

UNIVERSIDADE DE ÉVORA

Escola de Ciências e Tecnologia

Departamento de Informática

Mestrado em Engenharia Informática

Gateway SIP - Asterisk

Carlos Humberto Martins Chagas Guerra

Orientadora: *Professora Doutora Irene Rodrigues*

Évora, Novembro 2012

Tese submetida à Universidade de Évora para obtenção do grau de
Mestre em Eng. Informática.

UNIVERSIDADE DE ÉVORA
Escola de Ciências e Tecnologia
Departamento de Informática
Mestrado em Engenharia Informática

Gateway SIP - Asterisk

Carlos Humberto Martins Chagas Guerra

Orientadora: *Professora Doutora Irene Rodrigues*

Évora, Novembro 2012

Tese submetida à Universidade de Évora para obtenção do grau de
Mestre em Eng. Informática.

Agradecimentos

Existem pessoas que seria injusto da minha parte não mostrar o meu profundo agradecimento pela ajuda, força, apoio, dedicação e incentivo que sempre me deram e serviram de inspiração durante a realização desta tese.

Começo por agradecer à Professora Doutora Irene Rodrigues, primeiro por ter aceite ser minha orientadora, posteriormente pela disponibilidade, atenção, orientação e, acima de tudo a motivação que sempre transmitiu desde o primeiro momento até à conclusão deste trabalho.

Agradeço aos meus pais e irmão, com especial destaque à maravilhosa mãe que tenho que tudo fez para me ajudar nesta minha “aventura”.

Um eternamente grato por tudo o que me aturaram, pela força que sempre me deram quando parecia que a luz estava bem lá no fundo do túnel e na forma em particular que cada uma me ajudou. . . Obrigado Maria José Soares, Obrigado Betina São Pedro, obrigado Inês Alvarez, obrigado Cláudia Machado.

Para acabar, não posso deixar de agradecer ao Sr. Rui Pratas, responsável pelo sector informático da empresa Clidis, pela disponibilidade e interesse que demonstrou desde o primeiro momento, relativamente à minha participação no processo de implementação do sistema VoIP na empresa.

Gateway SIP - Asterisk

Resumo

Esta tese consiste num estudo realizado sobre a tecnologia VoIP (Voice over IP) com aplicação dentro da área dos PBX (Private Branch eXchange) e Gateways VoIP Open Source.

Em primeiro lugar foram abordados os conceitos, requisitos e elementos associados a esta tecnologia bem como a sua interligação com outras tecnologias de comunicação como a PSTN (Public Switched Telephone Network), ISDN (Integrated Services Digital Network) ou GSM (Group Special Mobile). Além disso, foi efectuado um estudo teórico e prático sobre o software Open Source Asterisk, tendo com objectivo explorar o seu modo de funcionamento e funcionalidades disponíveis, passíveis de serem utilizadas em ambiente empresarial.

Por fim, foi desenvolvida uma solução assente neste tipo de tecnologia na empresa Clidis - Laboratório de Análises Clínicas de Sines, onde ficou provado que a implementação de PBX/Gateways VoIP através de software Open Source é uma alternativa viável às reais necessidades de comunicação das empresas.

Palavras-chave: Voice over IP (Voip), PBX VoIP, Gateways VoIP, Asterisk.

Gateway SIP - Asterisk

Abstract

This thesis is a study on VoIP (Voice over IP) application within the area of PBX (Private Branch eXchange) and VoIP Open Source Gateways.

Firstly were addressed the concepts, requirements and elements associated with this technology and its interconnection with other communications technologies such as PSTN (Public Switched Telephone Network), ISDN (Integrated Services Digital Network) or GSM (Groupe Special Mobile). Furthermore, a study was carried out on the theoretical and practical Asterisk Open Source software, with the aim to explore its operation and features available, liable to be used in business environment.

Finally, we developed in the company Clidis (Laboratory of Clinical Analyses in Sines) a solution based on this technology, where it was proved that the implementation of PBX / VoIP Gateways through Open Source software is a viable alternative to the real needs of business communication.

Keywords: Asterisk, Voice over IP (Voip), IP PBX, SIP.

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.2	Objectivo do Estudo	2
1.3	Metodologia	2
1.4	Estrutura do trabalho	2
2	Estado da Arte	5
2.1	Voice Over Internet Protocol (VoIP)	5
2.1.1	Principais vantagens na utilização do VoIP	5
2.1.2	Contextos de aplicação do VoIP	6
2.1.3	Problemas na transmissão de voz sobre uma rede IP	8
2.2	Codecs	9
2.3	Protocolos	11
2.3.1	Transmissão de dados utilizando TCP/IP	11
2.3.2	Protocolos utilizados no VoIP	12
2.4	IP PBX ou PBX VoIP	14
2.5	Gateways VoIP	15
2.5.1	Interligação Rede IP/Rede telefónica analógica (PSTN)	15
2.5.2	Interligação Rede IP/Rede telefónica Digital (ISDN)	15
2.5.3	Interligação Rede IP/Redes Móveis (GSM)	16
2.6	Soluções PBX/Gateways VoIP existentes no mercado	17
3	Estudo teórico sobre o Asterisk	19
3.1	Asterisk, o que é?	19
3.2	Requisitos do Sistema	19
3.3	Funcionalidades do Asterisk	20
3.4	Arquitectura do Asterisk	21
3.5	Directórios e ficheiros de configuração	23
3.5.1	Secções nos ficheiros de configuração	24
3.5.2	Contextos nos ficheiros do plano de chamadas	24
3.6	Registo de utilizadores no Asterisk	25

3.7	Componentes base do plano de chamadas	26
3.7.1	Contextos	26
3.7.2	Extensões	26
3.7.3	Prioridades	28
3.7.4	Aplicações	28
3.8	Outros elementos do plano de chamadas	28
3.8.1	Variáveis	28
3.8.2	Expressões	29
3.8.3	Funções	29
3.8.4	Condições no plano de chamadas	29
3.8.5	Macros	30
3.8.6	A base de dados do Asterisk (AstBD)	30
4	Estudo prático sobre o Asterisk	31
4.1	Ambiente de testes	31
4.1.1	Como trabalhar com o Asterisk	32
4.2	Comunicação entre Utilizadores	33
4.3	Criação de utilizadores	33
4.3.1	Utilizadores SIP	34
4.3.2	Utilizadores IAX	34
4.4	Plano de chamadas básico	35
4.4.1	Chamadas entre utilizadores	35
4.4.2	Personalizar o atendimento	37
4.4.3	Alterar o idioma de atendimento	37
4.5	Gestão de chamadas recebidas	38
4.5.1	Transferência de chamadas entre extensões	38
4.5.2	Captura de chamadas	39
4.5.3	Colocação de chamadas em espera	40
4.5.4	Música em espera	41
4.6	Voice Mail	42
4.7	Não perturbe	44
4.8	Atendimento automático de chamadas	45
4.8.1	Configuração do IVR	45
4.8.2	Atendimento automático personalizado ao tempo	45
4.9	Salas de conferência	45
4.10	Gravação de detalhes das chamadas	46
4.11	Distribuição automática de chamadas	47
4.12	Trunking entre servidores	47
5	Case Study na empresa Clidis - Análise de requisitos	49
5.1	Apresentação da empresa Clidis	49
5.2	Infraestrutura de rede/comunicações existente na empresa	49
5.2.1	Interligação informática entre a sede e postos de colheita	50
5.2.2	Tipos de acessos/comunicações contratados	50
5.3	Requisitos de comunicações da empresa Clídis	50
5.4	Requisitos impostos pela empresa para a implementação da solução de comunicações	51

6	Case Study - Implementação da solução	53
6.1	Soluções de comunicações consideradas	53
6.2	Solução adoptada - Asterisk / FreePBX	54
6.3	Softwares de base instalados no servidor PBX	55
6.4	Hardware utilizado no servidor PBX	55
6.5	Infraestrutura de comunicações e terminais VoIP utilizados	56
6.6	Sistema de atendimento criado	56
6.7	Plano unificado de comunicações	57
7	Conclusão e trabalhos futuros	59
A	Ficheiros de configuração do Asterisk	61
B	Configuração de utilizadores SIP	63
C	Configuração de utilizadores IAX	65
D	Configuração de um utilizador num Softphone	67
D.1	Configuração no X-Lite	67
D.2	Configuração no 3CX	68
D.3	Configuração no Zoiper Communicator	68
E	Aplicações do Asterisk	71
F	Variáveis de canal do Asterisk	73
G	Configuração do CDR	75

Lista de Figuras

2.1	Ligação VoIP entre computadores [2]	6
2.2	Telefone IP da marca Linksys	7
2.3	Ligação VoIP entre telefones IP [2]	7
2.4	Ligação VoIP entre computador e telefone comum [2]	7
2.5	Ligação VoIP entre telefones analógicos [2]	8
2.6	Ligações FXO e FXS[21]	15
2.7	Gateway FXO [21]	16
2.8	Gateway FXS [21]	16
2.9	Adaptador ATA [21]	16
3.1	Requisitos de Hardware [23]	20
3.2	Interligação entre Núcleo e Módulos do Asterisk [25]	23
D.1	Aceder ao menu de configuração do X-Lite 4	67
D.2	Aceder ao menu de configuração do 3CX	68
D.3	Configuração da conta do utilizador no 3CX	68
D.4	Aceder ao menu de configuração do Zoiper	69
D.5	Configuração da conta do utilizador no Zoiper	69

Lista de Acrónimos

ADPCM Adaptative Differential PCM

ATA Analog Telephone Adapter

BRI Basic Rate Interface

CELP Code-Excited Linear Prediction

CS-ACELP Conjugate-Structure Algebraic-Code-Excited Linear Prediction

FXO Foreign eXchange Office

FXS Foreign eXchange Subscriber

GSM Groupe Special Mobile

IETF Internet Engineering Task Force

IAX/IAX2 Inter-Asterisk Exchange Protocol

IP Internet Protocol

ISDN Integrated Services Digital Network

ISP Internet Service Provider

ITU Internacional Telecommunication Union

iLBC Internet Low Bitrate Codec

IVR Interactive Voice Response

MG Media Gateway

MGC Media Gateway Controller

MGCP Media Gateway Controller Protocol

NAT Network Address Translator

PABX Private Automatic Branch eXchange

PBX Private Branch eXchange

PCM Pulse Code Modulation

PoE Power Over Internet

PRI Primary Rate Interface

PSTN Public Switched Telephone Network

QoS Quality of Service

RFC Request for Comments

RSVP Resource Reservation Protocol

RTCP Real Time Control Protocol

RTSP Real-time Streaming Protocol

RTP Real Time Protocol

SDP Session Description Protocol

SIP Session Initiation Protocol

STCP Stream Control Transmission Protocol

TCP Transmission Control Protocol

UDP User Datagram Protocol

VoIP Voice over IP

VPN Virtual Private Network

1. Introdução

A constante evolução na área das telecomunicações, com especial destaque para o aumento das velocidades de acesso à Internet disponibilizado por parte dos ISP (Internet Service Providers), aliado à cada vez maior fiabilidade e qualidade dos serviços prestados (QoS - Quality of Service) a custos cada vez mais reduzidos, além de ter permitido um incremento na adesão por parte de novos utilizadores e empresas à Internet, possibilitou igualmente o aparecimento e utilização de novas tecnologias e serviços. O VoIP (Voice over IP) é um dos exemplos dessas tecnologias, permitindo que seja possível a realização de chamadas telefónicas e outros serviços associados através da utilização de uma rede de dados, como a Internet.

A evolução destas tecnologias está a revolucionar a forma como se processam as comunicações hoje em dia, principalmente no que diz respeito à redução de custos para a realização das mesmas. São cada vez mais as empresas que disponibilizam soluções/aplicações baseadas nesta tecnologia a custos reduzidos ou inexistentes, se não se contabilizar o custo de acesso à Internet. Empresas como Skype, VoipBuster, VoipCheap, Jajah, Fring entre outras, disponibilizam todo um conjunto de serviços de comunicações a custo zero entre os seus utilizadores, bem como permitem a realização de chamadas para a rede fixa ou móvel de grande parte dos países do mundo igualmente a custo zero ou através de uma taxação reduzida.

A possibilidade de interligar através de *gateways*, as tecnologias VoIP com outras tecnologias de comunicação como as redes PSTN (Public Switched Telephone Network) ou GSM (Groupe Special Mobile), permitiu criar todo um conjunto de novas necessidades de comunicação dentro das organizações.

A resposta a essas necessidades é dada através de soluções de PBX VoIP, existindo no mercado diversas opções consoante as funcionalidades desejadas. Essas soluções podem ser implementadas através de soluções proprietárias, quer por hardware, quer por software, ou então através de soluções por software Open Source.

Dentro das soluções Open Source existentes, são cada vez mais os projectos que surgem no mercado com implementações “amigáveis” do tipo “tudo-em-um”. Através destas implementações e de uma forma relativamente acessível, é possível a instalação de raiz de todo um sistema, desde a instalação do S.O. (linux) até à instalação dos ambientes gráficos a utilizar para a configuração do sistema telefónico desejado. Neste tipo de softwares, o utilizador não tem necessidade de ter conhecimentos sobre como funciona o “core” do sistema. Assim, fica apenas limitado à implementação das funcionalidades previstas nos ambientes gráficos disponíveis.

Projectos como o Elastix e o FreePBX são exemplos de referência para este tipo de aplicações. Contudo, apesar de serem projectos distintos, o “core” base de ambos (e de outros projectos similares) é o software Asterisk. Este software foi desenvolvido pela *Digium* e é uma referência mundial dentro dos PBX VoIP Open Source.

1.1 Motivação

As soluções em comunicações VoIP estão a assumir um papel de relevância na vida das instituições, sendo cada vez mais as organizações a adoptar este tipo de tecnologias, não só pelas diversas funcionalidades passíveis de implementar fazendo uso das mesmas, mas também tendo em vista a redução de custos. Com base nisso, a motivação para a realização deste trabalho reside principalmente em duas vertentes. A primeira, estudar este tipo de tecnologias assentes em softwares Open Source disponíveis no mercado. A segunda porque, sendo esta uma área de negócio em expansão e com espaço de manobra dentro do mercado empresarial/institucional, explorar este tipo de tecnologias representa a possibilidade futura de desenvolver soluções por medida capazes de dar resposta às reais necessidades de comunicação das organizações e ser uma alternativa a sistemas proprietários com elevados custos de implementação.

1.2 Objectivo do Estudo

O objectivo deste trabalho, numa primeira fase, é fazer um estudo teórico sobre as tecnologias VoIP. De seguida e com base nisso, efectuar uma análise, primeiro teórica e posteriormente prática, sobre o software Open Source Asterisk, tendo em vista a compreensão do seu modo de funcionamento e exploração das suas funcionalidades. O objectivo final é comprovar a validade deste tipo de solução dentro do mercado empresarial e verificar se realmente representa uma alternativa viável aos outros tipos de soluções proprietárias existentes. A concretização deste objectivo será desenvolvido na empresa Clidis - Clínica de Diagnósticos de Sines, em Sines.

1.3 Metodologia

Tendo em conta os objectivos traçados, este trabalho assumiu as seguintes fases:

- Em primeiro lugar, foi efectuado um estudo teórico sobre o VoIP, nomeadamente sobre protocolos e codecs utilizados dentro deste tipo de tecnologia. Foi igualmente abordada a temática dos PBX Voip e Gateways Voip;
- Na fase seguinte, tendo em vista a exploração do software Asterisk, foi realizado um estudo teórico e prático, em ambiente de rede local, sobre o modo de funcionamento e implementação de funcionalidades utilizando este tipo de software;
- Por fim, foi desenvolvida uma solução baseada neste tipo de software, de forma a testar se esta é realmente uma solução que dá resposta às reais necessidades das empresas e se pode ser considerada uma alternativa viável, quer a nível de desempenho, quer a nível de custos de implementação, a outras soluções existentes no mercado.

1.4 Estrutura do trabalho

Este trabalho está estruturado da seguinte maneira:

- O capítulo 1 apresenta uma breve introdução, identifica a motivação e clarifica os objectivos do presente trabalho.
- O capítulo 2 apresenta os conceitos base sobre VoIP, com a definição dos protocolos de sinalização e de transporte utilizados no VoIP e a apresentação das vantagens e desvantagens da utilização deste tipo de tecnologia. Além disso, são igualmente apresentadas as funcionalidades que se esperam da utilização de um PBX VoIP e de um Gateway VoIP.
- No capítulo 3 apresenta uma análise teórica sobre a estrutura e funcionamento do software open Source Asterisk.
- No capítulo 4 apresenta todo o conjunto de testes práticos realizados, em ambiente de rede local, com o software Open Source Asterisk por forma a testar as suas funcionalidades de PBX VoIP.
- O Capítulo 5 trata da apresentação a análise de requisitos realizada dentro do âmbito do Case Study realizado na empresa Clídis.
- O Capítulo 6 apresenta as soluções adoptadas e implementadas na realização do Case Study.
- O Capítulo 7 apresenta as conclusões, assim como algumas sugestões de trabalhos futuros, para a continuidade desta pesquisa.

2. Estado da Arte

Este capítulo faz a apresentação do estudo teórico realizado sobre a tecnologia VoIP. São também abordadas as vantagens deste tipo de tecnologia dentro do âmbito das comunicações empresariais, e, além disso, a forma de interligar este tipo de tecnologia com as outras principais tecnologias de comunicação.

2.1 Voice Over Internet Protocol (VoIP)

VoIP ou voz sobre o protocolo Internet, é a tecnologia que permite estabelecer comunicações de voz ou vídeo através de uma rede de dados (incluindo a Internet), codificando/convertendo sinais analógicos num conjunto de sinais digitais sob a forma de pacotes de dados com endereçamento IP, para que estes sejam transportados pela rede em tempo real até ao seu destino e aí serem novamente descodificados/convertidos. A codificação/descodificação dos sinais e respectivo transporte dos pacotes fica a cargo de codecs e protocolos que suportam a tecnologia VoIP.

2.1.1 Principais vantagens na utilização do VoIP

Existem vantagens na utilização desta tecnologia sendo de destacar:[1]:

Redução de custos: Associado ao modo diferenciado como são taxados os acessos à rede Internet (com tráfego ilimitado) face às chamadas telefónicas realizadas através dos operadores telefónicos, com especial incidência sobre o volume e tipo de chamadas (locais ou internacionais) realizadas, assim será o tempo de retorno a nível de investimento feito numa implementação VoIP. Contudo, e apesar do maior ou menor tempo que possa existir nesse retorno, as comunicações VoIP representam uma real redução de custo com as comunicações;

Infraestrutura única: Trabalhando a tecnologia VoIP sobre uma rede de dados deixa de ser necessário existir nas empresas infraestruturas diferentes quer para os dados, quer para a voz, passando toda a informação a circular sobre uma mesma infraestrutura;

Mobilidade: No VoIP, ao contrário do sistema de telefone tradicional, não existe uma posição fixa do dispositivo utilizado para realizar chamadas (telefone ou softphone), basta apenas ter acesso à rede de dados utilizada para a realização das chamadas (rede local ou até mesmo através da Internet);

Novas funcionalidades: O VoIP, associado a outras tecnologias informáticas (ex. bases de dados, servidores de e-mail, etc.), possibilitou o desenvolvimento de um conjunto de novas funcionalidades não disponíveis em sistemas tradicionais. Exemplos dessas funcionalidades são o envio de correio de voz para o e-mail, atendimento automático interativo através de acesso a bases de dados, entre outras. Estas novas funcionalidades foram desenvolvidas principalmente para dar resposta a necessidades empresariais de atendimento telefónico.

2.1.2 Contextos de aplicação do VoIP

Existem diversos contextos ou formas de aplicação da tecnologia VoIP. Contudo, em todos eles é necessário que exista uma rede do tipo IP. De seguida, são apresentados alguns desses contextos de utilização.

Comunicação entre computadores

Para que exista comunicação VoIP entre computadores, quer em rede local, quer através da utilização da Internet, é necessário que se cumpram alguns requisitos:

- Os computadores estarem dotados de *kits multimédia* (microfone e colunas de som ou headsets);
- Os computadores terem instalado um *terminal VoIP do tipo Softphone* (software que permite realizar ligações VoIP através de uma rede IP). Existem no mercado diversos softphones free como é o caso do Skype, Google Talk, VoIPBuster, entre outros;
- Tem de existir uma ligação entre os computadores (rede local ou Internet) com uma largura de banda razoável.

Esta é a forma mais simples de se estabelecer uma comunicação VoIP, pois cada um dos computadores é visto como se de um telefone se tratasse. A figura 2.1 mostra este tipo de ligação com recurso à rede Internet.

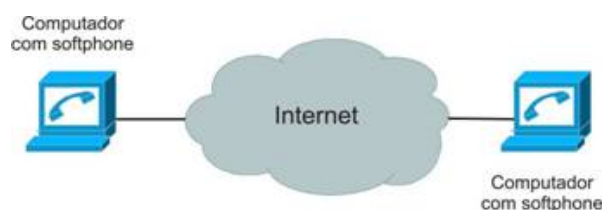


Figura 2.1: Ligação VoIP entre computadores [2]

Comunicação entre telefones IP

Os telefones IP são **terminais VoIP** desenvolvidos para serem utilizados em comunicações VoIP sobre redes IP. Existe no mercado uma grande variedade deste tipo de equipamentos, variando o seu preço de aquisição consoante as funcionalidades disponíveis em cada um desses equipamentos. Quando da aquisição de um telefone IP deve ser tido em conta os protocolos por ele suportados,

pois existem no mercado telefones com baixo custo de aquisição, mas que só funcionam com protocolos proprietários.



Figura 2.2: Telefone IP da marca Linksys

Caso um telefone IP não esteja inserido dentro de nenhuma organização que faça recurso a um PBX VoIP, então para que se possa realizar e receber chamadas poderá ser necessário contratar junto de um ISP um acesso ao serviço VoIP. Nestes casos, a gestão de todos os processos necessários à realização de chamadas fica a cargo do ISP, bastando ligar o telefone à rede Internet para que este se registre perante o ISP e a partir daí ficar habilitado para fazer e receber chamadas.

Este tipo de telefones são, regra geral, utilizados em ambientes corporativos, ligados à rede da empresa onde existe em funcionamento um PBX VoIP. A figura 2.3 mostra este tipo de ligação.



Figura 2.3: Ligação VoIP entre telefones IP [2]

Comunicação entre computador e telefone analógico

A utilização de telefones analógicos dentro deste contexto assume que este telefone está diretamente ligado à rede PSTN. Neste caso, para que seja possível a realização de chamadas é necessária a utilização de Gateways. Os utilizadores que participam neste tipo de contexto de comunicação nem se apercebem da existência dessa Gateway. Contudo, ela é necessária para que seja possível interligar a tecnologia PSTN com a rede IP. A figura 2.4 mostra este tipo de ligação.

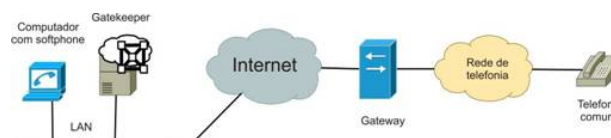


Figura 2.4: Ligação VoIP entre computador e telefone comum [2]

Comunicação entre telefones analógicos

Através da utilização de adaptadores ATA (Analog Telephone Adapter), os telefones analógicos conseguem realizar chamadas VoIP. Este tipo de adaptador é um **terminal VoIP** que funciona como uma Gateway, convertendo os sinais analógicos do telefone para dados digitais da VoIP e vice-versa. Através da utilização deste tipo de equipamento os telefones analógicos passam a ser reconhecidos na rede como telefones IP. A figura 2.5 mostra este tipo de ligação.

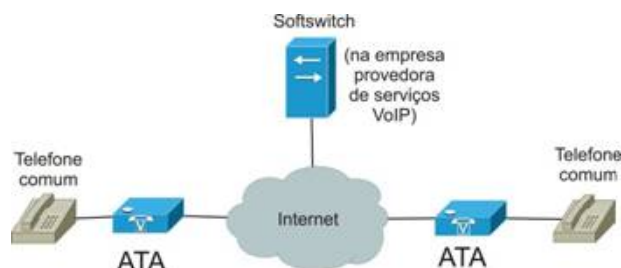


Figura 2.5: Ligação VoIP entre telefones analógicos [2]

2.1.3 Problemas na transmissão de voz sobre uma rede IP

As redes de dados IP não foram inicialmente projectadas para a transmissão de dados de aplicações com requisitos de funcionamento em tempo real, como o VoIP. Devido aos requisitos necessários neste tipo de tecnologia, existem algumas questões a ter em consideração quando se pretende realizar transmissão de voz sobre este tipo de rede, sendo de destacar:

Perda de Pacotes: Um dos principais problemas quando se trabalha com a transmissão de voz através da utilização de redes de dados é a perda de pacotes, ou que os mesmos cheguem com um atraso tal que já não seja possível a sua utilização. Se num outro tipo de aplicação esse problema fica resolvido através da retransmissão dos pacotes perdidos, nas aplicações de voz em tempo real tal solução não é possível. A perda de pacotes numa rede de dados está fortemente relacionada com o congestionamento que possa existir na mesma, onde a consequência desta situação, caso exista a perda de um grande número de pacotes, será o "picotamento da voz". Alguns codecs utilizam estratégias de compensação para reduzir esse efeito de perda, como a repetição do último pacote recebido. Contudo, uma solução para este tipo de situação é garantir a QoS (Quality Of Service) para pacotes de voz, ou seja, garantir que este tipo de pacotes tenha prioridade de envio face a outros tipos de pacotes de dados nos dispositivos transmissores existentes na rede.

Atrasos na rede: É a quantidade de tempo que a voz demora desde que sai da boca de quem fala até chegar ao ouvido de quem ouve. Sendo a voz transportada na rede através de pacotes, existe todo um conjunto de atrasos desde a codificação inicial da voz em pacotes de dados, processamento de envio desses pacotes, roteamento dos mesmos até atingir o seu destino e, por fim, todo o processamento de descodificação no destinatário para que os pacotes sejam novamente convertidos em voz. Segundo a recomendação G.114 da ITU (Internacional Telecommunication Union) [3], algumas aplicações com recurso a tempo

real podem ser afectadas por atrasos de 100 ms. Contudo, a maioria das aplicações não é seriamente afectada se esse atraso for mantido abaixo de 150 ms.

Jitter: Quando numa rede de dados são enviados pacotes de voz em intervalos regulares, pode existir atraso na rede e esses mesmos pacotes não chegam ao destino com o intervalo de tempo com que foram enviados. A diferença entre o tempo real de chegada de um pacote e o tempo para o qual essa chegada estava prevista dá-se o nome de *jitter*. Este problema pode ser resolvido através da utilização de *buffer jitter* nos dispositivos de recepção final.

2.2 Codecs

Para que os sinais analógicos sejam transportados em pacotes numa rede IP, é necessário que sejam codificados no envio e decodificados aquando da sua recepção no destino. Essa codificação/decodificação é realizada através da utilização de codecs, que mais não são do que algoritmos de compactação e descompactação específicos, consoante o tipo de formato de ficheiro desejado.

Existem diversos codecs que podem ser utilizados para a codificação da voz/video. Contudo, a escolha do codec está intimamente associado às suas principais características, sendo elas:

- **Taxa de bits (Codec Bit Rate) (Kbps):** Quantidade de informação que precisa ser transmitida, em bits por segundo, para ser entregue um pacote de media (audio ou video);
- **Intervalo da amostra (Codec Sample Interval) (ms):** Quantidade de tempo utilizado pelo codec, em milisegundos, para realizar a amostra para a codificação do sinal analógico;
- **Tamanho da amostra (Codec Sample Size)(bytes):** Número de bytes capturados em cada intervalo de amostra.
- **Tamanho do *payload* de voz (Voice Payload Size) (bytes/ms):** Quantidade de bytes transportados em cada pacote de dados. Este valor influencia directamente o tamanho da largura de banda a ser utilizada e o *delay* ou atraso da conversa. Quanto maior for este valor, maior será o tamanho do pacote e maior será o tempo para ele chegar ao seu destino e ser decodificado. A maioria dos codecs tem o seu *payload* a variar entre 10 e 40 ms.

Além das características dos codecs, factores como o *delay* das comunicações, o *jitter* ou a perda de pacotes têm grande importância para a qualidade final obtida.

Uma das escalas mais utilizadas pela ITU para avaliação de codecs é a escala Mean Opinion Score (MOS). Esta é uma técnica subjectiva[4], onde um determinado número de pessoas é exposta ao resultado produzido pelo codec em avaliação, e com base na sua opinião, avaliam esse resultado utilizando uma escala que varia entre 1 (ruim) e 5 (excelente). A tabela 2.1 apresenta a escala de avaliação MOS para audio [5].

Codecs utilizados no VoIP

Abaixo são apresentadas as características dos principais codecs em utilização para a codificação de voz [6].

- **G.711:** É o principal codec utilizado pela rede PSTN. Faz recurso do algoritmo padrão de codificação PCM (Pulse Code Modulation). A sua implementação depende da localização,

Tabela 2.1: Escala de valores MOS

Pontuação MOS	Entendimento da Voz	Distorção
5	Excelente	Imperceptível
4	Boa	Apenas perceptível, sem incomodar
3	Regular	Perceptível, leve perturbação
2	Pobre	Perturbando, mas audível
1	Ruim	Perturbando muito, inaudível

ou seja, este é implementado através do G.711 μ para a América do Norte e Japão e o G.711a para o resto do mundo. Utiliza fracos recursos de processamento e o atraso induzido pelo algoritmo é algo insignificante. Faz uso de uma taxa de compressão de 64 Kbps e tem uma qualidade de voz excelente. Este codec não requer licenciamento de utilização.

- **G.726:** Utiliza a codificação ADPCM (Adaptative Differential PCM) e pode funcionar com diversas taxas de compressão (16/24/32/40 Kbps). Oferece uma qualidade de codificação praticamente idêntica ao G.711, mas faz recurso apenas de metade da sua largura de banda. Apesar de utilizar mais recursos de processamento face ao G.711 também tem um atraso induzido pelo algoritmo insignificante. Consoante a taxa de compressão utilizada assim será a qualidade de voz obtida (boa a 40 Kbps e moderada a 24 Kbps). Este codec não requer licenciamento de utilização.
- **G.729a:** Utiliza a codificação CS-ACELP (Conjugate-Structure Algebraic-Code-Excited Linear Prediction) e tendo em conta a baixa largura de banda que utiliza, este codec consegue ter uma boa qualidade de voz. Tem uma grande taxa de compressão contudo origina uma grande carga computacional de processamento. Este codec é patenteado e requer licenciamento pago de utilização.
- **GSM:** Este codec (Global System for Mobile Communications) é a codificação utilizada para os telefones móveis e tem uma qualidade de som razoável não exigindo grandes capacidades de processamento. Tem uma taxa de compressão relativamente baixo (13 Kbps). Este codec não requer licenciamento de utilização.
- **iLBC:** Este codec (Internet Low Bitrate Codec) opera a 13.3 Kbps ou 15.2 Kbps com intervalos de amostra de, respectivamente, 30 e 20 ms e é um codec muito robusto no que diz respeito à perda de pacotes e recuperação de erro. Apesar de ser um codec gratuito, a sua utilização obriga a um licenciamento de utilização.
- **Speex:** Este é um codec open-source projectado para voz muito adoptado em aplicações a funcionarem na Internet. Suporta uma taxa de compressão variável entre os 2 e os 44 kbps em amostras de 20ms, podendo a cada amostra realizada alterar a taxa de compressão consoante a disponibilidade da largura de banda disponível. Baseia-se no algoritmo de codificação CELP (Code-Excited Linear Prediction)[7].

De seguida são apresentados os codecs para a codificação de video.

- **H.261:** Este codec foi definido para sistemas de videoconferência a funcionar sobre a rede ISDN (canais de 64kbps). Suporta dois formatos de imagem (CIF e QCIF) e utiliza duas formas de codificação dos blocos de dados.

- **H.263:** Este codec foi baseado no codec H.261 e foi projectado também para videoconferências, mas através de redes com mais taxas de transmissão. Conseguir atingir uma grande taxa de compressão e suporta cinco tipos de imagem (CIF, QCIF, SQCIF, 4CIF, 16CIF); Este codec pode substituir o codec H.261 em qualquer largura de banda e serviu de base para o codec H.264;
- **H.264:** Este codec também designado por H.264/AVC ou MPEG-4 part 10, utiliza a estrutura base dos padrões anteriores (H.261, MPEG-1, MPEG-2 H.262, H.263 ou MPEG-4 part 2) mas com uma maior eficiência da codificação, reduzindo o número de bits no vídeo codificado. Além disso, apresenta uma maior robustez a falhas na transmissão.

2.3 Protocolos

O VoIP é uma tecnologia que está assente sobre uma rede de dados de comutação de pacotes onde o protocolo de rede utilizado é o protocolo IP (Internet Protocol).

2.3.1 Transmissão de dados utilizando TCP/IP

A arquitectura TCP (Transmission Control Protocol)/IP (Internet Protocol) é um conjunto de protocolos de comunicação entre computadores em rede. O conjunto de protocolos pode ser visto como um modelo de camadas, onde cada camada é responsável por um conjunto de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. Inicialmente, este tipo de arquitectura não foi desenvolvida para funcionar com aplicações de requisitos em “tempo real”. Contudo, através da utilização conjunta de vários protocolos é possível atingir tais objectivos.

O protocolo IP pertence à camada de rede e é responsável pela entrega dos pacotes gerados pela camada superior (camada de transporte). É um protocolo não orientado à conexão e por isso não provê nenhum mecanismo de qualidade de serviço, ou seja, não existe garantia de entrega, verificação de erros ou verificação da sequência de entrega dos pacotes no destino. Assim, para que seja possível garantir os requisitos fundamentais das comunicações VoIP (qualidade de voz e entrega em “tempo real” dos pacotes), este protocolo trabalha em conjunto com outros, caso do RTP (Real Time Protocol) e RTCP (Real Time Control Protocol) (apresentados no ponto 2.3.2 deste trabalho).

A nível da camada de transporte são três os protocolos que podem ser utilizados para transportar os dados:

- **Protocolo TCP (RFC793[8]):** Orientado à conexão utilizando mecanismos de recuperação de dados perdidos por retransmissão. Apesar de ser um protocolo confiável não está de acordo com os requisitos VoIP, devido aos atrasos que pode adicionar numa comunicação;
- **Protocolo UDP (User Datagram Protocol)(RFC768[9]):** Não é orientado à conexão, não causando por isso atrasos no envio dos pacotes. Contudo, também não consegue garantir a entrega dos mesmos. Apesar disso, a sua utilização conjunta com os protocolos RTP e RTCP faz com que seja dos protocolos de transporte mais utilizados dentro das comunicações VoIP;

- **Protocolo STCP (Stream Control Transmission Protocol)(RFC4960)[10]:** É um protocolo de transporte confiável que foi desenvolvido para resolver as limitações que o protocolo TCP tem em transportar dados de sinalização PSTN sobre redes IP[11]. Amplia as capacidades dos protocolos TCP e UDP por integrar componentes de ambos e adiciona dois novos conceitos [12]. O primeiro, **Multi-homing**, permite que os sistemas que possuem múltiplas interfaces para redundância alternem a sua utilização em caso de falha de uma delas, sem ter de esperar e de forma transparente para a aplicação. O segundo, **Multistreaming**, permite que numa mesma ligação existam simultaneamente múltiplos fluxos de dados.

2.3.2 Protocolos utilizados no VoIP

Para que se possa realizar uma comunicação VoIP é necessário que existam processos de sinalização (inicialização, manutenção e finalização das comunicações) e processos relacionados com transporte e controlo do envio dos pacotes de dados. Assim, além dos protocolos de transporte apresentados no ponto anterior deste trabalho, é necessária a utilização de outros protocolos para a realização de comunicações VoIP.

Recomendação H.323

A recomendação H.323, criada para ITU, tem por objectivo definir especificações para a inicialização, controlo e finalização de comunicações multimédia, sendo por isso uma recomendação que especifica todo um conjunto de protocolos de sinalização e controlo de comunicações audio (suporte obrigatório), video e dados (no caso destas, a sua implementação é opcional), em redes baseadas em pacotes onde não esteja garantida a qualidade de serviço da mesma.

Esta recomendação foi igualmente projectada para que equipamentos desenvolvidos por fabricantes diferentes tivessem a capacidade de comunicar entre si. A sua implementação é algo complexa e pesada devido ao números de protocolos utilizados. Actualmente, esta recomendação já não está a ser muito utilizada, visto que não foi projectada para trabalhar sobre a rede Internet. Dos protocolos utilizados por esta recomendação são de referir:

- **Protocolos Video:** H.261 e H.263;
- **Protocolos Audio:** G.711, G.712, G.723, G.728 e G.729;
- **Protocolos Dados:** T.122, T.124, T.125, T.126 e T.127;
- **Protocolos de Controlo:** H.225, H.235, H.245, H.246, H.248 e H.450;
- **Protocolos de Transporte:** RTP e RTCP;

Protocolo SIP (Session Initiation Protocol)

Este protocolo é definido pela (RFC3261)[13] e opera sobre a camada de aplicação da arquitectura TCP/IP. Devido à sua simplicidade face ao H.323, passou a ser o protocolo de excelência na Internet, no que a comunicações VoIP diz respeito. Foi desenvolvido com base noutros protocolos existentes para funcionar na Internet, como os protocolos SMTP e HTTP. Suporta o transporte de qualquer tipo de carga nos seus pacotes e como a sua arquitectura é do tipo cliente/servidor, apenas utiliza métodos do tipo “requisição-resposta”.

O SIP é um protocolo de sinalização cujo objectivo principal é a criação, modificação e finalização de sessões de comunicações interactivas, como audio, video ou mensagens de texto. Uma sessão é considerada uma troca de dados entre participantes, existindo a possibilidade de adicionar novos participantes numa sessão já estabelecida, permitindo assim a realização de conferências multicast.

O SIP utiliza vários protocolos para conseguir efectuar uma comunicação entre participantes numa mesma sessão. De entre eles destacam-se os protocolos RTC e RTCP para o transporte em “tempo real”, os protocolos MGCP (Media Gateways Control Protocol) e MEGACO/H.248 (Gateway Control Protocol) para o controlo de *Media Gateways* e a forma como estas devem comunicar e o protocolo SDP (Session Description Protocol) para a negociação de parâmetros de sessões multimédia (os protocolos em causa são apresentados nos pontos seguintes deste trabalho).

Apesar da grande utilização deste protocolo em aplicações a funcionar na rede Internet, por fazer uso de duas portas distintas de comunicação (uma para sinalização e outra para transporte), ele apresenta algumas dificuldades na sua configuração quando utilizado em redes que utilizem NAT (Network Address Translator).

Protocolo IAX/IAX2 (Inter-Asterisk Exchange Protocol)

Este é um protocolo Open Source desenvolvido pela *Digium* (empresa que desenvolveu o software Asterisk) com a finalidade de possibilitar a comunicação entre servidores Asterisk. Actualmente, está na sua versão 2 (IAX2). Este protocolo utiliza uma única porta UDP (4569) quer para o canal de sinalização, quer para os fluxos de media, resolvendo desta forma os problemas em relação ao NAT que possam existir em relação ao protocolo SIP.

Outra vantagem apresentada por este protocolo é a possibilidade de, num mesmo fluxo de dados, ser possível efectuar a transmissão de diversos tipos de dados e de diversas sessões para um mesmo destino, representando isso uma redução na largura de banda requerida e latência nas comunicações.

Apesar de ter sido inicialmente desenvolvido para a interligação de PBX IP Asterisk, este é um protocolo que começa a ser utilizado cada vez mais em terminais VoIP, como os softphones e telefones IP (hardphones).

Protocolos RTP (Real-time Transport Protocol) e RTCP (Real-time Transport Control Protocol)

Estes protocolos são utilizados no transporte de pacotes de media (áudio e video) com requisitos de “tempo real” e no controlo da sua transmissão.

O protocolo RTP (RFC1889)[14] é responsável pela transmissão de fluxos de media. As suas principais características são a sincronização dos pacotes pela hora de transmissão e a capacidade de alterar a codificação durante uma transmissão de forma a adaptar-se à largura de banda existente. Este protocolo não reserva recurso da rede, uma vez que utiliza o protocolo RTCP (RFC 3605)[15] para controlar e monitorizar a qualidade de serviço entre todos os participantes de uma comunicação, recorrendo ao envio periódico de pacotes de controlo através do protocolo UDP.

Protocolo SDP

Este protocolo está definido através da RFC2327[16] e é utilizado na descrição e negociação de parâmetros de inicialização para transmissão multimédia entre os participantes da transmissão e pode trabalhar em conjunto com outros protocolos tais como o SIP, RTP e RTCP.

Protocolo RTSP (Real Time Streaming Protocol)

Este protocolo está definido através da RFC2326[17] e é utilizado para fornecer controlo sobre a entrega em “tempo real” de fluxos de media. Fornece igualmente os meios para a escolha de canais de distribuição (como o UDP, multicast UDP e TCP) e mecanismos de execução baseado no protocolo RTP.

Protocolo RSVP (Resource Reservation Protocol)

Este protocolo está definido através da RFC2205[18] e é utilizado para garantir QoS através da reserva de recurso da rede. Este protocolo emula dentro da rede um circuito “dedicado” por onde deve circular a comunicação de voz, tendo esta prioridade perante o outro tipo de dados, dentro desse circuito.

Protocolos MGCP (Media Gateway and Control Protocol) e MEGACO/H.248

Estes dois protocolos são utilizados para controlar as comunicações nas *Gateways* existentes nos sistemas VoIP.

O protocolo MGCP (RFC2705)[19] é um protocolo de sinalização e controlo de chamadas que foi desenvolvido para interligar a rede PSTN com a rede IP.

O protocolo MEGACO/H.248 (RFC3015)[20] tem por base a arquitectura do protocolo MGCP e é um protocolo de sinalização e controlo utilizado entre *Media Gateways* e *Media Gateways Controllers*.

Protocolo SCCP (Skinny Client Control Protocol)

Este é um protocolo de sinalização proprietário da Cisco utilizado para controlar as comunicação entre terminais VoIP e de Call Manager PBX da Cisco. Apesar de ser um protocolo proprietário, existem outras soluções de PBX VoIP, como o Asterisk, que suportam o protocolo em causa.

2.4 IP PBX ou PBX VoIP

Os termos PBX (Private Branch Exchange) ou PABX (Private Automatic Branch Exchange) servem para designar sistemas, manuais ou automáticos, de gestão de chamadas, regra geral só em funcionamento em organizações com alguma dimensão.

Com a evolução da tecnologia VoIP, surgiram no mercado os IP PBX ou PBX VoIP, sistemas telefónicos desenvolvidos para a transmissão de voz (e posteriormente vídeo) sobre uma rede de dados IP.

Face aos sistemas analógicos estes novos sistemas apresentam todo um conjunto de vantagens, onde as principais já foram apresentadas no ponto 2.1.1 deste trabalho.

Adicionalmente, estes equipamentos têm a possibilidade de se poderem interligar com os principais tipos de tecnologias de comunicação actuais (PSTN, GSM, ISDN) para a realização

de chamadas. Esta interligação veio criar toda uma revolução dentro do panorama das comunicações, com especial relevo no ambiente empresarial.

2.5 Gateways VoIP

A interligação da tecnologia VoIP com as outras tecnologias de comunicação só é possível através da utilização de **gateways**. O papel deste tipo de dispositivo é a conversão dos sinais existentes entre as tecnologias a interligar e, consoante a interligação que se queria implementar assim será o tipo de gateways a utilizar.

2.5.1 Interligação Rede IP/Rede telefónica analógica (PSTN)

A interligação da tecnologia IP com o sistema de rede telefónico analógico é feita através da utilização de dispositivos FXO (Foreign eXchange Office) e FXS (Foreign eXchange Station). Estes funcionam sempre aos pares (FXO/FXS), ou seja, a ligação de uma porta FXO será feita a uma porta FXS.

A porta FXS é a interface que fornece a linha analógica, disponibilizando o tom da marcação, corrente de energia e som. Exemplo deste tipo de porta é a tomada de parede onde irá ser ligado o dispositivo analógico como o telefone ou fax, através da sua porta FXO. A figura 2.6 apresenta este tipo de gateway.

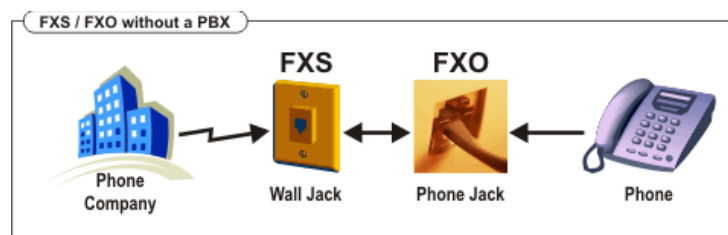


Figura 2.6: Ligações FXO e FXS[21]

Existem diversas formas de implementar este tipo de ligação consoante o contexto de comunicações existente:

Gateways FXO Este tipo de gateway permite que um sistema IP possa utilizar uma linha telefónica analógica para realizar chamadas. A figura 2.7 apresenta este tipo de gateway.

Gateways FXS Este tipo de gateway permite que um sistema analógico possa utilizar a rede IP para realizar chamadas. A figura 2.8 apresenta este tipo de gateway.

Gateways ATA Este tipo de gateway funciona como um adaptador FXS que permite ligar telefones analógicos a uma rede IP. A figura 2.9 apresenta este tipo de gateway.

2.5.2 Interligação Rede IP/Rede telefónica Digital (ISDN)

A interligação da rede IP com a rede ISDN (Integrated Services Digital Network) é feita através da utilização de placas BRI (Basic Rate Interface) e PRI (Primary Rate Interface).

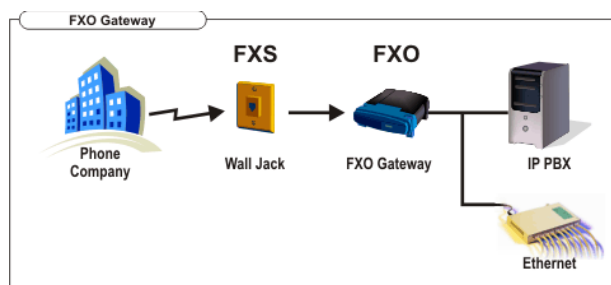


Figura 2.7: Gateway FXO [21]

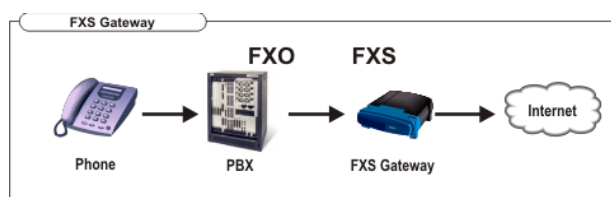


Figura 2.8: Gateway FXS [21]

A diferença entre estes dois tipos de placa reside no número de canais disponibilizados. Assim, as placas do tipo BRI disponibilizam dois canais de dados (B channels) de 64 Kb/s e um canal de sinalização de 16 Kb/s (D channel). Por sua vez, as placas do tipo PRI disponibilizam trinta canais de dados (B channels) de 64 Kb/s e um canal de sinalização de 64 Kb/s (D channel).[22].

2.5.3 Interligação Rede IP/Redes Móveis (GSM)

A interligação da rede IP com as redes móveis é feita através de dispositivos específicos existindo no mercado um conjunto de soluções que vão desde a possibilidade de interligar só uma rede, até soluções bem mais avançadas que permitem integrar em simultâneo diversas redes móveis.

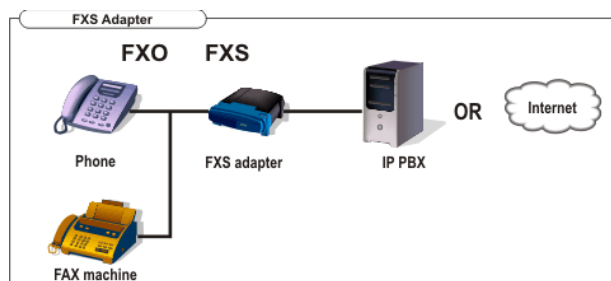


Figura 2.9: Adaptador ATA [21]

2.6 Soluções PBX/Gateways VoIP existentes no mercado

Este tipo de soluções está disponível no mercado de duas formas distintas. A primeira, fazendo uso de implementações por hardware, disponibilizado por marcas como a Cisco, HP, Siemens, Matra, NEC entre outras, sendo de salientar que estes sistemas além de terem um custo de implementação considerável, são sistemas proprietários das marcas que os fornecem, ou seja, o cliente acaba por “ficar preso” à marca que forneceu o sistema. A segunda, é através da implementação de sistemas por software. Interessa referir que ainda dentro deste tipo de soluções, podem ser identificados dois tipos distintos: soluções fornecidas por empresas que realizam toda a configuração necessária, mas que salvaguardam do cliente final o software desenvolvido. Neste caso, mais uma vez, o cliente acaba por “ficar preso” à empresa que procedeu à implementação de todo o sistema. O outro tipo de implementação faz uso de soluções Open Source, onde além de não existir custos de aquisição do software existe total possibilidade de alteração e configuração do seu código base, sendo assim possível criar um produto final à medida e capaz de dar resposta às necessidades de atendimento desejadas.

3. Estudo teórico sobre o Asterisk

Este capítulo apresenta todo o estudo teórico realizado sobre o software Open Source Asterisk. O intuito deste estudo é perceber o modo de funcionamento do Asterisk e a forma como os seus elementos se interligam entre si.

3.1 Asterisk, o que é?

O Asterisk, actualmente na sua versão 1.10, é um software Open Source para a implementação de PBX/Gateways VoIP, desenvolvido por Mark Spencer (inicialmente projectado para dar resposta ao elevado custo de aquisição de um PBX para a sua empresa de Help Desk) e funciona sobre o sistema operativo Linux e permite a utilização de vários protocolos (SIP, H.323, IAX2, etc.) e diversos codecs de forma completamente transparente para o utilizador.

Além de ter sido projectado para desempenhar funções de PBX VoIP, permite igualmente desempenhar funções de gateway por forma a ser possível a sua interligação com as outras redes de comunicação (PSTN, GSM ou ISDN).

3.2 Requisitos do Sistema

O Asterisk é bastante flexível no que diz respeito à distribuição Linux onde pode ser instalado, podendo funcionar na maioria das distribuições existentes hoje em dia.

Sendo uma aplicação que corre em tempo real, tem necessidade de acesso prioritário, quer em relação ao processador, quer ao bus do sistema, logo, os processos relacionados com o processamento das chamadas do Asterisk devem ter uma prioridade alta de execução face a outros processos que possam estar a correr no servidor. Há em conta que, com o aumento do número de chamadas em simultâneo, a carga de processamento aumenta, existindo maiores dificuldades em manter as ligações existentes, o que, para um PBX é uma situação crítica.

Como requisitos de sistema ao nível de hardware, estes variam basicamente consoante dois aspectos: número de utilizadores/canais em simultâneo que se pretende utilizar e alto desempenho do sistema e tolerância a falhas. A figura 3.1 apresenta os requisitos mínimos do sistema a ter em conta a nível de hardware, tendo em conta as funcionalidades/número de canais que se pretendem instalar.

Além do número de utilizadores a que o sistema venha a dar resposta, em sistemas de grande desempenho (ex. Call-Centers) onde é fundamental a não ocorrência de falhas e existir uma

Cenário	Número de canais em simultâneo	Requisitos mínimos recomendado
Sistema em Casa	< 5	400 MHz x86, 256 MB RAM
SOHO (Small Office/Home Office)	5 to 10	1 GHz x86, 512 MB RAM
Pequenas Empresas	Até 25	3 GHz x86, 1 GB RAM
Médias e Grandes Empresas	> 25	Instalação de múltiplos servidores com funcionalidades repartidas

Figura 3.1: Requisitos de Hardware [23]

resposta pronta por parte do(s) servidor(es), deve ter-se também em atenção outros factores, tais como [23] :

Codecs utilizados: Poderá exigir mais ou menos carga de processamento consoante os codecs utilizados e o que isso implicará no desempenho do servidor;

Chamadas em conferência: O servidor tem de processar cada fluxo de áudio de entrada nos diversos fluxos de saída consoante o número de participantes, o que em tempo real, irá gerar um acréscimo de carga computacional no CPU;

Processos a correr no servidor: Preferencialmente, nada mais deve estar a correr no servidor a não ser o sistema operativo e os processos do Asterisk. Por vezes, é recomendada a instalação de vários servidores para repartir as tarefas e aumentar o desempenho total do sistema.

Outros factores a ter em conta, são a motherboard a utilizar, a velocidade de resposta do CPU, o tráfego existente na rede, a optimização do sistema operativo, a qualidade e desempenho das placas utilizadas, irão de igual forma ter importância no desempenho total do sistema.

3.3 Funcionalidades do Asterisk

Entre as principais funcionalidades que se pretende obter com a utilização de um servidor IP PBX com estas características, destacam-se as seguintes [24]:

Receber uma chamada e efectuar o seu encaminhamento: É possível ao receber uma chamada, analisar alguns dos atributos da mesma e saber para onde esta deve ser reencaminhada, se para uma extensão, se para várias, entre outros;

Sistema de correio de voz: É possível que quando uma chamada não é atendida, seja emitido um sinal a avisar que a chamada vai ser reencaminhada para o correio de voz. Adicionalmente, pode ainda ser enviado para o email do destinatário da chamada, uma notificação dessa chamada perdida, com a possibilidade de ser anexada a mensagem de áudio gravada.

Sistema de música em espera: Permite configurar o sistema para tocar determinados ficheiros de música quando uma chamada é colocada em espera.

Sistema de Interactive Voice Response (IVR): Sistema que permite configurar o atendimento, para que de forma automática e sem intervenção humana, o chamador consoante as informações que lhe são transmitidas, consiga decidir o que pretende. Esta interacção entre

o originador da chamada e o Asterisk, é normalmente feita através do toque nas teclas do telefone. Contudo, já começam a estar disponíveis soluções de interacção através da voz.

Sistema de Call Detail Record (CDR): Sistema que permite armazenar numa base de dados todos os detalhes de uma chamada quando esta é realizada, ou seja, a qualquer momento é possível verificar quantas chamadas foram feitas por um utilizador, o seu destino e duração. É com base nestes dados que, utilizando aplicações externas ao Asterisk (ex.A2Billing), é possível criar um sistema de Billing.

Sistema de Conferência: Permite que vários utilizadores possam comunicar em conferência. Esta pode ser restrita a determinados utilizadores e ter uma palavra de acesso associada.

Atendimento Automático e Filas de Chamadas: Esta funcionalidade está vocacionada para Call-Centers. O Asterisk faz a gestão das chamadas, estejam elas a decorrer ou em espera. Quando recebe uma chamada, se todas as extensões estiverem ocupadas, esta é colocada em espera. Quando uma das chamadas finaliza, então a que estava em espera é reencaminhada para a extensão que ficou livre.

Interligação com outras Gateways: Através da interligação com outros sistemas Asterisk ou Gateways VoIP já implementados (como por exemplo entre uma empresa mãe e uma sua filial), aumenta as potencialidades na realização de comunicações.

3.4 Arquitectura do Asterisk

A arquitectura interna do Asterisk é constituída por um núcleo que é formado pelo **PBX Switch Core, Codec Translator, Application Launcher, I/O Manager and Scheduler, Dynamic Module Loader** e pelo **CDR Core** [25].

A função de cada elemento do núcleo é a seguinte:

PBX Switching Core: É através deste componente que o Asterisk realiza a ligação de chamadas entre os vários utilizadores, mesmo que feitas através de diferentes interfaces (Hardware ou Software). É igualmente responsável pela automatização de tarefas.

Codec Translator: Este componente é o responsável pela codificação/descodificação dos vários formatos de codecs de compressão que o Asterisk é capaz de operar.

Application Launcher: É através deste componente que o Asterisk consegue colocar em execução os serviços para o qual é configurado, como por exemplo o atendimento automático de chamadas.

I/O Manager and Scheduler: Componente que tem a seu cargo a gestão das tarefas de baixo nível. É também responsável pelo desempenho optimizado do sistema, em qualquer das situações de funcionamento.

Dynamic Module Loader: É o componente que funciona quando o Asterisk inicia, responsável pela leitura e inicialização dos drivers dos canais, formatos dos ficheiros com que o Asterisk vai operar, codecs e aplicações disponíveis, entre outros, fazendo a ligação às API apropriadas.

CDR Core: É o responsável pelo armazenamento dos detalhes de uma chamada.

Este núcleo é responsável pela interligação de quatro módulos fundamentais ao funcionamento do Asterisk, que quando carregados no seu arranque, irão permitir ao núcleo uma transparência no processamento das chamadas, ou seja, o núcleo não tem necessidade de se preocupar com detalhes de uma chamada, como por exemplo, qual o codec necessário, qual o tipo de ligação, etc.

Os quatro módulos necessários ao funcionamento do Asterisk são:

Asterisk Channels API: É responsável pela manipulação do tipo de conexão que um emissor de uma chamada efectua, independentemente do tipo desta (VoIP, ISDN, PRI, etc.). Cada chamada no Asterisk é feita sobre uma interface num canal distinto.

Codecs Translation API: É responsável pelo carregamento dos módulos de codecs, permitindo a codificação de áudio a processar no formato desejado. É possível trabalhar com vários formatos de áudio codificado, como o GSM, G.711, G.722, Mp3, etc.

Asterisk Applications API: É responsável pelo carregamento dos módulos necessários ao funcionamento das várias aplicações/funcionalidades a implementar no sistema.

File Format API: É responsável pela manipulação de ficheiros nos vários formatos para leitura, escrita e armazenamento de dados fisicamente no sistema. O Asterisk permite a utilização de ficheiros em diversos formatos (Wav, Mp3, GSM, etc.) para serem utilizados como toques de chamada, música em espera, entre outros.

Para melhor compreensão da interligação dos elementos enunciados (núcleo e módulos), a figura 3.2 mostra a forma como estes se relacionam.

Outra forma de analisar a arquitectura do Asterisk, é olhar para a sua forma de funcionamento, sendo esta baseada em:

Canais: São responsáveis pelas conexões das chamadas, independentemente da tecnologia que está a ser utilizada (PSTN, ISDN, VoIP, etc.). Todas as chamadas são tratadas em canais distintos. O Asterisk suporta diversos tipos de canais (SIP, IAX, H.323, etc.) estabelecendo conexões entre eles.

Codecs: Para melhor se aproveitarem as estruturas de rede e velocidade de transmissão de dados, são utilizadas codificações de compressão de voz. Esta compressão da voz está a cargo dos codecs. Consoante o codec utilizado, assim será a largura de banda ocupada e a menor ou maior velocidade na transmissão dos pacotes de dados. Não esquecer que, quando se utilizam codecs, tem que se ter em conta a carga computacional que o mesmo irá causar no servidor, ou seja, não vale a pena utilizar um codec com grande taxa de compressão, se o mesmo causar uma enorme carga de computação no servidor quando for trabalhar com o codec em causa.

Protocolos: São os responsáveis entre outras coisas, pelas tarefas de sinalização para se estabelecer uma conexão, quais os requisitos a serem utilizados na conexão, como a mesma irá decorrer, etc. O protocolo mais utilizado actualmente pelo Asterisk é o protocolo SIP (Session Initiation Protocol). Apesar disso, foi desenvolvido e é cada vez mais utilizado

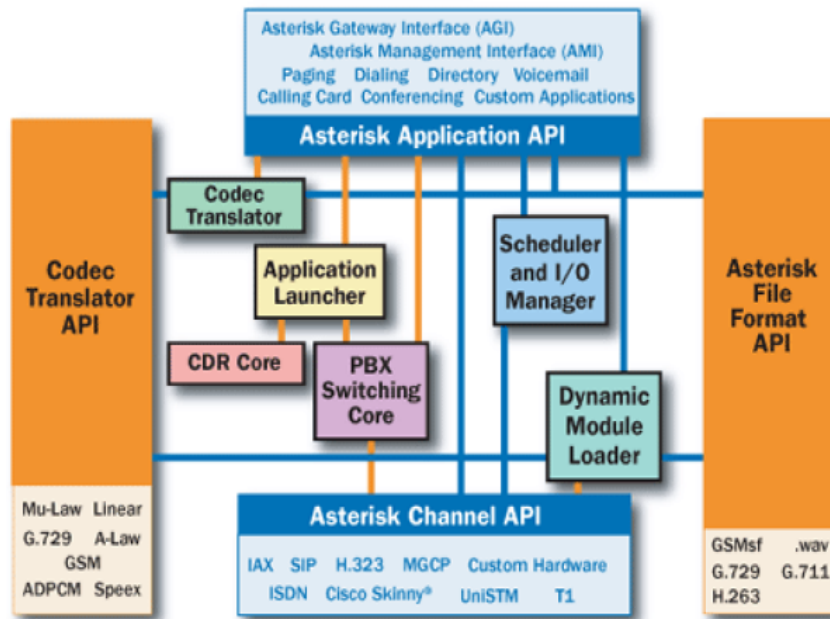


Figura 3.2: Interligação entre Núcleo e Módulos do Asterisk [25]

um novo protocolo chamado IAX (Inter-Asterisk eXchange protocol). Este é um protocolo aberto e foi desenvolvido com o propósito de definir um modelo de comunicação entre servidores Asterisk. Este vem resolver principalmente problemas de Trunking entre servidores e facilitar o tratamento de problemas com firewall's e NAT.

Aplicações: Estas determinam as acções a serem realizadas pelo Asterisk, ou seja, são as tarefas definidas e programadas para serem executadas dentro de um plano de chamadas.

3.5 Directórios e ficheiros de configuração

Quando se instala o Asterisk, este, por defeito, fica instalado em várias directorias, sendo de destacar:

`/etc/asterisk/` Contém os ficheiros de configuração do Asterisk.

`/usr/lib/asterisk/modules/` Contém os módulos do Asterisk.

`/var/lib/asterisk/moh/` Contém os arquivos para a funcionalidade da música em espera

`/var/lib/asterisk/sounds/` Contém os arquivos de áudio a serem utilizados pelo Asterisk nas diversas funcionalidades definidas.

`/var/spool/asterisk/` Directório onde são armazenados os arquivos criados pelo funcionamento do Asterisk, como a gravação de chamadas ou voice-mail.

O Asterisk é configurado através da manipulação de ficheiros em formato ASCII. Basta alterar o conteúdo de determinados parâmetros dentro desses ficheiros, reiniciar a leitura desses ficheiros

pelo Asterisk e as novas funcionalidades passam a estar disponíveis pelo servidor. No apêndice A é apresentado uma listagem dos principais ficheiros de configuração.

3.5.1 Secções nos ficheiros de configuração

Um ficheiro de configuração é constituído por vários segmentos, aos quais se dá o nome de **secções**. Estas são identificadas através de um nome, aparecendo este dentro de parêntesis rectos. É através das secções que se definem os diversos parâmetros de configuração a utilizar pelo Asterisk. Estes são definidos através da instrução **parâmetro = valor**. É possível criar, em certos casos, secções padrões (definições gerais que podem vir a ser herdadas por outras secções). Nestes casos, a seguir ao nome da secção devem aparecer os símbolos (!). Para que uma secção possa herdar herdar as definições de uma secção padrão, basta colocar entre parêntesis curvos o nome da secção padrão a seguir ao seu nome.

Exemplo do conteúdo de um ficheiro de configuração é apresentado abaixo.

```
[secção1] Identificação de uma secção
Parâmetro1 = valor
Parâmetro2 = valor

[secção2](!) Identificação de uma secção padrão
Parâmetro1 = valor
Parâmetro2 = valor

[secção3](secção2) A secção3 herda as definições da secção 2
Parâmetro1 = valor
Parâmetro2 = valor
```

3.5.2 Contextos nos ficheiros do plano de chamadas

No que diz respeito à configuração dos ficheiros relacionados com o plano de chamadas, as secções criadas têm o nome de **contextos**. Estes também podem herdar funcionalidades uns dos outros, bastando para isso utilizar a instrução **include => nome_contexto** dentro do contexto que quer herdar as funcionalidades já implementadas noutra contexto.

Exemplo do conteúdo de um ficheiro de configuração do plano de chamadas.

```
[contexto1] Identificação do contexto1
exten => 2000, 1, Dial(SIP/2000)
exten => 2000, n, Hangup()

[contexto2] Identificação do contexto2
include => contexto1; Inclusão do contexto1 dentro do contexto2
exten => 3000, 1, Dial(IAX/3000)
exten => 3000, n, Hangup()
```

Muitas vezes para uma melhor e mais fácil gestão das funcionalidades a implementar no plano de chamadas, estas são criadas através de novos contextos em ficheiros distintos. Nestes casos, para que a funcionalidade a implementar esteja disponível no plano de chamadas (`extensions.conf`), é necessário dar a localização do ficheiro a incluir dentro do `extensions.conf`. Esta indicação é feita através da directiva **#include directório/nome_ficheiro** no início do ficheiro `extensions.conf`. Ter em atenção que a directoria de referencia é a `/etc/asterisk`. Para se utilizar algum dos novos contextos disponiveis dentro do plano de chamadas, basta utilizar a instrução **include => nome_contexto** .

Exemplo da inclusão de ficheiros no plano de chamadas e utilização dos contextos definidos nele.

Ficheiro funcionalidades.conf

```
[contexto3] Identificação do contexto3
exten => 4000, 1, Dial(ZAP/1/4000)
exten => 4000, n, Hanguap()
```

Ficheiro extensions.conf

```
#include funcionalidades.conf Inclusão de ficheiro no plano de chamadas
```

```
[contexto1] Identificação do contexto1
include => contexto3 Inclusão do contexto3 dentro do contexto1
exten => 2000, 1, Dial(SIP/2000)
exten => 2000, n, Hanguap()
```

```
[contexto2] Identificação do contexto2
include => contexto1 Inclusão do contexto1 dentro do contexto2
exten => 3000, 1, Dial(IAX/3000)
exten => 3000, n, Hanguap()
```

3.6 Registo de utilizadores no Asterisk

Para que um utilizador possa efectuar chamadas, primeiro é necessário criar o seu registo no servidor. Consoante o tipo de utilizador (**SIP, IAX, etc**) assim será o ficheiro de configuração necessário configurar (**sip.conf, iax.conf, etc**). Após a configuração dos ficheiros adequados, estes devem ser reinicializados no servidor, caso este esteja em funcionamento, pois só dessa forma os utilizadores passam a estar registados no servidor.

Independentemente do tipo de utilizador, entre os vários parâmetros necessários para a sua criação no servidor (apêndices **B** e **C**), existe um que é de extrema importância para toda a estrutura do Asterisk, o parâmetro **context**. É este parâmetro que vai relacionar um utilizador a um contexto existente no plano de chamadas, logo, o valor aqui definido tem obrigatoriamente de ser o nome de um dos contextos existentes no plano de chamadas, caso contrário, o Asterisk irá terminar todas as chamadas desse utilizador por falta de contexto conhecido.

É através dos dados de registo do utilizador no servidor (`userID` e `password`) com alguns dados do servidor (`endereço IP`) que se configura o equipamento a utilizar na realização de chamadas

(telefones VoIP, softphone, etc.). No apêndice D são exemplificados os passos necessários na configuração de um utilizador num softphone.

3.7 Componentes base do plano de chamadas

O plano de chamadas do Asterisk, implementado através do ficheiro **extensions.conf**, é visto como se fosse o coração de todo o sistema. Nele são definidas todas as instruções e funcionalidades a implementar, especificando o que fazer em cada situação. O Asterisk é altamente personalizável, permitindo por isso dar resposta às diversas necessidades de atendimento.

Um plano de chamadas é constituído por quatro conceitos base: **Contextos**, **Extensões**, **Prioridades** e **Aplicações**. Todos estes conceitos trabalham em conjunto para que todo o sistema funcione na perfeição.

3.7.1 Contextos

Os contextos são secções dentro do plano de chamadas, desenvolvidas para dar resposta às necessidades para o qual o servidor foi implementado. A menos que exista interacção entre os contextos existentes, estes são independentes uns dos outros. Os contextos representam as extensões/números para a qual um utilizador pode ligar. Servem igualmente para segmentar o plano de chamadas em funcionalidades disponíveis pelo sistema.

Entre outros, existem dois contextos especiais dentro do plano de chamadas, **[general]** e o **[globals]**. O contexto **[general]** é utilizado para se definirem as opções gerais do plano de chamadas. O contexto **[globals]** é utilizado para definir as variáveis globais a implementar, ficando estas disponíveis para todo e qualquer contexto existente dentro do plano de chamadas. A definição de variáveis globais facilita a alteração dentro de um plano de chamadas. Imagine-se a alteração do nome de um canal dentro de todo o plano de chamadas, ou seja, em vez de ser utilizado o nome do canal nas diversas extensões, pode ser utilizada uma variável criada para esse fim, logo, a qualquer momento, alterando o conteúdo da variável (novo nome canal), todo o plano de chamadas fica actualizado e funcional.

Quando um utilizador inicia uma chamada, é criado um canal entre ele e o plano de chamadas (consoante o tipo de utilizador assim será o canal criado, utilizador SIP canal SIP, utilizador IAX canal IAX, etc.). O Asterisk ao receber os dados do utilizador através desse canal, consulta todo o plano de chamadas e verifica se existe um contexto com o mesmo nome do contexto associado ao utilizador. Em caso afirmativo, é percorrido todo esse contexto para verificar se o utilizador está autorizado a efectuar ligações para o número digitado. Havendo essa autorização, é então criado um novo canal para o destinatário da chamada. É possível numa mesma comunicação serem utilizados canais de diferentes tipos.

3.7.2 Extensões

Uma extensão representa o número do destinatário de uma chamada. Dentro do ambiente do Asterisk, além desse conceito, uma extensão é o conjunto de passos ou acções que o servidor deve realizar quando recebe uma chamada. A sintaxe de uma extensão completa no Asterisk é a seguinte:

`exten => extensão, prioridade(rótulo), aplicação.`

A extensão representa o número disponível no plano de chamadas e para o qual um utilizador pode efectuar chamadas.

Extensões especiais

Existem três extensões especiais no Asterisk que são utilizadas para dar resposta às seguintes situações:

Número inexistente no contexto: Quando um utilizador realiza uma chamada para um número que não está definido no seu contexto, pode fazer ouvir-se uma mensagem de erro a alertar para tal facto. Neste caso deve ser utilizada a extensão **s**.

Opção inválida no atendimento automático: Quando um utilizador é atendido pelo atendimento automático e confrontado com as diversas opções existentes, este digita uma opção inválida ou não prevista pelo atendimento. Estes casos são resolvidos através da utilização da extensão **i**.

Excedeu o tempo no atendimento automático: Quando confrontado com as opções do atendimento automático, o utilizador não digita nada durante o espaço de tempo definido para tal, o servidor para não ficar com essa chamada em linha, finaliza-a. Estas situações são resolvidas através da utilização da extensão **t**.

Um exemplo de implementação destes casos está desenvolvido no ponto 4.8 deste trabalho.

Definição de padrões na extensão

É possível criar padrões de extensões para a marcação de chamadas. Os padrões simplificam a definição dos números que um utilizador tem à sua disposição para marcação, deixando de ser necessário a especificação individual de todos esses números. Quando definido mais do que um padrão num mesmo contexto, o Asterisk só irá executar a extensão mais específica tendo em conta o número digitado.

Todos os padrões de extensão começam por um underscore (`-`) seguido de um ou mais dos seguintes caracteres:

X Substitui um qualquer número de 0 a 9.

Z Substitui um qualquer número de 1 a 9

N Substitui um qualquer número de 2 a 9

[1,5-7] Substitui um qualquer número dentro do intervalo especificado, neste caso, poderia ser o número 1,5,6 ou 7.

. (**período**) Substitui um ou mais caracteres quaisquer

! Substitui zero ou mais caracteres quaisquer

Um exemplo de implementação fazendo uso da definição de padrões está desenvolvido no ponto 4.4 deste trabalho.

3.7.3 Prioridades

As prioridades representam a ordem sequencial com que cada extensão irá ser realizada. A primeira extensão tem início na prioridade 1 e por motivos de facilidade de manutenção e alteração do plano de chamadas, todas as extensões seguintes podem ser definidas com prioridade **n** em vez de forma sequencial (2, 3, ...). Apesar de opcional, pode ser definido um **rótulo (identificação)** a uma prioridade. Esta situação possibilita a criação de planos de chamadas condicionais mesmo utilizando prioridades **n**, como abordado no ponto 3.8.4.

3.7.4 Aplicações

As aplicações são as responsáveis pela realização das acções dentro do plano de chamadas. Cada aplicação realiza uma acção específica no canal que está a ser utilizado. Existem aplicações que necessitam de argumentos para funcionarem. Nestes casos, estes devem ser especificados sequencialmente dentro dos parêntesis da aplicação separados por uma vírgula.

O Asterisk tem disponível um grande número de aplicações para fazer face às diversas funcionalidades que podem ser implementadas neste tipo de servidor. Apesar da extensa lista de aplicações disponíveis pelo Asterisk, no anexo E é apresentada uma listagem das aplicações com que habitualmente mais se trabalha dentro de um plano de chamadas.

3.8 Outros elementos do plano de chamadas

Para se implementarem planos de chamadas mais abrangentes, por forma a dar resposta às reais necessidades de atendimento das empresas, há necessidade de utilizar outros elementos além dos descritos até este ponto.

3.8.1 Variáveis

Através da utilização de variáveis passa a ser possível basicamente duas coisas:

- Redução na escrita de extensões.
- Adicionar clareza e lógica ao plano de chamadas.

No Asterisk, as variáveis são case sensitive, ou seja, VAR é diferente de Var. A atribuição de valores às variáveis é feita através da aplicação `Set()`. Quando se quer utilizar o valor da variável tem que se utilizar a sintaxe `${Variável}`.

Variáveis Globais

Estas variáveis podem ser utilizadas dentro de todos os contextos do plano de chamadas. A sua declaração tem que ser feita dentro do contexto especial `[globals]`. Apesar de se atribuir um valor inicial aquando da sua criação, é possível alterar esse valor dentro do plano de chamadas utilizando a função `GLOBAL(var)` dentro da aplicação `Set()`.

Variáveis de Canal

Estas variáveis são criadas pelo Asterisk quando se inicia uma chamada e só estão disponíveis para os canais que participam na chamada. O valor destas variáveis está dependente das chamada em curso e o seu valor só existe enquanto a mesma decorrer.

O Asterisk disponibiliza todo um conjunto de variáveis de canal sendo possível através da sua utilização saber-se o que se passa com os canais em utilização numa chamada. No anexo F é apresentado uma listagem das variáveis de canal mais utilizadas.

Variáveis de ambiente

Apesar de não serem muito utilizadas, com este tipo de variável é possível dentro do Asterisk aceder a variáveis de ambiente do Linux. Para isso deve utilizar-se a seguinte sintaxe: `ENV(var)`.

3.8.2 Expressões

As expressões permitem aumentar consideravelmente as funcionalidades dentro de um plano de chamadas. Estas são representadas através de `expressão` e tal como noutras linguagens de programação, são constituídas por operandos e operadores. Os operandos podem ser valores especificados ou variáveis e os operadores podem ser do tipo **booleanos** (`—` - ou, `&` - e), **relacionais** (`<`, `<=`, `>`, `>=`, `=`, `!=`) e **matemáticos** (`+`, `-`, `*`, `/` - divisão inteira, `%` - resto divisão inteira).

3.8.3 Funções

As funções em conjunto com as aplicações disponíveis pelo Asterisk, permitem aumentar as capacidades do plano de chamadas. Com elas passa a ser possível trabalhar com datas, horas, tamanhos de extensões, entre outros, permitindo criar funcionalidades extras no plano de chamadas. A sua sintaxe é `NOME_FUNÇÃO(argumento)` e é possível colocar funções dentro de funções, onde o resultado de uma representa o argumento de entrada da outra.

3.8.4 Condições no plano de chamadas

Uma das grandes funcionalidades num plano de chamadas é tornar este “inteligente”, ou seja, ser capaz de decidir qual a extensão a executar consoante determinadas condições. Um exemplo da aplicação deste tipo de funcionalidade é a implementação de um atendimento automático de chamadas.

Este tipo de funcionalidade é implementado através da utilização da aplicação `GotoIf()` e a sua sintaxe é `GotoIf(expressão?destino1:destino2)`. Esta aplicação efectua a avaliação da `expressão` e caso dê verdadeiro, o plano de chamadas desloca-se para o `destino1`. Caso contrário, irá para o `destino2`.

Os destinos são referenciados através das seguintes situações:

- **Rótulo** de uma prioridade existente numa extensão do contexto actual;
- **Extensão e rótulo** de uma prioridade existente do contexto actual;
- **Contexto, extensão e rótulo** de uma prioridade existente dentro do plano de chamadas

É possível personalizar o atendimento automático à hora, dia da semana, dia do mês e mês com que uma chamada ocorre. Esta funcionalidade é criada através da aplicação `GotoIfTime()` onde se deve utilizar a sintaxe `GotoIfTime(Horas,Dias_da_semana, Dias_do_mês, meses?destino)`. Caso a chamada recebida esteja dentro dos parâmetros temporais definidos, então irá existir um salto no plano de chamadas definido através do `destino`. No apêndice E são apresentadas com maior detalhe estas duas aplicações.

3.8.5 Macros

É possível a utilização de macros (conjunto de acções/extensões a automatizar) no plano de chamadas. Estas são definidas como os contextos, mas precedidas da palavra **macro-** antes do nome da macro ([**macro-teste**]). Para se utilizar uma macro é necessário trabalhar com a aplicação **Macro(nome macro)**. As macros podem ter argumentos associados, sendo estes especificados após o nome da macro, separados por vírgulas.

Exemplo **Macro(nome macro, arg1, arg2, ...)**

Existe um conjunto especial de variáveis associadas às macros, sendo de destacar:

`\${MACRO_CONTEXT}`: Guarda o nome do contexto de onde a macro foi chamada.

`\${MACRO_EXTEN}`: Guarda a extensão de onde a macro foi chamada.

`\${MACRO_PRIORITY}`: Guarda a prioridade da extensão de onde a macro foi chamada.

`\${ARG n }`: Argumentos passados para a macro aquando da sua chamada.

3.8.6 A base de dados do Asterisk (AstBD)

Por vezes é necessário guardar determinados valores dentro de um plano de chamadas. O Asterisk tem disponível uma base de dados interna, chamada **Asterisk Database ou AstDB**. A informação aí contida está organizada sob a forma de **Famílias** e dentro destas são definidas **Chaves**, sendo nestas que se guarda a informação desejada. Quando se vai armazenar um valor na AstDB tem que se fazer referência à família e à chave a utilizar. Caso não exista a chave referenciada dentro da família especificada, então essa chave é criada. Caso a chave já exista, o seu valor será actualizado.

O armazenamento de valores nas chaves é realizado através a aplicação **Set()** em conjunto com a função **DB()**.

Exemplo: **exten => extensão, 1, Set(DB(família/chave)=valor)**

A função **DB()** é também utilizada para obter o valor armazenado dentro das chaves. Para isso basta fazer referência à família e chave que se desejada utilizar.

Exemplo: **exten => extensão, 1, Set(VAR=\${DB(família/chave)})**.

Para apagar valores dentro da AstDB existem duas maneiras distintas: ou se apaga apenas a chave e se utiliza a aplicação **DB_DELETE(família/chave)**, ou se utiliza a aplicação **DBdeltree(família)** e é apagada toda a família.

Exemplo: **exten => extensão, 1, DB_DELETE(família/chave)**.

4. Estudo prático sobre o Asterisk

Este capítulo descreve todo o estudo prático realizado, em ambiente de rede local, sobre o software Asterisk, com o intuito de explorar as diversas funcionalidades de PBX VoIP disponíveis por este software.

4.1 Ambiente de testes

O ambiente de testes criado pretendeu simular a instalação de um servidor Asterisk para funcionar numa empresa, capaz de dar resposta às principais necessidades de atendimento desta. Além da instalação do servidor principal, foi instalado um segundo servidor Asterisk para simular e testar ligações *Trunk* (canal de comunicação) entre servidores VoIP.

Foram utilizados sistemas diferentes na instalação dos dois servidores Asterisk. O primeiro foi instalado num computador “real” onde se testaram todas as funcionalidades que se queriam implementar no Asterisk. O segundo servidor foi implementado em ambiente de máquina virtual (Oracle VM Virtual Box) num computador diferente do primeiro servidor.

O servidor principal só “corria” o sistema operativo, o Asterisk e os programas associados ao seu funcionamento. Esse computador tinha um processador Intel Core 2 T5800 e 4 GB de memória RAM, requisitos suficientes tendo em conta o apresentado no ponto 3.2. O segundo servidor foi instalado numa máquina virtual onde foi definida a existência de apenas 512 MB de memória RAM. O computador onde “corria” a máquina virtual tinha um processador Pentium P6000 a 1.87 GHz com 8 GB de memória RAM, considerando-se por isso ser uma máquina capaz de suportar o funcionamento da máquina virtual e a utilização dos softphones sem comprometer o ambiente de testes.

Em relação ao software, por motivos de ordem técnica foram utilizadas versões de 32 bits (máquina virtual) e 64 bits (computador real). Apesar das versões utilizadas, em ambos os servidores foi instalado o sistema operativo linux CentOS release 5.7 e o Asterisk na sua versão 1.6.2.22. Já em relação aos softphones, foram utilizadas três aplicações distintas: 3CXPhone (ver.6.0.20943.0), X-Lite 4.1 (ver.63214) e o Zoiper Communicator(ver.2.05.11136).

Os testes de realização de chamadas entre utilizadores foi feita através de hardphones (Grandsream - GXP285) e softphones (X-Lite, 3CX Phone e Zoiper), sendo estes instalados directamente no computador onde se instalou o segundo servidor Asterisk, simulando a situação de estarem instalados num dos computadores da empresa.

4.1.1 Como trabalhar com o Asterisk

O Asterisk pode ser inicializado como um serviço quando o sistema operativo arranca ou então na shell do Linux através do comando **“asterisk”**. Enquanto serviço, é possível utilizar os comandos **“service asterisk start”**, **“service asterisk restart”** ou **“service asterisk stop”** para fazer a gestão do serviço no servidor, contudo, é preciso ter cuidado com a sua utilização em servidores a funcionar no terreno, pois irão interromper o serviço, o que poderá causar alguns problemas.

O Asterisk disponibiliza uma ferramenta muito útil chamada *Command Line Interface (CLI)*. Esta ferramenta é uma consola de acesso ao servidor que está em funcionamento permitindo, em tempo real, obter informações sobre quais as tarefas que estão a decorrer, verificar quais as configurações existentes, executar comandos no servidor entre outras. Uma das principais vantagens na utilização da CLI é a possibilidade de visualizar todo um conjunto de mensagens que o servidor disponibiliza quando realiza uma acção, ou seja, é através destas mensagens que se consegue realizar uma correcta avaliação e detecção de erros que possa estar a acontecer no servidor quando se está a implementar um plano de chamadas.

A CLI pode ser inicializada quando se arranca com o servidor Asterisk através da shell do Linux utilizando o comando **“asterisk -r”**. Caso o servidor já esteja a funcionar, pode utilizar-se o comando **“rasterisk”**. Além do parâmetro **“- r”** existem outros que podem ser utilizados no arranque do Asterisk alterando a forma como este é inicializado. Dos parâmetros disponíveis há que destacar:

- h** Mostra a ajuda dos parâmetros disponíveis.
- c** Inicia o Asterisk e uma CLI como processo de utilizador e não como daemon do sistema.
- p** Arranca como pseudo-realtime. Executa o Asterisk com prioridade de tempo real.
- q** Não visualiza as mensagens do servidor.
- v** Activa a visualização/verbose de mensagens do servidor. O seu nível varia de 0 a 9 e quantos mais v forem utilizados na linha de comandos, maior será a quantidade de mensagens apresentadas na CLI.
- g** Faz com que o Asterisk descarregue o núcleo em caso de *segment violation*.

Com base nos parâmetros apresentados, uma das formas de fazer arrancar o Asterisk seria, por exemplo, através do comando **“rasterisk -vvvgc”**, onde o servidor seria inicializado com o nível 3 de verbose, iria ser aberta uma CLI como processo e em caso de *segment violation* todo o núcleo do Asterisk seria descarregado.

Comandos na CLI

Dentro da CLI existe todo um conjunto de comandos disponíveis que permite efectuar todas as operações de manutenção e análise do servidor. Estes comandos estão divididos por áreas de acção, existindo comandos para o plano de chamadas, comandos para gerir o core do Asterisk, comandos relacionados com os utilizadores SIP, com os utilizadores IAX entre outros. A listagem de todos esses comandos e suas funcionalidades é obtida através do comando **“help”** dentro da CLI. Apesar disso, destaco os seguintes comandos:

core show version: Mostra a versão do Asterisk.

core show applications: Mostra a lista de aplicações disponíveis para serem utilizadas no plano de chamadas do Asterisk.

core show channels: Mostra a lista de canais de comunicação activos.

core show translation: Mostra informações sobre o tempo de transcodificação entre os codecs habilitados no Asterisk.

dialplan show: Mostra o plano de chamadas definido no servidor.

dialplan reload: Reinicializa o plano de chamadas.

sip show users: Mostra a lista dos utilizadores SIP.

sip show peers: Mostra a lista de canais de comunicação SIP activos.

sip reload: Faz a reinicialização do ficheiro sip.conf.

iax2 show users: Mostra a lista dos utilizadores IAX.

iax2 show peers: Mostra a lista de canais de comunicação IAX activos.

iax2 reload Faz reinicialização do ficheiro iax.conf.

4.2 Comunicação entre Utilizadores

Num servidor como o Asterisk a funcionalidade mais básica que se pode implementar é a realização de chamadas entre utilizadores. É aqui que todo o plano de chamadas começa e, à medida que se quer implementar novas funcionalidades, vão-se adicionando as novas instruções/extensões ao plano de chamadas existente.

Para se realizar uma chamada existem duas coisas que têm obrigatoriamente que acontecer. A primeira é existir um plano de chamadas com *contextos* criados dentro do ficheiro *extensions.conf*. A segunda é haver *utilizadores* registados no servidor cujo o seu contexto de utilização exista dentro do plano de chamadas.

Não é importante a ordem pela qual estas duas condições são implementadas no servidor, pois uma para funcionar precisa da outra e as definições de uma implicam as definições da outra, pois o nome ou número que é atribuído a um utilizador irá representar uma extensão de ligação dentro do contexto no plano de chamadas. O contexto associado a um utilizador irá representar todas as funcionalidades disponíveis no plano de chamadas para esse utilizador.

4.3 Criação de utilizadores

O Asterisk permite a realização de chamadas entre utilizadores de diferentes tipos (SIP, IAX, etc.), bastando para isso que ambos os utilizadores estejam registados no servidor (chamadas internas) e exista a correcta configuração do plano de chamadas.

4.3.1 Utilizadores SIP

O registo deste tipo de utilizadores no servidor é feito através da configuração do ficheiro *sip.conf*. É neste ficheiro que se definem todos os parâmetros necessários à criação de utilizadores SIP. Conforme o ponto 3.5.1, além da secção **[general]** é possível criar padrões para simplificar a criação de utilizadores, sendo depois só necessário definir os parâmetros específicos de cada utilizador, como o username, password, etc.. No apêndice B é apresentado todo o ficheiro *sip.conf* criado para a realização deste trabalho.

Exemplo do ficheiro sip.conf.

```
[general] Secção geral
context=default
allow=gsm
language=pt

[interno](!) Criação de secção padrão
nat=yes
allow=alaw
allow=ulaw
call-limit=1;

[2000](interno) Criação do utilizador SIP
username=2000
calleid= nome do utilizador <2000>
secret=password
type=friend
host=dynamic
context = ChamadasInternas definição do contexto do utilizador
```

É possível efectuar alterações ao ficheiro sip.conf quando o servidor está em funcionamento, mas para que tais alterações sejam reconhecidas pelo servidor, é necessário que o Asterisk volte a carregar a informação no ficheiro sip.conf. Essa operação deve ser realizada na consola do Asterisk (CLI) através do comando “*sip reload*”.

4.3.2 Utilizadores IAX

O processo de criação de utilizadores IAX é em todo semelhante ao descrito no ponto anterior, sendo neste caso utilizado o ficheiro “*iax.conf*”. No apêndice C é apresentado todo o ficheiro *iax.conf* criado para a realização deste trabalho.

Exemplo do ficheiro iax.conf.

```
[general] Secção geral
bindport=4569
allow=gsm
call-limit=1;
language=pt
```

```
[interno](!) Criação de secção padrão
type=friend
host=dynamic
nat=yes
allow=all

[3000](interno) Criação do utilizador IAX
username=3000
calleid= nome do utilizador <3000>
secret=password
context = ChamadasInternas definição do contexto do utilizador
```

O comando a utilizar para voltar a carregar os dados do ficheiro `iax.conf` no servidor é `“iax2 reload”`.

4.4 Plano de chamadas básico

O plano de chamadas configurado através do ficheiro `extensions.conf` é o local onde se define o que o Asterisk deve fazer quando recebe uma chamada. Como referido no ponto 3.5.2, um plano de chamadas é constituído por contextos e dentro destes são implementadas todas as funcionalidades desejadas.

4.4.1 Chamadas entre utilizadores

Na implementação do primeiro plano de chamadas entre utilizadores, vai desde já ter-se em consideração o seguinte:

- Os pontos 4.3.1 e 4.3.2, onde é apresentado o registo de utilizadores SIP e IAX;
- Os utilizadores SIP vão ficar configurados com as *extensões 2000* e considera-se a existência inicial de dois utilizadores (*2000* e *2001*). O seu contexto associado dentro do ficheiro `extensions.conf` chama-se `userSIP`;
- Os utilizadores IAX vão ficar configurados com as *extensões 3000* e considera-se a existência inicial de dois utilizadores (*3000* e *3001*). O seu contexto associado dentro do ficheiro `extensions.conf` chama-se `userIAX`;
- O ponto 3.7.1, onde para já não se considera a existência dos contextos `[globals]` e `[general]` dentro do plano de chamadas;

A divisão de extensões consoante o tipo de utilizador resulta da necessidade de especificar o canal pelo qual deve ser realizada a chamada, ou seja, se o destinatário da chamada for um utilizador SIP, o Asterisk tem que criar um canal SIP, se for um utilizador IAX, o canal a criar já será um canal IAX. O tipo de canal de destino pode ser diferente do tipo de canal de emissor da chamada, daí ser perfeitamente possível a realização de chamadas entre utilizadores de tipos distintos.

Tendo por base as considerações assumidas e o ponto 3.7, um primeiro plano de chamadas é apresentado de seguida.

```
[ChamadasInternas]
exten => 2000,1,Dial(SIP/2000,20) pode marcar-se para a extensão 2000
exten => 2000,n,HangUp() desliga a chamada

exten => 2001,1,Dial(SIP/2001,20) pode marcar-se para a extensão 2001
exten => 2001,n,HangUp()

exten => 3000,1,Dial(IAX2/3000,20) pode marcar-se para a extensão 3000
exten => 3000,n,HangUp()

exten => 3001,1,Dial(IAX2/3001,20) pode marcar-se para a extensão 3001
exten => 3001,n,HangUp()
```

O plano apresentado é constituído por um contexto (`ChamadasInternas`) e dentro deste são definidas as extensões de atendimento, que não é mais do que o `userID` dos utilizadores criados. As extensões definem os números para os quais se podem efectuar chamadas. São utilizadas duas aplicações, a `Dial()` e `HangUp()`, onde a primeira é a responsável pela realização da chamada e a segunda pela finalização da mesma. Neste caso específico a aplicação `Dial()` utiliza dois argumentos, o `SIP/XXXX` ou `IAX2/XXXX` para especificar o tipo de canal e extensão de destino e `20` para especificar a duração em segundos que uma chamada fica a tocar no destinatário antes de ser desligada pelo servidor.

Interessa salientar duas situações face ao plano apresentado:

- Utilizadores de tipos diferentes conseguem comunicar entre si;
- Como foi considerada a existência de apenas dois utilizadores por tipo, rapidamente foram definidas as extensões de cada um. Facilmente, se percebe que este plano apresenta problemas de escalabilidade quanto à adição de novos utilizadores, pois sempre que é criado um novo utilizador, é necessário alterar o plano de chamadas.

Para resolver a limitação identificada, com base nos pontos **3.7.2** e **3.8.1**, o plano de chamadas deverá sofrer as seguintes alterações:

```
[ChamadasInternas]
exten => _2XXX,1,Dial(SIP/${EXTEN},20) (1)
exten => _2XXX,n,HangUp()

exten => _3XXX,1,Dial(IAX2/${EXTEN},20) (1)
exten => _3XXX,n,HangUp()
```

O problema da escalabilidade fica resolvido através da criação de dois padrões de extensões (`_2XXX` e `_3XXX`) em simultâneo com a utilização da variável de canal `${EXTEN}` (1). O padrão criado representa qualquer número de 2000 a 2999 e de 3000 a 3999, definindo a possibilidade de existirem 1000 utilizadores SIP e IAX sem haver necessidade de alterar o plano de chamadas. A variável `${EXTEN}` guarda o número digitado que deu origem à chamada, logo utilizando o nome do canal em conjunto com o conteúdo da variável tem-se referenciado tudo o que é necessário para se efectuar uma chamada.

4.4.2 Personalizar o atendimento

O Asterisk permite personalizar a forma como é feito o atendimento e mesmo para um plano básico como o que se apresenta, existem duas situações que devem ser consideradas:

- O destinatário não atende a chamada e, neste caso, pode ser reproduzido um áudio a alertar a sua ausência;
- O destinatário tem uma chamada em linha e, neste caso, pode ser reproduzido um áudio a alertar que a linha está ocupada;

Para implementar estas novas funcionalidades, com base nos pontos 3.8.1 e 3.8.4, o plano de chamada deverá sofrer as seguintes alterações:

```
[ChamadasInternas]
exten => _2XXX,1,Dial(SIP/${EXTEN},20)
exten => _2XXX,n,Gotoif("${DIALSTATUS}"="CHANUNAVAIL"?ocupado) (1)
exten => _2XXX,n,PlayBack(vm-nobodyavail) (2)
exten => _2XXX,n,HangUp()
exten => _2XXX,n(ocupado),PlayBack(is-in-use) (3)
exten => _2XXX,n,HangUp()

exten => _3XXX,1,Dial(SIP/${EXTEN},20)
exten => _3XXX,n,Gotoif("${DIALSTATUS}"="CHANUNAVAIL"?ocupado) (1)
exten => _3XXX,n,PlayBack(vm-nobodyavail) (2)
exten => _3XXX,n,HangUp()
exten => _3XXX,n(ocupado),PlayBack(is-in-use) (3)
exten => _3XXX,n,HangUp()
```

A principal alteração é a introdução de uma avaliação ao estado do canal que se quer aceder para realizar a chamada (1). Essa avaliação é feita à variável de canal `${DIALSTATUS}` através da aplicação `Gotoif()`. Caso a chamada tenha sido recusada pelo destinatário ou o tempo definido para atendimento tenha sido excedido, será reproduzido o áudio “*vm-nobodyavail*” para assinalar a indisponibilidade do destinatário (2). Caso o destinatário esteja a participar numa chamada, então é realizado um salto no plano de chamadas para a extensão com rótulo “ocupado” sendo reproduzido o áudio “*is-in-use*” sinalizando que a linha está a ser utilizada (3). A reprodução dos áudios é realizada através da utilização da aplicação `PlayBack()`.

Nos anexos E e F são apresentadas com mais detalhe as principais aplicações e variáveis de canal disponíveis no Asterisk.

4.4.3 Alterar o idioma de atendimento

O Asterisk tem todo um conjunto de áudios que podem ser utilizados na configuração do plano de chamadas. Esses áudios estão disponíveis por defeito na directoria `/var/lib/asterisk/sound/en` em língua inglesa. É perfeitamente possível a integração de outros idiomas existindo para isso duas configurações a ter em conta:

- Definir o idioma de cada utilizador através do parâmetro `language=idioma` quando se cria um utilizador.

- A referência utilizada para definir o “*idioma*” deverá ser obrigatoriamente o nome da directoria a criar dentro de `/var/lib/asterisk/sound/` e onde deverão existir todos os novos áudios que se queriam utilizar.

Para uma rápida e fácil integração de vários idiomas num mesmo plano de chamadas, basta manter o mesmo nome dos ficheiros áudio com mensagens iguais nos vários idiomas que se queria utilizar. Desta forma, sem ser necessário efectuar alterações no plano de chamadas consegue-se que este utilize os áudios correctos no idioma associado ao utilizador.

4.5 Gestão de chamadas recebidas

Existem duas funcionalidades de extrema utilidade que devem ser implementadas dentro do plano de atendimento numa empresa. A primeira é possibilitar a transferência de chamadas entre extensões e a segunda poder colocar-se uma chamada em espera por algum motivo e haver a possibilidade de a conseguir recuperar.

4.5.1 Transferência de chamadas entre extensões

Esta funcionalidade vem dar resposta a duas situações. A primeira, é quando um cliente liga para a empresa e por algum motivo ligou para o departamento errado. A segunda, é quando um cliente ligou para a empresa, trata dos assuntos que tem a tratar, mas ainda precisa de algo num departamento diferente para o qual está a falar. Estes casos são resolvidos através da transferência de chamadas, evitando-se assim que o cliente tenha necessidade de realizar uma nova chamada para a empresa.

A implementação desta funcionalidade dentro do Asterisk é realizada através da configuração do ficheiro `features.conf` existindo duas formas de realizar uma transferência de chamada:

- A chamada é transferida sem haver uma consulta ao destinatário da mesma. Este procedimento é designado por **transferência às cegas** ou **blind transfer**;
- Existe uma consulta ao destinatário da chamada para se saber se este quer ou não atender a chamada, antes desta ser transferida. Este procedimento é designado por **transferência assistida** ou **attend transfer**.

Apesar de existirem configurações por defeito (teclas definidas) no Asterisk para a realização de transferências de chamadas, através da configuração de alguns parâmetros no contexto `featuremap` dentro do ficheiro `features.conf`, essas configurações podem ser alteradas.

```
[featuremap]
blindxfer => # Tecla para transferência às cegas
atxfer => *2 Tecla para transferência assistida
disconnect => *0 Tecla para desligar uma chamada
...
```

Nota: Para que estas funcionalidades possam funcionar dentro do plano de chamadas, é necessário colocar como terceiro parâmetro dentro das aplicações `Dial()` ou `Queue()` as letras **K**

e/ou **k**, **T** e/ou **t** e **H** e ou **h**.

Após a recepção de uma chamada e face ao apresentado acima, um utilizador para a transferir pode optar por:

- Carregar em **#** e após receber uma indicação áudio que irá fazer uma transferência de chamada, deve digitar o número da extensão para a qual quer realizar a transferência da chamada. Neste caso, a chamada é transferida às cegas para a nova extensão deixando o receptor inicial da chamada de participar na mesma após a realização da transferência.
- Carregar em ***2** e após receber uma indicação áudio que irá fazer uma transferência de chamada, deve digitar o número da extensão para a qual quer realizar a transferência da chamada. Neste caso, enquanto é realizado uma segunda chamada para o destinatário da transferência a saber se este quer ou não atender a chamada inicial, a primeira chamada fica em espera. Caso o destinatário queira atender a chamada inicial, então a transferência será efectuada e só a partir daí o receptor inicial da chamada fica liberto da mesma.

4.5.2 Captura de chamadas

A captura de chamadas representa a possibilidade de um utilizador transferir uma chamada que está a tocar numa outra extensão para a sua. Existem duas formas de realizar a captura de uma chamada: captura em grupo ou captura de uma extensão específica.

Captura em grupo

Para se implementar a captura em grupo, existem duas configurações que têm que ser feitas. A primeira é a configuração do parâmetro **pickupexten** do contexto **general** dentro do ficheiro **features.conf**. Este por defeito tem o valor ***8**. A segunda configuração é a definição dos parâmetros **callgroup** (definição do grupo de chamadas do utilizador) e **pickupgroup** (definição de quais grupos um utilizador pode capturar chamadas) no registo do utilizador. Após as alterações nos ficheiros de configuração, caso o Asterisk esteja em funcionamento, este deve ser reinicializado.

Ficheiro *features.conf*

```
[general]
...
pickupexten = *8 Tecla para capturar chamada
...
```

Ficheiro *sip.conf*

```
[2000] (padraoSIP)
...
callgroup=1 pertence ao grupo 1
pickupgroup=1,2,3,4 pode capturar chamadas dos grupos 1,2,3 e 4
```

Para se realizar a captura de uma chamada a tocar noutra extensão, basta carregar *8 no telefone onde se quer atender a chamada. Este tipo de captura só funciona apenas para canais de comunicação com o mesmo protocolo, ou seja, um canal de comunicação SIP somente pode capturar uma chamada a tocar noutra canal de comunicação SIP.

Captura específica

Neste tipo de captura especifica-se a extensão que se deseja capturar, independentemente do seu grupo e protocolo de comunicação. Esta funcionalidade é implementada através da utilização da aplicação **PickUp()** dentro do plano de chamadas, onde a sintaxe a utilizar deverá ser *PickUp(extensão_a_capturar@contexto_da_extensão_capturar)*.

Para se implementar esta funcionalidade no trabalho desenvolvido, foi criado um ficheiro chamado **funcionalidades.conf** e dentro deste um contexto chamado **captura** como se mostra de seguida.

Ficheiro *funcionalidades.conf*

```
[captura]
exten => _600XXXX,n,Pickup(${EXTEN:3}@PICKUPMARK)
```

Neste contexto foi definida a extensão 600 para esta funcionalidade, ou seja, um utilizador para capturar uma chamada específica, marca 600 + número da extensão a capturar. De notar o recurso à variável `${EXTEN:3}` (variável que guarda a extensão digitada, e que, neste caso foram excluídos os 3 primeiros dígitos para cortar o 600 para a aplicação `PickUp()`) e a string especial `PICKUPMARK` (variável que guarda o contexto da extensão que se pretende capturar a chamada).

Para colocar esta funcionalidade activa no plano de chamadas, é necessário fazer o `include` do ficheiro criado (`funcionalidades.conf`) dentro do ficheiro `extensions.conf` e depois fazer referência do contexto `captura` dentro do contexto `ChamadasInternas` como apresentado de seguida.

Ficheiro *extensions.conf*

```
#include IPPBX/funcionalidades.conf inclusão do ficheiro funcionalidades.conf

[ChamadasInternas]
include => captura inclusão do contexto criado no funcionalidades.conf

exten => _2XXX,1,Dial(SIP/${EXTEN},15,KkTtHh)
...
```

4.5.3 Colocação de chamadas em espera

Esta funcionalidade permite colocar uma chamada temporariamente em espera, existindo a possibilidade de serem reproduzidos áudios pré-configurados, evitando assim que quem realizou a chamada esteja a ouvir o ruído de fundo que possa existir enquanto espera pela resposta ao seu assunto.

A configuração de um parque para chamadas em espera é definido através do contexto **general** dentro do ficheiro **features.conf**. Por defeito do Asterisk, esta funcionalidade tem a extensão 700 e ficam reservadas 20 extensões (701 à 720) para a colocação de chamadas em espera. Está igualmente definido o nome de um contexto (**parkedcalls**) que deve ser incluído dentro do plano de chamadas para que se possa implementar esta funcionalidade.

Ficheiro *features.conf*

```
[general]
parkext => 700 Definição da extensão para chamadas em espera
parkpos => 701-720 Definição das extensões do parque de chamadas
context => parkedcalls Definição do nome do contexto do parque chamadas
...
```

Ficheiro *extensions.conf*

```
#include IPPBX/funcionalidades.conf

[ChamadasInternas]
include => captura
include => parkedcalls inclusão do contexto para chamadas em espera

exten => _2XXX,1,Dial(SIP/${EXTEN},15,KkTtHh)
...
```

Para se colocar uma chamada em espera tem que se fazer o mesmo processo da transferência de chamadas, ou seja, tem que se carregar em **#** seguido do número da extensão das chamadas em espera (700). Após isso é reproduzido um áudio indicando qual o número da extensão atribuída à chamada. Este número deverá ser a extensão a utilizar para recuperar a chamada. Caso a chamada não seja recuperada durante um espaço de tempo (45 segundos por defeito), então a chamada volta a tocar na extensão que a colocou em espera.

4.5.4 Música em espera

O Asterisk permite que seja configurado um conjunto de áudios a serem reproduzidos quando uma chamada se encontra em espera. A implementação desta funcionalidade é realizada através de duas configurações distintas. A primeira é a criação de classes dentro do ficheiro **musiconhold.conf**, classes essas que depois serão referenciadas quando se pretende activar as músicas. A segunda configuração é a alteração do parâmetro **parkedmusicclass** no contexto geral dentro do ficheiro **features.conf**.

Ficheiro *musiconhold.conf*

```
[espera] Nome da classe criada
mode=files
directory=/var/lib/asterisk/moh Directoria onde ficam armazenados
sort=random Reprodução aleatória
```

Ficheiro *features.conf*

```
[general]
...
parkedmusicclass = espera Nome da classe criada em musiconhold.conf
...
```

Desta forma, quando uma chamada é colocada em espera será reproduzido um áudio. Esta funcionalidade pode ser aproveitada pela empresa para reproduzir áudios promocionais dos serviços da empresa.

4.6 Voice Mail

O Voice Mail, muito difundido nas comunicações móveis, é a possibilidade que um utilizador tem, quando por algum motivo não possa atender uma chamada, esta ser redireccionada para a sua caixa de correio de voz. Desta forma, quem está a realizar a chamada tem a possibilidade de deixar uma mensagem de voz para o destinatário da mesma, para que este quando disponível, possa aceder à sua caixa de mensagens e ouvir as mensagens aí existentes.

Adicionalmente a esta situação, o Asterisk permite efectuar a integração do Voice Mail com o correio electrónico dos utilizadores, permitindo gerar um email com o áudio da mensagem em anexo para o destinatário da chamada. Esta funcionalidade é particularmente interessante em situações em que um funcionário está fora do escritório, conseguindo desta forma ter acesso às chamadas deixadas no seu Voice Mail da empresa.

Para se implementar esta funcionalidade implica a configuração dos ficheiros dos canais de comunicação (*sip.conf*, *iax.conf* ou outro) e o arquivo de configuração **voicemail.conf**, onde a configuração de um implica a configuração do outro. Nos ficheiros de configuração dos canais, para cada utilizador que se queira activar esta funcionalidade, é necessário definir o parâmetro **mailbox**.

Exemplo do ficheiro **sip.conf**

```
[2000] (interno)
...
mailbox => 2000@classe_do_correio_voz
...
```

O ficheiro **voicemail.conf** é onde se configura a funcionalidade do correio de voz, sendo este constituído por três secções: **general**, onde se definem as configurações do envio do e-mail; **zonemessages**, onde se especificam os formatos de datas e hora, equivalentes aos fusos horários e por último, **classe do correio de voz**, onde se definem classes para grupos distintos de utilizadores. É esta classe que é referenciada dentro da definição do parâmetro mailbox nos ficheiros de configuração do canal.

Exemplo do ficheiro **voicemail.conf**

```
[general]
```

```

format=wav49 Formato de gravação das mensagens de voz
serveremail=asterisk.cwar Servidor de email
attach=yes Activa o anexar ou não do arquivo de voz ao e-mail
maxmsg=50 Quantidade máxima de mensagens por caixa postal
minsecs=3 Tamanho mínimo da mensagem, em seg.
maxsilence=10 Quantidade em seg. de silêncio durante a gravação da mensageml
emailsubject=[IPPABX:] Nova mensagem de voz: ${VM_MSGNUM}
emailbody=${VM_NAME} Existe uma mensagem de voz (${VM_MSGNUM}) na sua caixa
correio (${VM_MAILBOX}) com ${VM_DUR} segundos enviada por ${VM_CALLERID}, em
${VM_DATE}. Obrigado! Servidor IP PABX

```

```
[default]
```

```

2000=>1234,Extensao 2000,mail_de_destino@dominio.com, attach=no, delete=no
2001=>1234,Extensao 2001,mail_de_destino@dominio.com
2002=>1234,Extensao 2002,mail_de_destino@dominio.com
3004=>1234,Extensao 2003,mail_de_destino@dominio.com
3005=>1234,Extensao 2004,mail_de_destino@dominio.com

```

Interessa referir que o Asterisk disponibiliza todo um conjunto de variáveis que funcionam com o voicemail, permitindo saber-se algumas informações úteis, como por exemplo na criação do e-mail a enviar para o utilizador como apresentado.

Após as configurações enunciadas acima, é necessário proceder a alterações no plano de chamadas existindo duas funções que devem ser implementadas para se funcionar correctamente com o voicemail. A primeira é o desvio de chamadas para o correio de voz na indisponibilidade de se atender uma chamada. A segunda é permitir que os utilizadores consigam consultar a sua caixa de mensagens.

Assim, para se implementar a funcionalidade do voicemail foi criado dentro do ficheiro **funcionalidades.conf** dois novos contextos, um chamado **voicemail** e outro **escutavm**. No primeiro contexto desenvolveu-se as funcionalidades de gravação de mensagens através da criação de um atendimento automático. No segundo desenvolveu-se a possibilidade do utilizador ouvir as chamadas existentes no seu voice mail através da criação de mais uma extensão (**60X**). No ponto 4.8 deste trabalho é explicado em detalhe como funciona um atendimento automático.

Para finalizar é necessário fazer duas pequenas alterações no plano de chamadas (1). A segunda é adicionar dois saltos para o contexto voicemail (2)(3) dentro do contexto ChamadasInternas como apresentado de seguida:

```
#include IPPBX/funcionalidades.conf
```

```

[ChamadasInternas]
include => captura
include => parkedcalls
include => voicemail (1)
include => escutavm (1)

```

```
[ChamadasInternas]
```

```

exten => _2XXX,1,Dial(SIP/${EXTEN},20)
exten => _2XXX,n,Gotoif("${DIALSTATUS}"="CHANUNAVAIL"?ocupado)
exten => _2XXX,n,PlayBack(vm-nobodyavail)
exten => _2XXX,n,Goto(voicemail,s,1) (2)
exten => _2XXX,n,HangUp()
exten => _2XXX,n(ocupado),PlayBack(is-in-use)
exten => _2XXX,n,Goto(voicemail,s,1) (3)
exten => _2XXX,n,HangUp()

exten => _3XXX,1,Dial(SIP/${EXTEN},20)
exten => _3XXX,n,Gotoif("${DIALSTATUS}"="CHANUNAVAIL"?ocupado)
exten => _3XXX,n,PlayBack(vm-nobodyavail)
exten => _3XXX,n,Goto(voicemail,s,1) (2)
exten => _3XXX,n,HangUp()
exten => _3XXX,n(ocupado),PlayBack(is-in-use)
exten => _3XXX,n,Goto(voicemail,s,1) (3)
exten => _3XXX,n,HangUp()

```

4.7 Não perturbe

Esta funcionalidade muito útil em ambiente empresarial, é a possibilidade de um utilizador colocar a sua extensão em “não perturbe” por exemplo se estiver em reunião. Na implementação desta funcionalidade vai ter-se em atenção os pontos 3.8.5 e 3.8.6 do trabalho.

Para se implementar esta funcionalidade, em primeiro lugar, foi criado mais um contexto dentro do ficheiro **funcionalidades.conf** chamado **nao-perturbe** sendo definido mais uma extensão (60X) dentro do plano de chamadas. Este contexto basicamente verifica e altera o valor de uma variável (DND) armazenada dentro da base de dados do Asterisk. Esta variável tem dois valores possíveis: o valor 0 quando a funcionalidade “não perturbe” está desactivada e o valor 1 quando esta está activada. O contexto criado é apresentado de seguida:

```

[nao-perturbe]
exten => 622,1,Set(DND=${DB(DND/${CALLERID(number)})})
exten => 622,n,GotoIf("${DND}"="0"?ativa:desativa)
exten => 622,n(ativa),Answer()
exten => 622,n,Wait(1)
exten => 622,n,Set(DB(DND/${CALLERID(number)})=1)
exten => 622,n,Playback(is-set-to)
exten => 622,n,Playback(activated)
exten => 622,n,Hangup()
exten => 622,n(desativa),Answer()
exten => 622,n,Wait(1)
exten => 622,n,Set(DB(DND/${CALLERID(number)})=0)
exten => 622,n,Playback(is-set-to)
exten => 622,n,Playback(de-activated)
exten => 622,n,Hangup()

```

Após a criação do contexto **nao-perturbe** foi desenvolvida uma macro para verificar o estado

da variável DND. Esta macro é utilizada antes de se iniciar uma chamada, pois caso o “não perturbe” esteja activo para uma extensão, então não deve existir comunicações para ela. A macro criada é apresentada de seguida:

```
[macro-dnd]
exten => s,1,Set(DND=${DB(DND/${ARG1})})
exten => s,n,GotoIf("${DND}"= "1"?dndout:sai)
exten => s,n(dndout),Playback(im-sorry)
exten => s,n,Playback(is-set-to)
exten => s,n,Playback(do-not-disturb)
exten => s,n,Playback(auth-thankyou)
exten => s,n,Hangup()
exten => s,n(sai),NoOp(Voltando para o Contexto do Plano de Discagem.)
```

4.8 Atendimento automático de chamadas

O atendimento automático de chamadas ou Interactive Voice Response (IVR) representa hoje em dia para as empresas que disponibilizam “help desk” aos seus clientes via telefone, uma funcionalidade imprescindível dentro do seu sistema de atendimento de chamadas. Este tipo de atendimento permite a interacção automática entre o sistema de atendimento e quem realiza a chamada, interacção essa feita através da reprodução de áudios, onde são apresentadas ao utilizador as opções disponíveis e este, através das teclas do seu telefone ou através da fala, escolhe as opções desejadas. Esta interacção permite fazer uma despistagem de assunto pelo qual o cliente que fez a chamada quer ser atendido, podendo haver uma resposta automática por parte do sistema ou ser transferida a chamada para atendimento através de operador.

4.8.1 Configuração do IVR

Para se implementar um IVR, a primeira coisa que deve ser feita é analisar todas as funcionalidades a que este deve dar resposta, bem como a forma como as mesmas devem ser apresentadas ao utilizador que irá receber essas instruções.

4.8.2 Atendimento automático personalizado ao tempo

Adicionalmente ao apresentado no ponto anterior, é possível consoante a hora a que decorre a chamada personalizar a forma como o atendimento automático irá decorrer. Esta situação é particularmente útil por exemplo, nas situações em que um utilizador liga já fora do horário de expediente da empresa e o atendimento emite um áudio a alertar para essa situação.

4.9 Salas de conferência

Devido aos avanços tecnológicos, hoje em dia existe uma maior descentralização das grandes empresas, onde conceitos como o teletrabalho ou trabalho à distância passam a fazer cada vez mais sentido. Apesar da distância geográfica que possa existir, nunca foi tão fácil comunicar em grupo como nos dias de hoje, como é o caso da vídeo ou áudio conferência.

O Asterisk permite a implementação de salas de conferência ou Meetme entre utilizadores. Essas salas podem ser reservadas a determinados utilizadores ou serem de carácter geral, podendo

igualmente ser criada uma palavra passe de acesso às mesmas, limitando dessa forma o acesso às mesmas.

4.10 Gravação de detalhes das chamadas

A gravação de detalhes das chamadas ou Call Details Record (CDR) é a possibilidade de integrar bases de dados num IP PBX, passando a ser possível armazenar numa base de dados todo um conjunto de informações referentes a uma chamada quando esta é realizada. Esta funcionalidade associada ao desenvolvimento de softwares capazes de manipular esses mesmos dados (ex. A2Billing, ASTPP, FreeSide, Asterisk-Stat, etc.) veio permitir que empresas como Skype, Voip-Buster, Jajah entre muitas outras, colocassem à disposição dos seus clientes todo um conjunto de soluções no que diz respeito à realização de chamadas pela Internet a custos reduzidos ou inexistentes. Não foi intenção deste trabalho explorar a forma como se trabalha com algum dos softwares de Billing disponíveis no mercado, mas sim analisar e testar a integração do Asterisk a uma base de dados, mais especificamente ao sistema de base de dados MySQL, sistema livre e muito difundido dentro da área das bases de dados.

Para se armazenar os dados das chamadas numa base de dados, existem três situações a realizar para que tal passe a ser possível:

- Criar uma base de dados e uma tabela para armazenamento dos dados no MySQL;
- Configurar o ficheiro **cdr_mysql.conf**;
- Alterar o plano de chamadas para activar a gravação dos dados.

Em resposta ao primeiro item apresentado, após a instalação do MySQL foi criada uma base de dados “*asterisk*” e dentro desta uma tabela chamada “*cdr*” [26]. No anexo G é apresentado o código utilizado para a criação da base de dados e tabela bem como uma explicação dos campos utilizados.

No que diz respeito ao segundo item, as alterações realizadas foram no contexto **[global]** dentro do ficheiro de configuração em causa. Após as alterações apresentadas abaixo, caso o Asterisk esteja em funcionamento, este deve ser reiniciado para sincronizar com o MySQL.

```
[global] hostname=127.0.0.1
dbname=asterisk
table=cdr
user=root
password=passwordroot
port=3306
sock=/tmp/mysql.sock
```

Por fim, é necessário proceder a alterações dentro do plano de chamadas para que os dados das mesmas passe a ser armazenado dentro da base de dados. Caso o servidor esteja em funcionamento, após as alterações ao plano de chamadas, é necessário efectuar a reinicialização do mesmo.

4.11 Distribuição automática de chamadas

A distribuição automática de chamadas está directamente relacionada com os sistemas de atendimento automático (IVR), sendo estas funções direccionadas principalmente para “Call-Centers”. A distribuição automática das chamadas implementa a funcionalidade de colocar chamadas em fila de espera. Quando uma chamada é recebida pelo sistema de atendimento este, através do IVR, faz a despistagem do assunto e reencaminha a chamada para a área de acção que o utilizador deseja ser atendido. Caso o(s) responsável(eis) pelo atendimento desse tipo de chamadas esteja(am) com a sua linha ocupada, a chamada é colocada em fila de espera. Quando uma chamada que estava a ser atendida finaliza, a chamada que estava na fila de espera é reencaminhada para a extensão que entretanto ficou livre.

4.12 Trunking entre servidores

Uma das grandes vantagens na utilização do Asterisk em empresas com alguma dimensão onde esta tenha vários escritórios, filiais, armazéns, fábricas ou afins, mas afastadas geograficamente, é a possibilidade de se integrar os diversos servidores de IP PBX existentes num único sistema de atendimento, ficando todos estes interligados e passando a fazer comunicações como se de um ambiente local se tratasse.

Esta funcionalidade é designada como Trunking em servidores, ou seja, é criado um registo de interligação dos servidores existentes de forma a que quando se quer realizar uma chamada para uma filial, basta ligar uma extensão interna e todo o trabalho de reencaminhamento é realizado de forma transparente para o utilizador, pelo Asterisk.

5. Case Study na empresa Clidis - Análise de requisitos

Este capítulo começa por fazer a apresentação da empresa, a forma como está organizada e a sua infraestrutura de rede. Por fim, são apresentados os requisitos de comunicações da empresa.

5.1 Apresentação da empresa Clidis

A Clídis, Clínica de diagnósticos de Sines, foi criada em 1977 e desenvolve diversas actividades na área da saúde, sendo de destacar a prestação de serviços nas áreas de análises clínicas, imagiologia, clínica médica, bloco operatório, medicina dentária, dermatologia estética, cardiologia e pneumologia.

A empresa está sediada em Sines e, actualmente, tem vinte e dois postos de colheita. Apesar das principais valências disponibilizadas pela empresa ocorrerem na sua sede, uma parte importante da actividade da mesma é desenvolvida nos postos colheita, estando estes espalhados principalmente pelo litoral Alentejano.

5.2 Infraestrutura de rede/comunicações existente na empresa

A sede da empresa Clídis funciona num edifício de 5 andares (do piso -2, ao piso 2), estando no piso -2 instalado o “Data Center” da empresa.

Tendo em conta as necessidades informáticas dos diversos departamentos da empresa, a infraestrutura de rede existente assenta em:

- A infraestrutura principal de rede ou “Backbone”, é constituída através de links de fibra óptica de dois canais. Esta infraestrutura faz a interligação dos pisos do edifício;
- Num dispositivo concentrador de fibra óptica (HP Procurve 4104GL) no Data Center, sendo utilizado para interligar os links de fibra óptica existentes para cada um dos pisos;
- Num bastidor por piso onde se centraliza (de forma passiva) todos os pontos de rede existentes nesse piso. A ligação dos pontos de rede ao bastidor é feita através de cablagem categoria 6 (Gbit);

- Em cada um desses bastidores existem switches com gestão (HP procure 1700) para desempenhar as seguintes funções:
 - Interligar os pontos de rede do piso (de forma activa) ao “Backbone”;
 - Ponto de ligação entre o bastidor de piso e o Data Center através de link de fibra óptica;
- Além dos links de fibra óptica em funcionamento, existe a ligação de mais catorze links através de cablagem Cat.6 entre cada bastidor e o data center. Estes links possibilitam, caso necessário, a criação de mais catorze redes físicas independentes;

5.2.1 Interligação informática entre a sede e postos de colheita

Esta empresa devido ao elevado número de postos de colheita que tem, aliado à especificidade dos serviços prestados (muitos deles a funcionar na sua sede), apresenta como requisito de desempenho, sempre que possível, a possibilidade de interligar as filiais ao sistema informático (aplicações específicas) a funcionar nos servidores da sede. Esses acessos estão implementados via Internet através da utilização de VPN (Virtual Private Network). Este tipo de acessos faz com que um computador a funcionar numa filial seja capaz de aceder a recursos disponíveis na rede informática da sede.

5.2.2 Tipos de acessos/comunicações contratados

Devido à organização interna da empresa, existe uma grande necessidade de interligar os sistemas informáticos da sede com os postos de colheita. Esta interligação é feita através da utilização da Internet, existindo em caso de falha no acesso, todo um transtorno no funcionamento da empresa.

Assim, por forma a evitar esse tipo de situações, a empresa tem contratado acessos à Internet através de tecnologias e operadores distintos. Esta solução previne, dentro do possível, a salvaguarda de acesso à Internet, pois caso um operador falhe, o outro, por trabalhar com tecnologias de acesso diferente, irá à partida conseguir garantir esse acesso. Está igualmente implementado um sistema de “Load Balance” (balanciamento de tráfego pelas ligações existentes) por forma a garantir uma melhor aproveitamento das ligações existentes. Abaixo são apresentados os acessos contratados:

- Dois acessos básicos ISDN à empresa NOVIS (funciona como acesso à rede telefónica e à Internet);
- Um acesso ADSL de 8Mbits através da SAPO (só funciona como acesso à Internet);
- Um acesso via Feixes Hertzianos através da Vodafone de 16 Mbits sincronos (só funciona como acesso à Internet). Este acesso está principalmente direccionado para as VPN de ligação entre o sistema da sede e os seus postos de colheita.

5.3 Requisitos de comunicações da empresa Clídis

Devido à área de acção da empresa e a forma como está estruturada, foram definidos requisitos fundamentais para a implementação da solução de comunicações, sendo eles:

5.4. REQUISITOS IMPOSTOS PELA EMPRESA PARA A IMPLEMENTAÇÃO DA SOLUÇÃO DE COMU

Gestão no atendimento de chamadas recebidas: Existe todo um conjunto de valências a funcionar na sede, como a realização de exames, consultas, entre outros, serviços esses efectuados em sectores diferentes da empresa. Como a marcação para esses serviços pode ser realizado via telefone, é fundamental a implementação de um sistema interactivo de atendimento de chamadas, reduzindo ao máximo a intervenção humana no reencaminhamento das chamadas recebidas até chegar ao sector correspondente;

Interligação entre a sede e postos de colheita: Agilizar todo o processo de comunicação entre a sede e os postos de colheita, onde se pretende que cada posto de colheita passe a ser considerado como uma extensão de atendimento dentro da estrutura de comunicações da empresa.

Implementação de um plano de chamadas unificado: Tendo por base os diversos acessos de comunicações que a empresa dispõe, por forma a rentabilizar esses mesmos acessos a nível de custos e existir um controlo das chamadas realizadas, é essencial criar um plano unificado de chamadas, ou seja, definir-se quais os acessos a utilizar na realização de cada tipo de chamadas.

5.4 Requisitos impostos pela empresa para a implementação da solução de comunicações

A empresa estabeleceu como requisitos de implementação os seguintes factores:

Redução de custos em telecomunicações: Este foi o principal requisito imposto, pois com o elevado número de postos de colheita e o tipo de serviços prestados, esta empresa tem necessidade de grandes volumes de comunicações, representando isso um custo considerável para a empresa;

Evitar soluções proprietárias: Este requisito foi justificado com base numa anterior experiência da empresa. Soluções proprietárias obrigam a que a empresa fique “presa” a quem implementa a solução e, em casos de necessidade de manutenção ou outra qualquer intervenção sobre a solução, regra geral, implica custos consideráveis;

Ser uma solução flexível/escalável: Ser uma solução que permita ser adaptada às realidades de comunicação da empresa ao longo dos tempos, conseguindo acompanhar o constante crescimento da empresa e das suas necessidades em comunicações.

6. Case Study - Implementação da solução

Este capítulo apresenta as considerações e o trabalho realizado na implementação da solução de comunicações na empresa Clídis.

6.1 Soluções de comunicações consideradas

Face aos requisitos apresentados nos pontos 5.4 e 5.3, a solução a adoptar passa pela implementação de um PBX/Gateway VoIP. Com base nisso, foi realizada uma pesquisa sobre soluções existentes no mercado. Existe um grande número de soluções dentro desta área capazes de desempenhar as funcionalidades requeridas (pelo menos de forma anunciada), contudo, diferem quanto ao seu modo de implementação.

No que diz respeito às soluções analisadas, estas basicamente são implementadas através de:

Hardware: Estas soluções são projectadas para que, utilizando um único dispositivo de hardware, este desenvolva funcionalidades de PBX e permita integrar as diversas tecnologias de comunicação. Consoante o sistema em causa e a finalidade para que se destina (pequenas, médias ou grandes empresas), assim é o número de recursos e tecnologias de comunicação disponíveis (FXS/FXO, BRI, PRI, GSM), número máximo de utilizadores, capacidade de processamento, entre outros. No que respeita ao sistema de base PBX utilizado, estas soluções fazem uso de sistemas proprietários (desenvolvidos pelo fabricante) ou então, como acontece em muitos sistemas, são baseados no software Asterisk. A configuração das funcionalidades do sistema, regra geral, é feita via browser através da utilização de ambientes gráficos desenvolvidos para esse efeito.

Este tipo de soluções, fornecidas através de empresas como a Cisco, Xorcom, Digium, Alcatel e muitas outras, pela forma como se apresentam (solução integrada de hardware e software), têm custos de aquisição elevados (ex. a solução empresarial da Xorcom - XR3000 - tem um custo de aquisição superior aos 5 000,00 euros[27]). Aliado a isso, apesar de serem muitas vezes escaláveis, quase sempre são soluções “fechadas” à integração de novas funcionalidades não previstas inicialmente na solução, tendo por isso problemas em acompanhar a evolução tecnológica que possa vir a existir dentro da sua área de acção.

Software: Neste tipo de soluções não está incluído nenhum valor relacionado com hardware. Assim, este tipo de soluções requer a utilização de um sistema informático para poder

funcionar. Importa ainda referir que as soluções por software só conseguem desenvolver funcionalidades de gateway se utilizarem uma das seguintes situações:

- Utilização de um operador VoIP para a realização de comunicações com as outras redes, desempenhando estas as funções de Gateway;
- Integração de dispositivos/placas gateways no sistema informático onde irá ser instalada a aplicação.

Assim, consoante a situação que se queira implementar poderá ser necessária a aquisição de gateways para o tipo de rede desejado. Existem no mercado todo um conjunto de gateways para as diversas tecnologias, onde a aquisição de uma placa PCI de acesso básico com 2 linhas pode ter um custo perto dos 270,00 euros (ex. placa da marca sangoma modelo A500BRM)[28].

Independentemente da aquisição ou não de dispositivos gateway, as soluções por software apresentam-se no mercado de duas formas distintas:

Software com custos de aquisição: Estas soluções são aplicações finais implementadas através de código proprietário. São aplicações que não permitem a implementação de novas funcionalidades a não ser através de actualizações do sistema, regra geral, pagas. Nestes casos fica-se “preso” à solução e à forma como esta irá ser desenvolvida no futuro.

Software Open Source: Este tipo de soluções não acarreta custos de aquisição, existindo igualmente a possibilidade de adaptar o seu código às necessidades requeridas. Uma referência neste tipo de soluções é o Asterisk, desenvolvido pela Digium. Devido à grande estabilidade e funcionalidades que consegue implementar sobre ele, aliado à compatibilidade com muitos dispositivos de gateways, este software é adoptado como base em muitas soluções existentes, quer por software quer por hardware, de PBX/Gateways VoIP. Devido à complexidade de configuração na sua forma base (não utiliza ambientes gráficos), surgiram no mercado diversas soluções do tipo “pacote completo”, ou seja, são soluções que instalam todo o software necessário, deste o sistema operativo (linux), sistema Asterisk, bases de dados e sistemas de ambientes gráficos para a configuração das diversas funcionalidades disponíveis. Duas soluções de referência neste tipo de implementação são o AsteriskNow (suportada pela Digium) e o Elastix.

6.2 Solução adoptada - Asterisk / FreePBX

Tendo por base o ponto anterior, foi considerado que a solução a implementar iria ser feita através de software Open Source. Esta decisão é justificada por três motivos:

- Redução de custos de implementação;
- O sistema a implementar tem que ter capacidade de se adaptar ao crescimento da empresa;
- Não se fica “preso” a soluções proprietárias.

Assim, dentro das soluções analisadas, foi considerado que a implementação assentaria no software AsteriskNow tendo por base os seguintes critérios:

- Devido ao facto de a instalação do Asterisk de base ser um processo com alguma especificidade, a instalação de um pacote integrado iria garantir o funcionamento e interligação de todos os seus componentes;
- Devido às características apresentadas e por utilizar o software FreePBX, passa a ser possível trabalhar com ambientes gráficos e o processo de implementação já não seria tão complexo e moroso;
- Por esta solução ser suportada pela própria Digium, à partida seria garantia de qualidade na implementação das funcionalidades requeridas;

6.3 Softwares de base instalados no servidor PBX

O AsteriskNow é uma solução que instala de forma integrada todos os sistemas necessários para a implementação de um PBX/Gateway VoIP. Assim, a solução foi implementada tendo sido utilizados os seguintes softwares principais:

- Sistema Operativo Linux CentOS 5.8;
- Sistema de PBX Asterisk 1.8;
- Sistema de Base de Dados MySQL Ver. 14.12 Distribuição 5.0.95;
- Ambiente gráfico de configuração através do FreePBX 2.10;

O sistema FreePBX é uma ferramenta que permite a configuração das funcionalidades do Asterisk de forma mais “user friendly”. O seu funcionamento está assente sobre a base de dados MySQL, onde todas as opções de configuração efectuadas ficam guardadas na base de dados. Este software através do seu sistema de ambiente gráfico (via web), disponibiliza todo um conjunto de funcionalidades que podem ser implementadas sobre o Asterisk. Quando se procede à configuração de uma determinada funcionalidade, ao validar essa configuração, o FreePBX grava dentro da base de dados as opções seleccionadas. Após isso faz a reconstrução dos ficheiros de configuração (ficheiros de texto) essenciais para o Asterisk funcionar, sendo por isso mais fácil e rápido utilizar este sistema para proceder à configuração das funcionalidades requeridas.

6.4 Hardware utilizado no servidor PBX

A implementação do servidor foi realizada num computador com as seguintes características:

- Processador: Pentium 4 - 3000 ;
- Memória Ram: 2048 Gb;
- Capacidade de Disco Rígido: 160 Gb;
- Placa de Rede a Gbit;
- Placa Beronet BRI de 2 acessos para ligação à rede ISDN da Novis;

6.5 Infraestrutura de comunicações e terminais VoIP utilizados

Apesar da empresa ter capacidade, a nível de infraestrutura de rede, para criar até catorze redes físicas independentes dentro da infraestrutura geral de rede, a solução de comunicações está assente sobre a infraestrutura normal de rede.

Tendo em vista uma ligação mais “limpa” dos terminais VoIP utilizados (hardphones) dentro do edifício da sede, foram instalados nos bastidores de piso injectores PoE (Power Over Internet), tendo estes a função de injectar energia dentro do cabo de rede utilizado para fazer passar dados, deixando por isso de ser necessário utilizar os alimentadores de energia em cada telefone utilizado. Os terminais VoIP instalados nos diversos gabinetes são telefones IP Thomson ST 2020, com excepção feita na recepção onde foi instalado um telefone IP Thomson ST2030 (telefone com funcionalidades de central).

No que diz respeito aos postos de colheita, estes estão ligados via Internet através de VPN à rede da empresa, tendo sido utilizado como terminal VoIP o softphone X-Lite.

6.6 Sistema de atendimento criado

Devido à estrutura interna da empresa, foi definido um padrão a seguir para a criação das extensões telefónicas. Assim, ficou estabelecido que as extensões seriam compostadas por três dígitos, sendo que o primeiro deles está associado ao número do piso onde essa extensão irá funcionar, ou seja, as extensões 0XX (piso 0), 1XX (piso 1) e 2XX (piso 2). Para as extensões do piso -1 foi adoptado o padrão 4XX. No caso dos postos de colheita as extensões criadas utilizaram o padrão 8XX.

Por forma a dar resposta às necessidades de atendimento da empresa, nomeadamente no que diz respeito à recepção e reencaminhamento de chamadas de forma automática, foi efectuada uma análise de como esse atendimento deveria funcionar, no que diz respeito às opções a serem apresentadas e ao encadeamento das mesmas.

Assim, foi implementada uma solução de atendimento automático e interactivo (IVR), onde o utente através das opções disponibilizadas, ao seleccionar a opção desejada, a sua chamada é directamente redireccionada para o serviço desejado. Além desse reencaminhamento, o sistema implementado tem a particularidade de estar associado ao horário de funcionamento desses serviços, ou seja, caso se deseje aceder a um serviço e o mesmo não esteja no seu horário de funcionamento, então é apresentada uma mensagem ao utente alertando para essa situação. Abaixo são apresentadas as ferramentas e lógica utilizada na implementação do IVR principal:

- A primeira situação é redireccionar as chamadas de entrada para um menu de opções inicial. Esta configuração é realizada através da criação de um *inbound route*, utilização as variáveis de canal definidos para trabalhar com a ISDN e redireccionar esse canal para um IVR .
- O menu principal é constituído por quatro opções e foram projectadas três situações de resposta:
 - É digitada uma opção válida e nesse caso a chamada é reencaminhada;
 - É digitada uma opção inválida e, nesse caso, é emitida uma mensagem de aviso e volta a ser apresentado o menu;
 - Não é digitada qualquer opção e volta a ser apresentado o menu. Caso não volte a ser digitada nenhuma opção, então a chamada é reencaminhada para a recepção da empresa (extensão 001);

- Após ser digitada uma opção válida, é testado se a chamada está dentro do horário de atendimento do serviço desejado. Este teste é implementado através de *time conditions*, sendo que:
 - Se estiver no horário de atendimento a chamada é reencaminhada para o serviço desejado. Caso a extensão de atendimento do redireccionamento esteja ocupada, então a chamada fica em fila de espera. Este reencaminhamento é realizado através da definição de *Queues*.
 - Se não estiver no horário de atendimento, é apresentada uma mensagem a alertar para o horário do serviço em causa e após isso, o atendimento é redireccionado para o menu inicial. Esta implementação faz recurso a *Announcements*.

A solução implementada além do sistema de IVR apresentado acima, tem a funcionar todas as principais funcionalidades disponíveis por PBX VoIP, sendo de destacar a transferência de chamadas entre extensões, colocação de chamadas em espera, captura de chamadas, funções de não perturbe, entre outras.

De salientar que apesar do número de extensões existentes (cerca de 50) e do volume de comunicações realizadas pela empresa, a solução está assente num servidor com baixo custo de aquisição, conseguindo desempenhar de forma estável as funções para o qual foi projectado.

6.7 Plano unificado de comunicações

O plano unificado de comunicações define a forma e recursos a serem utilizados na realização de chamadas, tendo como objectivo o controlo e a redução de custos associados às mesmas.

Se dentro da rede as chamadas entre extensões não apresentam qualquer custo para a empresa, pois fazem uso da infraestrutura de rede existente (rede local e VPN), o problema reside nas chamadas para o exterior. Dos recursos existentes a nível de acessos de comunicação (apresentadas no ponto 5.2.2), foi definido que a realização de chamadas para fora da rede interna iria estar assente principalmente no acesso ADSL da Sapo e em situações específicas, na rede ISDN da Novis.

Assim, tendo em conta a realização de chamadas via VoIP, foram subscritas duas contas no ISP VoIP VoIPCheap. A utilização de duas contas está relacionada com a intenção de controlo na realização de chamadas para a rede nacional ou para o estrangeiro. A escolha deste operador foi feita com base nas baixas taxas cobradas para a realização de chamadas, sendo de destacar que chamadas realizadas para Portugal para a rede fixa não é taxado qualquer tipo de valor e para a rede móvel é taxado 0,1 euros, independentemente da rede utilizada.

Assim, foi definido um plano unificado de comunicações com os seguintes critérios:

- Chamadas realizadas para a rede fixa nacional são feitas via VoIP, utilizando a conta do VoIPCheap criada para esse efeito. Contudo, no caso de falha desta conta, de forma automática, o sistema faz utilização da segunda conta;
- Devido ao elevado custo de aquisição de uma gateway VoIP/GSM, as chamadas realizadas para a rede móvel nacional são feitas via VoIP, utilizando a conta do VoIPCheap criada para as chamadas nacionais;
- Chamadas realizadas para o estrangeiro são feitas via VoIP, utilizando a conta do VoIP-Cheap criada para esse efeito. Contudo, no caso de falha desta conta, de forma automática, o sistema faz utilização da outra conta existente;

- Como salvaguarda na realização de chamadas, através da utilização de um prefixo inicial (88), todas as chamadas passam a ser realizadas através da rede ISDN;
- Chamadas realizadas para números iniciados em 800 são feitas através da linha ISDN;
- O fax utiliza a linha ISDN;

Este plano unificado de comunicações foi implementado no servidor através da utilização de “trunks” entre o servidor VoIP instalado e o ISP VoIPCheap e canal ISDN instalado no sistema.

7. Conclusão e trabalhos futuros

Actualmente, são cada vez mais as empresas com necessidades de optimização dos seus recursos dentro da área das comunicações, quer a nível de tempo gasto na recepção e reencaminhamento de chamadas, quer na redução de custos, onde as tecnologias VoIP têm todo um potencial para ser explorado através da utilização da Internet.

Existem no mercado diversas soluções utilizando a tecnologia VoIP e destinadas a resolver os problemas de comunicações das empresas, soluções essas implementadas quer por hardware, quer por software. Dentro das soluções por software, existe cada vez mais a utilização de soluções Open Source, sendo uma área de negócio a explorar.

Assim, numa primeira fase, através de um estudo teórico e prático sobre a tecnologia VoIP assentes em soluções Open Source, foi possível explorar as potencialidade e funcionalidades de um dos softwares de referência neste tipo de aplicação - Asterisk.

Apesar do estudo realizado, sem a sua aplicação num ambiente empresarial real, onde fosse possível testar as suas verdadeiras funcionalidades, a validação deste tipo de soluções não seria comprovada.

Com o objectivo de efectivar esse tipo de validação, no âmbito deste trabalho foi possível projectar e desenvolver toda uma solução assente em tecnologia VoIP Open Source para funcionar em ambiente empresarial, onde o principal requisito apresentado pela empresa foi a optimização dos seus recursos, a nível da agilização do processo de atendimento existente e principalmente na redução de custos em comunicações.

Com base nisso foi desenvolvida uma análise sobre a solução a implementar e a forma como a mesma iria decorrer, tendo sido projectada e implementada uma solução que ia ao encontro das necessidades da empresa, permitindo rentabilizar os seus recursos, quer a nível de redução de custos associados a comunicações, quer a nível da eficácia no processo de atendimento telefónico.

Após a instalação da solução e os testes realizados sobre a mesma, foi possível comprovar que sistemas de PBX/Gateway VoIP assentes em software Open Source, neste caso Asterisk com ambiente gráfico FreePBX, são sistemas válidos, capazes de dar resposta às necessidades de atendimentos das empresas, aliando ainda um custo relativamente baixo (face a outros sistemas) de implementação.

Apesar de todo o trabalho desenvolvido ter servido para melhor compreender e explorar as tecnologias VoIP, esta é uma área em constante evolução, incentivada pelas cada vez maiores velocidades de acesso à Internet disponível nos dispositivos móveis, tecnologias de codificação de

vídeo entre outras.

Assim, o trabalho futuro a desenvolver dentro do âmbito desta tese passa pela exploração de tecnologias como a Unified Communications, sistemas que integram todo um conjunto de serviços, como serviço telefónico, chat, videoconferência, entre outros num só dispositivo.

A. Ficheiros de configuração do Asterisk

O Asterisk é configurado através de ficheiros de texto onde dentro destes devem ser definidos os parâmetros de configuração desejados. Consoante o ficheiro utilizado assim será a funcionalidade que se pretende configurar, existindo por isso necessidade de configurar diversos ficheiros na implementação de um completo plano de chamadas. Estes ficheiros por defeito devem estar dentro da directoria */etc/asterisk* e têm extensão *.conf*

Para uma descrição detalhada de todos os ficheiros de configuração existentes e suas opções de configuração, pode ser consultado o livro “Asterisk - Future of Telephony” [29]. A listagem abaixo apresenta apenas os ficheiros de configuração utilizados na realização deste trabalho.

sip.conf Ficheiro para configuração de todas as opções do protocolo SIP.

iax.conf Ficheiro para configuração de todas as opções do protocolo IAX.

extensions.conf Ficheiro para configuração do plano de chamadas.

meetme.conf Ficheiro para configuração das salas de conferência.

musiconhold.conf Ficheiro para configuração para a música em espera.

voicemail.conf Ficheiro para configuração do Voice Mail.

queues.conf Ficheiro para configuração para listas de atendimento ou Call Centers.

features.conf Ficheiro para configuração para colocação de chamadas em espera.

agents.conf Ficheiro para configuração para listas de atendimento ou Call Centers.

B. Configuração de utilizadores SIP

Devido à extensa lista de parâmetros que podem ser utilizados na configuração de utilizadores SIP, abaixo será apenas apresentado o conteúdo do ficheiro **sip.conf** criado para a realização deste trabalho. Para saber-se todos os parâmetros de configuração disponíveis e sua função, deverá ser consultado o ficheiro sip.conf original.

Abaixo é apresentado o conteúdo do ficheiro sip.conf criado.

```
[general]
context=default
allowoverlap=no
udpbindaddr=0.0.0.0
bindport=5060
bindaddr=0.0.0.0
tcpenable=no
tcpbindaddr=0.0.0.0
srvlookup=yes
maxexpiry=600
defaultexpiry=300
disallow=all
allow=alaw
allow=ulaw
allow=gsm
dtmfmode=rfc2833
```

```
[padrao](!)
type=friend
qualify=yes
nat=yes
host=dynamic
disallow=all
allow=alaw
allow=ulaw
```

```
allow=gsm
allow=ilbc
context=userSIP
canreinvite=no
musicclass=moh
qualify=1000
dtmfmode=rfc2833
call-limit=1
```

```
[2000] (padrao)
username=2000
callerid= Server B 2000 <2000>
secret=cbr1100xx
mailbox=2000@default
language=br
callgroup=1
pickupgroup=1,2,3,4
```

```
[2001] (padrao)
username=2001
callerid= Server B 2001 <2001>
secret=cbr1100xx
mailbox=2001@default
language=en
callgroup=2
pickupgroup=1,2,3,4
```

```
[2002] (padrao)
username=2002
callerid= Server B 2002 <2002>
secret=cbr1100xx
mailbox=2002@default
language=pt
callgroup=3
pickupgroup=1,2,3,4
```

C. Configuração de utilizadores IAX

Devido à extensa lista de parâmetros que podem ser utilizados na configuração de utilizadores IAX, abaixo será apenas apresentado o conteúdo do ficheiro **iax.conf** criado para a realização deste trabalho. Para saber-se todos os parâmetros de configuração disponíveis e sua função, deverá ser consultado o ficheiro sip.conf original.

Abaixo é apresentado o conteúdo do ficheiro iax.conf criado.

```
[general]
bindport=4569
bindaddr=0.0.0.0
disallow=lpc10
disallow=all
allow=gsm
allow=ulaw
language=br
jitterbuffer=no
forcejitterbuffer=no
maxjitterbuffer=500
maxexcessjitterbuffer=100
autokill=yes

[padrao](!)
type=friend
context=userIAX
host=dynamic
allow=all

[3004](padrao)
username=3004
secret=cbr1100xx
callerid= Jose Silva <3004>
mailbox=3004
qualify=yes
```

```
[3005] (padrao)
username=3005
secret=cbr1100xx
callerid= Antonio Ramalho <3005>
mailbox=3005
qualify=yes
```

D. Configuração de um utilizador num Softphone

Na realização deste trabalho foram utilizados três softphones distintos para a realização das chamadas entre utilizadores e testes das restantes funcionalidades. A configuração de um utilizador em qualquer um dos softphones é muito semelhante sendo necessário entre outros, os dados do `userid` e `password` do utilizador no servidor e o IP do servidor Asterisk. Se os dados estiverem correctos e o servidor aceitar o pedido de registo dos softphones, passa a ser possível a realização de chamadas.

D.1 Configuração no X-Lite

O registo do utilizador é feito acedendo ao menú *Softphone* e dentro deste escolher a opção *Account Settings*. Na janela de configuração do utilizador, os campos *User ID*, *Domain* e *Password* são para preencher com os dados de registos e com o endereço IP do servidor Asterisk. Os campos *Account name* e *Display name* são campos que irão ser utilizados dentro do X-lite para identificação do utilizador em uso.

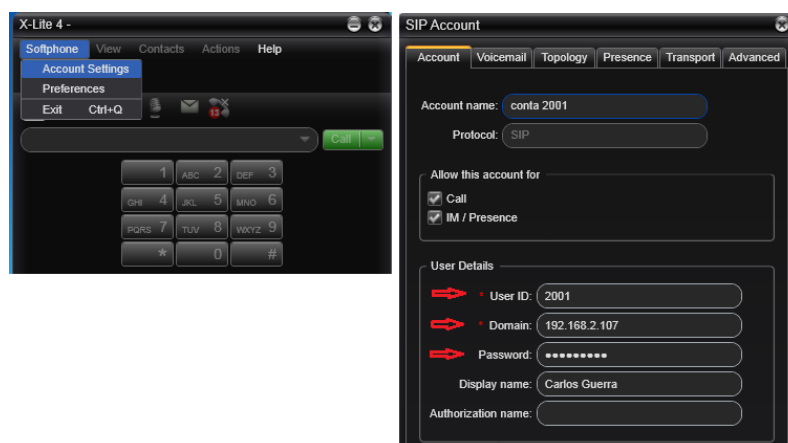


Figura D.1: Aceder ao menu de configuração do X-Lite 4

D.2 Configuração no 3CX

Para se aceder ao menú de configurações do 3CX é necessário carregar no botão inferior e depois escolher a opção **conecção** como mostra a figura D.2.



Figura D.2: Aceder ao menú de configuração do 3CX

Os campos a preencher com os dados do registo, neste caso, são os campos **ID**, **Senha** e **IP do servidor**.

Account name:	Conta 2000
Caller ID:	Carlos
Credenciais	
Insira as credenciais de sua conta SIP	
Ramal:	2000
ID:	2000
Senha:	*****
Minha posição	
Especifique o IP do seu servidor de PBX/SIP	
<input checked="" type="radio"/> Eu estou no escritório - IP local	192.168.2.107 do PBX
<input type="radio"/> Eu estou fora do escritório - IP ext	do PBX

Figura D.3: Configuração da conta do utilizador no 3CX

D.3 Configuração no Zoiper Communicator

O Softphone Zoiper Communicator é o único dos três softphones utilizados que permite o registo de utilizadores SIP e IAX. Para se configurar um utilizador tem de se aceder ao menú **Confi-**

gurações e dentro deste **Preferências** com mostra a figura D.4.

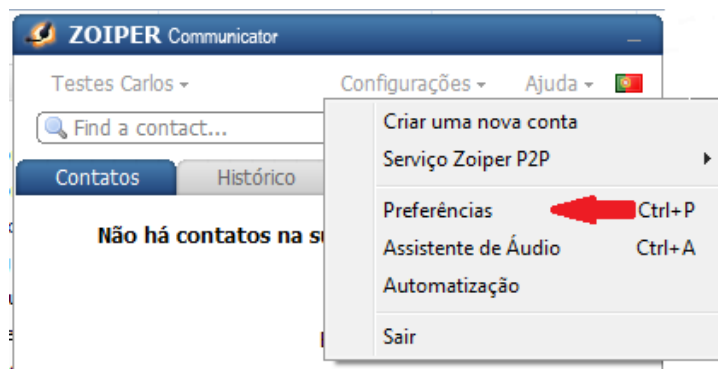


Figura D.4: Aceder ao menu de configuração do Zoiper

Dentro da janela das **Preferências** primeiro deve ser escolhido qual o tipo de utilizador a registar (lado esquerdo da janela) e depois disso preencher os campos **Utilizador**, **Senha** e **IP do Servidor** com os dados de registo e IP do servidor.

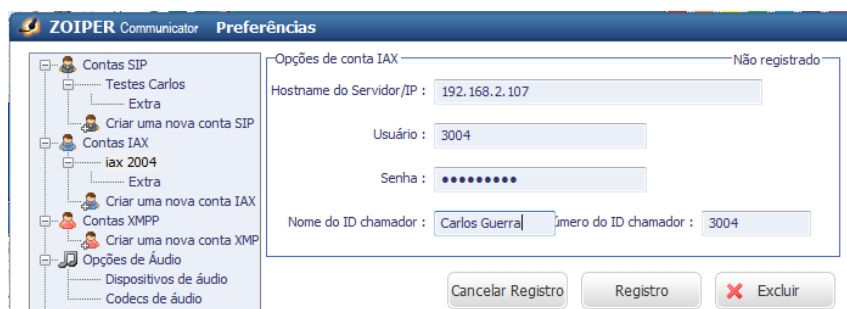


Figura D.5: Configuração da conta do utilizador no Zoiper

E. Aplicações do Asterisk

As aplicações dentro de um plano de chamadas são as responsáveis pela realização de acções. Abaixo é apresentada uma listagem das aplicações utilizadas na realização deste trabalho. Através do site da Digium[30] pode ser obtida uma listagem completa das aplicações disponíveis no Asterisk.

Answer() Aplicação que efectua o atendimento automático de uma chamada

Dial() Aplicação que efectua a chamada para um determinado canal.

Playback() Aplicação utilizada para reproduzir ficheiros de áudio.

Background() Aplicação utilizada para reproduzir ficheiros de áudio.

WaitExten() Aplicação utilizada para reproduzir ficheiros de áudio.

Hangup() – aplicação utilizada para desligar uma chamada.

Goto() – Aplicação que efectua um salto dentro do plano de chamadas.

Gotoif() – Aplicação que efectua um salto dentro do plano de chamadas associado a uma condição.

NoOp() – Aplicação que permite emitir mensagens na consola de linhas de comandos do Asterisk.

F. Variáveis de canal do Asterisk

As variáveis de canal são utilizadas pelo Asterisk para armazenar informação sobre os canais que estão a ser utilizados na realização de uma chamada. Abaixo é apresentada uma listagem parcial das variáveis de canal disponíveis pelo Asterisk. A listagem completa pode ser obtida através da Internet no site da Digium[31].

`${CDR(accountcode)}` - Account code (if specified)

`${CALLERID(name)}` - Caller ID Name only

`${CALLERID(num)}` - Caller ID Number only

`${CHANNEL}` - Current channel name

`${ENV(VAR)}` - Environmental variable VAR

`${EXTEN}` - Current extension

`${HANGUPCAUSE}` - Asterisk cause of hangup (inbound/outbound)

`${HINT}` - Channel hints for this extension

`${HINTNAME}` - Suggested Caller*ID name for this extension

`${INVALID_EXTEN}` - The invalid called extension (used in the “i” extension)

`${LANGUAGE()}` - Current language

`${LEN(VAR)}` - String length of VAR (integer)

`${PRIORITY}` - Current priority in the dialplan

`${STRFTIME(${EPOCH} ,,%Y%m%d-%H%M%S)}` - Current date time in the format:
YYYYMMDD-HHMMSS

`${TRANSFER_CONTEXT}` - Context for transferred calls

`\${FORWARD_CONTEXT}` - Context for forwarded calls

`\${UNIQUEID}` - Current call unique identifier

`\${SYSTEMNAME}` - value of the systemname option of asterisk.conf

G. Configuração do CDR

A criação da base de dados e tabela para armazenamento dos dados das chamadas foi feita através da importação de um ficheiro dentro do MySql.

Conteúdo do ficheiro criado com código SQL (**asterisk.sql**)

```
CREATE DATABASE asterisk;
```

```
GRANT INSERT  
ON asterisk.*  
TO asterisk@localhost  
IDENTIFIED BY 'cbr1100xx';
```

```
USE asterisk;
```

```
CREATE TABLE 'cdr' (  
'calldate' datetime NOT NULL default '0000-00-00 00:00:00',  
'clid' varchar(80) NOT NULL default '',  
'src' varchar(80) NOT NULL default '',  
'dst' varchar(80) NOT NULL default '',  
'dcontext' varchar(80) NOT NULL default '',  
'channel' varchar(80) NOT NULL default '',  
'dstchannel' varchar(80) NOT NULL default '',  
'lastapp' varchar(80) NOT NULL default '',  
'lastdata' varchar(80) NOT NULL default '',  
'duration' int(11) NOT NULL default '0',  
'billsec' int(11) NOT NULL default '0',  
'disposition' varchar(45) NOT NULL default '',  
'amaflags' int(11) NOT NULL default '0',  
'accountcode' varchar(20) NOT NULL default '',  
'uniqueid' varchar(32) NOT NULL default '',  
'userfield' varchar(255) NOT NULL default '');
```

Para melhor compreensão dos campos criados na tabela, é apresentado de seguida uma listagem dos campos e sua função.

- **calldate** - Identificação da data e hora a que decorreu a chamada
- **clid** - Identificação da chamada
- **src** - Identificação da origem
- **dst** - Identificação do destino
- **dcontext** - Contexto de destino
- **channel** - Canal utilizado
- **dstchannel** - Canal de destino
- **lastapp** - Última aplicação executada
- **lastdata** - Parâmetro utilizado em lastapp
- **duration** - Duração total da chamada em segundos
- **billsec** - Tempo desde o atendimento da chamada
- **disposition** - Resposta da chamada em segundos
- **amaflags** - Indica forma como os dados são recolhidos
- **accountcode** - Identificação da conta utilizada;
- **uniqueid** - Identificação única para cada chamada realizada.
- **userfield** - Campo personalizado pelo utilizador.

Bibliografia

- [1] A. Keller, *Asterisk na prática - Segunda Edição*. Novatec Editora Ltda.
- [2] C. S. Ribeiro, “Voz sobre ip: Tecnologias e protocolos?.” http://www.teleco.com.br/tutoriais/tutorialvoipconv/pagina_4.asp, 2011. [Recuperado em 2-Setembro-2012].
- [3] I. T. Union, “G.114 one-way transmission time.” <http://www.itu.int/rec/T-REC-G.114/en>, 2012. [Recuperado em 2-Setembro-2012].
- [4] M. B. S. K. S. M. Jonathan Davidson, James Peters, *Fundamentos do VoIP - Segunda Edição*. ARTMED EDITORA S.A.
- [5] W. E. Bruno Wanderley, “Voip e qos: Análise de parâmetros de qos para chamadas intercontinentais.” http://www.teleco.com.br/tutoriais/tutorialvoipqos/pagina_3.asp, 2008. [Recuperado em 2-Setembro-2012].
- [6] R. B. Leif Madsen, Jim Van Meggelen, *AsteriskTM: The Definitive Guide*. O'REILLY.
- [7] the xiph open source community, “speex.” <http://www.speex.org/>, 2012. [Recuperado em 2-Setembro-2012].
- [8] I. E. T. Force, “Transmission control protocol.” <http://www.ietf.org/rfc/rfc793.txt>, 1981. [Recuperado em 8-Setembro-2012].
- [9] I. E. T. Force, “User datagram protocol.” <http://www.ietf.org/rfc/rfc768.txt>, 1980. [Recuperado em 8-Setembro-2012].
- [10] I. E. T. Force, “Stream control transmission protocol.” <http://www.ietf.org/rfc/rfc4960.txt>, 2007. [Recuperado em 8-Setembro-2012].
- [11] S. L. Cechin, “Protocolo sctp stream control transmission protocol.” <http://www.inf.ufrgs.br/~cechin/Net/sctp/sctp.html>, ????. [Recuperado em 9-Setembro-2012].
- [12] P. Stalvig, “Introdução ao protocolo de controle de fluxo de transmissão (sctp), a próxima geração do protocolo de controle de transmissão (tcp).” <http://www.f5networks.com.br/pdf/white-papers/>

- `introducao-ao-stream-control-transmission-protocol-sctp-wp.pdf`, ??? [Recuperado em 9-Setembro-2012].
- [13] I. E. T. Force, “Sip: Session initiation protocol.” <http://www.ietf.org/rfc/rfc3261.txt>, 2002. [Recuperado em 8-Setembro-2012].
- [14] I. E. T. Force, “Rtp: A transport protocol for real-time applications.” <http://www.ietf.org/rfc/rfc1889.txt>, 1996. [Recuperado em 8-Setembro-2012].
- [15] I. E. T. Force, “Real time control protocol (rtcp) attribute in session description protocol (sdp).” <http://www.ietf.org/rfc/rfc3605.txt>, 2003. [Recuperado em 8-Setembro-2012].
- [16] I. E. T. Force, “Sdp: Session description protocol.” <http://tools.ietf.org/html/rfc2327>, 1998. [Recuperado em 8-Setembro-2012].
- [17] I. E. T. Force, “Real time streaming protocol (rtsp).” <http://www.ietf.org/rfc/rfc2326.txt>, 1998. [Recuperado em 8-Setembro-2012].
- [18] I. E. T. Force, “Resource reservation protocol (rsvp).” <http://www.ietf.org/rfc/rfc2205.txt>, 1998. [Recuperado em 8-Setembro-2012].
- [19] I. E. T. Force, “Media gateway control protocol (mgcp) - version 1.0.” <http://tools.ietf.org/html/rfc2705>, 1999. [Recuperado em 8-Setembro-2012].
- [20] I. E. T. Force, “Megaco protocol - version 1.0.” <http://tools.ietf.org/html/rfc3015>, 2000. [Recuperado em 8-Setembro-2012].
- [21] 3CX, “O que significam os termos fxs e fxo?.” <http://www.3cx.com.br/voip-sip/fixs-fxo.php>, 2012. [Recuperado em 27-Fevereiro-2012].
- [22] I. Wester, “Tecnologias de rede.” <http://www.infowester.com/isdn.php>, 2012. [Recuperado em 1-Março-2012].
- [23] J. V. Meggelen, L. Madsen, and J. Smith, *Asterisk, The Future of Telephony - 2nd Edition*. O'REILLY.
- [24] Digium, “About the asterisk project.” <http://www.asterisk.org/asterisk>, 2012. [Recuperado em 22-Fevereiro-2012].
- [25] Digium, “Architecture.” <http://www.asterisk.org/architecture>, 2012. [Recuperado em 22-Fevereiro-2012].
- [26] Voip-Info.org, “Asterisk cdr mysql.” <http://www.voip-info.org/wiki/view/Asterisk+cdr+mysql>, 2012. [Recuperado em 24-Março-2012].
- [27] VoIPango, “Solution: Ip pbx.” http://www.voipango.com/index.php?lang=1&listorderby=oxvarminprice&listorder=desc&lang=1&force_sid=803472b23a0ec4c9f10604f59f829965&cl=alist&searchparam=&cnid=01a46f7ce3226ace5.28752638, 2012. [Recuperado em 13-Maio-2012].

- [28] VoIPango, “Sangoma bri.” <http://www.voipango.com/en/ISDN-Interfaces/Sangoma/Sangoma-BRI/Sangoma-A500BRM-Basecard-2-24-Port-BRI-PCI.html>, 2012. [Recuperado em 22-Fevereiro-2012].
- [29] J. V. Meggelen, L. Madsen, and J. Smith, *Asterisk, The Future of Telephony - 2nd Edition*. O'REILLY.
- [30] Digium, “Dialplan applications.” <https://wiki.asterisk.org/wiki/display/AST/Dialplan+Applications>, 2012. [Recuperado em 2-Março-2012].
- [31] Digium, “Asterisk standard channel variables.” <https://wiki.asterisk.org/wiki/display/AST/Asterisk+Standard+Channel+Variables>, 2012. [Recuperado em 2-Março-2012].
- [32] Digium, “Glossary.” <http://www.asterisk.org/glossary>, 2012. [Recuperado em 2-Março-2012].
- [33] Voip-Info.org, “Asterisk.” <http://www.voip-info.org/wiki/view/asterisk>, 2012. [Recuperado em 13-Dezembro-2011].

