. Várias soluções são propostas para

superar essa dificuldade, sendo a *QoS* (*Quality of Service*) uma linha de pesquisa muito complexa e em pleno desenvolvimento, a qual se propõe a encontrar métodos e maneiras para atenuar essa dificuldade.

A *QoS* pode ser definida como o conjunto de características de um sistema necessárias para atingir uma determinada funcionalidade. O processamento da *QoS* em um sistema distribuído começa com o estabelecimento dos parâmetros exigidos pelo usuário. Esses parâmetros são mapeados e negociados entre os componentes do sistema, assegurando que todos podem atingir um nível de *QoS* aceitável. “Recursos são então alocados e monitorados, havendo possibilidade de renegociação caso as condições do sistema se alterem” (KUROSE, 2010, p.87).

Um ponto importante e preocupante quanto ao uso do *VoIP*, está na necessidade do balanceamento de carga, bem como das chamadas, ou seja, à medida que o número de chamadas aumenta essas devem ser distribuídas entre os servidores *VoIP* de forma que eles tenham um nível de utilização justo, propiciando um melhor aproveitamento dos recursos disponíveis, quer seja hardware ou mesmo da banda de enlace disponível.

A motivação do uso dos esquemas de *QoS* se dá pela necessidade do oferecimento de confiabilidade e qualidade que uma aplicação *VoIP* necessita para que essa ofereça uma qualidade satisfatória. Em contrapartida, na telefonia convencional um canal (circuito fim-a-fim) é disponibilizado para o estabelecimento de uma chamada, e este se mantém até a finalização da mesma. Ao contrário, as aplicações *VoIP* utilizam canais compartilhados, onde não trafegam apenas datagramas contendo voz, e sim diversos tipos de dados que são multiplexados e transferidos por demanda.

Um ponto interessante a ser abordado é o balanceamento de chamadas em servidores *VoIP*, que poderá ser baseado nas características do enlace de dados, nas capacidades de processamento dos *ITSPs* (*Internet Telephony Service Provider*), nas localizações específicas das terminações de *PSTNs*, e fazendo uso da variação dos atrasos de propagação da voz, conhecida como *jitter*, a qual é um parâmetro de *QoS*.

“A utilização dos parâmetros de *QoS* permitirá a definição de políticas para admitir e recusar chamadas, e mecanismos para estabelecimento de rotas e chamadas para os *ITSPs*” (KUROSE, 2010, p.88).

2.5.2 Jitter

O Retardo da voz é gerado por diversos fatores relacionados à transmissão da voz, como *codecs*, e processos como enfileiramento dos pacotes, os quais podem gerar sérios problemas de inteligibilidade da ligação. “Estudos comprovam que atrasos maiores que 150 ms na transferência de voz são sensíveis para os interlocutores” (TANENBAUM, 2011, p.55). Nos padrões de telefonia tem-se como valor estipulado 40 ms para ligações continentais e 80 ms intercontinentais. Na comunicação de Voz por *IP* existe diferentes tipos de retardo ao longo do sistema: O Retardo por *codec*, retardo de empacotamento, enfileiramento e serialização e o retardo de comutação de rede e propagação.

O retardo de *codec* representa o atraso causado pelo processo de digitalização da voz analógica, compressão para a sua transmissão e o seu processo inverso de descompressão. Esse valor varia de acordo com o *codec* utilizado. O retardo referente ao empacotamento, enfileiramento e serialização representa os atrasos no tratamento do pacote contendo dados de voz. Ele sofrerá um atraso para que se complete um pacote para envio, terá um tempo de espera no *buffer* que dependerá da capacidade do canal e da fila no *buffer*. Ao final, existirá um atraso fixo para sincronização de relógios na transmissão. Por último tem-se o retardo de comutação de rede e propagação que identificam os tempos gastos nos elementos que compõem a rede que trafega a voz por *IP*, como os enlaces envolvidos e equipamentos que fazem as multiplexações nas etapas intermediárias da transmissão.

Esses retardos geram o que se conhece como retardo variável denominado também como *Jitter*. A definição de *Jitter*: “É o desvio de tempo médio entre pacotes subsequentes” (TANENBAUM, 2011, p.55). Então, deve-se calcular o *Jitter* de modo dinâmico, calculando o desvio de tempo entre cada pacote enviado, gerando um valor acumulado.

2.6 ASTERISK

O nome *Asterisk* vem do símbolo “*”, muito utilizado no mundo da telefonia. Ele foi desenvolvido, inicialmente, apenas para a plataforma *Unix*, mas, no entanto,

devido aos bons resultados obtidos, ele foi rapidamente difundido como solução de código aberto (*open source*) no cenário convergente do *VoIP* e com isso foram desenvolvidas versões para diversos outros sistemas operacionais, como, por exemplo: *BSD*, *OS X*, *Solaris* e até mesmo para plataforma *Windows*.

VoIP (*Voice over IP* - Voz sobre IP) é o setor de telecomunicações que mais cresce. Seu crescimento está ocorrendo a uma taxa mais veloz do que o crescimento da telefonia móvel. Fabricantes, operadoras e gerentes precisam se adaptar mais rápido do que nunca, mas a curva de aprendizagem é bastante íngreme e requer de fato uma estratégia educacional [...](HERSENT, GUIDE, PETIT, 2002, p.118).

O *Asterisk* é uma implementação em software de uma central telefônica completa, criado pela Digiun (*The Asterisk Company*), que basicamente funciona como um *CTI* (*Computer Telephony Integration*), ou seja, uma ponte de ligação entre a telefonia e o computador. Apresenta a capacidade de gerenciamento e integração de redes de telefonia *IP* e redes telefônicas convencionais, tendo como serviços básicos: conferência, chamada em espera, *URA* (Unidade de Resposta Audível), transferência e distribuição automática de chamadas, caixa postal, dentre outros.

O *Asterisk* é muito flexível, podendo ser customizado para diversas situações. Essas alterações são executadas através do *Script* do plano de discagem. O software é modulado e foi desenvolvido na linguagem C, ou através de *APIs* independentes ou utilizando o *AGI* (*Asterisk Gateway Interface*). A integração com a *RTPC* é feita através de placas de expansão que precisam de certos módulos carregados para dar carga completa no sistema, conforme o cenário solicitado.

A customização do *Asterisk* é toda feita pela edição de parâmetros ou variáveis em um conjunto de arquivos do tipo texto. Essa configuração define todo o comportamento e abrangência do software. Cada tipo de protocolo a ser utilizado define a forma e funcionalidades que podem ser executadas pelos ramais. Com o intuito de facilitar a configuração do *Asterisk*, ele aceita a acoplagem de interfaces gráficas desenvolvidas por terceiros.

2.6.1 Arquitetura e funcionamento

A arquitetura do *Asterisk* tem como principais características a simplicidade e a flexibilidade, tendo como base para essa condição uma gama de *API's* (*Application Programming Interface*) que interconectam todas essas funcionalidades do núcleo do PBX. Internamente o *Asterisk* trata de forma independente todas as conexões de protocolos, *codecs*, *interfaces* (*software e hardware*), assim como as aplicações de telefonia de outros fabricantes. A execução dos módulos de forma desacoplada, deve-se a forma dinâmica de carregamento dos módulos que implementam as *API's*, flexibilizando, dessa forma, a compatibilidade com qualquer camada de *software ou hardware*.

Segundo Meggelen, (2002, p.84), são quatro as *APIs* definidas para os módulos carregáveis, de modo a facilitar a abstração de *hardware* e protocolos:

API de Formato de Arquivo: Trata a leitura e escrita de vários formatos de arquivo no sistema de arquivos;

API de Tradução de Codecs: Carrega módulos de *codecs* para suporte aos mais variados formatos de codificação;

API de Canal: Trata os tipos de conexão de entrada e saída de uma chamada, seja ela uma conexão *VoIP*, *RTPC* ou outro tipo de tecnologia. Módulos dinâmicos são carregados para lidar com os detalhes de nível mais baixo dessas conexões;

API de Aplicação: Possibilita que vários módulos sejam executados para exercer várias funções. Conferências, atendimento automático, correio de voz, e qualquer outra atividade que um *PBX* possa executar, agora ou futuramente, são gerenciados por esses módulos;

Ao utilizar esse sistema de módulos, o núcleo do *Asterisk* não precisa saber os detalhes de como a conexão de uma chamada é feita, ou quais *codecs* serão utilizados. As funções exercidas pelo núcleo do *PBX* são:

Núcleo de Comutação *PBX*: É ele o responsável pela comutação das chamadas entre os usuários. Todas as solicitações de entrada e saída são gerenciadas nesse núcleo, de forma transparente para o usuário;

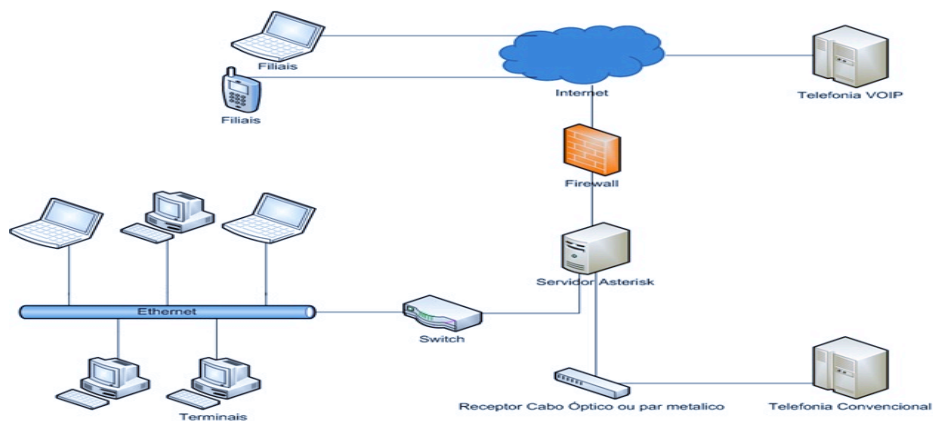
Lançador de Aplicações: Executa as aplicações que realizam serviços como correio de voz, reprodução de arquivos de som, conferências;

Tradutor de *Codecs*: Executa módulos de codec para codificar e decodificar vários formatos de compressão de áudio utilizados na indústria telefônica;

Agendador e Gerenciador de *E/S*: Controla o agendamento de tarefas de baixo nível e o gerenciamento dos recursos do sistema para melhor desempenho em várias condições de carga.

A grande capacidade de customização do *Asterisk*, permite que ele possa carregar e implementar, de forma dinâmica, novos módulos, sempre dentro de uma padronização. Isso agrega uma grande capacidade de acoplamento às novas tecnologias e até a possibilidade de desenvolvimento de novas funcionalidades, conforme demanda. A Figura 4 apresenta uma perspectiva básica da Arquitetura do *Asterisk* e a interligação entre os diferentes componentes.

Figura 4 - Arquitetura do *Asterisk* e interligação entre os diferentes componentes



Fonte: Adaptado de Hersent, Guide, Petit (2002, p. 120).

2.6.2 Protocolo IAX

O protocolo *IAX (Inter Asterisk Exchange)*, tem como função básica a sinalização e o transporte de dados de mídia, conforme descreve a Figura 5. Ele foi desenvolvido pela *Digium*, com o objetivo de realizar a comunicação entre servidores *Asterisk*. O *IAX*, originalmente ficaria responsável por interligar os servidores *Asterisk*. O *IAX* é um protocolo de transporte (como o *SIP*), que utiliza uma única porta *UDP* para as comunicações do tipo: *streams* de sinalização de chamadas e *RTP*. “Essa condição torna o protocolo mais maleável no que se diz respeito a comunicação com *firewalls* e *NAT (Network Address Translation)*” (HERSENT, GUIDE, PETIT, 2002, p.86).

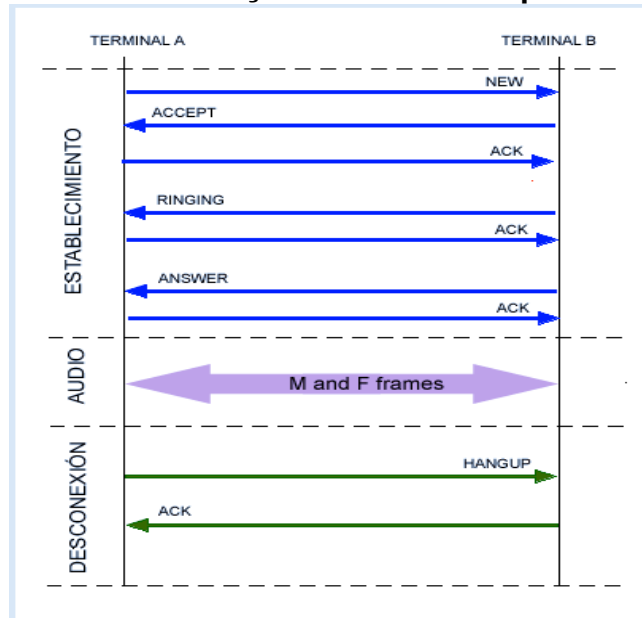
O *IAX* suporta entroncamentos de chamadas, ou seja, ele gerencia a multiplexação de várias sessões em uma única instancia do software. Garante, dessa forma, uma diminuição relevante no *overhead*, e nos canais individuais, sem comprometer os indicadores de qualidade da transmissão. “Essa vantagem é bem significativa em transmissões *VoIP*, nas quais os cabeçalhos dos pacotes ocupam grande porcentagem da banda disponível” (HERSENT, GUIDE, PETIT, 2002, p.87).

O *IAX* é mais eficiente do que o *RTP*, para qualquer número de ligações e qualquer *codec*. O benefício é algo como 2.4 *Kbps* para uma única chamada podendo até triplicar o número de chamadas possíveis a cada 1 *Mbps* com o *codec G.729*. Essa medição é feita no nível físico da rede e utilizando um canal em modo *trunk*.

Conforme sinalizado na Figura 5 pode-se observar que as trocas das mensagens feitas pelo protocolo *IAX*, são executadas de forma binária, viabilizando dessa forma um ganho em relação a utilização da largura de banda. Com essa prática, também evita-se a utilização de programas analisadores sintáticos utilizados para verificação de mensagem de texto, que podem sofrer ataques de tipos diversos.

No entanto, é importante ressaltar que mensagens de natureza binária devem possuir campos bem definidos para armazenamento dos dados, pois é necessário evitar problemas relacionados a numeração de sequência, ordenação de pacotes e tempos relativos das mensagens que são utilizados pelo *IAX*. Devido ao baixo consumo de recursos (banda ao multiplexar), o transporte dos pacotes do tipo mídia e sinalização das chamadas, foram definidos pelos parâmetros do protocolo como três formatos de mensagens classificados como frames.

Figura 5 - Comunicação básica com o protocolo IAX

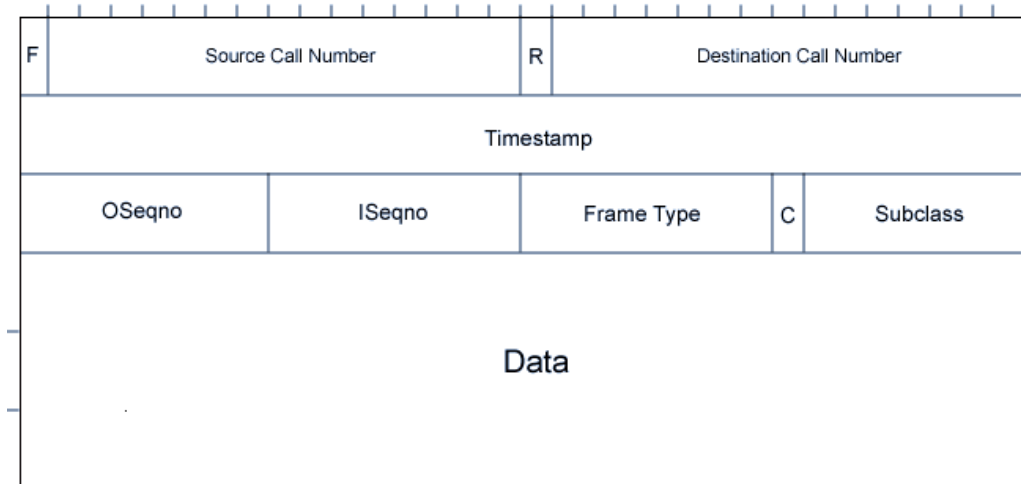


Fonte: Adaptado de Hersent, Guide, Petit (2002, p. 72).

A troca de mensagens no *IAX* é feita através de pacotes denominados frames, que se apresentam de formas: *frames* completos, *mini frames* e *meta frames*.

Frames completos são enviados de forma confiável. Dessa forma, todos os *frames* completos necessitam de uma confirmação imediata após seu recebimento. Essa confirmação pode ser explícita, através de uma mensagem *ACK* (*acknowledgement*), ou implícita, baseada no recebimento de uma resposta apropriada ao *frame* completo enviado. *Frames* completos podem enviar dados de sinalização ou mídia. Geralmente, eles são usados para controlar o início, a preparação e a terminação de chamadas *IAX*, mas também podem ser usados para transportar dados de *streams*, embora isso não seja eficiente (HERSENT, GUIDE, PETIT, 2002, p.88). A Figura 6 apresenta uma visão da estrutura de um frame do protocolo *IAX*.

Figura 6 - Estrutura de um *frame* completo *IAX proxy*



Fonte: Adaptado de Tanenbaum (2011, p.151).

O cabeçalho de um *frame* completo é formado por doze octetos e divididos em 10 campos, que guardam as informações, como, por exemplo, tipo do *frame*, código do chamador e do destinatário, retransmissão, *timestamp*, números de sequência para *frames* que entram e saem (*inbount* e *outbount*) tipo de mensagens enviadas, e algumas informações adicionais.

A seguir são apresentadas as descrições dos campos de um *frame IAX*:

F: É usado para indicar se uma moldura é uma moldura completa ou não. Um valor de 1 neste campo indica que o quadro é um quadro completo e um valor 0 indica que o quadro é algo diferente de um quadro completo ;

Source Call Number (Número de Chamada de Fonte): Trata-se de um número inteiro não assinado de 15 bits que é usado para rastrear um ponto de extremidade de fluxo de mídia no host de origem;

R: Foi definido para o valor 1 se este quadro está sendo retransmitido e o valor 0 para a transmissão inicial;

Destination Call Number (Número de chamada de destino): O mesmo que o número de chamada de origem, mas com destino em vez de fonte;

Timestamp (selo do tempo): Foi criado para carimbar o tempo de cada pacote;

OSeqno: É o número de sequência, da sequência de saída. O campo *OSeqno* sempre começa com 0 e aumenta monotonicamente;

ISeqno: É semelhante ao *OSeqno*, exceto que ele é usado para rastrear a ordenação de quadros de mídia de entrada;

Frame Type (Tipo de estrutura): Define o tipo do quadro;

C: Se C é definido como 1, o valor da Subclasse é interpretado como uma potência de dois. Se C for definido como 0, o valor da Subclasse é interpretado como um simples valor inteiro sem sinal de 7 bits;

Subclass (Subclasse): Tipo de subclasse da mensagem;

Data (Dados): Dados enviados em formato binário.

Mini *frames* são chamados assim pois seu cabeçalho é composto apenas de quatro octetos. Mini *frames* não transportam dados de sinalização ou controle, seu único propósito é transportar um *stream* de mídia numa chamada *IAX* previamente estabelecida, e são enviados sem confiabilidade. Essa decisão foi tomada porque chamadas *VoIP* normalmente podem perder vários *frames* sem que haja degradação significativa na qualidade da chamada, enquanto que o *overhead* criado em transmissões confiáveis aumenta o consumo de banda e diminui a quantidade de mídia trafegada. “Além disso, como chamadas *VoIP* são normalmente transmitidas em tempo real, os frames perdidos tornam-se velhos muito rapidamente para serem reinseridos no *stream* de áudio quando conseguem ser recuperados” (HERSENT, GUIDE, PETIT, 2002, p.120). Na Figura 7 pode-se observar a estrutura de um mini *frame* completo *IAX proxy*.

Figura 7 - Estrutura de um mini *frame* completo IAX proxy



Fonte: Adaptado de Hersent, Guide, Petit (2002, p. 35).

O cabeçalho de um mini *frame* possui apenas três campos: identificador de tipo de frame, número do chamador/remetente e *timestamp*.

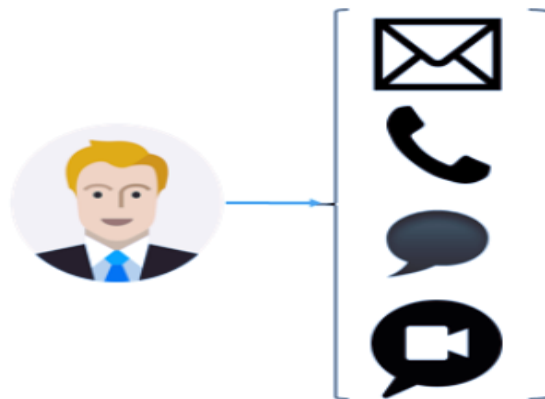
Os meta *frames* servem a dois propósitos: Meta *frames* de vídeo possibilitam a transmissão de *streams* de vídeo com um cabeçalho otimizado, “tendo um propósito similar aos mini *frames*”; meta *frames* de tronco são utilizados no entroncamento (multiplexação) de vários *streams IAX* entre dois *endpoints* em apenas um cabeçalho, para reduzir ainda mais o consumo de banda” (MEGGELEN, 2005, p.38).

3 PROBLEMÁTICA

3.1 DESCRIÇÃO DO PROBLEMA

A ausência no mercado de uma ferramenta para integração dos canais de comunicação (métodos utilizados pelas empresas para entrar em contato com o seu público-alvo) dificulta o gerenciamento dos recursos de acesso ao cliente para os gestores envolvidos no processo de captação e fidelização do cliente. A Figura 8 ilustra a necessidade da convergência dos canais.

Figura 8 - Convergência dos canais de comunicação com o cliente



Fonte: Elaborado pelo autor.

A grande relevância em manter o contato com o cliente é que, a partir dele, pode-se estabelecer uma relação de confiança, ou seja, melhorar o relacionamento entre os serviços prestados pelas empresas e os clientes. Dessa forma, pode-se citar como os principais canais de comunicação com o cliente:

- **Telefone** - geralmente as ligações são feitas em larga escala através de *call center*;
- **Redes Sociais** - esse canal tem ganhado bastante visibilidade no mercado nos últimos tempos, visto que a maior parte da população brasileira possui conta em pelo menos uma mídia social, como, por

exemplo: *Facebook, LinkedIn, Twitter*;

- **Chat Online** - normalmente é acessado diretamente através do site da empresa. É um canal direto por onde a empresa pode tratar dúvidas referentes aos serviços prestados ou aos produtos. Dessa forma, pode-se relacionar como serviços similares: o *Whatsapp*, o *Skype* e outras ferramentas de comunicação instantânea (como o bate-papo do *Facebook*);
- **E-mail** - geralmente é mais utilizado para comunicação formal, é menos utilizado que as ferramentas Online, porém não menos importante;
- **Canais de reclamação** – É necessário que a empresa disponibilize um meio por onde o cliente possa prestar reclamações ou sugestões, como SAC (Serviço de atendimento ao cliente), ouvidoria, pesquisas de opinião, etc;
- **Página de dúvidas frequentes (FAQ)** – é uma forma bem prática de resolver a maior parte das dúvidas mais frequentes.

Observa-se que as ferramentas ou aplicativos na área de soluções para *Call Centers*, ofertadas atualmente pelo mercado de softwares, podem não atender a realidade financeira de micro e pequenas empresas. Por outro lado, soma-se a isso os altos custos com infra-estrutura específica que geralmente é exigida como premissa pelos grandes fornecedores (fabricantes) de soluções para esse seguimento. A Figura 9 mostra a integração de todos os canais de comunicação em uma única ferramenta.

Figura 9 – Integração de canais de comunicação



Fonte: Elaborado pelo autor.

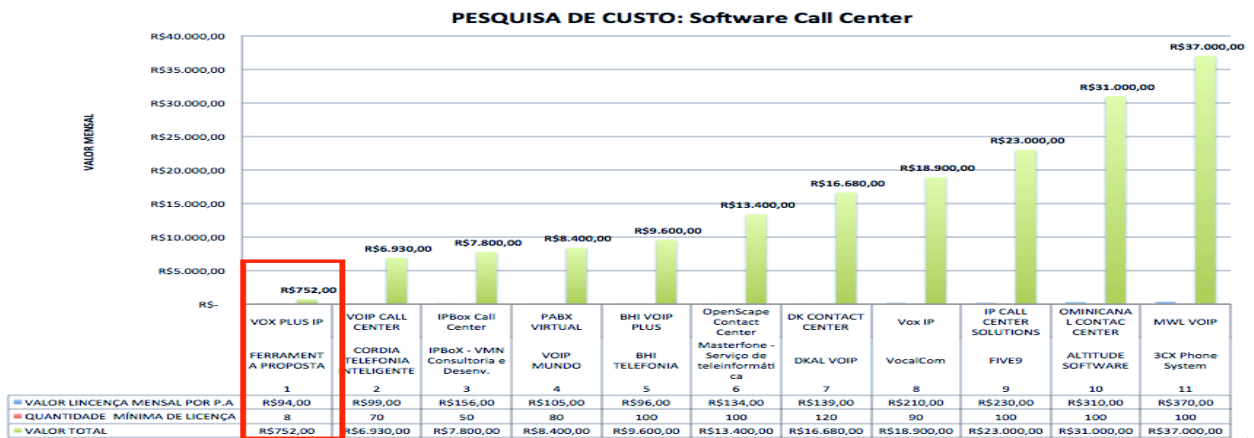
Logo, o desenvolvimento de uma ferramenta de baixo custo, que exija pouca infra-estrutura, mas que funcione de forma satisfatória, é o foco deste trabalho. Adicionalmente busca-se garantir uma melhoria relevante no gerenciamento dos processos envolvidos. A seguir é apresentado um gráfico baseado na Tabela 1 exemplificando um *ranking* de custos em relação a quantidade de licenças mínimas para a aquisição de um solução *VoIP* para integração de um *Call Center*. Para isso foi feita uma pesquisa mercadológica para gerar uma relação de custo benefícios em relação as melhores ferramentas ofertadas pelo mercado. A Figura 10 mostra um gráfico do ranking de custos relacionados as licenças de cada software.

Tabela 1 - Ranking de custo do valor da licença mensal de cada software

NOME FABRICANTE	NOME PRODUTO	VALOR LICENÇA MENSAL POR P.A	QUANTIDADE DE MÍNIMA DE LICENÇAS	VALOR TOTAL
FERRAMENTA PROPOSTA	VOX PLUS IP	R\$ 94,00	8	R\$ 752,00
CORDIA TELEFONIA INTELIGENTE	VOIP CALL CENTER	R\$ 99,00	70	R\$ 6.930,00
IPBoX - VMN Consultoria e Desenv.	IPBox Call Center	R\$ 156,00	50	R\$ 7.800,00
VOIP MUNDO	PABX VIRTUAL	R\$ 105,00	80	R\$ 8.400,00
BHI TELEFONIA	BHI VOIP PLUS	R\$ 96,00	100	R\$ 9.600,00
Masterfone - Serviço de teleinformática	OpenScape Contact Center	R\$ 134,00	100	R\$13.400,00
DKAL VOIP	DK CONTACT CENTER	R\$ 139,00	120	R\$16.680,00
VocalCom	Vox IP	R\$ 210,00	90	R\$18.900,00
FIVE9	IP CALL CENTER SOLUTIONS	R\$ 230,00	100	R\$23.000,00
ALTITUDE SOFTWARE	OMINICANAL CONTAC CENTER	R\$ 310,00	100	R\$31.000,00
3CX Phone System	MWL VOIP	R\$ 370,00	100	R\$37.000,00

Fonte: Elaborada pelo autor.

Figura 10 - Ranking de custo do valor da licença mensal de cada software



Fonte: Elaborado pelo autor.

3.2 TRABALHOS RELACIONADOS

Esta seção tem como objetivo, mostrar trabalhos encontrados que focaram na problemática da integração de canais de comunicação em um *Contact Center*. Espera-se, com este estudo, obter subsídios para a formulação de contribuições relevantes para esta dissertação analisando as principais características de cada abordagem.

Silva (2009) apresenta em seu trabalho um *gateway* de voz para integração entre as redes *IP* e o universo das redes públicas de telefonia tradicional. A principal contribuição de seu trabalho está na integração das tecnologias de redes convencionais com a nova geração, através da utilização de bibliotecas e plataformas abertas, como também ao desenvolvimento de aplicações que possibilitam a geração de novos serviços para empresas e instituições que se utilizem de tal estrutura. O aplicativo desenvolvido para essa solução, foi todo baseado nas redes de nova geração.

Foram criados e testados, como forma de validação da proposta de integração *IP-PSTN*, cinco aplicativos (*Agent Authorization*, *Automatic Dialer*, *Charging*, *Callback* e *Employee Control*), os quais foram integrados com uma plataforma *VoIP* de formato aberto. Todos os aplicativos foram desenvolvidos na linguagem C, possibilitando uma total adesão a estrutura das NGNs, dado que atuam de forma modulada e podem se conectar e desconectar do núcleo do gateway principal conforme circunstância. A seguir é apresentada uma breve descrição das funcionalidades das aplicações desenvolvidas:

Agent Authorization: Serviço de autorização e autenticação onde qualquer ramal logado no gateway, ao solicitar uma linha, terá que se identificar através de um código de usuário e senha para completar a operação.

Automatic Dialer: Serviço de listagem e indexação dos dados de diversos usuários destino que são repassados para o gateway com o objetivo de serem contactados.

Charging: Serviço de tarifação das chamadas executadas pelos usuários do

gateway.

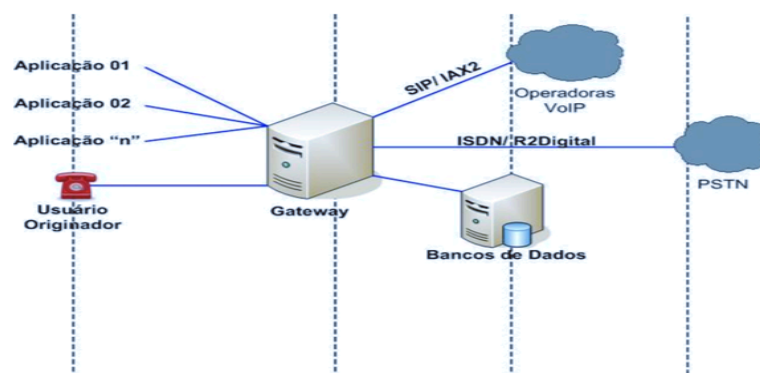
Callback: Tem a função de prover o fluxo de retorno das ligações direcionadas para o SAC (serviço de atendimento ao consumidor).

Employee Control: Serviço de controle de ponto dos funcionários da corporação.

Toda a proposta foi desenvolvida utilizando como lastro principal a estrutura do *Asterisk* com ênfase nas bibliotecas *libpri*, *dahdi* e *chan_ss7*. O foco principal foi validar a integração com as principais operadoras de telecomunicações utilizando para isso padrões de sinalização *ISDN* e *R2Digital*, assim como a intercomunicação com provedores de serviço *VoIP* que por sua vez utilizam os protocolos *SIP* e *IAX2*.

Por fim, foram executados testes com o *Gateway* utilizando a sinalização *SS7 Over IP* para viabilizar uma análise comportamental desses protocolos em um cenário mais heterogêneo (rede das operadoras de telecomunicações), pois normalmente [esse tipo de sinalização é utilizado](#) em ambiente interno. A Figura 11 apresenta uma visão básica da topologia utilizada na solução proposta.

Figura 11 - Topologia do Gateway em ambientes corporativo

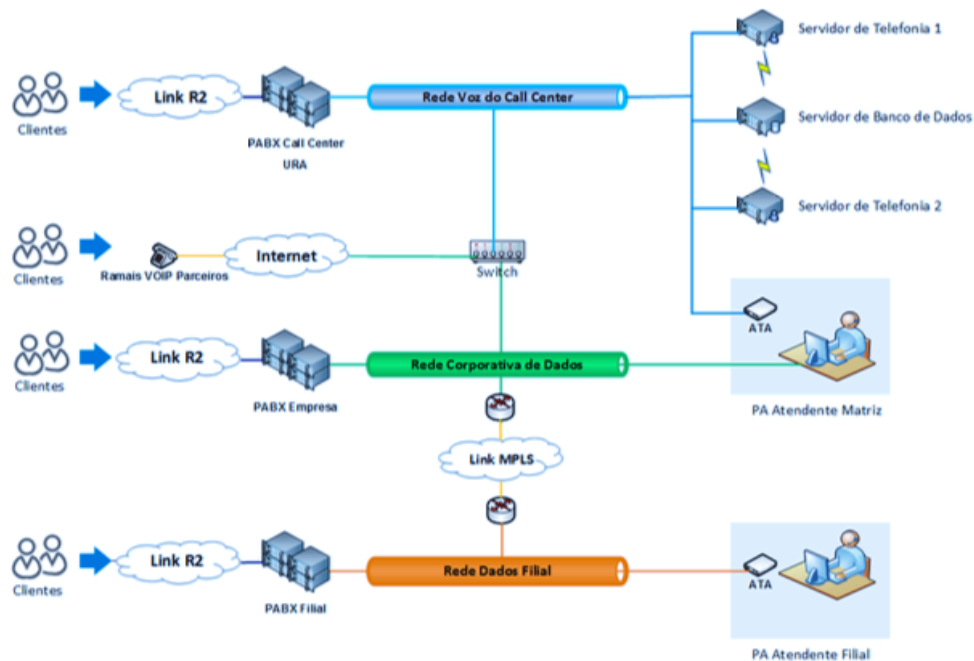


Fonte: Silva (2009, p.60).

Stahelin (2014) propõe a implantação de uma ferramenta de monitoramento para uma rede VoIP corporativa. A ferramenta coleta e fornece dados que servem de

parâmetro para aferir a qualidade dos pacotes *VoIP* que trafegam na rede de dados. Dessa forma, foram elaborados testes voltados exclusivamente para a qualidade do serviço *VoIP*. Os dados resultantes da análise da ferramenta fornecem indicadores de qualidade que colaboram diretamente com o funcionamento e tomada de decisões em um *Call Center*. A Figura 12 apresenta uma representação básica da estrutura que foi utilizada como cenário corporativo para coleta de dados e monitoramento dos pacotes *VoIP*.

Figura 12 - Estrutura de rede do *Call Center* que foi monitorado



Fonte: Stahelin (2014, p.30).

Nesse modelo os servidores do *Call Center* ficam alocados fisicamente no mesmo ambiente, em geral na matriz da empresa. A estrutura que atende as PAs é segmentada entre a matriz e as filiais. A conexão entre as unidades da empresa (matriz e filial) foram feitas com um *link MPLS (Multiprotocol Label Switching)*.

O sistema de *Call Center* está totalmente integrado ao *ERP* da empresa. Na matriz todo o tráfego de dados que é gerado pelas aplicações que compõem o sistema do *Call Center* foi separado através de *VLANs*, de modo que uma das *VLANs* ficou

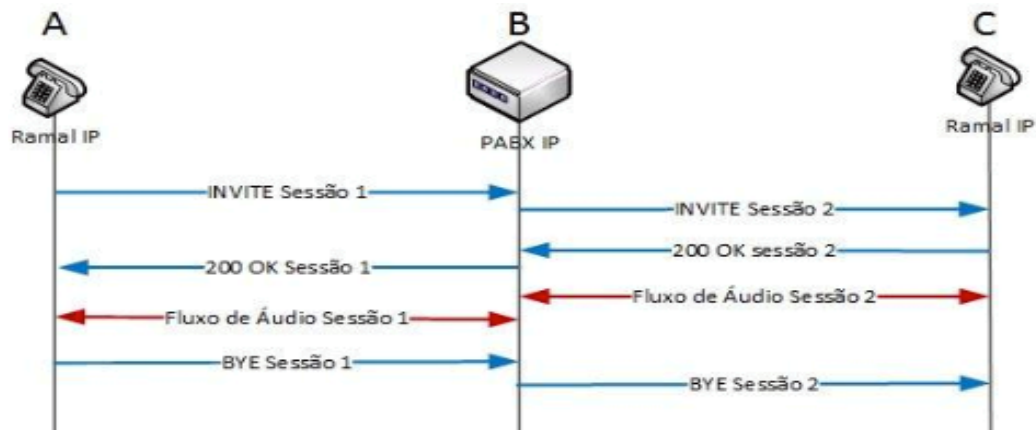
específica para o fluxo de dados *VoIP*, pois somente dessa forma foi possível garantir as métricas necessárias para um bom funcionamento dos sistemas *VoIP*.

Na seção 2.3, Stahelin (2014) fez uma abordagem teórica sobre as técnicas de monitoramento, dentre elas o monitoramento passivo. Um monitoramento passivo é executado basicamente por uma sonda que coleta e reporta informações a um sistema. Já na seção 2.4, foi descrito o funcionamento da ferramenta *VoiPMonitor*, que tem como método de análise de performance o monitoramento de sondas passivas.

Essa aplicação categoriza, classifica e garante a persistência dos dados (pacotes) dos protocolos envolvidos na sessão *VoIP* (*SIP*, *RTCP*, etc.). Pode-se afirmar que esse comportamento do *VoiPMonitor* descreve a principal característica da ferramenta, uma vez que melhora de forma significativa o diagnóstico dos pacotes *VoIP* que circulam na rede.

Os parâmetros monitorados ou indicadores de qualidade (*MOS*, *Jitter*, *perda de pacotes*, etc.) são observados por sessão *VoIP* estabelecida. Nos casos onde uma chamada telefônica requisita mais de uma sessão *VoIP*, *pode-se extrair relatórios bem mais complexos, como, por exemplo, o controle de ligações e de indicadores de qualidade*. A Figura 13 descreve melhor o funcionamento dessa ferramenta de monitoramento. Trata-se de um exemplo de um sistema simplificado com dois ramais *VoIP* e um *PABX IP*. Para esse cenário foi definido que toda a troca de mensagens entre os pontos A e C precisam passar pelo ponto B.

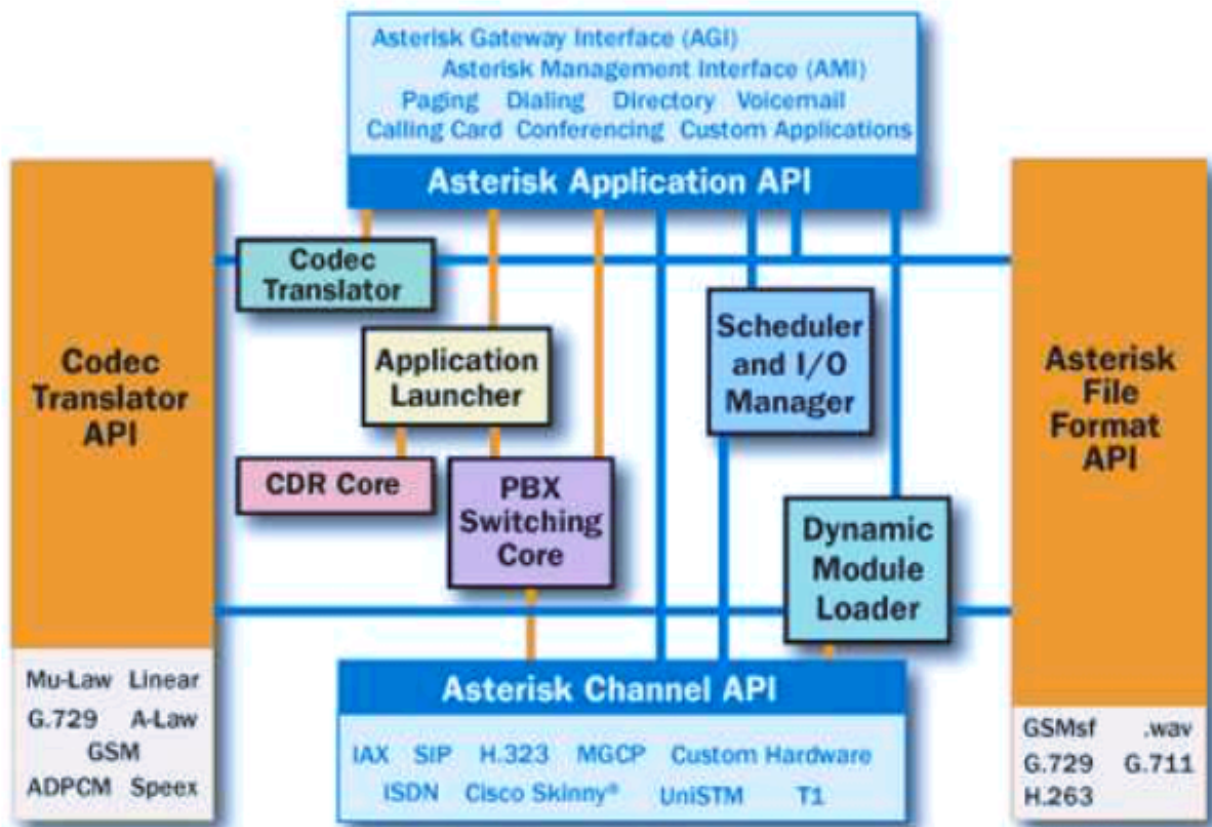
Figura 13 - Funcionamento do *VoiPMonitor*



Fonte: Stahelin (2014, p.39).

Moraes (2010) apresenta em seu trabalho um estudo abrangente das tecnologias de telefonia *IP* existentes e os protocolos mais utilizados. O autor teve como foco principal os seguintes aspectos: estudar o *PBX Asterisk*, desenvolver um sistema de telefonia *VoIP* simples e expansível e documentar o processo para que o mesmo servisse de referência para futuros estudos e implantações. A Figura 14 apresenta a integração entre as *APIs* (*Application Programming Interface*) e as funções do núcleo na arquitetura do *Asterisk* estão ilustradas.

Figura 14 - Arquitetura do *Asterisk*, com as integrações entre os diferentes componentes



Fonte: Moraes (2010, p.44).

O trabalho foi embasado na integração de duas ou mais redes de telefonia (*VoIP*) privadas simples, de tal forma que os resultados coletados possam ser utilizados

como base de conhecimento para futuras implementações reais, por exemplo a comunicação entre duas unidades de uma empresa (matriz e filial) separadas geograficamente. Para dar sustentação real ao ambiente de teste criado para validar o cenário proposto foi proposto um plano de numeração de ramais para que seja possível a identificação dos usuários na interação com as duas centrais *Asterisk*.

De acordo com Moraes (2010) as centrais *Asterisk* estão instaladas em uma máquina virtual em uma máquina física com sistemas operacionais *GNU/Linux 64 bits*. A central A1 usa o *Ubuntu Server 10.04 kernel 2.6.32-22-server*, e está instalada numa máquina com processador *AMD Phenom II X4 940*, com quatro núcleos, e 8 GB de *RAM DDR2-1066*; e a central A2 usa o *Debian 5.0 kernel 2.6.26-2-amd64*, que está executada numa máquina virtual utilizando o programa de virtualização *Virtual Box*, da Oracle.

Observa-se que tanto a escolha do hardware quanto a do sistema operacional colaboram para um ambiente computacional balanceado e otimizado. A utilização de máquinas virtuais para compor o cenário de testes facilitou a manipulação desses componentes. Porém a proposta apenas trata a implementação de um ambiente de rede *VoIP*, baseado na plataforma do *Asterisk*.

ROCHA, FONTANA e BARRÉRE (2012) apresentam as vantagens da integração de um *Call Center* com as tecnologias *VoIP (Voice over Internet Protocol)*, identificando as tendências de investimentos no setor e na logística de atendimento baseado no serviço *offshore*, levando em consideração as questões relacionadas ao fuso horário de cada estado ou país. A partir dessa visão pode-se escolher parâmetros de descentralização do serviço ou a terceirização do mesmo, sendo possível executar essa ferramenta apenas com computadores e uma internet de banda larga. Assim que os parâmetros estão definidos, a central é disponibilizada para o acesso das informações do *Call Center* e para transferir ligações recebidas através da tecnologia *VoIP*. A Figura 15 apresenta a visão de integração entre as *APIs (Application Programming Interface)* e as funções do núcleo na arquitetura do *Asterisk*.

Figura 15 - Distribuição de unidades ou filiais de um Call Center.



Fonte: Rocha, Fontana e Barrère (2012, p.7).

No entanto, a solução proposta por ROCHA, FONTANA e BARRÉRE (2012), apenas soluciona o problema da conectividade telefônica entre dois pontos separados geograficamente. O trabalho tem como premissa a existência de um software de Call Center, já integrado com as tecnologias VoIP, ou seja, nesse cenário todos os problemas internos referentes ao funcionamento de um Call Center já foram solucionados. O foco do trabalho é a otimização dos custos em relação aos gastos com telefonia. Basicamente, a solução proposta teve como objetivo a utilização das tecnologias VoIP para interligar diferentes sites de um Call Center. Através dessa central pode-se definir parâmetros de dispersão geográfica para descentralizar os serviços, possibilitando desse modo uma possível terceirização do mesmo.

De acordo com ROCHA, FONTANA e BARRÉRE (2012), para o cliente todo o processo é transparente, ou seja, os impactos dessa descentralização não são perceptíveis. Com a possibilidade de redirecionar as demandas para qualquer filial, independente do posicionamento geográfico, pode-se otimizar a utilização dos recursos envolvidos, como, por exemplo, a necessidade de pessoas trabalharem no período

noturno. Essa flexibilidade impacta de forma positiva nos custos da empresa, assim como em uma melhor disponibilidade dos serviços ofertados.

Pode-se observar que a proposta de ROCHA, FONTANA e BARRÉRE (2012), tem como foco a resolução do problema de transbordo das ligações de um *Call Center* para outro e não aborda nenhuma solução de integração com o *CRM* do *Call Center* ou falhas internas do sistema que possibilite uma intergração do canais de comunicação com o cliente externo.

Inserido nesse contexto, verifica-se, que a ferramenta de integração de canais de comunicação proposta neste trabalho, resolve uma gama de problemas que se apresentam de forma local e interna no cenário de um *Call Center*. Pode-se citar as seguintes características:

- Integração dos principais canais de comunicação com o cliente;
- Viabiliza a otimização de vários processos internos da empresa, mesmo sem a presença de um *CRM*;
- **Permite** integração com os principais *CRM* do mercado;
- Visão analítica dos principais indicadores de produtividade e qualidade em um *Call Center* (logado, disponível, ocupado, pausa, fila, recebidas, atendidas, abandonadas);
- Controle em tempo real do TMA (tempo médio de atendimento) dos operadores;
- Controle de *CallBack* para todas as chamadas entrantes;
- Gravação e monitoramento das chamadas;
- Controle simultâneo de parâmetros de atendimento para vários DACs (Distribuidor automático de chamadas).

A ferramenta tem como foco viabilizar, uma integração entre todos os canais de comunicação com o cliente, assim como facilitar a execução dos processos internos do *Call Center*, como, por exemplo, a integração do computador com o telefone e a integração com a base de dados já existente.

4 DESENVOLVIMENTO

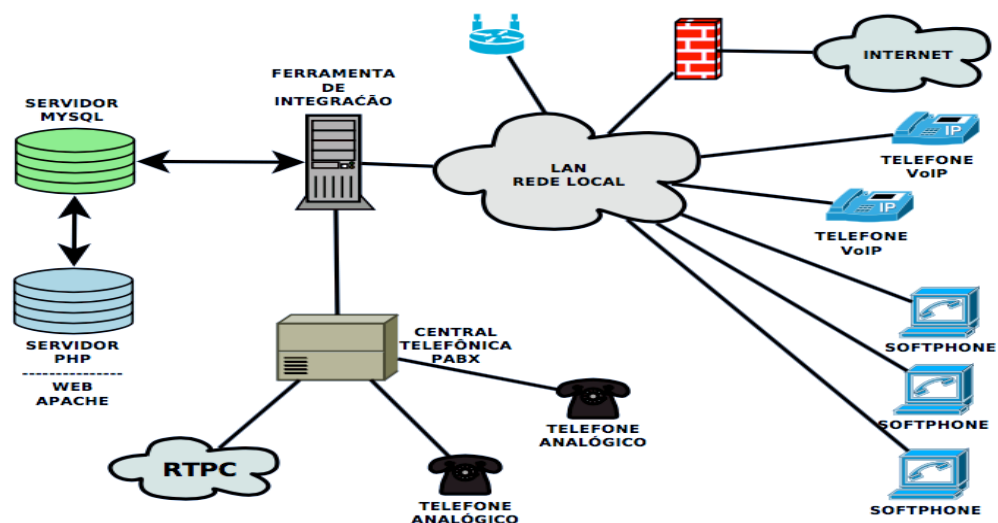
Neste capítulo é definido o escopo do projeto das redes, juntamente com as estruturas envolvidas no processo. São relacionados os componentes a serem utilizados, como computadores, periféricos, cabeamento, sistemas operacionais e programas relacionados. A distribuição dos ramais foi definida de acordo com a necessidade do cenário de testes.

A configuração da central é feita em etapas, acrescentando gradativamente as funcionalidades, de tal forma que seja possível fazer um mapeamento claro dos passos que foram executados. As configurações são implementadas através da edição de arquivos (tipo texto) que contém parâmetros de ajustes das funcionalidades.

4.1 ARQUITETURA E PROJETO

A ferramenta de integração a qual este trabalho se baseia, tem por objetivo integrar os principais canais de comunicação com o cliente (telefone, e-mail, redes sociais, aplicativos de mensagens, CRM, URA, sms, recursos de PBX IP etc.), em um cenário de um *Call Center*, centralizando. Dessa forma, todos os recursos de comunicação em uma única interface. A Figura 16 mostra uma visão geral da ferramenta de integração de canais.

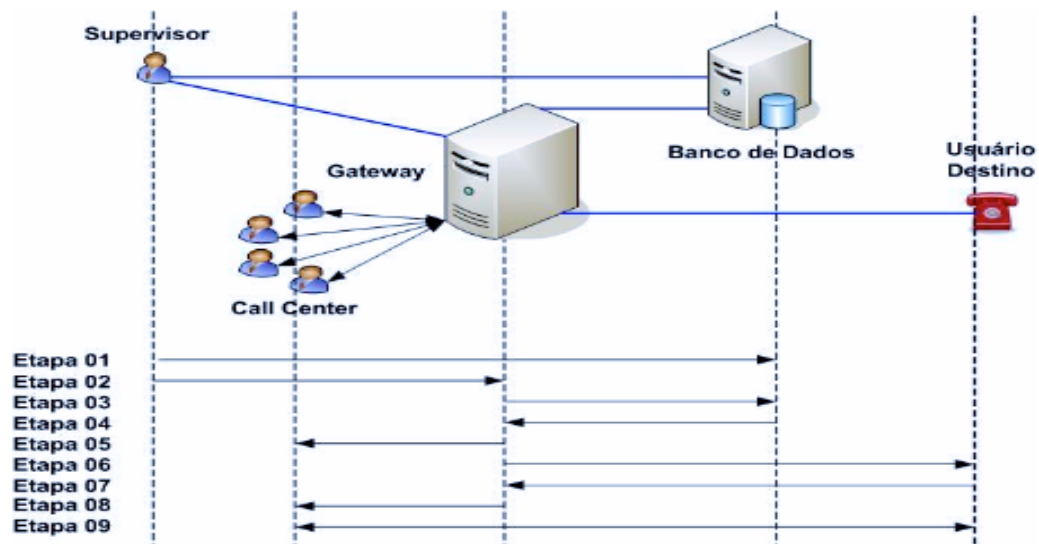
Figura 16 – Visão geral da ferramenta de integração



Fonte: Elaborado pelo autor.

A ferramenta proposta, por sua vez, é implementada através da utilização de bibliotecas e plataformas abertas, possibilitando o acesso de micro e pequenas empresas a soluções tecnológicas que antes só estavam disponíveis, para grandes corporações. É utilizado o *PBX IP Asterisk*, como base para toda a estrutura de telefonia *VoIP* e integração com as redes de telefonia pública comutada (RTPC), conforme Figura 17.

Figura 17 - Estrutura básica de funcionamento da Ferramenta



Fonte: Adaptado de Hersent, Guide, Petit (2002, p. 155).

A seguir, são descritas as etapas relacionadas ao desenvolvimento do fluxo de comunicação da ferramenta, conforme é mostrado na Figura 17.

Na Etapa 01 o analista de tráfego do *Call Center* executa uma carga na base de dados. O arquivo inserido no banco de dados deve ter as seguintes características: , relação dos nomes, código identificador de cada indivíduo destino, número do telefone que se deseja contactar.

Na Etapa 02 o analista de tráfego inicia o processo da discagem para os clientes.

A Etapa 03 é o momento em que a ferramenta solicitará à Base de dados a listagem dos clientes destino para que possam ser contactados.

Na Etapa 04 ocorre o retorno da listagem.

Na Etapa 05 a ferramenta verifica quais operadores estão disponíveis para receber o fluxo de ligações.

Na Etapa 06 é iniciado o processo de discagem conforme as informações contidas na listagem fornecida na Etapa 04. Nesse momento a listagem já está em memória para o processamento.

Na Etapa 07 a ferramenta recebe o retorno do estado da chamada. Nesse cenário o estado da chamada é ligação atendida ou não atendida por parte do destino.

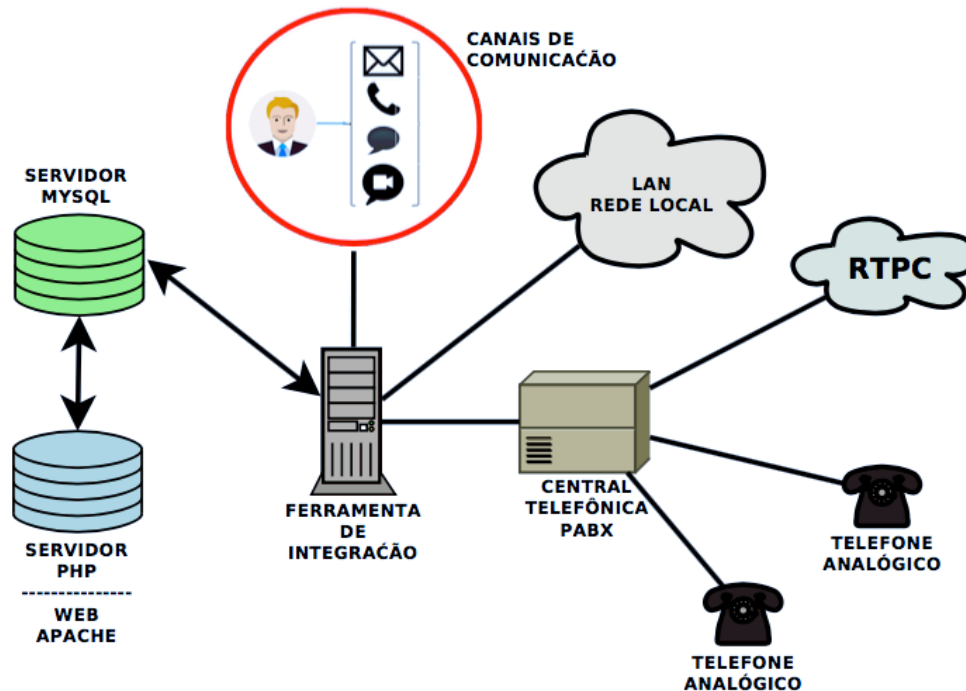
Na Etapa 08 a ferramenta direciona a chamada para o *Call Center* e, segundo a estratégia definida para a campanha, um operador disponível executará o atendimento.

Na Etapa 09 o operador inicia o tratamento da chamada. É o momento da efetivação do atendimento.

4.2 COMPONENTES

A ferramenta está instalada em uma máquina virtual que, por sua vez, está em uma máquina física com o sistema operacional *GNU/Linux 64bits*. A Central *Asterisk*, assim como todo o switch de programas (*Apache HTTP Server*, PHP 5.3.1, *MySQL 5*, *phpMyAdmin 3.2.5*) de apoio e desenvolvimento, estão fisicamente instalados numa máquina modelo *PowerEdge T110 II*, da marca *Dell*, com o Processador *Intel® Xeon® E3-1220v2*, 8GB de memória e 2HDs de 1TB *Raid 1*, que utiliza a última versão da distribuição Linux Debian, que está sendo executada em uma máquina virtual utilizando o programa de virtualização *Hypervisor Xen*, que foi desenvolvido na Universidade de *Cambridge* (Reino Unido) conforme Figura 18.

Figura 18 - Estrutura dos componentes da Ferramenta de integração de canais



Fonte: Elaborado pelo autor.

Foi definido o uso de tecnologias de virtualização para que se pudesse ter um cenário de fácil portabilidade e manipulação. No entanto, em ambientes de produção real é fortemente aconselhável a utilização de máquinas físicas, para evitar problemas como, por exemplo, elevado processamento.

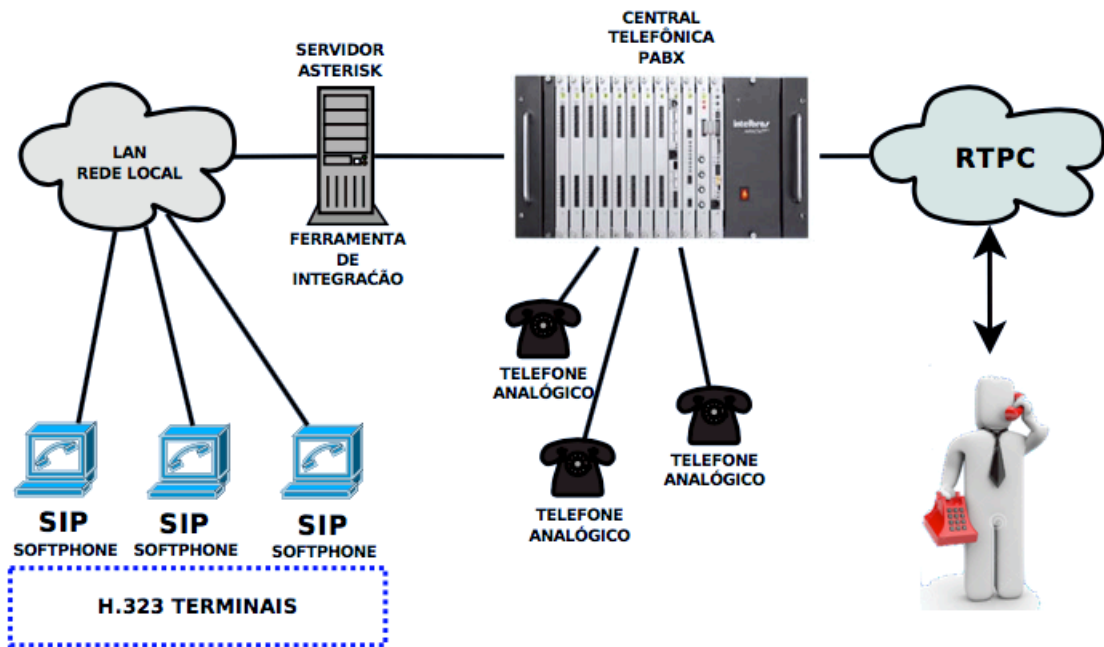
São utilizados para ativação dos ramais o *softphone X-Lite*, em conjunto com adaptadores para *headset* e *headfone IP*.

A rede interna (*LAN* padrão *Ethernet 100BASE-TX*), é baseada em cabeamento metálico do tipo *UTP Cat5e*. A preferência para este tipo de cabeamento deve-se ao fato de se tratar de um cenário compatível com mais de setenta por cento das redes internas (*LAN*) de empresas de médio e pequeno porte. A central *Asterisk* está conectada à internet por meio de um link do tipo Dedicado com a velocidade de 15 *Mbps* de forma síncrona (*down/up*).

A integração (interface de convergência) da rede de telefonia *VoIP* com a rede RTPC, foi feita através de uma central *PABX* modelo: Intelbras - Impacta 300 Híbrida. Dessa forma, os ramais analógicos e *SIP* (*Session Initiation Protocol*) podem solicitar uma linha externa para executarem chamadas, assim como podem receber as

ligações vindas da RTPC de forma transparente para o usuário. Essa é uma configuração básica para o funcionamento de uma central *PBX*, porém serve como demonstração para a capacidade e abrangência dos componentes do *Asterisk*. A figura 19 apresenta uma visão da integração das interfaces de convergência.

Figura 19 - Integração Interface de convergência



Fonte: Elaborado pelo autor.

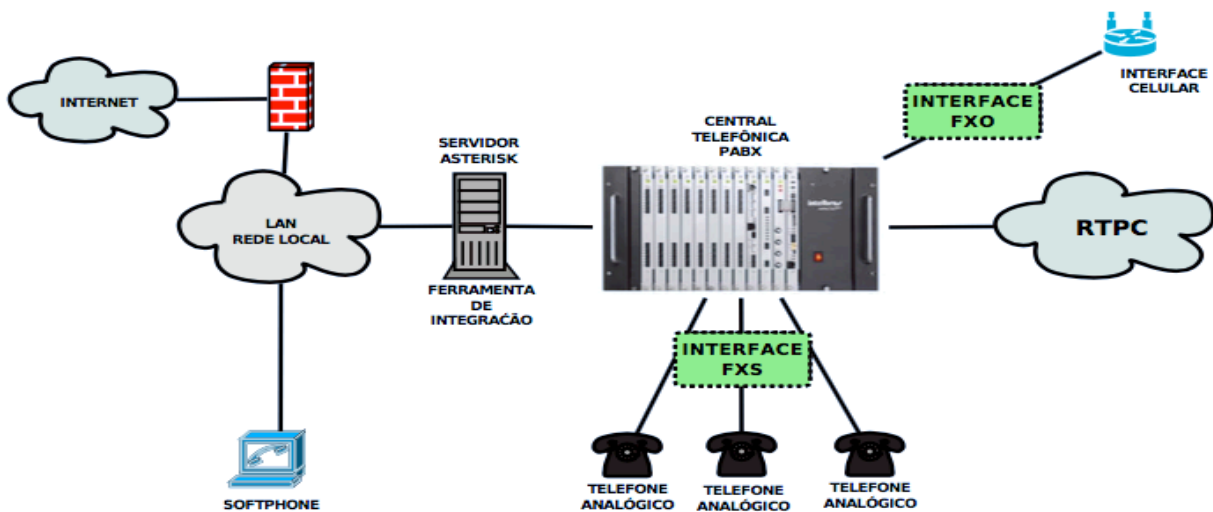
Existem dois tipos de interfaces no sistema de telefonia comutada:

FXO (Foreign Exchange Office), são interfaces que correspondem analogamente a um telefone analógico ou a um tronco analógico de um *PABX*. A interface *FXO* pode ser conectada a um ramal de um *PABX* externo. Tem um comportamento (modo de operação) similar ao um telefone analógico *DTMF (Dual Tone MultiFrequencial)*, fechando o circuito e discando para sistemas externos, no caso de chamadas originadas no *PABX*.

A interface *FXS (Foreign Exchange Station)*, corresponde a uma posição de ramal do *PABX*. Um dispositivo *FXS* fornece alimentação elétrica e tom de discagem. Essa interface foi projetada para ser diretamente ligada a um aparelho telefônico

análogo *DTMF* ou então a qualquer posição de tronco analógico de um *PABX* externo. Ela reconhece discagens *DTMF* e gera toque de campainha para aparelho telefônico analógico. Ou seja, são interfaces que provêm energia elétrica para a linha telefônica, disponibilizam tom de discagem e geram tensão de sinalização de chamada. As centrais telefônicas (*PABX*) são exemplos de interfaces *FXS*. A figura 20 apresenta uma visão das interfaces *FXS* e *FXO*.

Figura 20 - O Asterisk e as interfaces *FXS* e *FXO*



Fonte: Elaborado pelo autor.

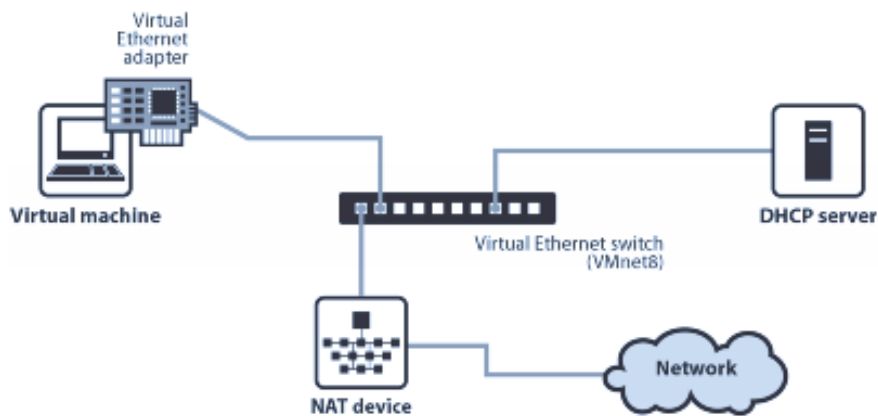
Em relação a segurança na comunicação *VoIP*, é visível e justificável a preocupação do usuário final, assim como das operadoras (prestadoras de serviços) com relação aos riscos vinculados a esse tipo de serviço, considerando a enorme quantidade de variáveis envolvidas no processo. Essa preocupação é herdada diretamente do modelo inicial das telecomunicações (RTPC), onde se tinha uma rede toda projetada para um único fim, a propagação da voz. Para isso ser possível, era necessário toda uma infraestrutura dedicada, que naquele momento já era sujeita a uma série de ameaças. Com a consolidação da integração: *VoIP* x RTPC, verifica-se uma concatenação de todas as características (positivas e negativas) da pilha *TCP/IP* para a telefonia convencional.

Os riscos ganham uma forte visibilidade, devido à grande expectativa, não negociável, de se manter o nível de serviço equivalente àquele ofertado outrora pelas grandes companhias de telecomunicações, estritamente baseadas na telefonia convencional (RTPC).

Para realizar um comparativo desse tipo, é necessário executar uma análise do estado da arte do cenário atual da telefonia RTPC como um todo. Seria necessário a enumeração de cada um dos serviços versus os mecanismos existentes para prevenir e mitigar os ataques de toda natureza.

Visando garantir um mínimo possível de segurança e economia de recursos na rede interna, essa estará protegida por um *firewall* funcionando em conjunto com um *NAT*. Faz-se necessário, dessa forma, o direcionamento das portas e um servidor *STUN* (*Session Traversal Utilities for NAT*) conforme Figura 21, para facilitar a travessia dos clientes pelo *NAT*. Baseados nessa visão de segurança todo o tráfego entre servidores *Asterisk* está direcionado para protocolo *IAX*, pois dessa forma é mais fácil manter um nível aceitável de segurança.

Figura 21 - Rede com interface NAT



Fonte: Adaptado de Kurose (2010, p.35).

4.3 PLANO DE NUMERAÇÃO

Este planejamento tem como objetivo, aprovisionar e identificar os ramais que serão registrados na central *Asterisk*, além de definir os ramais específicos para implementação dos testes das funcionalidades da ferramenta *proposta* neste projeto. Os ramais *VoIP* são compostos por quatro dígitos e estão agrupados da seguinte forma:

- O primeiro dígito: identifica a central a qual o ramal se reporta (registro), além de funcionar como sinalizador para realização de conferências;
- O segundo e o terceiro dígito: definem a seção à qual o ramal está acoplado;
- O quarto dígito: identifica o ramal.

O primeiro ramal de uma seção é utilizado para sinalizar simultaneamente em múltiplos ramais. Ou seja, ele funciona como um *gateway* para um determinado grupo de ramais. Este planejamento poder ser observado com maiores detalhes na Tabela 2.

Tabela 2 - Plano de numeração

Ramal	Descrição
2010	Representa o grupo de ramais do 2011 até o 2020
2011	Ramal pertencente ao grupo de ramais do cenário de teste
2012	Ramal pertencente ao grupo de ramais do cenário de teste
2013	Ramal pertencente ao grupo de ramais do cenário de teste
2014	Ramal pertencente ao grupo de ramais do cenário de teste
2015	Ramal pertencente ao grupo de ramais do cenário de teste
2016	Ramal pertencente ao grupo de ramais do cenário de teste
2017	Ramal pertencente ao grupo de ramais do cenário de teste
2018	Ramal pertencente ao grupo de ramais do cenário de teste
2019	Ramal pertencente ao grupo de ramais do cenário de teste
2020	Ramal pertencente ao grupo de ramais do cenário de teste

Fonte: Elaborado pelo autor.

4.4 CONFIGURAÇÃO DO AMBIENTE

Os sistemas operacionais utilizados foram:

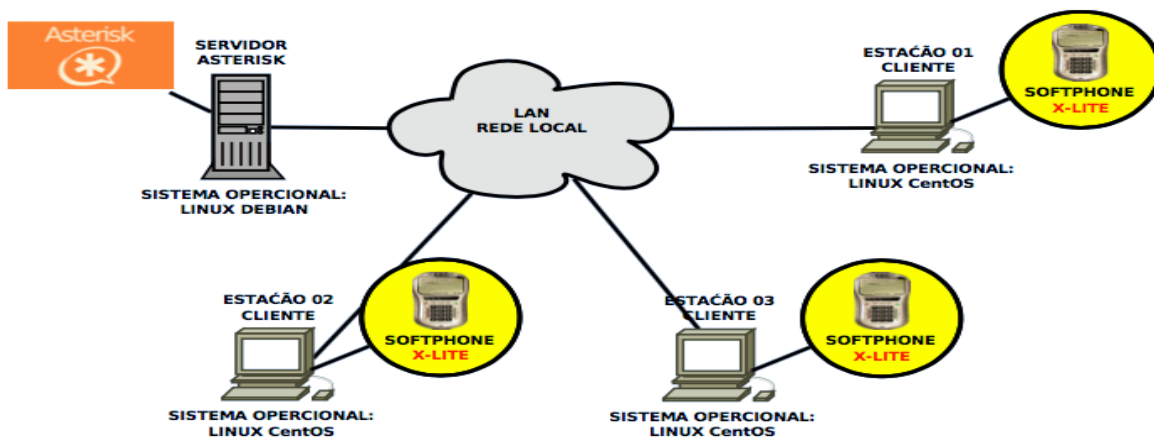
- *Linux Debian*: Sistema operacional gratuito de código aberto instalado no computador utilizado como servidor.
- *Linux CentOS*: Sistema operacional gratuito de código aberto instalado nos computadores utilizados como estações cliente.

Os softwares utilizados foram:

- *Trixbox*: Software que possui em seu conteúdo o sistema operacional *CentOS* e o *Asterisk*.
- *Asterisk*: Software emulador de *PABX IP* instalado no computador de código aberto. Com ele, o computador funciona como um *PABX IP*.
- *Wireshark*: Software de monitoração das trocas de mensagens das chamadas.
- *X-Lite* : É um *Softphone* que centraliza em uma única *interface*, *voz* e *vídeo*.

A Figura 22 apresenta uma visão da configuração do ambiente, tendo como foco, a distribuição dos sistemas operacionais utilizados no desenvolvimento da ferramenta de integração de canais.

Figura 22 – Configuração do ambiente



Fonte: Elaborado pelo autor.

Na implantação dos sistemas operacionais são utilizadas as opções padrão para instalações em servidores e estações, sem ambiente gráfico e com um servidor *SSH*. Após a finalização da instalação dos sistemas operacionais, são criados os usuários e grupos, assim como os privilégios relacionados. A implantação do servidor do *Asterisk* foi instalado usando o gerenciador de pacotes *Aptitude* para instalar a versão disponível nos repositórios de pacotes com o comando “*sudo aptitude install asterisk*”. O *Aptitude* é uma interface em modo texto para o sistema de pacotes do *Debian GNU/Linux* que permite ao usuário/administrador visualizar, de forma fácil, as listas de pacotes e executar operações como instalação, atualização e remoção de pacotes. O gerenciador de pacotes instala automaticamente as dependências necessárias para que o *Asterisk* funcione corretamente.

As configurações do *Asterisk* são feitas através dos arquivos de configuração *.conf*, e podem ser executados através de comandos diretamente no console. As configurações também podem ser alteradas e salvas a partir do console, mas isto requer que duas opções sejam previamente configuradas no arquivo *extensions.conf*. Os arquivos *.conf* possuem extensa documentação interna e podem ser consultados para esclarecer dúvidas relativas aos registros.

São executados os seguintes passos para configuração:

1. Registro dos ramais *SIP* e configuração das chamadas;
2. Configuração da transferência de chamadas;
3. Configuração da sinalização múltipla e da captura de chamadas;
4. Configuração das salas de conferência;
5. Configuração da integração com a RTPC;
6. Configuração da comunicação entre centrais *Asterisk*.

4.4.1 Registro dos Ramais *SIP* e Configuração das Chamadas

Os ramais *SIP* (*Session Initiation Protocol*) são configurados no arquivo *sip.conf*. Cada ramal é configurado em uma seção chamada contexto. Os arquivos de configuração possuem uma sintaxe de comparação de padrões que possibilita a simplificação da declaração de regras, otimizando dessa forma a quantidade de

informação redundante com regras e parâmetros semelhantes. Na figura 23 é possível verificar um exemplo de configuração do arquivo *sip.conf*.

Figura 23 - Exemplo de registro de um ramal no arquivo de configuração: *sip.conf*

```
[2011]                ;nome do contexto
type      =friend     ;pode receber e realizar chamadas
username  =2011       ;nome do usuario SIP
secret    =2011       ;senha do usually SIP
callerid  =2011       ;identificador de chamada SIP
host      =dynamic    ;ip do cliente é dinamico
context   =INTERNO   ;contexto onde será registrado
disallow  =all        ;desabilita todos os codecs
allow     =alaw       ;habilita o codec A-law
allow     =ulaw       ;habilita o codec U-law
allow     =g711       ;habilita o codec G.711
allow     =g729       ;habilita o codec G.729
```

Fonte: Adaptado de Hersent, Guide, Petit (2002, p. 97).

Nas configurações do arquivo *sip.conf* são definidos o tipo de *endpoint*, o nome de usuário, senha, identificador de chamada, contexto principal de discagem ao qual ele pertence e *codecs* que podem ser utilizados. A Figura 23 ilustra o registro de configurações no arquivo *sip.conf*, os registros dos outros ramais são semelhantes.

Para que os ramais *VoIP* registrados possam executar chamadas, faz-se necessário o registro no plano de discagem (*dial plan*) principal, que está contido no arquivo *extensions.conf*. O registro é feito com o comando *exten =>* descrição do ramal, prioridade de execução, função.

As declarações *exten*, em conjunto com as funções de chamadas, definem toda a lógica e fluxo de uma chamada a um ramal. Nessa condição, o ramal deve chamar o cliente *VoIP* correspondente, levando em consideração um tempo de espera para que o cliente atenda.

Depois de atendida a condição do tempo de espera, a próxima declaração na lista de prioridades do ramal é executada. A Figura 24 apresenta a configuração do plano de discagem da central *PBX*.

Figura 24 - Exemplo de registro de contexto de chamadas em *extensions.conf*

```
[general]
;permite que as configs sejam
;alteradas e salvas atraves do console
static=yes
writeprotect=no
;termina chamada se nao for atendida
autofallthrough=yes

[central_a1]           ;contexto da central A1
;registra ramais 1011 a 1019 para chamadas
;SIP, com espera de 30 segundos
;e desliga a chamada se nao for atendida
exten => _101Z, 1, Dial(SIP/${EXTEN},30)
exten => _101Z, 2, Hangup()

;ramal de eco, apenas responde com o audio de entrada
exten => 9999, 1, Answer()
exten => 9999, 2, Echo()
exten => 9999, 3, Hangup()
```

Fonte: Adaptado de Hersent, Guide, Petit (2002, p. 35).

4.4.2 Configuração da Transferência de chamadas

Para configurar a função de transferência de chamadas se faz necessário a configuração dos seguintes arquivos: *sip.conf*, *extensions.conf* e *features.conf*.

A função *blindxfer* (dar suporte as características das chamadas entre ramais), que é configurada no arquivo *features.conf*, como se pode ver na Figura 25. Se faz necessário adicionar o parâmetro *t* ([habilita transferência](#)) à chamada da função *Dial*, no arquivo *sip.conf*. Dessa forma, garante-se que apenas os ramais envolvidos na chamada atual, fiquem com a função transferência habilitada, como é exemplificado na Figura 26.

Figura 25 - Transferência de chamadas em *features.conf*

```
[featuremap]
blindxfer => 1#           ;habilita transferência de chamadas
```

Fonte : adaptado de Meggelen (2005, p.46).

Figura 26 - Transferência de chamadas em *extensions.conf*

```
[central_PBX]
.
.
.
;habilita transferência de chamadas
exten => _101Z,1,Dial(SIP/101Z,30,t)
exten => _101Z,2,Hangup()
```

Fonte : adaptado de Meggelen (2005, p.47).

Por último, porém não menos importante é necessário desabilitar a funcionalidade de reenvio de pacotes (função *INVITE*) nas chamadas *SIP* (*Session Initiation Protocol*), em *sip.conf*, para que dessa forma o *Asterisk* possa controlar a transferência das chamadas, como apresentado na Figura 27.

Figura 27 - Transferência de chamadas em *sip.conf*

```
[2011]
.
.
.
;comando que especifica que o cliente não pode mais enviar pacotes
INVITE após o estabelecimento da chamada, deixando que o Asterisk
controle as transferencias de chamadas

canreinvite = no
```

Fonte : adaptado de Meggelen (2005, p.47)

4.4.3 Configuração de Sinalização Múltipla e da Captura de Chamadas

Faz-se necessário definir grupos de sinalização múltipla para que vários ramais toquem simultaneamente quando o grupo receber uma chamada. Para que essa condição seja atendida é necessário configurar no arquivo *queues.conf*, que está ilustrado na Figura 28, quais são os membros de cada grupo, e no arquivo

extensions.conf, informa-se para qual grupo a chamada deve ser sinalizada, conforme e exemplificado na Figura 29.

Figura 28 - Grupos de sinalização múltipla configurado no arquivos *queues.conf*

```
[2011]
music      = default
strategy   = ringall
|
member     = SIP/2011
member     = SIP/2012
member     = SIP/2013
member     = SIP/2014
```

Fonte : adaptado de Meggelen (2005, p.49).

Tem-se também a possibilidade de definir um grupo de monitoramento ou captura de chamadas, que viabiliza um ramal capturar uma chamada que está tocando em outro ramal do mesmo grupo. Essa configuração é habilitada no arquivo *sip.conf*, onde são definidas as características de cada ramal, apresentada na Figura 29.

É de extrema importância ressaltar que os grupos de monitoramento ou captura somente funcionam quando utilizado o mesmo protocolo. Dessa forma tem-se que um ramal *SIP* (*Session Initiation Protocol*) e um ramal *H.323* não podem pertencer ao mesmo grupo (bloco) de ramais, assim como não podem capturar chamadas entre si, ou seja, entre protocolos diferentes.

Figura 29 - Sinalização múltipla de chamadas configurado no arquivo *extensions.conf*

```
[central_PBX]
-|
-
-
;o comando Queue diz que a chamada será sinalizada no
grupo 2080 por 30 segundos

exten      => _1010, 1, Queue(2080,,,,30)
exten      => _1010, 2, Hangup()
exten      => _101Z, 1, Dial(SIP/101Z,30,t)
exten      => _101Z, 2, Queue(2080,t,,,30)
exten      => _101Z, 3, Hangup()
```

Fonte : adaptado de Meggelen (2005, p.52).

4.4.4 Configuração dos cenários de conferência

A funcionalidade de conferência permite que três ou mais terminais conversem entre si simultaneamente. É possível definir salas de conferência simples com e sem autenticação no arquivo *meetme.conf* e registrar os ramais correspondentes em *extensions.conf*, como exemplificado, respectivamente, nas Figuras 30 e 31.

Figura 30 - Grupo de monitoramento ou captura de chamadas em *sip.conf*

```
[1011]
:
;define a qual grupo de captura o ramal pertence
callgroup=1
;define grupos dos quais pode capturar chamada
pickupgroup=1

[1012]
:
callgroup=1
pickupgroup=1

[1021]
:
callgroup=2
pickupgroup=2
```

Fonte : adaptado de Meggelen (2005, p.54).

Figura 31 - Salas de conferência em *meetme.conf*

```
[rooms]
;definicao: conf => sala, senha, senhadmin
conf => 1
conf => 2
conf -> 3, 1234, 4321
```

Fonte : adaptado de Meggelen (2005, p.55).

A sala de conferência consiste basicamente na configuração do número da sala no arquivo *meetme.conf* e de sua chamada no plano de discagem programado (*dialplan*). Com isso tem-se diversas configurações possíveis para a chamada da sala no *dialplan*. São mostradas a seguir as mais relevantes para este projeto.

Para isso, faz-se necessário verificar se os módulos *ztdummy* ou *dahdi_dummy* estão devidamente inicializados. Essa função, depois de ativa, funciona

como um contador (*clock*) para o *Asterisk* que é de vital importância para implementação das salas de conferência.

Para garantir que os módulos estejam sendo executados é necessário inicializar o seguinte comando: *lsmmod ! grep ztdummy*.

As salas de conferência devem ser descritas na *tag* [*rooms*] do arquivo *meetme.conf*. A sintaxe é a seguinte: *conf => sala, senha, senha do admin*, conforme exemplo a seguir:

[rooms]

conf => 501

conf => 502,4433

conf => 503,8787,9909

Tendo criado as salas, deve-se criar um *dialplan* para acessar as salas:

[conferencia]

exten => _50[1-3],1,MeetMe(\${EXTEN})

Existe uma lista considerável de opções para o comando *meetme*, sendo aqui relacionadas as mais importantes para o projeto:

c – Anuncia a contagem de usuários na conferência;

i – Anuncia entrada e saída de usuários da conferência;

l – Anuncia entrada e saída de usuários da conferência sem informar o nome do usuário;

M – Habilita *musicOnHold* na conferência. Quando esta *flag* é habilitada, o *Asterisk* tocará música de espera sempre que houver apenas um usuário na sala;

m – Habilita o modo monitor, fazendo com que os usuários entrem mudos na conferência.

Dessa forma, pode-se criar uma sala de conferência onde os usuários sejam anunciados ao entrar, caso haja apenas um usuário, este ficará ouvindo música de espera, para isso a configuração do parâmetro *dialplan* fica desta forma:

[conferencia]

```
exten => _50[1-3],1,MeetMe(${EXTEN},iM)
```

A Figura 32 apresenta os comandos necessários para a configuração de uma sala de conferência.

Figura 32 - Descrição da configuração das salas de conferência no arquivo *extensions.conf*

```
[conferencias]
exten => 9001, 1, MeetMe(1, iMp)
exten => 9001, 2, Hangup()
exten => 9002, 1, MeetMe(2, iMp)
exten => 9002, 2, Hangup()
exten => 9003, 1, MeetMe(2, iMp, 1234)
exten => 9003, 2, Hangup()
```

Fonte : adaptado de Meggelen (2005, p.57).

4.4.5 Configuração Da Integração Com A RTPC

A configuração das placas de telefonia fixa é descrita no arquivo *zaptel.conf*. O *Asterisk* suporta a configuração de várias placas de telefonia com múltiplas interfaces FXS e FXO conforme a ordenação física do *hardware* envolvido, mas, no caso específico deste projeto, a *central PBX* utiliza apenas uma interface *FXO*. A Figura 33 demonstra a definição de uma placa *FXO*.

Figura 33 - Configuração das placas *FXS* e *FXO* em *zaptel.conf*

```
;carregar uma lista de tons de comunicacao separados por virgula
loadzone=br,us
;utilizar os seguintes tons por padrao
defaultzone=br

;definir uma placa fxo, que recebe sinalizacao fxs
fxsks=1

;definir uma placa fxs, que recebe sinalizacao fxo
;fxoks=1           ;nao sera utilizada
```

Fonte : Adaptado de Meggelen (2005, p.57).

O canal *FXO* é configurado no arquivo *zapata.conf*, como na Figura 34. Existem outras muitas opções de configuração dos canais em *zapata.conf*, como ganho do sinal, identificação de chamada, grupos de captura, cancelamento de eco, chamada em espera, mas a central *PBX* utiliza somente as opções básicas para utilizar o modem como interface *FXO*.

Figura 34 - Configuração dos canais no arquivo *zapata.conf*

```
[channels]
;contexto de recebimento de chamadas da rtpc
context=rtpc
;metodo de comutacao
switchtype=national
;metodo de sinalizacao do canal
signalling=fxs_ks ;recebe sinalizacao fxs
channel => 1
```

Fonte : Adaptado de Meggelen (2005, p.58).

Finalmente, no arquivo *extensions.conf* é definido o contexto que atende as chamadas provenientes da RTPC e a configuração de saída para a RTPC, exemplificado na Figura 35:

Figura 35 - Configuração dos canais no arquivo *extensions.conf* na sessão RTPC

```
[rtpc] ;contexto de recebimento da rtpc
;a expressao 's' eh usada quando nao existe padrao no numero
;do ramal ou quando nao ha um ramal a ser chamado
exten => s, 1, Answer()
exten => s, 2, Dial(SIP/1011,30,t)
exten => s, 3, Queue(1000,t, , ,30)
exten => s, 4, Hangup()

[central_a1]
:
;suprime o 0 discado para obter linha externa quando
;quando enviar o numero a operadora de telefonia fixa
exten => _0., 1, Dial(Zap/1/${EXTEN:1})
channel => 1
```

Fonte : adaptado de Meggelen (2005, p.58).

4.4.6 Configuração Da Comunicação Entre Centrais Asterisk

Para viabilizar a comunicação entre as centrais *PBX*, é utilizado o protocolo *IAX*, que foi configurado no arquivo *iax.conf*, registrando o protocolo em uma interface de rede e solicitando a autenticação da central atual na central remota. A Figura 36 exemplifica a configuração *IAX* da Central *PBX*:

No arquivo *extensions.conf* é configurado o direcionamento dos ramais 2xxx para a *central_PBX*, como ilustrado na Figura 37. É necessário incluir o redirecionamento da porta do *IAX* nos respectivos *NATs* e *firewalls* para que a comunicação funcione.

Figura 36 - Configuração da central *PBX* no arquivo *iax.conf*

```
[general]
port=4569                ;porta padrao do IAX
;registra o protocolo IAX em todas as interfaces de rede
bindaddr=0.0.0.0
;registra a central_a1 com a senha 'senha123'
;no endereco da central_a2
;isto serve para que a central_a2 conheca o endereco
;da central_a1, que eh dinamico
register => central_a1:senha123@enderecocentrala2

[central_a2]             ;contexto da central_a2
type=peer
secret=senha321
auth=md5
context=central_a1
host=dynamic
```

Fonte: Adaptado de Meggelen (2005, p.63).

Figura 37 - Ramal da central *PBX* configurado em *extensions.conf*

```
[central_a1]
:
exten => _2XXX, 1,
Dial(IAX2/central_a1:senha123@enderecocentrala2/$EXTEN@central_a2)
```

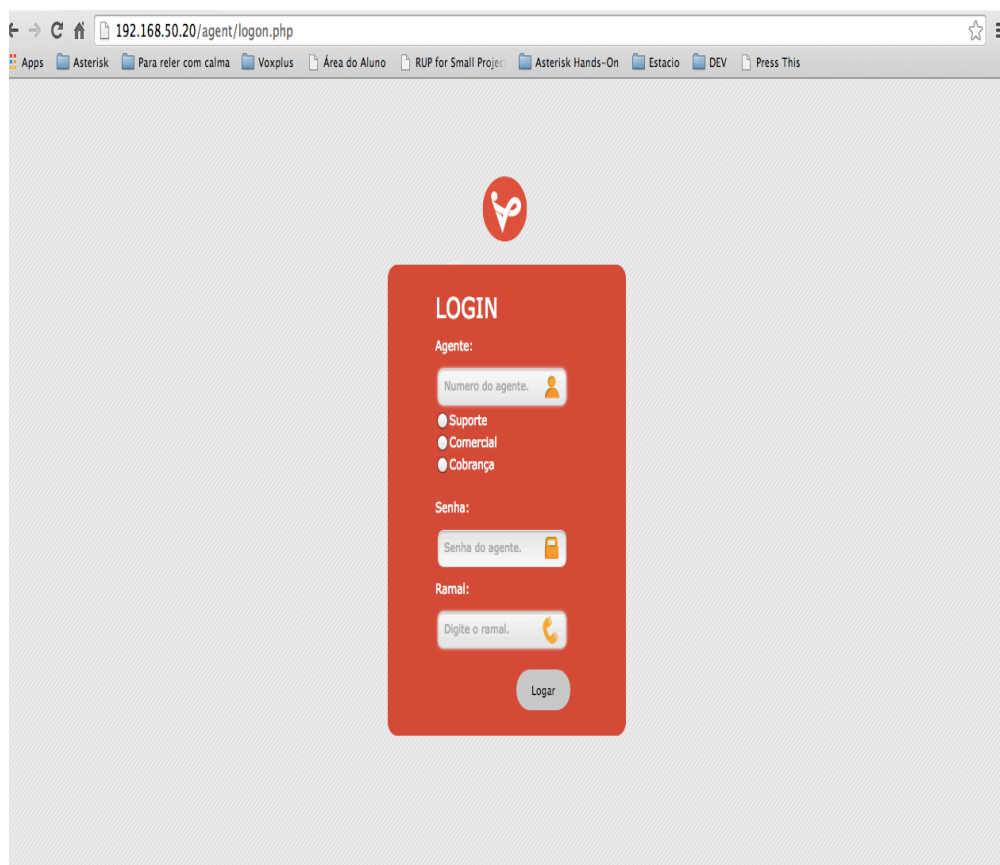
Fonte: Adaptado de Meggelen (2005, p.58).

4.5 TELAS DO SISTEMA

Todo o sistema foi planejado para que as consultas, inserções, edições e a eliminação de registros fossem executadas com o *plugins* do *PHP*. Desta forma, a utilização do sistema fica mais simples e proporcionando agilidade no processamento dos comandos, tanto no visual como na codificação.

As telas de *cadastro* (*login*) são de usuário, supervisor, operação de monitoramento de agentes e *callback*. Na figura 38 tem-se o login de operação (operador ou agente), como exemplo.

Figura 38 - Tela de login do sistema, módulo operador



The screenshot shows a web browser window with the address bar displaying "192.168.50.20/agent/logon.php". The browser's tab bar shows several open tabs, including "Apps", "Asterisk", "Para rerer com calma", "Voxplus", "Área do Aluno", "RUP for Small Projec", "Asterisk Hands-On", "Estacio", "DEV", and "Press This". The main content area of the browser displays a login form. At the top center of the form is a red circular logo with a white stylized figure. Below the logo, the word "LOGIN" is written in white capital letters on a red background. The form consists of several input fields: "Agente:" with a dropdown menu showing "Suporte", "Comercial", and "Cobrança"; "Senha:" with a password input field; and "Ramal:" with a text input field. A "Logar" button is positioned at the bottom of the form.

Fonte: Elaborado pelo autor.

A tela de *login* permite que o agente (operador) efetue *login* no sistema de acordo com a operação de atendimento desejada ou definida para o mesmo (suporte, comercial, cobrança). O sistema foi desenvolvido baseado em quatro níveis de

privilégios:

- 01- usuário (operador) - com acesso a tela de atendimento;
- 02- supervisor - com acesso a tela de monitoria e relatórios da operação;
- 03- gerente - com acesso a tela de usuários, monitoria e relatórios;
- 04- administrador - com acesso a cadastros, monitoria, atendimento e relatórios.

Figura 39 - Tela de atendimento do módulo agente (operador)

Fonte:



Elaborado pelo autor.

Na tela de atendimento do operador tem-se em primeiro lugar um *popup* solicitando o início do atendimento dos clientes da operação, conforme Figura 39, logo após (ícone de uma ampulheta), um sub-nível que mostra o histórico de tempo das ligações, esse indicador é atualizado sempre que o operador recebe um novo cliente ou chamada.

Figura 40 - Tela de *CallBack* comum a todos os agentes



Fonte: Elaborado pelo autor.

A tela de *CallBack*, permite que o operador possa ter acesso a uma lista de números de clientes, que, por algum motivo, foram desconectados dos sistema depois de entrarem, ou não, na fila de atendimento. Da mesma forma, se o cliente estiver em atendimento na URA (Unidade de Resposta Audível) o operador poderá retornar a ligação para aqueles que foram desconectados inesperadamente, conforme Figura 40.

Figura 41 - Tela de monitoria do sistema. Módulo de monitoria de DAC (Distribuidor automático de chamadas)

The screenshot shows a web browser window with the URL 10.10.4.56/voicerwayDev/dashboard.php. The dashboard displays a table with 10 columns: Operação, Nível de Serviço, Logado, Disponível, Ocupado, Pausa, Fila, Recebidas, Atendidas, and Aband. Each row represents a different operation, with icons for each status and numerical values.

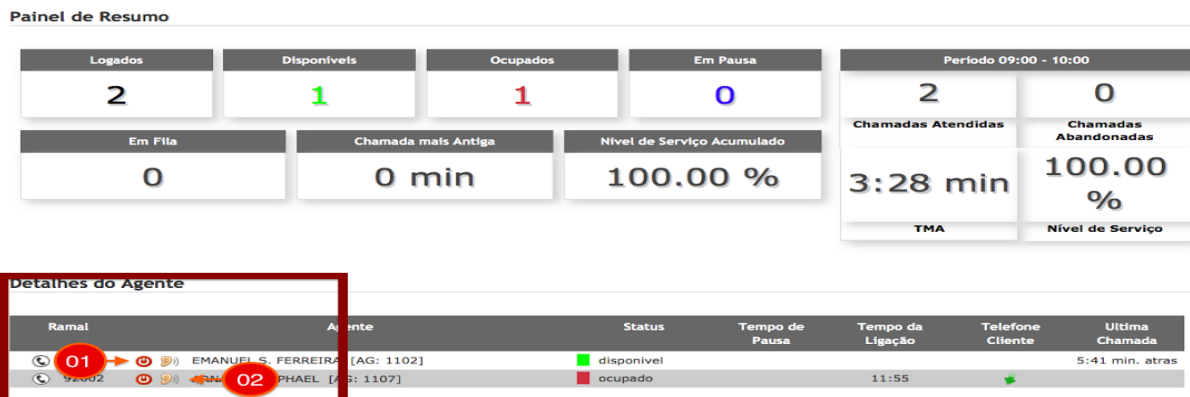
Operação	Nível de Serviço	Logado	Disponível	Ocupado	Pausa	Fila	Recebidas	Atendidas	Aband.
detran	10.00%	7	0	3	4	3	30	26	4
redesplan	63.64%	6	0	6	0	0	11	11	0
cobranca-redesplan	100.00%	0	0	0	0	0	0	0	0
jobredesplan	100.00%	0	0	0	0	0	0	0	0
verdesmares	100.00%	5	4	0	1	0	3	3	0
solar	11.11%	17	0	13	4	1	27	21	6
unilab	100.00%	0	0	0	0	0	0	0	0
saoluiz	100.00%	1	1	0	0	0	0	0	0

VoicerWay Supervisor 2.0.08 © Copyright 2013 Way Tecnologia

Fonte: Elaborado pelo autor.

Na tela de monitoria, estão todos os DAC's, como exemplo da Figura 41, onde pode-se definir a informação buscando diretamente da tabela de registro de detalhes conforme a disposição das colunas (logado, disponível, ocupado, pausa, fila, recebidas, atendidas, abandonadas).

Figura 42 - Tela de monitoria. Ouvindo as ligações 01 e 02



Fonte: Elaborado pelo autor.

No relatório de monitoria, pode-se ouvir as gravações feitas pelo sistema, conforme a Figura 42 mostra. Este relatório trabalha com quartis, onde cada ligação pode receber uma nota através de um questionário. Existem quatro quartis, que são faixas de nota, o primeiro vai 0% até 50% do valor total do questionário de monitoria utilizado; o segundo vai dos 50% até 75%; o terceiro dos 75% até 95% e o quarto é para notas acima de 95%. Nesse relatório é possível visualizar quais operadores estiveram em cada quartil, como se pode ver na Figura 42 e, como medida gerencial, entender, definir e agir de forma corretiva e preventiva.

Figura 43 - Tela de login no módulo de relatórios

192.168.50.20/voxplus/reportsLogin.php

Alterar senha? | Usuário logado: Yuri Tanuska.

Início Realtime **Relatorios** Ferramentas

Ativo Ura Agentes Abandonadas Consolidadas Login

Relatório de Login

Agente: Escolha um agente... Data: De hoje De hoje Escolher Dia Busca

Ajuda

VoxPlus IP 1.0.4 © Copyright 2014

Fonte: Elaborado pelo autor.

Os relatórios permitem selecionar um determinado período (por data) e uma operação para visualizar, em seguida, um quadro com os dados do período. O sistema conta com relatório de monitoria. A Figura 43 apresenta a tela de seleção de parâmetros para os relatórios de monitoramento.

Figura 44 - Tela de Relatório de *login/pausa* por agente

Relatório LOGIN/PAUSAS para o agente Emanuel S. Ferreira (1102) .

Em: 10-03-2015 Utilizando o ramal: 92004.

Hora	Tempo Decorrido	Acumulado Pausa	Acumulado Total	Atividade
05:55:43	--	--		Efetuoou login
05:55:49	00:00:06	--	00:00:06	Em atendimento
07:01:53	01:06:04	--	01:06:10	Efetuoou login
07:06:07	00:04:14	--	01:10:24	Em atendimento
07:12:20	00:06:13	--	01:16:37	Pausado [Pausa - Outros]
07:22:03	00:09:43	--	01:26:20	Em atendimento
08:17:14	00:55:11	--	02:21:31	Efetuoou login
08:21:11	00:03:57	--	02:25:28	Em atendimento
08:37:33	00:16:22	--	02:41:50	Pausado [Pausa - Outros]
08:40:10	00:02:37	--	02:44:27	Em atendimento
10:19:34	01:39:24	--	04:23:51	Efetuoou login
10:22:53	00:03:19	--	04:27:10	Em atendimento
Hora	Tempo decorrido	Acumulado Pausa	Acumulado Total	Atividade

VoxPlus IP 1.0.4 © Copyright 2014

Fonte: Elaborado pelo autor.

O sistema permite ouvir as ligações e preencher um questionário sobre a qualidade do atendimento. No sub-nível da ligação é inserido o registro da monitoria. A Figura 44 apresenta um exemplo de relatório de Login/Pausa por agente.

5 TESTES

Neste capítulo são descritos os testes realizados, os resultados alcançados, e os problemas encontrados juntamente com suas soluções.

Uma forma de gerar e avaliar o resultados dos testes, é a utilização de aplicativos (*software sniffer*) que permitem a coleta de pacotes de uma rede. Todavia, sem uma interpretação devida, os pacotes que foram capturados ou coletados na rede, podem não parecer consistentes, pois o tipo de arquivo que é gerado (*log*) é composto basicamente por dados em formato binário.

É importante observar que existem outras ferramentas que complementam as de captura de pacotes, são elas: os monitores de rede e os analisadores de protocolo. Estes possibilitam a visualização do tráfego de forma organizada e estruturada. A utilização desse tipo de aplicativo permite também diagnosticar erros, gerar estatísticas além de uma visão gráfica do fluxo de dados (*VoIP*) que trafegam na rede. Pode-se citar, como exemplo, os seguintes aplicativos: o *TcpDump* - é um aplicativo de captura de pacotes. Ele fornece relatórios estatísticos sucintos. Outra característica importante é que depois da coleta da amostra (tráfego de rede), existe a possibilidade dos dados gerados serem direcionados para uma pasta, onde poderão ser analisados futuramente através de outros processos. Outra ferramenta muito utilizada para esse tipo de análise de dados é o *Wireshark*. Este utiliza alguns processos (motor de captura) similares aos do *TcpDump*. Basicamente o aplicativo tem dois tipos de filtros: de captura (baseado no *TcpDump*) e o de amostragem que é utilizado para controlar o que está sendo observado no momento da execução do monitoramento. O aplicativo possui uma interface gráfica amigável e intuitiva. Dessa forma, pode-se afirmar, que é imprescindível a utilização de um aplicativo de análise e monitoramento, além de um gerador de tráfego, na fase de testes.

Durante esta fase, os terminais utilizaram o *softphone X-lite*. Foram executados os seguintes testes:

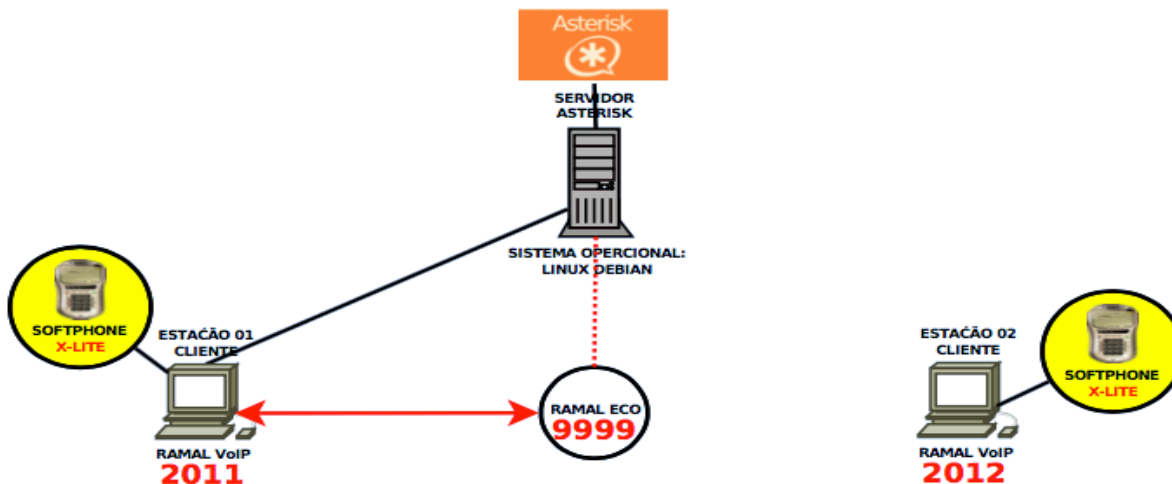
- Estabelecimento de uma chamada entre dois terminais (ramais);
- Estabelecimento de múltiplas chamadas simultâneas;
- Conferência;

- Transferência de chamada, captura de chamada e grupo de sinalização;
- Integração com a RTPC;
- Estabelecimento de chamadas entre as duas centrais (VoIP x RTPC).

5.1 ESTABELECIMENTO DE UMA CHAMADA ENTRE DOIS TERMINAIS

O primeiro passo deste teste foi realizar uma chamada para o ramal de eco, que apenas retorna o áudio de entrada da chamada. Assim, é possível testar o registro do cliente na central e a qualidade do áudio. A Figura 45 apresenta o teste com o ramal de eco.

Figura 45 – Teste de ramal de eco



Fonte: Elaborado pelo autor.

Após testar o ramal de eco, uma chamada foi estabelecida do terminal 2011 para o terminal 2012, com duração de 2 minutos, de acordo com a Tabela 3.

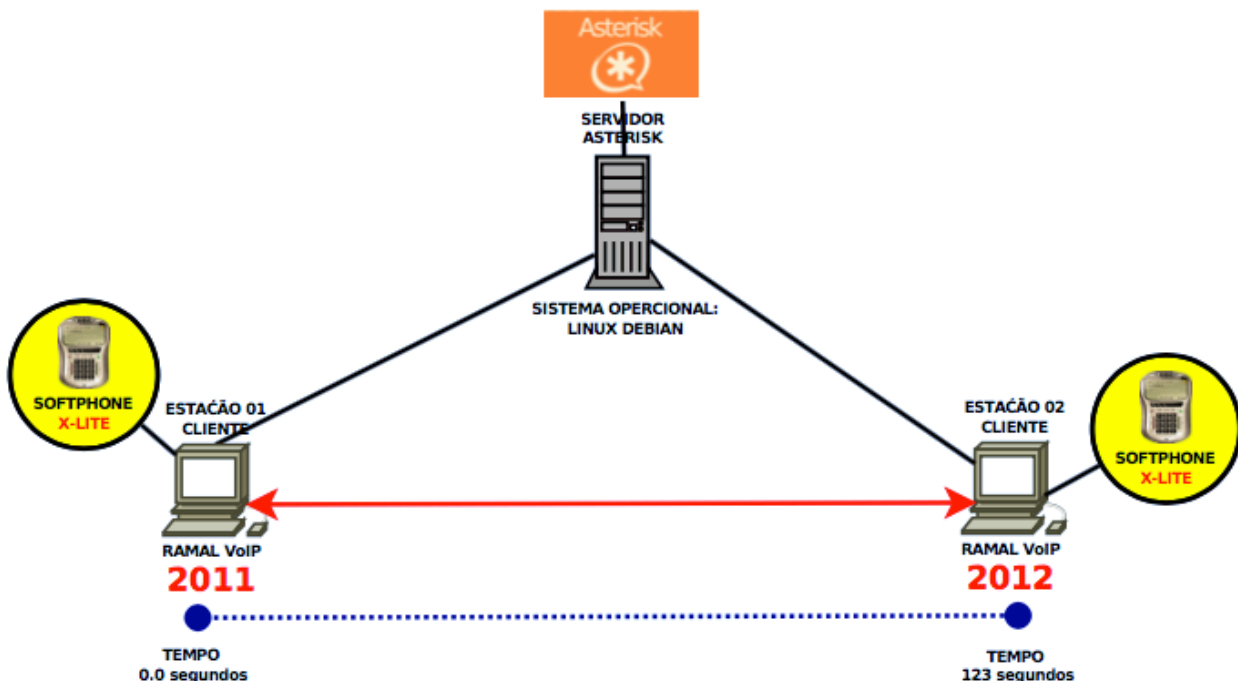
Tabela 3 - Teste de chamada entre dois terminais

Origem	Destino	Duração
2011	9999	29 segundos
2011	2012	123 segundos

Fonte: Elaborado pelo autor.

O resultados observados nas execuções (teste), corresponderam às expectativas. Os clientes *VoIP (softphone)* se conectaram facilmente, sem problemas. Não foram necessárias alterações no cenário da *LAN*. As chamadas entrantes foram estabelecidas, em geral, de forma normal e com excelente qualidade de áudio. A qualidade das ligações se manteve semelhante a da RTPC. A Figura 46 apresenta uma visão do teste executado entre os ramais *VoIP 2011* e *2012*.

Figura 46 – Teste executado entre os ramais *VoIP 2011* e *2012*



Fonte: Elaborado pelo autor.

5.2 ESTABELECIMENTO DE MÚLTIPLAS CHAMADAS SIMULTÂNEAS

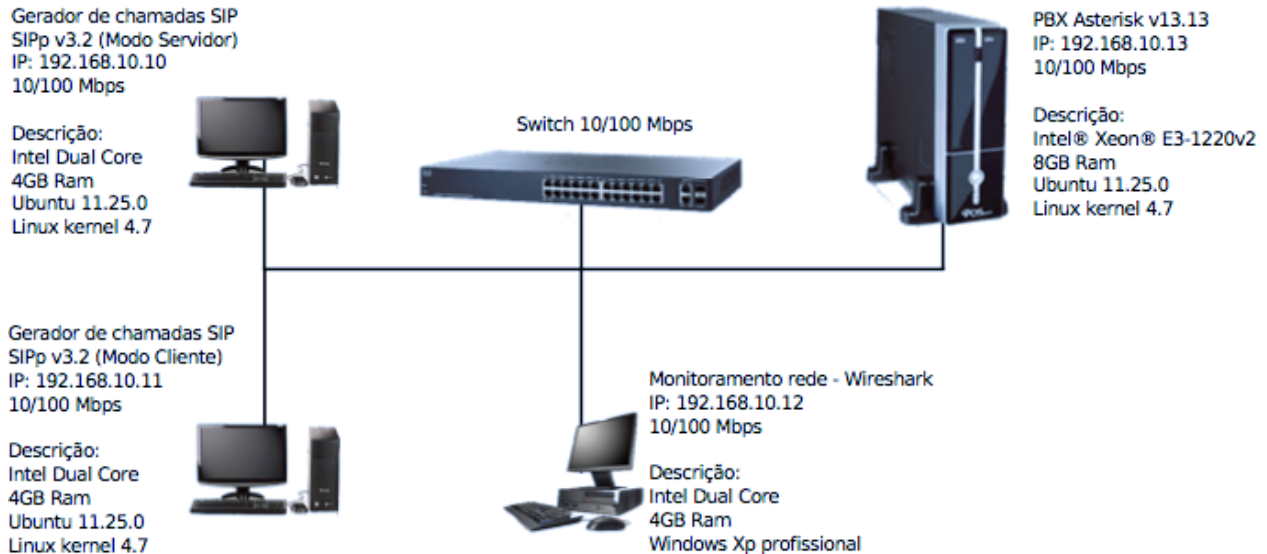
Este teste pretende definir se o servidor *PBX Asterisk* é capaz de trabalhar com múltiplas chamadas simultâneas sem perda na qualidade das chamadas. No cenário de múltiplas chamadas simultâneas, quando é executada uma chamada para um dado ramal (ponto a ponto), o *PBX Asterisk* direciona a chamada para o receptor, que foi previamente configurado (*script - arquivos de configuração*), e determina qual *codec* dever ser utilizado para atender ao receptor. Dessa forma, o *Asterisk* atua

também como um agente intermediador da chamada, podendo esse procedimento gerar um overhead no hardware do servidor *Asterisk*. Propõe-se, desse modo, estabelecer um parâmetro de qual o número máximo de chamadas simultâneas a partir do qual o *PBX Asterisk* consegue processar com essa configuração de hardware, e qual o impacto quando o *PBX Asterisk* executa os processos de decodificação, ou seja, viabilizar a comunicação entre dois terminais. Com intuito de facilitar a captura e a interpretação dos dados gerados neste teste, as estações de trabalho (origem e destino) utilizaram o mesmo codec (*G711a*) para o tráfego de voz.

Ambiente de teste:

Para analisar o desempenho do *PBX Asterisk* em um cenário de múltiplas chamadas simultâneas, foram efetuados testes de carga. Para isso, foram empregados os seguintes recursos: vinte e dois computadores conectados em uma rede de cabeamento metálico (*UTP Cat5e*), através de um switch (*Fast Ethernet*) de vinte e quatro portas com velocidade de 10/100 *Mbps*. Os testes foram executados utilizando como base para estação servidor uma máquina modelo *PowerEdge T110 II*, da marca *Dell*, com o Processador *Intel® Xeon® E3-1220v2*, 8GB de memória e 2HDs de 1TB *Raid 1*. Para as estações cliente, computadores *desktops* com processador *Intel Dual Core*, com 4GB de memória *Ram* e 500GB de HD (*Hard Disk*). Em cada conjunto de duas estações (transmissor/receptor) foi configurado uma das estações para gerar as chamadas *SIP*, utilizando para isso o módulo servidor da aplicação *SIPp*, de forma a gerar um tráfego no *PBX Asterisk*. A outra estação atua como o receptor da chamada *SIP*, executando a versão cliente da aplicação *SIPp*. Por fim, existe um computador que monitora todo o tráfego da rede, utilizando os recursos do software *Wireshark*. A Figura 47 ilustra de forma resumida a configuração de rede utilizada bem como as características de cada computador envolvido no processo.

Figura 47 – Resumo do cenário para os teste de múltiplas chamadas simultâneas



Fonte: Elaborado pelo autor.

Configuração do **PBX Asterisk**:

Seção: *extension.conf*

```
[sipp_pp]
```

```
exten => 222333444,1,Dial(SIP/sippuas)
```

Sempre que for executada uma chamada para a extensão 222333444 o **PBX Asterisk** vai encaminhar a chamada para o receptor que está utilizando o *sippuas*.

Seção: *sip.conf*

```
[sipp]
```

type=friend: Para que uma estação possa executar uma chamada para um ramal destino é necessário que o parâmetro "*sippuas*" receba os privilégios do perfil "*sipp*", que por sua vez contém o contexto *sipp_pp* (*context= sipp_pp*). Desse modo a estação pode receber e efetuar chamadas.

username=sip: informa o nome da estação de origem da chamada.;

host=dynamic: localização da rede da estação cliente. No caso do cliente ser do tipo “*dynamic*” ele pode registrar-se de forma independente do endereço *IP* utilizado;

context=sipp_pp: determina qual o contexto a ser utilizado pelo perfil, segundo o arquivo de configuração *extension.conf* , que contem o *dial plan*;

canreinvite=no: Por padrão o *PBX Asterisk* tenta redireccionar o tráfego *RTP* directamente do emissor para o receptor. Alguns Aplicativos *VoIP* não suportam esta funcionalidade;

insecure=very: Habilita este perfil para não utilizar autenticação e nem verificar o endereço *IP*;

nat=no: Não foi necessario a utilização de interface *NAT*, por se tratar de uma *LAN*;

```
[sippuas]
```

```
type=friend
username=sippuas
host=192.168.10.0
```

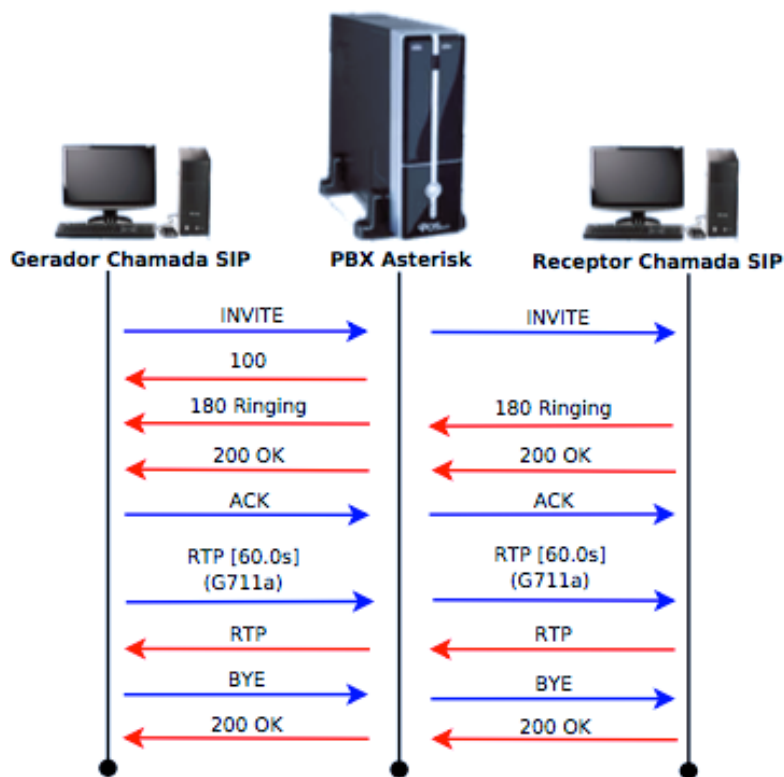
```
context=sipp_pp
insecure=very
canreinvite=no
nat=no
```

Esta seção localiza o ramal pelo *PBX Asterisk*, uma vez que a aplicação *SIPp* não registra de forma automatica os clientes.

Estrutura do teste:

Nos testes de múltiplas chamadas, o software *SIPp* foi configurado de forma a gerar chamadas *SIP* em uma estação (origem), assim como receber essas chamadas em uma estação destino. A Figura 48 ilustra o fluxo de comunicação de mensagens *SIP* e *RTP*, de acordo com o teste realizado, tendo como referência uma amostragem de uma estação origem para uma destino.

Figura 48 – Fluxo de comunicação de mensagens *RTP* e *SIP*



Fonte: Elaborado pelo autor.

Inicialmente o transmissor (gerador de chamadas *SIP*) executa uma chamada para o ramal de destino, essa chamada é recebida pelo *PBX* e redirecionada para o receptor (estação cliente). O transmissor envia então o tráfego *RTP* de voz durante 60 segundos, que está codificado com o codec *G711a*. A estação receptora recebe o tráfego.

O Asterisk permite que o codec que foi utilizado pelo transmissor para enviar o tráfego de voz através do protocolo *RTP* seja diferente do suportado pela estação cliente. Isso é possível graças a capacidade do Asterisk de transcodificar o tráfego entrante e nivelar ou traduzir para um codec que seja compatível com a estação receptora. No ambiente de teste utilizado foi adotado como padrão a ser utilizado nas estações (transmissor/receptor) o codec *G711a*.

Para testar um cenário de sobrecarga, foram geradas três chamadas a cada segundo. Foi adotado como parâmetro de avaliação o número de chamadas

simultâneas. Essa condição de carga na rede, serve para avaliar até onde o *PBX Asterisk* consegue processar o tráfego de voz entrante, sem haver retransmissão de mensagens devido ao timeout, por ausência de resposta do *Asterisk*. Com este teste pretende-se estabelecer um parâmetro de monitoramento de overhead customizado para esse hardware específico.

Para a estação transmissora, onde são geradas as chamadas, o *PBX Asterisk* poderá sempre responder com uma mensagem do tipo “*invite*” e “*bye*”. No caso da estação receptora das mensagens, o *PBX Asterisk* deverá enviar sempre às mensagens “*200 Ok*”. Conforme está ilustrado na Figura 48. Para a realização desse teste, foi capturado tráfego durante três minutos (180 segundos) utilizando para isso o aplicativo *Wireshark*.

Pretende-se:

1. Verificar qual o número máximo de chamadas, geradas de uma estação transmissora para uma receptora simultaneamente, que o *PBX Asterisk* pode suportar, ou seja, o número máximo de chamadas sem retransmissão de mensagens devido ao timeout;
2. Acompanhar o desempenho do *PBX Asterisk* utilizando o mesmo codec no transmissor e no receptor, verificar quais as principais vantagens de desvantagens de se utilizar um só tipo de *codec*;
3. Monitorar o comportamento do servidor *PBX Asterisk*, tendo como indicadores os logs que incidem sobre utilização da *CPU*, de memória, disco rígido, contabilização de processos, tráfego de rede, entre outros.

Propõe-se avaliar o desempenho do *PBX Asterisk* em relação ao número de chamadas recebidas e que foram tratadas corretamente. Para isso, foram utilizados vários parâmetros de configuração executados no aplicativo *SIPp* versão cliente. Foi definido como parâmetro variável o limite máximo de chamadas simultâneas.

Para se obter uma amostragem de dados mais precisa a respeito do tráfego gerado e recebido no servidor *Asterisk*, foi utilizado, também como base, as estatísticas geradas pelo comando “*ifconfig*” do *linux*. Dessa forma, esses dados irão complementar os que foram capturados pelo computador que está monitorando a rede (modo furtivo)

com o aplicativo wireshark. O processo de captura dos dados foi executado no intervalo de aproximadamente 180 segundos.

Procedimento executado:

No servidor com *PBX Asterisk* instalado *IP:192.168.10.13*;

1. Executar no console do servidor *PBX Asterisk*: > sudo /usr/sbin/asterisk –vc ;
2. Iniciar a execução de um script desenvolvido em *perl* que reseta as estatísticas antigas já arquivadas pelo comando *ifconfig* e escreve os resultados obtidos em uma pasta após a execução do teste.

Na estação geradora das chamadas *SIP*, aplicação *SIPp* : *IP:192.168.10.10*:

3. Executar a aplicação *Sipp* como receptor de chamadas *SIP*, isto é, em modo servidor, indicando os parâmetros como, nome do pasta *.xml* que contém o cenário a utilizar no teste, é necessário ativar a opção para ecoar o tráfego *RTP* de voz recebido.

Na estação receptora de chamadas *SIP*, aplicação *SIPp*: *IP:192.168.10.11*:

4. Executar a aplicação *Sipp* como gerador de chamadas *SIP*, isto é, em modo cliente, a partir de um script que recebe como parâmetros os dados essenciais como, por exemplo, número máximo de chamadas em simultâneo, número de chamadas geradas por segundo, duração do teste (em minutos) e o nome da pasta *.xml* que contém o cenário a utilizar no teste.

Computador de monitoramento do tráfego: *IP:192.168.10.12*:

5. Captura o tráfego utilizando o *Wireshark*.

Resultados:

Os testes foram executados com uma duração de aproximadamente 180 segundos. Esse valor é aproximado, porque o *SIPp* client “espera” que todas as chamadas com status do tipo pendente finalizem o processo para serem computadas. A

variável que foi utilizada para testar a sobrecarga do sistema foi o número máximo de chamadas executadas simultaneamente. Sendo que o *SIPp* foi configurado para gerar no máximo três chamadas por segundo.

Foram considerados apenas os valores “estáveis”, ou seja, os dados que não apresentaram retransmissão ou timeout de mensagem *SIP*. Porém podem ocorrer perdas ao nível de pacotes *RTP*, devido ao overhead na rede.

Foi definido o número máximo de chamadas simultâneas em **100**, todas utilizando o mesmo codec (*G711a*) no transmissor e no receptor. Foram obtidos resultados contidos na Tabela 4.

Na Tabela 5, são mostrados os valores máximos “estáveis” atingidos, para as chamadas ponto a ponto para o codec (*G711a*).

Tabela 4 - Resultados obtidos para o teste de carga para cem chamadas simultâneas

Parâmetro	Resultado
Codec	<i>G711a</i>
Máximo de chamadas simultâneas	100
Duração do teste em segundos	214,22
Total de chamadas	300

Fonte: Elaborado pelo autor.

Tabela 5 - Estatísticas de rede para o teste de chamadas simultâneas

Parâmetro	Resultado encontrado
<i>Downlink (MB)</i>	366,8
<i>Uplink (MB)</i>	366,1
<i>DownLink - Carga na rede (Mbit/s)</i>	14,37
<i>UpLink - Carga na rede (Mbit/s)</i>	14,34

Fonte: Elaborado pelo autor.

Tabela 6 - Estatísticas sobre fluxos de áudio de voz RTP para o teste de chamadas simultâneas

Parâmetros	Resultados
Jitter (ms) - Média	5,61
Jitter (ms) - Máximo	7,53
Jitter (ms) - Mínimo	0,8
Latência (ms) - Média	45,3
Latência (ms) - Máximo	62,78
Latência (ms) - Mínimo	26,63
Total N° de fluxos RTP analisados	256
Total N° de fluxos RTP com perdas	18
% média de perda por fluxo RTP	0,22

Fonte: Elaborado pelo autor.

Análise dos resultados obtidos

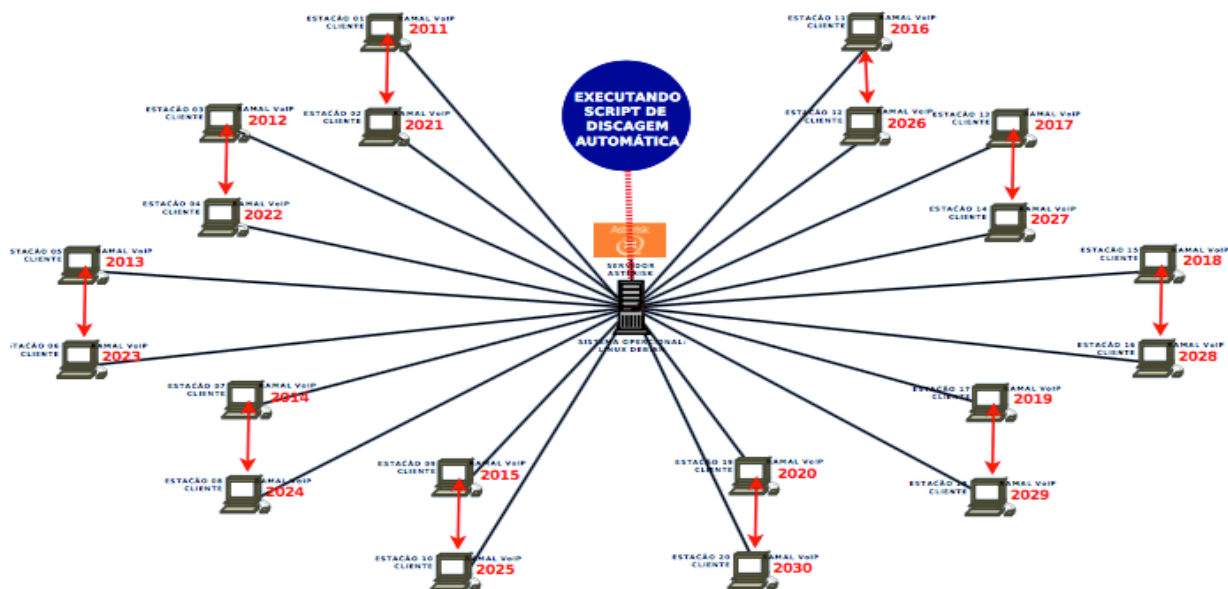
Pela análise das Tabelas 5 e 6, pode-se observar que, com a utilização do *codec G711a*, para executar chamadas simultâneas, o servidor *PBX Asterisk* suporta executar o encaminhamento de chamadas de forma que os atores envolvidos no processo (*transmissor e receptor*) possam se comunicar de forma estável e satisfatória para os padrões de compreensão da voz humana, para o máximo de 103 chamadas simultâneas. Observa-se que quando os interlocutores executam o mesmo *codec*, *G711a*, a quantidade de chamadas realizadas simultaneamente que o *PBX Asterisk* comporta, está diretamente relacionada com o número de bytes necessários para transmitir a informação da voz, isto é, um *codec* de baixa complexidade comprime menos o áudio. Logo, pode-se verificar que o consumo de largura de banda é maior para se transmitir a informação e que esse custo (consumo de largura de banda) diminui à medida que se utiliza *codecs* mais complexos que executam processos de compactação mais avançados.

Em relação ao fluxo de dados *RTP*, pode-se verificar, na Tabela 6, que o comportamento modal do *Jitter* (variação do tempo de atraso entre pacotes de dados

enviados de forma sucessiva) não ultrapassou os 15 ms. Como forma de atenuar os impactos do *Jitter* no receptor e conservar a taxa de entrega contínua, foi habilitado na aplicação (*SIPp*) a criação de um *buffer*. Esse recurso pode ser configurado para ter tamanho fixo ou dinâmico, conforme o comportamento do *Jitter*.

Os resultados obtidos com as chamadas múltiplas foram satisfatórios, e a qualidade do áudio ficou estável durante todo o período de teste. Não foram observados cortes ou atrasos na entrega dos pacotes *VoIP*, relacionados a chamadas de teste. A Figura 49 apresenta uma ilustração do estabelecimento das múltiplas chamadas simultâneas.

Figura 49 – Teste de estabelecimento de múltiplas chamadas simultâneas



Fonte: Elaborado pelo autor.

5.3 CONFERÊNCIA

Em um ambiente e chamadas em conferência, quando um usuário executa uma chamada para um dado ramal ou estação, a chamada é direcionada pelo servidor PBX Asterisk para uma "sala" de conferência anteriormente configurada no sistema. Dependendo dos parâmetros habilitados no arquivo de configuração, poderá ser solicitado ao terminal que originou a chamada a apresentação de um código de

autorização (senha) para ter acesso a conferência.

Na execução deste teste, quando um interlocutor entra em uma sala de conferência, o mesmo mantém-se conectado durante sessenta segundos, ao longo do qual é gerado tráfego de voz. O cenário criado representa um dos piores casos possíveis, pois, em um ambiente real, dificilmente seria observado um tráfego contínuo entre todos os participantes da conferência. Observa-se que, falando todos ao mesmo tempo, ninguém se entenderia impossibilitando assim a intercomunicação.

Pretendem-se verificar, neste caso, qual o comportamento do *PBX Asterisk*, tendo como principal variável global o desempenho do hardware do servidor em um cenário com chamadas em conferência. Os dados gerados neste teste, ajudarão a determinar a capacidade máxima de processamento de chamadas simultâneas, isto é, definir o número máximo de participantes que o *PBX Asterisk* pode suportar durante o intervalo de sessenta segundos, além de observar o que acontece quando existe mais de uma sala de conferência.

Para executar as chamadas em conferência foi habilitado o módulo *MeetMe* e o *Ztdummy*, que são nativos do Asterisk. O *Ztdummy* faz a função de um *time source* (temporizador) e é de fundamental importância para o funcionamento do módulo *MeetMe*, que serve de template para o ambiente da "conferência". A grande vantagem de se utilizar esses dois módulos é que ambos manipulam recursos do próprio Kernel do Asterisk, colaborando desse modo com a otimização dos tempos de resposta, haja visto que essa negociação sempre ocorrerá internamente no Asterisk, evitando consequentemente um acesso externo a outro aplicativo ou até mesmo a outro hardware, que teria possivelmente um custo superior no tempo de resposta.

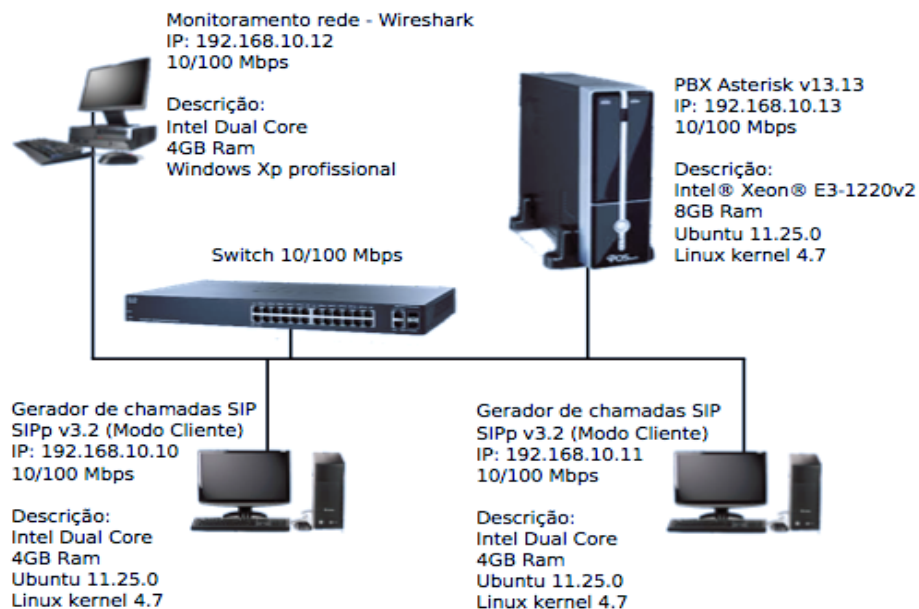
Ambiente de teste:

Para qualificar o comportamento do desempenho para o teste de chamadas em conferência, foram executados testes de carga. Para criar este cenário, foram utilizados os seguintes recursos: onze computadores conectados em uma rede de cabeamento metálico (*UTP Cat5e*), através de um switch (*Fast Ethernet*) de vinte e quatro portas com velocidade de 10/100 *Mbps*. Os testes foram executados utilizando como base para a estação servidor do *PBX Asterisk* uma máquina modelo *PowerEdge*

T110 II, da marca *Dell*, com o processador *Intel® Xeon® E3-1220v2*, 8GB de memória e 2HDS de 1TB Raid 1. E para as estações cliente, computadores *desktops* com processador *Intel Dual Core*, com 4GB de memória *Ram* e 500GB de *HD (Hard Disk)*.

Para gerar as chamadas *SIP*, foram utilizados dez computadores (estações *desktop*), executando no total até 02 instâncias da aplicação *SIPp*. Similar ao realizado no teste de chamadas simultâneas existe também um computador que monitora todo o tráfego de rede. A Figura 50 ilustra de forma resumida a configuração de rede empregada bem como a configuração de cada computador.

Figura 50 – Resumo do cenário de rede para o teste de chamadas em conferência



Fonte: Elaborado pelo autor.

Configuração do *PBX Asterisk*:

Arquivo: *meetme.conf*

```
[rooms]
```

```
; Criação das "salas" de conferência:
```

```
conf => 1234
```

```
conf => 1235
```

```
conf => 1236
```

```
conf => 1237
```

Arquivo: *extension.conf*

Sempre que seja efetuada uma chamada para o ramal 8500, 8600, o *PBX Asterisk* irá criar e adicionar um utilizador à chamada em conferência na “sala” 1234 e 1235.

```
[conferencia]
exten => 8500,1,Meetme(1234)
exten => 8600,1,Meetme(1235)
exten => 8700,1,Meetme(1236)
exten => 8800,1,Meetme(1237)
```

Arquivo: *sip.conf*

Para que um usuário possa participar de uma das “salas” de conferência é necessário que seja habilitado o perfil “*sipp*”, que contém o contexto para a conferência (*context=conferencia*).

```
[sipp]type=friend
;indica que este perfil pode receber e efectuar chamadas
context=conferencia

host=dynamic

username=sip

canreinvite=no

insecure=very

nat=no
```

Estrutura do teste:

Neste teste foi estabelecido como parâmetro máximo de execução, a discagem de duas chamadas por segundo, afim de gerar um overhead no servidor *Asterisk*. Da mesma forma que foi executado no teste anterior (chamadas simultâneas), foi limitado o número chamadas simultâneas para se verificar qual o limite ocorrência

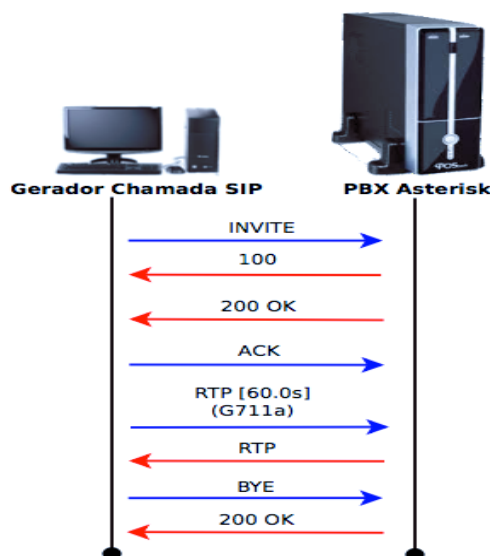
da retransmissão de mensagens *SIP* (*Invite* ou *Bye*) em virtude do timeout por ausência de resposta do *PBX Asterisk*. Quando o transmissor (aplicação *SIPp* em modo cliente) executa uma chamada para um dos ramais destino (8500 e 8600), a chamada é direcionada para uma sala de conferência.

A Figura 51 apresenta um detalhamento do fluxo de mensagens *SIP* e *RTP*, de acordo com o teste de carga executado para as chamadas em conferência. O tráfego de voz *RTP* estabelecido para o *PBX Asterisk* foi capturado de forma similar ao procedimento executado no teste de chamadas simultâneas.

Pretende-se:

1. Executar testes para um ambiente de conferência utilizando para isso o número máximo de conexões simultâneas, antes de haver retransmissão por falta de resposta do servidor *PBX Asterisk*. Executar o teste com tráfego de voz *RTP*, utilizando o codec *G711a* como codificador padrão.
2. Verificar qual o quantidade máxima de chamadas simultâneas em um cenário com mais de uma sala de conferência a ser executada de forma paralela.

Figura 51 – Fluxo de mensagens *RTP* e *SIP* para o teste de carga nas chamadas em conferência



Para uma melhor observação do desempenho do PBX Asterisk, tendo como referência o número de chamadas recebidas e corretamente tratadas, foram configurados vários procedimentos na aplicação cliente do *SIPp*. Para isso foi definido como parâmetro variável limite máximo de chamadas executadas simultaneamente.

O monitoramento do servidor *PBX Asterisk* e da rede utilizada para o teste, foi feito da mesma forma que no teste anterior (chamadas simultâneas).

Procedimento executado:

Computador (servidor) com PBX Asterisk instalado. *IP*:192.168.10.13

1. Executar no console do servidor *PBX Asterisk* o seguinte comando: `> sudo /usr/sbin/asterisk -vc`.
2. Iniciar a execução de um *script* em *perl* que reseta o histórico de estatísticas apresentadas pelo comando *ifconfig*, na sequência escreve os resultados obtidos em um arquivo de log após a finalização do teste.

Computadores (cliente) *SIPp*: *IP*:192.168.10.10:

3. Inicia a execução do aplicativo cliente *SIPp*, a partir de um *script* que foi configurado com os seguintes parâmetros: número máximo de chamadas simultâneas, número de chamadas geradas por segundo, tempo de duração do teste (em segundos) e o nome do arquivo **.xml*, que contém as configurações do cenário de teste.

Computador de monitoramento de tráfego : *IP*: 192.168.10.12:

4. Inicia a captura do tráfego de rede através do aplicativo *Wireshark*.

Resultados:

Foram executados testes com tempo de duração de aproximadamente de 180 segundos. O intervalo de tempo não é exato e sim aproximado devido o aplicativo cliente *SIPp* aguardar que todas as chamadas pendentes sejam finalizadas para encerrar o processamento, conforme a Tabela 7. A variável definida para testar a carga do sistema foi o número máximo de chamadas. Neste teste foram geradas por estação duas chamadas por segundo. Cada chamada conectada a sala de conferência tem a

duração aproximada de um minuto em áudio de voz (*RTP*) mais sinalização *SIP*, observa-se que em média cada chamada teve a duração de 01:00:027 minutos.

Tabela 7 - Estatísticas de rede obtidas pela informação do comando ifconfig e resumo da aplicação *SIPp*

Parâmetros	Teste 01	Teste 02	Teste 03
Nº de conferências em simultâneo	1	2	3
<i>Codec</i>	<i>G711a</i>	<i>G711a</i>	<i>G711a</i>
Máximo de chamadas simultâneas obtidas	90	40	25
Total de chamadas	270	240	225
<i>Bytes</i> recebidos (MB)	166,4	147,9	138,5
<i>Bytes</i> transmitidos (MB)	153	134,6	128,3
Carga na rede (<i>Mbit/s</i>) - <i>DownLink</i>	6,13	6,16	6
Carga na rede (<i>Mbit/s</i>) - <i>UpLink</i>	5,63	5,61	5,56
Intervalo de tempo do teste em segundos	228	201,4	193,69

Fonte: Elaborado pelo autor.

Análise dos resultados obtidos

Observa-se, que no cenário do teste de chamadas em conferência, é necessário que o sistema transcodifique o fluxo de áudio das chamadas entrantes em *PCM* linear (*Pulse-code modulation*) e, somente depois, iniciar o processo de recodificação para os múltiplos fluxos de saída. Pode-se afirmar que, se o *codec* é mais complexo, ou seja, imprime uma força de compactação maior no áudio de um fluxo *RTP* de voz, será maior o processamento envolvido na execução dessa atividade, conseqüentemente, gerando um *overhead* no sistema. Os *codecs* que utilizam maior compressão, possuem menor carga na rede, porém, o número de chamadas simultâneas processadas é menor em uma conferência.

Verificou-se que com o crescimento do número de conferências ocorrendo simultaneamente, o número total de chamadas que o *PBX Asterisk* processa, tende a diminuir. Nota-se que, cada conferência exige um processamento exclusivo, mesmo que temporário, que reduz, desse modo, o poder de processamento do servidor *Asterisk*. Mesmo no melhor caso, onde existe apenas uma conferência em execução,

pode-se observar uma sutil queda no rendimento do servidor *PBX Asterisk*.

Pode-se verificar, conforme Tabela 8 e 9, que a capacidade de processamento do hardware utilizado, atua diretamente como um fator de limitação para o crescimento do número de chamadas simultâneas.

Tabela 8- Estatísticas sobre fluxos de áudio de voz *RTP*, coletadas através do aplicativo *Wireshark*

Parâmetros	Teste 01	Teste 02	Teste 03
Nº de conferências em simultâneo	1	2	3
<i>Codec</i>	<i>G711a</i>	<i>G711a</i>	<i>G711a</i>
<i>Jitter</i> (ms) - média	5,71	3,6	5,69
<i>Jitter</i> (ms) - máximo	3,71	6,42	7,87
<i>Jitter</i> (ms) - mínimo	0,13	0,39	1,51
Latência (ms) - média	35,9	38,4	45,54
Latência (ms) - máxima	49,51	61,4	60,3
Latência (ms) - mínimo	13,46	22,37	20,22
Número de fluxos <i>RTP</i> analisados	208	186	168
Número de fluxos <i>RTP</i> com perdas	3	7	4

Fonte: Elaborado pelo autor.

Tabela 9- Percentagem média de utilização de *CPU* e memória *RAM*, na máquina onde está instalado o servidor *PBX Asterisk*

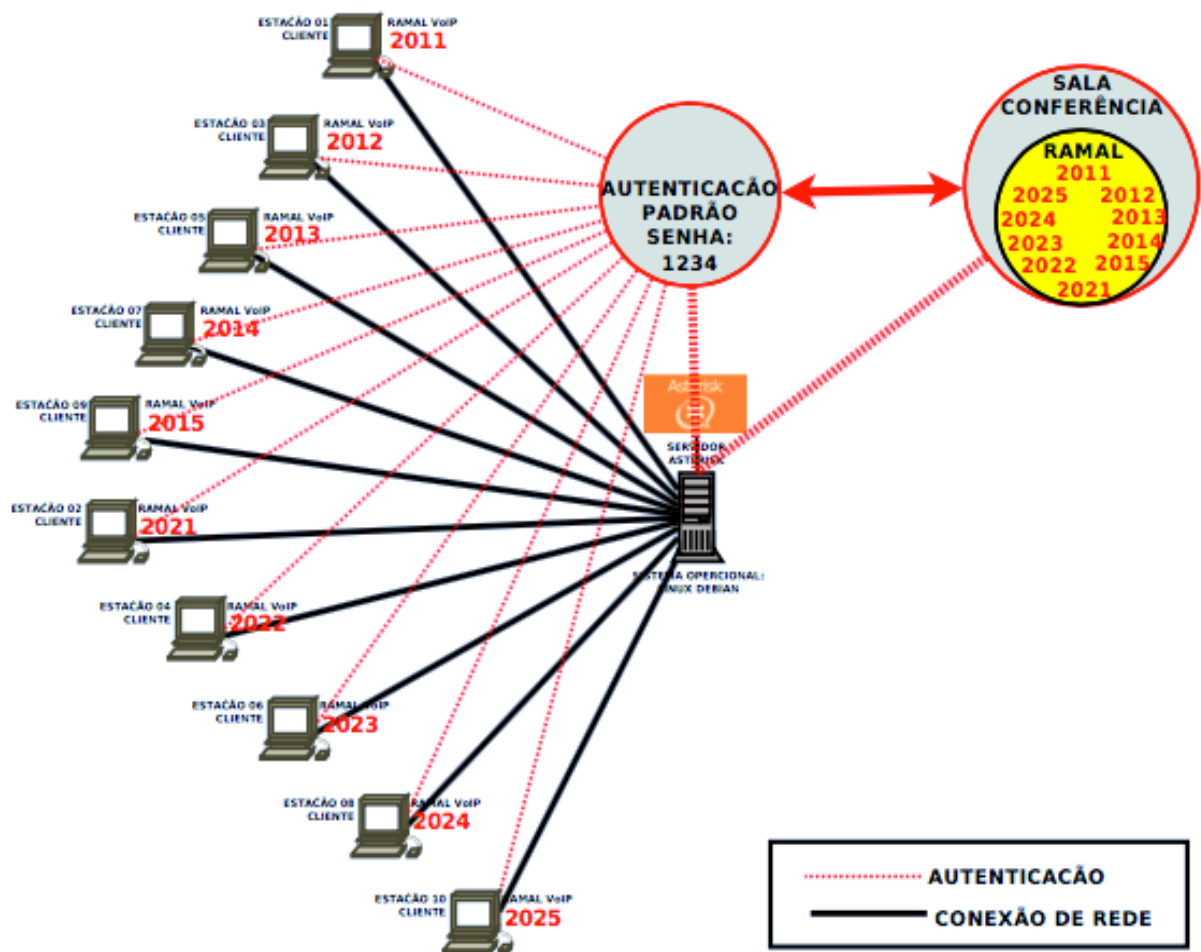
Parâmetros	Teste 01	Teste 02	Teste 03
Nº de conferências em simultâneo	1	2	3
<i>Codec</i>	<i>G711a</i>	<i>G711a</i>	<i>G711a</i>
Max. chamadas simultâneas obtido	90	40	25
Total de chamadas	270	240	225
Utilização média <i>CPU</i> (%)	63,47	77,78	80
Utilização média - Memória <i>RAM</i> (%)	47,26	66,14	66,4

Fonte: Elaborado pelo autor.

A execução dos testes relacionados as chamadas em conferência foi feita utilizando dez terminais em duas etapas:

01. Todos os terminais foram envolvidos na mesma conferência. A autenticação foi feita com a senha padrão 123456. A Figura 52 apresenta o cenário da primeira etapa.

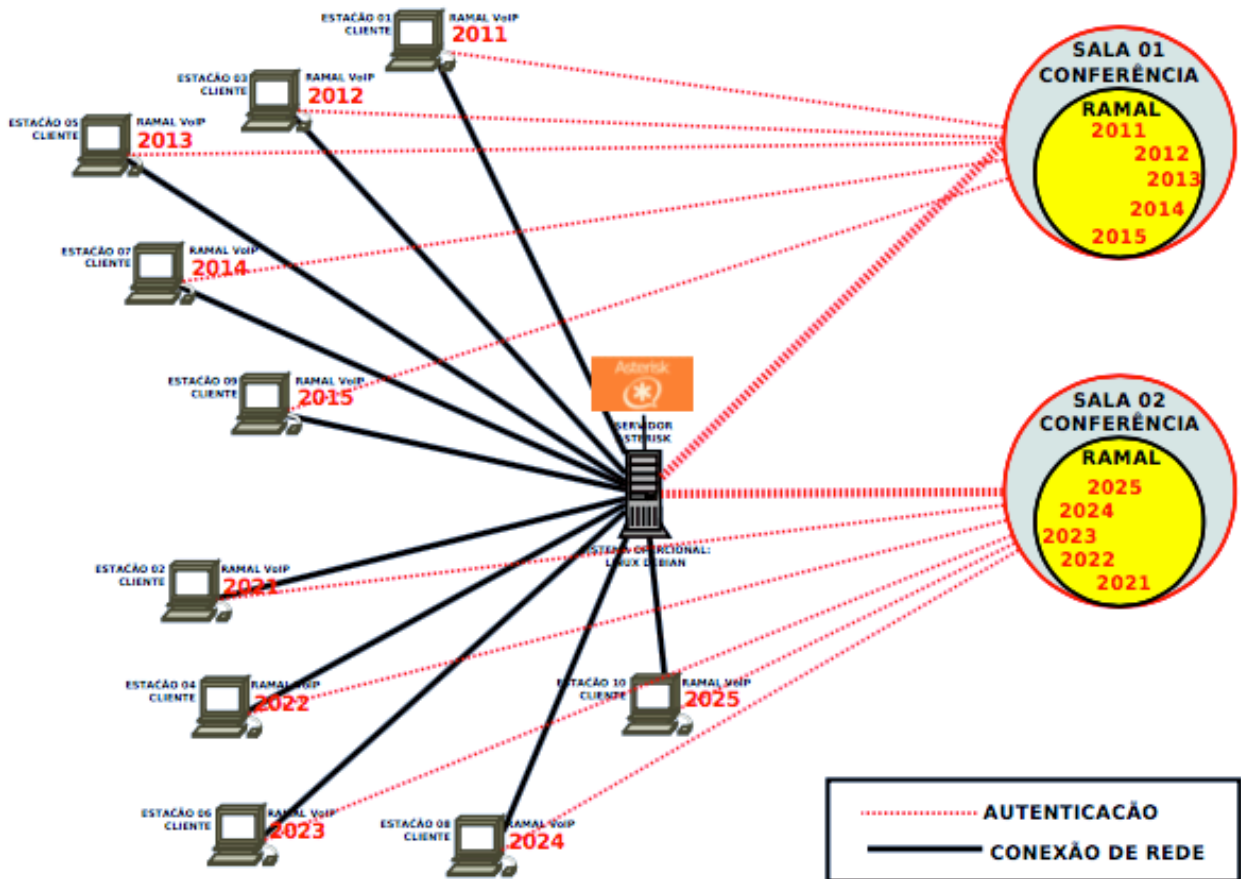
Figura 52 – Teste de conferência com todos os ramais



Fonte: Elaborado pelo autor.

02. Os terminais foram divididos em dois grupos entre duas conferências simultâneas sem autenticação. Nesse caso para sair da conferência, era necessário encerrar a chamada, ou enviar o sinal de *#. A Figura 53 apresenta o cenário da segunda etapa.

Figura 53 – Teste de conferência com duas salas e dois grupo de ramais



Fonte: Elaborado pelo autor.

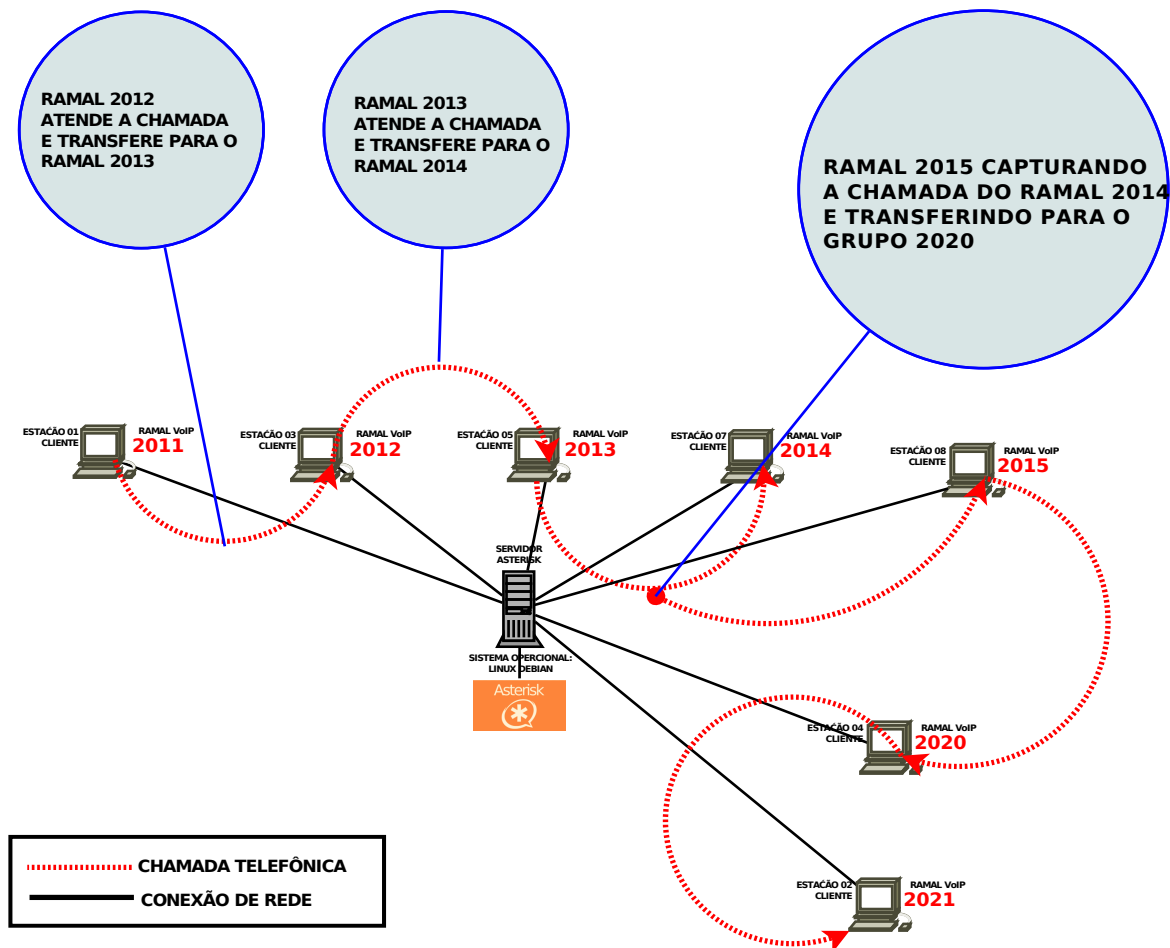
5.4 TRANSFERÊNCIA DE CHAMADA, CAPTURA DE CHAMADA E GRUPO DE SINALIZAÇÃO

A seguir são descritas as fases de execução do teste:

1. Ramal 2011 chama ramal 2012;
2. Ramal 2012 atende a chamada e transfere para o ramal 2013;
3. Ramal 2013 atende a chamada e transfere para o ramal 2014;
4. Ramal 2015 captura a chamada do ramal 2014 e transfere para o grupo 2020;
5. Ramal 2021 atende a chamada.

Todas as etapas foram bem sucedidas e as funcionalidades operam como esperado. A Figura 54 apresenta uma visão do teste de transferência de chamada, captura e sinalização de grupo de ramais.

Figura 54 – Transferência de chamada, captura e sinalização de grupo de ramais



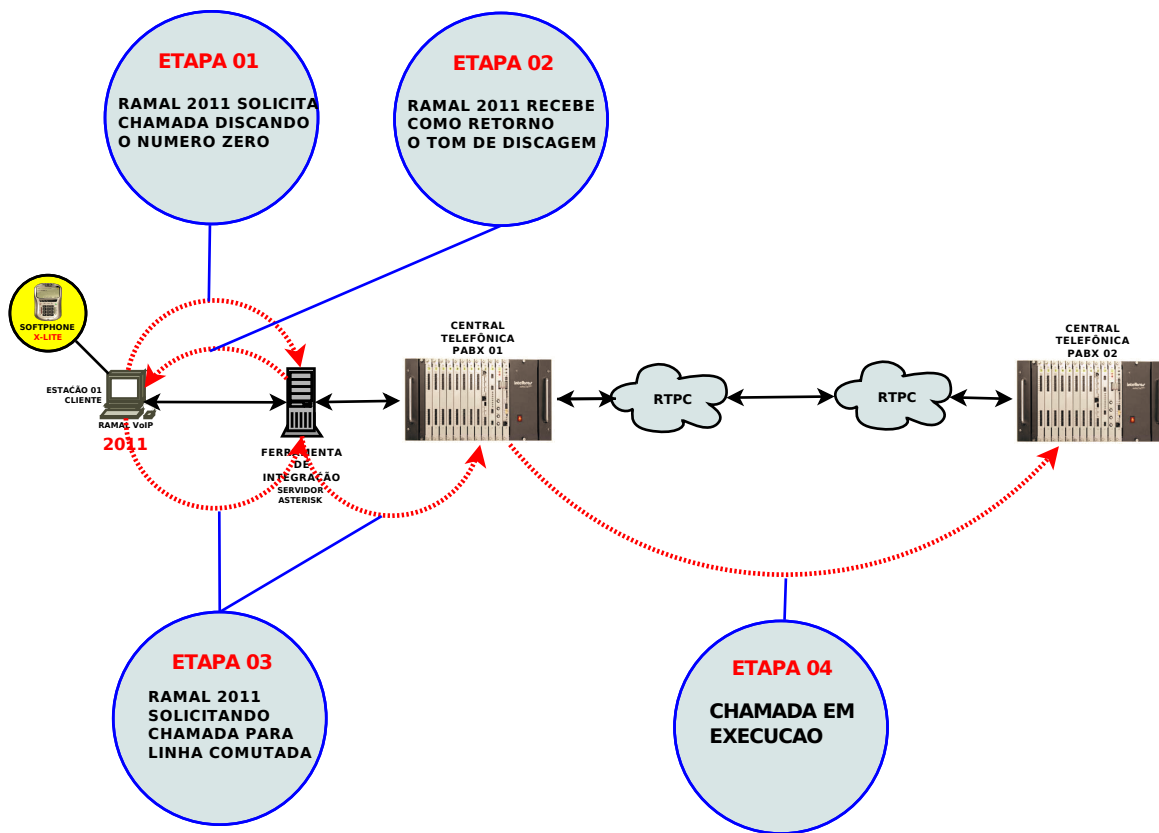
Fonte: Elaborado pelo autor.

5.5 INTEGRAÇÃO COM A RTPC

Os testes de integração com a RTPC, foram realizados em duas etapas. A primeira consistiu em estabelecer uma chamada para a RTPC e a segunda em receber uma chamada originada da RTPC. No primeiro ensaio, o terminal 2011 chamou o ramal 0 como forma de solicitar acesso a linha externa e obteve como retorno o tom de discagem. Logo na sequência foi executada uma outra chamada para uma linha

comutada local. No entanto, dessa vez os números transmitidos para a RTPC não incluíram o 0 no início, como havia sido configurado anteriormente. A Figura 55 apresenta o processo executado na primeira etapa.

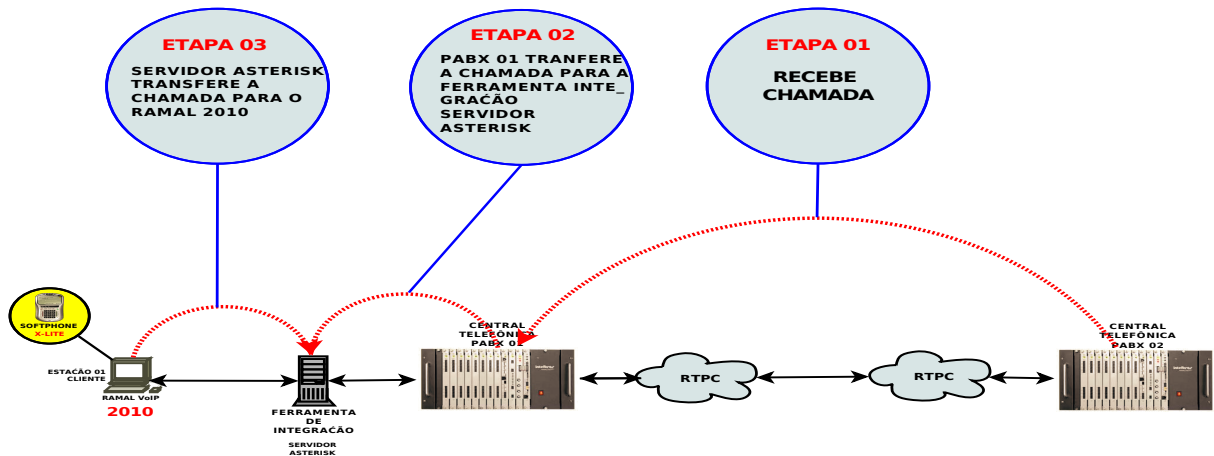
Figura 55 – Integração com a rede RTPC (primeira etapa)



Fonte: Elaborado pelo autor.

Na segunda etapa, uma chamada foi realizada para a linha comutada da central, que recebeu a chamada e transferiu para o grupo 2010. Na sequência o terminal 2011 atendeu à chamada. Cada amostragem de chamada teve a duração de aproximadamente 240 segundos. Em ambas as etapas a qualidade de áudio manteve-se similar à da RTPC, sem cortes e atrasos. A Figura 56 apresenta o processo executado na segunda etapa.

Figura 56 – Integração com a rede RTPC (segunda etapa)

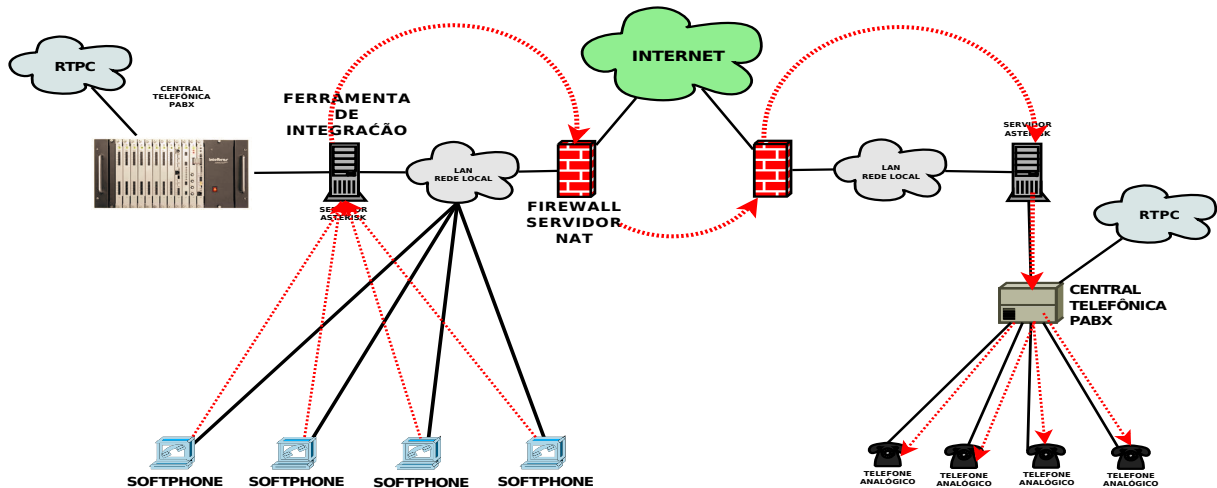


Fonte: Elaborado pelo autor.

5.6 ESTABELECIMENTO DE CHAMADAS ENTRE AS DUAS CENTRAIS

Com objetivo de executar testes de conectividade entre duas centrais, tendo como foco a avaliação do comportamento das chamadas em relação a qualidade de áudio, desempenho da transmissão de dados e transposição de ambientes com NAT e firewalls, foram executadas 04 chamadas simultâneas originadas de quatro ramais da central 01 para a central 02, com tempo de duração aproximado de 180 segundos com variações máxima de até 240 segundos, conforme a Figura 57.

Figura 57 – Estabelecimento de chamada entre duas centrais

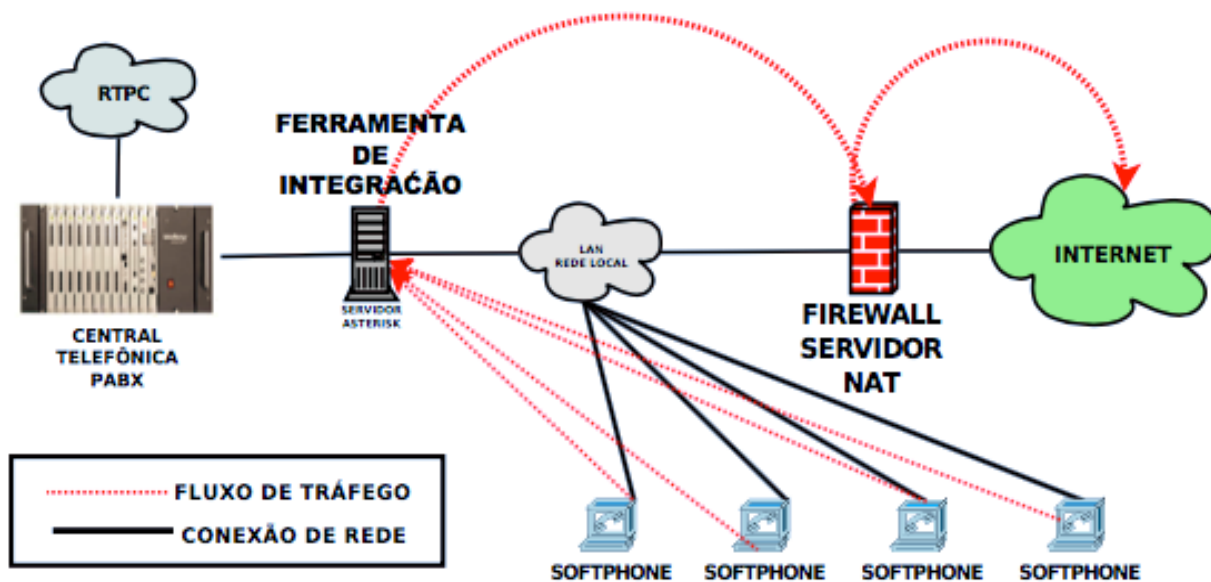


Fonte: Elaborado pelo autor.

Para se ter um cenário de amostragem, tendo como indicadores de qualidade a latência e a largura de banda disponíveis, foi inserido um tráfego adicional a rede local (LAN). Dessa forma foi possível visualizar o comportamento das chamadas (VoIP) em um ambiente com sobrecarga. Para simular este comportamento da rede, foi instalado um gerador de tráfego em dois hosts da rede, *host A* (servidor) e *host B* (cliente). A rota estabelecida entre os dois pontos (*hosts*), contemplou o maior número possível de ativos da rede. Logo após, foi inicializado um tráfego do tipo stream no qual foi possível simular um cenário de oscilação de fluxo de download e upload.

Foi observado que no entroncamento de chamadas do protocolo IAX, não houve problemas em atravessar os NATs e *firewalls* de ambos os lados, que foram devidamente configurados previamente. Todas as chamadas foram completadas com sucesso, porém ocorreram algumas variações na qualidade do áudio de acordo com o consumo de banda exigido pelos testes de trafego que foram executados com o aplicativo *Network Traffic Generator*. A Figura 58 apresenta uma visão do cenário com NAT.

Figura 58 - Topologia de uma rede implementando NAT



Fonte: Elaborado pelo autor.

Os testes no cenário de simulação de consumo de banda, mostram que as chamadas *VoIP* são bastante sensíveis a variações da latência de rede e da largura de banda. Foi verificado, claramente, que em condições de sobrecarga de uma rede corporativa, uma aplicação *VoIP* sofre impactos relevantes ao seu funcionamento. Nessa condição, é possível observar fatores como variações bruscas no *download* e *upload* da rede. Baseado nesse contexto pode-se afirmar que é de suma importância a implementação de uma política de *QoS* em uma rede com tráfego *VoIP*, principalmente pelas características que a voz exige. Pode-se verificar o comportamento desses eventos na Tabela 10.

Tabela 10 - Teste de Disponibilidade de banda em relação ao fluxo de chamadas *VoIP*

Disponibilidade de banda de rede	Resultado
Sem transferências (chamadas simples)	Não foi observado degradação das chamadas
Limite - 300/30 kbps (<i>Download/Upload</i>)	Não foi observado degradação das chamadas
Limite - 600/60 kbps (<i>Download/Upload</i>)	Pouca degradação e alguns cortes esporádicos
Limite - 900/90 kbps (<i>Download/Upload</i>)	Muita degradação e vários cortes (picotado no áudio)
Sem limitações (<i>Download / Upload</i>)	Muita degradação, vários cortes

Fonte: Elaborado pelo autor.

6 CONCLUSÃO

Embasado nas pesquisas executadas, e depois de analisar os resultados dos testes, conclui-se que a implementação de uma ferramenta para integração dos principais canais de comunicação com o cliente em um *Contact Center*, estruturada em uma plataforma baseada em um *PBX VoIP (Asterisk)*, é totalmente possível e viável, principalmente quando se analisa a perspectiva custo versus benefício, que em muitos casos inviabiliza a aquisição de uma solução desse tipo por empresas de médio e pequeno porte. Além disso, o *Asterisk*, assim como todos os aplicativos envolvidos no processo de desenvolvimento desta ferramenta, vem agregar uma grande flexibilidade devido a todos serem baseados em um domínio de visão *Open Source*.

O cenário criado foi perfeitamente capaz de satisfazer os objetivos dos testes, assim como simular o comportamento de uma rede de grande porte, como é normal encontrar em ambientes de *Contact Center*.

A qualidade das chamadas, e da voz, mantiveram-se em padrões aceitáveis para os indicadores de um ambiente real de *Contact Center*, levando em consideração o comportamento normal da voz. Os resultados obtidos com os testes via internet variaram de acordo com o consumo da banda disponibilizada no enlace (conexão). Para solucionar este problema foi criada uma regra no *PBX VoIP*, para limitar a quantidade de chamadas simultâneas, com destino na rede externa, originada de um único ramal ou de vários simultaneamente. Em complemento, foi criada uma regra para tratar esse tipo de evento no controle de QoS da rede (*LAN*).

No processo de desenvolvimento desta ferramenta, um dos principais problemas encontrados foi a documentação para integração do *PBX Asterisk*, com os *PABX* das principais empresas do segmento (*Intelbras, Panasonic, Leucotron, Siemens, Digital*). As *API's* disponibilizadas pelo fabricante, muitas vezes não possibilitam acesso completo a todos os recursos dos equipamentos. Em contrapartida a essa dificuldade inicial, a documentação técnica do *Asterisk* é disponibilizada de forma gratuita e distribuída, facilitando a aquisição do conhecimento necessário para o desenvolvimento. A distribuição da documentação do *Asterisk* segue o modelo da documentação de arquivos da plataforma *UNIX*.

Na perspectiva de hardware, um dos problemas observados, principalmente nas empresas de pequeno porte, foi a utilização de equipamentos de baixíssima qualidade que dificultam enormemente a garantia de uma qualidade mínima possível em cenário *VoIP*. Isso se deve, principalmente, aos PC's utilizados como estações de trabalho, que na maioria das vezes são montados com peças de baixa qualidade que refletem diretamente com o resultado final, além de dificultar a integração das chamadas *VoIP* com a RTPC. É fortemente recomendado a utilização de computadores compatíveis com o processamento necessário para viabilizar a utilização da plataforma *Asterisk*.

TRABALHOS FUTUROS

Como sugestões para trabalhos futuros e complementação deste projeto, propõe-se o desenvolvimento de um *CRM* acoplado a ferramenta de forma nativa utilizando o modulo *cdr_mysql*, facilitando dessa forma a atuação do *CTI*.

Implementação de um sistema para monitoramento da ferramenta, baseado em uma metodologia de análise de problemas com foco especial em inconsistências internas do sistema (erros de programação) assim como, na rede *VoIP*. Desenvolvimento de novas *API's* (*Application Programming Interface*) para possibilitar a compatibilidade com o hardware de outros fabricantes de *PABX* e facilitar a integração com os softwares de relacionamento com o cliente (*CRM - Customer Relationship Management*).

Desenvolvimento de um módulo específico para configuração e otimização das funcionalidades da URA, haja visto que esse recurso sofre constantes alterações, tendo como motivação a garantia da diminuição de falhas no processo de tratamento das ligações dos clientes pelos operadores. Certos tipos de ligações, por exemplo críticas, poderiam seguir para um fluxo de gravação da URA, essa ação evitaria o contato do cliente com o operador e diminuiria de forma relevante o tempo médio de atendimento (TMA).

REFERÊNCIAS

ADRIA, M. e Chowdhury, S. Centralization as a design consideration for the management of call centers. **Information & Management**, New York, v.41, n.33, p.497-507, out. 2004.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÃO. **Tarifação Fixa e Móvel**. Brasília, 2016. Disponível em: <<http://www.anatel.gov.br>>. Acesso em: 30 abr. 2016.

CISCO. **Academia Cisco**. Venezuela:[s.n], 2016. Disponível em: <<http://cisco.netacad.net>>. Acesso em: 25 abr. 2016.

COMER, DOUGLAS E. **Redes de computadores e internet**. Porto Alegre: Bookman, 2001.

CARDOSO, J. **Unified customer interactionTM**: gestão do relacionamento num ambiente misto de interação self e assistida. Lisboa:[s.n], 2000.

DELL. **Servidores Dell**. Disponível em: <<http://www.dell.com.br>>. Acesso em: 30 abr. 2016.

DIGIUM. **Interface E1**. Disponível em: <<http://store.digium.com>>. Acesso em: 30 abr. 2016.

GABALLA, A. e Pearce, W. Telephone sales manpower planning at Qantas. **Interfaces**, [S.l.], v. 9, n.3, p.1-9, jun. 1979.

HAWKINS, L., Meier, T., Nainis, W. S. e James, H. M. **The evolution of the call center to customer contact center**. [S.l.]: ITSC, 2001.

HERSENT, Oliver; GUIDE, David; PETIT, Jean-Pierre. **Telefonia IP**: comunicação multimídia baseada em pacotes. São Paulo: Makron Books, 2002.

INTERNET ENGINEERING TASK FORCE RFC. **RTP**: a transport protocol for real-time applications. Califórnia:[s.n], 1889. Disponível em: <<http://www.ietf.org/rfc/rfc1889.txt>>. Acesso em: 7 abr. 2016.

INTERNET ENGINEERING TASK FORCE RFC 3261. **SIP**: session initiation protocol. Califórnia:[s.n], 1889. Disponível em: <<http://www.ietf.org/rfc/rfc3261.txt>>. Acesso em: 10 abr. 2016.

INTERNET ENGINEERING TASK FORCE RFC 1122. **Requirements for internet hosts -- communication layers**. Califórnia:[s.n], 1889. Disponível em: <<http://tools.ietf.org/html/rfc1122>>. Acesso em: 23 maio 2016.

INTERNET ENGINEERING TASK FORCE RFC 5456. **IAX**: inter-asterisk eXchange version 2. Califórnia:[s.n], 1889.

.Disponível em:<<http://tools.ietf.org/html/rfc5456>>. Acesso em: 6 abr. 2016.

INTERNET ENGINEERING TASK FORCE RFC 791. **IP**: internet protocol.

Califórnia:[s.n], 1889. Disponível em: <<http://tools.ietf.org/html/rfc791>>. Acesso em: 22 maio 2016.

KUROSE, James F. e ROSS, Keith W. **Computer Networking: a top-down approach**. 5. ed. Boston: Addison-Wesley, 2010.

KOOLE, G. e Mandelbaum, A. Queueing models of call centers: an introduction. **Annals of Operations Research**, Boston, v.113, n.44, p.41-59, jun. 2002.

MEGGELEN, Jim V., SMITH, Jared e MADSEN, Leif. **Asterisk: the future of telephony**. [S.l]: O'Reilly Media, 2005.

MORAES, George Salvador. **Implementação de um PBX VoIP utilizando o PBX Asterisk**. Espirito Santo: [s.n], 2010.

MUNRO, J. **The Heroes of Telegraph**. New York:[s.n], 2002.

ODOM, W. **Implementing Cisco Quality of Service (QoS) v2.0**. New York: Cisco Press, 2004.

PICHITLAMKEN, J., Deslauriers, A., L'Ecuyer, P. e Avramidis, A. N.,. **Modelling and simulation of a telephone call center**: proceedings of the 2003 winter simulation conference. New Orleans:[s.n], 2003

ROCHA, Talita Ferreira; FONTANA, Eduardo Veiga; BARRÉRE, Eduardo. **VoIP como solução para Call Center descentralizado**. Rio de Janeiro: [s.n], 2012.

SILVA, Genevides Laureno da. **Gateway de voz para integração IP-PSTN aderente à arquitetura das redes de nova geração**. 2009. 93f. Dissertação (Mestrado em Informática Aplicada) – Centro de Ciências e Tecnologia, Universidade de Fortaleza, Fortaleza, 2009.

STAHELIN, Carlos Alberto. **Monitoramento, análise e diagnóstico de problemas de sistema VOIP em um Call Center**. Santa Catarina:[s.n], 2014.

TANENBAUM, Andrew S.; WETHERALL, David J. **Redes de Computadores**. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

TORRES, G. **Redes de Computadores**. Cruzeiro: Axcel Books, 2001.