UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FAET – FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

PABX IP DE ALTA DISPONIBILIDADE

UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FAET – FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

PABX IP DE ALTA DISPONIBILIDADE

por

Flávio de Castro Carneiro
2006440-7 – FAET – UNICEUB

Trabalho Final de Graduação

Prof. Antonio José Gonçalves Pinto – M.Sc.
Orientador

Brasília/DF, junho de 2005.

Agradecimentos

Aos meus pais, Beatriz e Flávio Carneiro, pela compreensão, pelo apoio, e pela dedicação.

Aos meus avós Maria Almerinda e João Luiz Saraiva de Castro pelo amor, pela amizade e pelos belos exemplos que, cotidianamente, me transmitiram.

A minha irmã, Vivianna de Castro Carneiro, por sua amizade, paciência, pela descontração que sempre me proporcionou.

A minha amada namorada, Flávia de Carvalho Ferreira Leite, pela sua paciência, pelo apoio nos momentos difíceis e, principalmente, pelo carinho e pela tranqüilidade que tanto me ajudam no meu dia-a-dia.

Ao meu professor-orientador, António José Gonçalves Pinto, pelo conhecimento técnico transmitido, pela paciência e, principalmente, por sua orientação precisa, me direcionando sempre para o caminho correto.

A toda minha família, incluindo primos, tios e avós, pela convivência familiar, balizada pela união, pelo respeito e pela paz.

Aos meus amigos que me acompanharam na jornada acadêmica, em todos os momentos de alegria e tristeza, mas, acima de tudo, irmanados e confiantes.

Resumo

A Rede Mundial de Computadores surgiu sobre a telefonia e, sobre aquela, essa se desenvolve, minimizando os custos e multiplicando as possibilidades do seu emprego.

Este trabalho visa aumentar a confiabilidade de um sistema de telefonia IP, focando no *gerenciador de chamada* (callmanager) ou *Central Automatica de Comunicação Interna e Externa* (PABX , Private Automatic Branch eXchange) de alta disponibilidade. O projeto consiste na implementação de um ambiente de *voz sobre protocolo de Internet* (VoIP) com 2 callmanager em cluster, ativo-passivo, para aumentar a disponibilidade do ambiente.

Este projeto aborda as seguintes atividades: Implementação de uma rede VoIP utilizando o Protocolo de Inicialização de Sessão (SIP), a implementação de 2 callmanager em Cluster e testes de falhas nos callmanagers.

O ambiente desenvolvido neste projeto possui uma maior disponibilidade e um baixo custo se comparado com as outras soluções de PABX IP.

Palavras Chave: Disponibilidade de um ambiente VoIP.

Abstract

The World-wide Net of Computers appeared of the net of telephony and the net of telephony "*Voice over Internet Protocol'* (VoIP) if it develops on the computer network, minimizing the costs and multiplying the possibilities.

This work has as objective to increase the trustworthiness of a system of telephony IP, implementing the call manager in high availability. The project implements of a VoIP environment with 2 to call manager in to increase high availability to increase the trustworthiness of the environment.

This project approaches the following activities: Implementation of a VoIP environment using the protocol "*Session Initiation Protocol*" (SIP), Implementation of 2 to call manager in high availability and tests of imperfections in call managers.

Key Words: Availability of a VoIP environment.

Sumário

Lista de Figuras

Listas de Tabelas

LISTA DE ACRÔNIMOS

Asterisk - Software de PABX GNU

CallManager - Servidor de Gerência de Chamadas

CRC - Cyclic Redundance Check

CRLF - Carriage Return, Line Feed

CSRC - Contributing Source

CVS - Sistema de Controle de Versões

DHCP - Dynamic Host Configuration Protocol

DS - Differentiated Services

FEC - Forward Error Correction

GPL - General Public License

HTTP - Hypertext Transfer Protocol

IETF - Internet Engineering Task Force

IIS - Internet Integrated Services

IP - Internet Protocol

Ipv4 - Internet Protocol 4

ITU-T – International Telecom Union

LAN - Local Area Network

MGCP - Media Gateway Control Protocol

MMUSIC - Multiparty Multimedia Session Control

NTP - Network Time Protocol

PABX - Central Automatica de Comunicação Interna e Externa

PCM - Pulse Code Modulation

PSTN - Public Switched Telephony Network

QoS - Quality of Service

RFC - Request For Comments

RSVP - Resource Reservation Protocol

RTP - Real-Time Transport Protocol

SDP - Session Description Protocol

SIP - Session Initiation Protocol

SS7 - Signaling System 7

SSRC - Synchronization Source

TCP - Transmission Control Protocol

ToS - Type of Sevice

UDP - User Datagram Protocol

VoIP - Voz sobre IP

WEB - World Wide Web

Capítulo 1 - Introdução

1.1. MOTIVAÇÃO

As afirmações de John Hall, presidente da *Open-Source Linux International*, listadas a seguir constituíram o cerne da motivação desse trabalho:

"Eu acredito que nos próximos três anos, VoIP usando soluções Open Source, como o Asterisk, irão gerar mais negócios que todo o mercado Linux de hoje."

"Hoje as soluções PABX são incrivelmente caras, fechadas e proprietárias. O Asterisk é aproximadamente um décimo do preço de um PABX proprietário."[Jon Hal, 2004][1]

Figura 1-1 - John Hall

A tecnologia de telefonia *"Internet protocols"* Protocolo de Internet (IP) tem despertado o interesse de várias empresas devido à redução de custo, segurança, mobilidade, aplicações avançadas, escalabilidade e outras facilidades. Porém, a sua rápida adoção tem encontrado obstáculos devido aos problemas associados à falta de confiança na disponibilidade do serviço. Existem vários estudos para melhoria da qualidade da voz, mas quase não há pesquisas na área de disponibilidade do sistema.

Para muitas empresas, a parada parcial ou total do sistema de telefonia gera prejuízos e até mesmo a parada da empresa.

1.2. OBJETIVOS E BENEFÍCIOS DO TRABALHO

Este trabalho visa atender a uma necessidade do mercado. Com o crescente interesse na tecnologia VoIP, cresce, também, a necessidade de aumentar a qualidade e a disponibilidade do serviço, o que demanda um grande investimento. Esse trabalho aborda a implementação de um ambiente VoIP de baixo custo que oferece um aumento na disponibilidade do serviço.

1.3. ABRANGÊNCIA

Para o desenvolvimento do projeto será implementado uma rede VoIP utilizando o protocolo de Inicialização de Sessão (SIP). A rede será constituída de dois callmanagers utilizando o sistema Asterisk em alta disponibilidade. O Asterisk é um software livre que roda em uma plataforma Linux, neste caso, o Suse 9.2.

O protocolo utilizado é o SIP. As redes VoIP utilizam protocolos de controle de sinalização que tem por função negociar o início e o fim de uma transmissão, a codificação de áudio, a localização dos usuários, o redirecionamento de mensagens, entre outras sinalizações, que possibilitam a transmissão de voz sobre IP.

[1] JON Hall 06/04/2004 **DZNet Inglaterra**

Para esta função será utilizado o Protocolo de Inicialização de Sessão (Session Initiation Protocol – SIP) que está definido na “Request For Comments” (RFC) 3261 do grupo de trabalho “Multiparty Multimedia Session Control” (MMUSIC) do “Internet Engineering Task Force” (IETF).

Os Callmanagers são servidores responsáveis por receberem e encaminharem os pedidos. Eles podem ou não mudar os parâmetros da mensagem antes de passar adiante e também podem decidir mandar uma resposta ao cliente gerada através de funções implementadas no servidor.

Para solucionar o problema de disponibilidade foi utilizado um “cluster” entre os dois servidores Linux.

1.4. ESTRUTURA DO TRABALHO

Este trabalho está divido em forma de capítulos descritos a seguir:

O primeiro capítulo expõe uma justificativa da origem deste trabalho, os seus objetivos, como também, uma síntese do que será tratado no desenvolvimento.

No segundo, explana a Telefonia IP, suas funcionalidades, bem como a sua aplicação no contexto das redes de computadores.

O terceiro capítulo apresenta os conceitos, padrões e procedimentos utilizados pelo protocolo SIP de Telefonia IP.

No quarto, apresenta a especificação, as funcionalidades e configurações feitas para o desenvolvimento do servidor de Callmanager, utilizando o Asterisk.

O quinto capítulo aborda a implementação, configuração e o monitoramento do anbiente de alta disponibilidade.

No sexto, será feita uma análisa dos resultados obtidos durante a implementação do ambiente.

Capítulo 2 - Telefonia IP

2.1. HISTÓRICO DA TELEFONIA

Os serviços telefônicos estão evoluindo constantemente. Nos anos 50, a introdução de cabos transatlânticos possibilitou as ligações internacionais; nos anos 60, as centrais e transmissões digitais melhoraram o sinal de áudio; os serviços de chamadas em espera e de discagem por tons foram viabilizados na década de 70, pelas centrais programáveis; o sistema de sinalização em canal comum como o "Signaling System 7" (SS7) possibilitou serviços com números 0800. Os anos 90 foram marcando definitivamente a trajetória de transmissão e sinalização telefônica analógica e uma infra-estrutura baseada em redes de pacotes. [FERNANDES, 2003] [2]

As redes telefônicas "Public Switched Telephony Network" (PSTN) constituem a maioria das redes telefônicas, tendo poucas alterações em relação aos equipamentos de trasmissão. Sua principal característica é estabelecer um circuito entre dois assinantes para que possa haver comunicação. [ALENCAR, 1998][3]

As redes de computadores são redes baseadas em pacotes e tem tido um grande avanço em relação aos equipamentos de transmissão de dados, possibilitando usar esta mesma rede para transferência de voz. Com isto, houve um crescente número de aplicações voltadas para a transferência de voz sobre os protocolos de redes, chamadas Voz sobre IP (VoIP). [ALENCAR, 1998][4]

Segundo Hersent, A tecnologia de VoIP é a área que mais cresce no setor de telecomunicações superando a taxa de crescimento de telefonia móvel. A tecnologia atrás da telefonia IP pode parecer trivial, no entanto ela é muito mais complexa do que a transmissão unidirecional usada na transmissão de TV ou de rádio nas redes de computadores, pois a taxa de transferência entre o locutor e o ouvinte deve permanecer constante e baixa. [HERSENT, 2002][5]

2.2. O QUE É VoIP?

VoIP é a sigla de "Voz sobre IP" (Voice over IP). É um termo utilizado para caracterizar o serviço que consiste em transmitir informação de voz através do Protocolo IP (Internet Protocol) - TCP ou UDP.

De uma forma geral, significa enviar informação de voz em formato digital dentro de pacotes de dados, ao invés do tradicional protocolo de comutação de circuitos utilizado há décadas pelas companhias telefônicas.

A maior vantagem da tecnologia VoIP é a redução dos custos de utilização dos serviços de telefonia comum, principalmente em ambientes corporativos. As redes de

[2] FERNANDES, Nelson Luiz Leal.**Voz sobre Ip**
[3] ALENCAR, Marcelo Sampaio de. **Telefonia digital**.
[4] ALENCAR, Marcelo Sampaio de. **Telefonia digital**.
[5] HERSENT, Oliver. **Telefonia IP.**

dados já instaladas passam também a transmitir voz e, dessa forma, os custos podem ser reduzidos independente do dia da semana, da hora e duração da chamada.

Diversas empresas já estão oferecendo Voz sobre IP corporativo. Companhias com filiais em qualquer parte do mundo já estão falando entre si com o custo zero ou quase isso. Esses benefícios não se restringem ao segmento empresarial. A cada dia, aumenta o número de usuários domésticos a descobrir as vantagens de se utilizar VoIP para a redução de custos em chamadas.

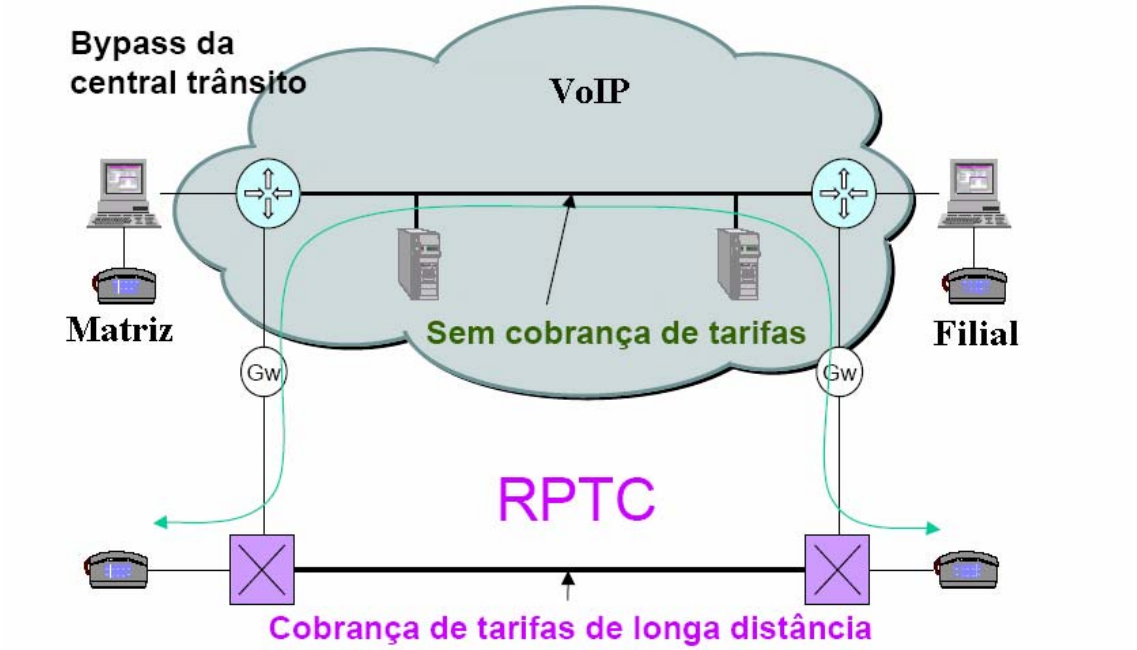

Figura 2-1 - Topologia VoIP e PSTN.

Na figura 2-1, temos uma topologia de implementação de uma rede VoIP em paralelo de uma rede de telefonia convencional. As chamadas de longa distância são redirecionadas para a rede IP, com isso não há bilhetagem nessas ligações.

Serviços como ligações para telefones fixos e móveis da rede pública, FAX e até para vigilância remota podem ser implementados através de um “gateway” de rede como o Asterisk, esse serviço também pode ser contratado junto aos provedores VoIP.

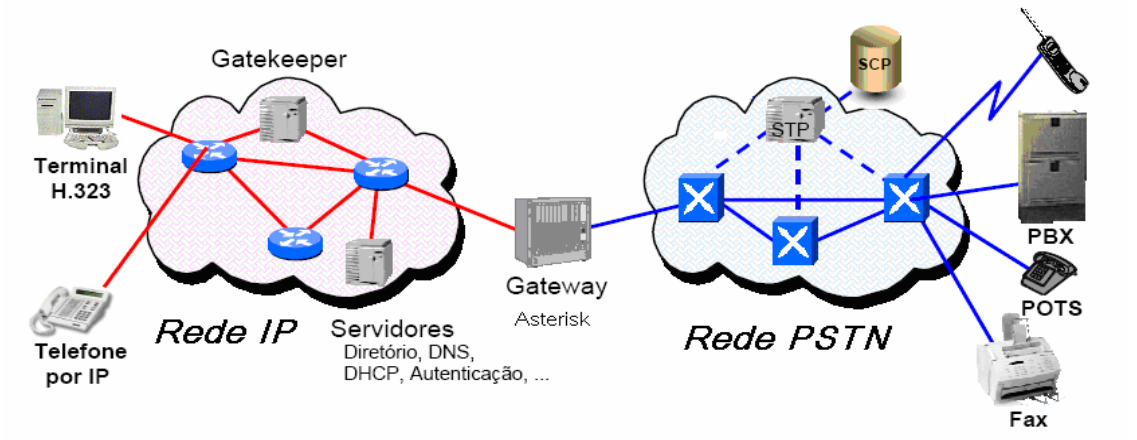

Figura 2-2 - Interligação da rede VoIP com a rede PSTN

2.3. CODIFICAÇÃO E TRANSPORTE DA VOZ

A técnica de codificação “Pulse Code Modulation” (PCM), modulação por codificação de pulsos, que consiste em 8000 amostragens do sinal de voz contínuo por segundo, representa um valor amostrado de 8 bits, o que, para a transmissão de cada canal de voz, implica na necessidade de um canal digital de 64 kbps. Este tipo de codificação possui baixo atraso e pequena complexidade, mas requer uma taxa de transmissão elevada. [FERNANDES, 2003][6]

A conversação humana é uma forma de onda com freqüências principais na faixa que vai de 300 Hz a 3.4 kHz, com alguns padrões de repetição devido ao timbre de voz e aos fonemas emitidos durante a conversação. O problema da telefonia em geral é reproduzir a qualidade da voz humana em um terminal à distância. [SOUZA, 2001][7]

Com a telefonia digital, a voz é codificada em formato digital, que pode ser multiplexado no tempo, de forma a compartilhar o meio de transmissão. Esse fluxo de bits é encapsulado em datagramas UDP que são encapsulados em pacotes IP como mostra a figura abaixo.

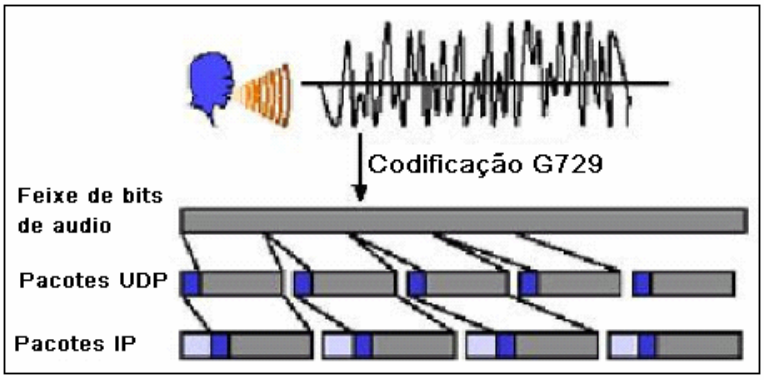

Figura 2-3 - Codificação VoIP. [ALVES, 2002]

[8]

Para os modelos de codificação de voz, foram desenvolvidas novas técnicas de codificação. Estas técnicas fazem a segmentação de sinal analógico em intervalos periódicos.

Segundo Alves, não é suficiente pegar amostras do sinal digital e colocá-los no pacote que será enviado a rede. Para que seja possível transmitir um pacote de voz numa rede de dados, deve-se levar em conta os seguintes fatores: [ALVES, 2002][9]

a) A construção do pacote não é feita em tempo nulo. No caso de amostras de 8 khz, é necessário esperar um tempo até que se tenha uma quantidade suficiente para colocar em um pacote de dados. Neste caso, está se introduzindo um atraso na comunicação que é proporcional ao tamanho do pacote.

---

[6] FERNANDES, Nelson Luiz Leal.**Voz sobre Ip**
[7] SOUZA, José Marcio de. Protótipo de um sitema de VoIP(Voz sobre IP).
[8] ALVES,Victor Manuel Golçalves.**Apresentação e análise da Ip Telephony,**
[9] ALVES,Victor Manuel Golçalves.**Apresentação e análise da Ip Telephony**,

b) Nos equipamentos de pacotes numa rede IP (routers), existem filas de espera. Se um pacote de voz, sensível às variações de atraso, for colocado numa fila desses equipamentos atrás de um número variável de pacotes que não têm esses requisitos, podem ocorrer variações no atraso (jitter), resultando na distorção do sinal;

c) O valor mais comum de normas de digitalização de voz é de 64 kbps. A unidade básica para uma largura de banda de parte dos routers é de 64 kbps, com isso, os pacotes de voz podem saturar estas ligações, sem contar os bits dos cabeçalhos dos pacotes de voz introduzidos pela pilha de protocolos;

d) A norma Modulação por Código de Pulso (PCM) de 64kbps é de utilização constante, mesmo quando há pausa na conversação; enquanto que redes de dados são concebidas para tratar tráfego com características de rajada. Se o tráfego de silêncio não for enviado, caracterizando o modo rajada, pode trazer desconforto ao receptor, ficando uma sensação de que a chamada foi interrompida;

e) O protocolo TCP/IP é um protocolo orientado à conexão, que caracteriza o reenvio de pacotes perdidos, o que em uma conversação em tempo real não pode acontecer, pois redunda em perdas no diálogo. Contudo, existem mecanismos que tentam ultrapassar essas questões:

I) redução dos tamanhos dos pacotes IP;

II) uso de diferentes prioridades e pacotes com diferentes requisitos;

III) o recurso de compressão, para que o sinal tenha um débito menor que os 64Kbps, pois os pacotes de voz ficam com características mais apropriadas para serem transportados;

IV) introduzir, no lado do receptor, ruídos para atenuar o efeito de perda do som quando há pausas no diálogo;

V) utilizar protocolos mais adequados para transmissão de voz como o “Real-Time Transport Protocol” (RTP), que é semelhante ao TCP, mas não tem reenvio de pacotes.

2.3.1.PROCESSO DE CODIFICAÇÃO

Segundo Oliver, a conversão analógica digital é o processo de representar um sinal por um conjunto finito de números a uma taxa de amostragem constante. Em outras palavras, consiste em garantir um sinal analógico a uma taxa constante. [OLIVER, 2002][10]

[10] OLIVER, Sérgio, 2002. **Telefonia IP para ambientes móveis usáveis**

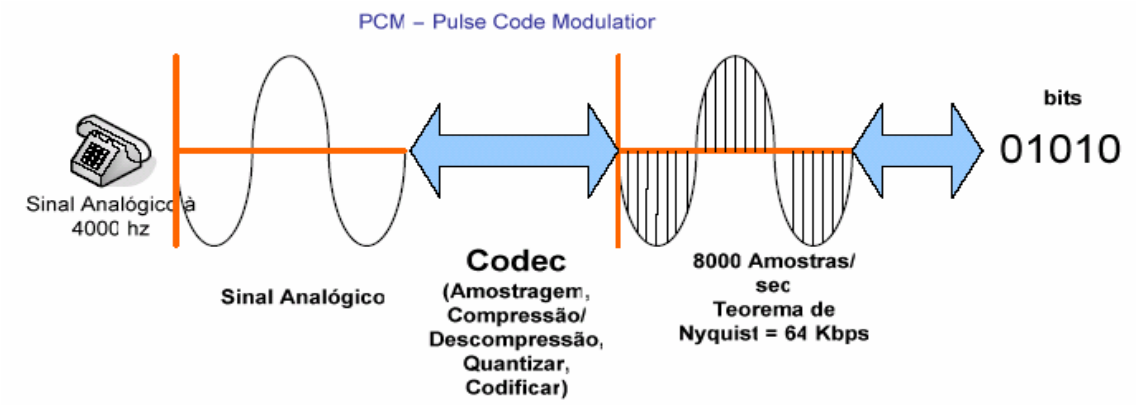

Figura 2-4 - Analógico Digital (Teorema de Nyquist)

O fator taxa de amostragem pode causar um grande impacto na relação qualidade e taxa de transmissão por segundo. Se aumentarmos a taxa de amostragem, melhoramos a qualidade de representação do sinal, porém será necessária uma maior largura de banda para transmisão em tempo real. Por outro lado, se diminuir a taxa de amostragem, a qualidade do sinal diminui, diminuindo também, a quantidade de banda necessária para transmissão em tempo real.

As figuras seguintes ilustram o principio da amostragem :

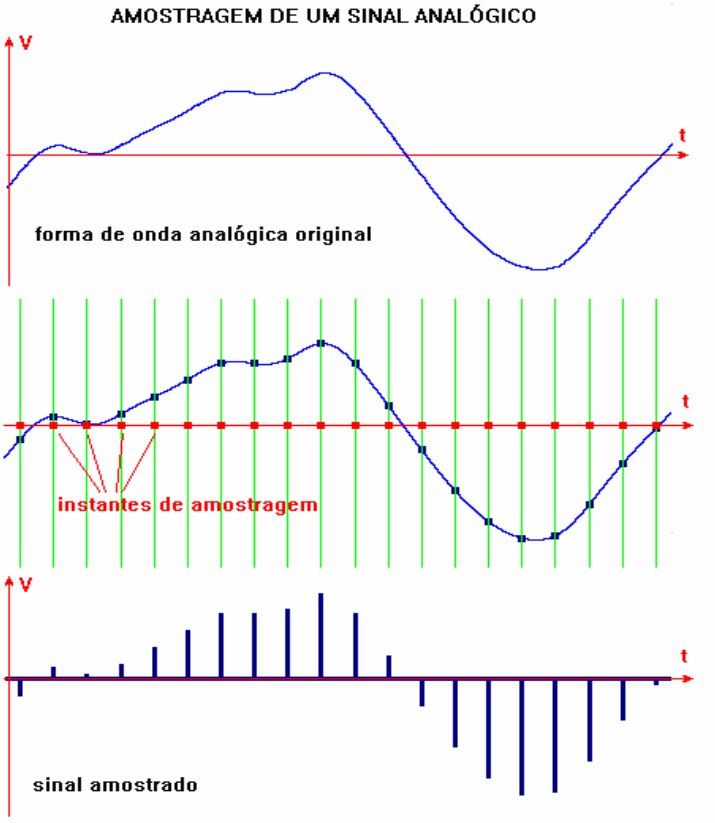

Figura 2-5 - Codificação da voz sinal amostrado

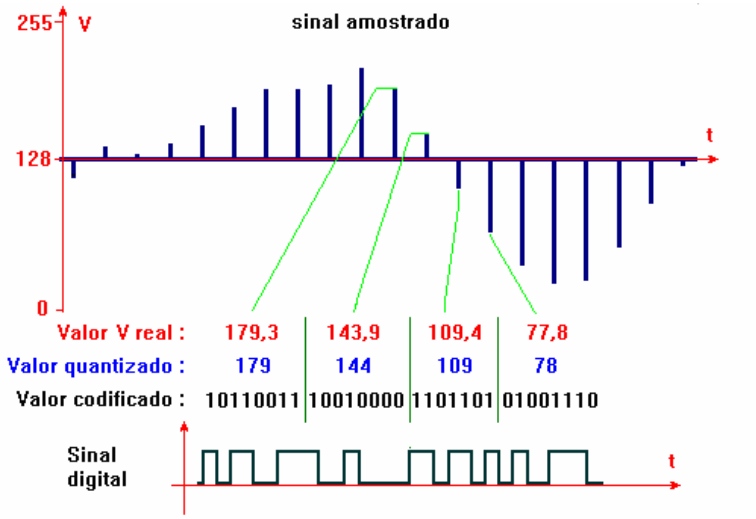

Figura 2-6 - Codificação de sinal amostrado para sinal digital

Como o sinal analógico é continuo no tempo e em nível, contém uma infinidade de valores e, como o meio de comunicação tem banda limitada, obriga a transmissão de apenas uma certa quantidade de amostras deste sinal, como enunciado no Teorema de Nyquist.

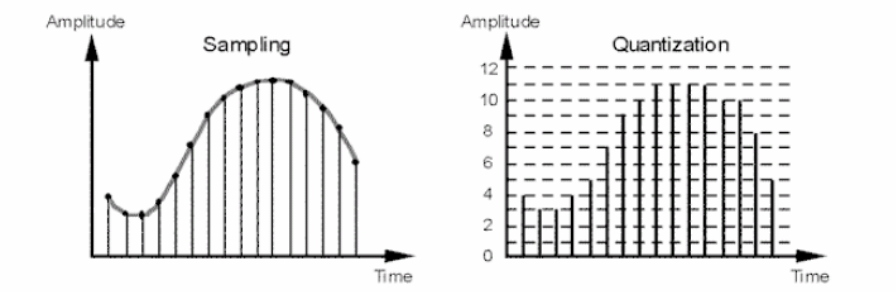

Figura 2-7 - Quantização

É obvio, que quanto maior for a freqüência de amostragem, mais fácil será reproduzir o sinal original, mas haverá desperdício de banda ocupada sem nenhuma melhoria sensivel na qualidade da voz.

Tipos	Largura de banda de transmissão	Freqüência de amostragem	Taxa de bits em Kbits/s	Principais aplicações
Fala telefônica	300 –3400 Hz	8 kHz	96 ou 104	Redes PSTN e ISDN, telefones celulares digitais.
Transmissão de voz e áudio em banda larga	50 – 7000 Hz	16 kHz	224 ou 240	Vídeo e áudio-conferência, rádio FM.
Transmissão de voz e áudio de alta qualidade	30 -15000Hz	32 kHZ	512	Som digital para TVs analógicas (NICAM)
	20 -20000Hz	44.1 kHZ	706	Áudio CD Player
	10 -22000Hz	48 kHZ	1152	Áudio profissional

Tabela 2-1 Faixas regulamentadas para Transmissão de sinais.

2.3.2.AMBIENTE PARA VoIP

Para que se possa oferecer o mínimo de qualidade na telefonia IP, tem que se atentar para alguns fatores de transmissão de voz.

2.3.3.CONSUMO DE BANDA

Para cada codificação, há uma largura de banda. Esse fator deve ser bem analisado quando se faz a escala de codificação que será utilizada, uma vez que vários canais de voz compartilham o mesmo canal digital (a rede).

As técnicas mais usadas para minimizar o requisito de banda são a compressão de cabeçalhos dos pacotes IP e a supressão de silêncio. [FERNANDES, 2003][11]

2.3.4.COMPRESSÃO DE CABEÇALHO DOS PACOTES

As soluções de VoIP utilizam o "Real-Time Transport Protocol" (RTP) e o "User Datagram Protocol" (UDP). Para um pacote de voz, somente o cabeçalho ocupa 40 bytes. Uma transmissão utilizando a implementação de codificação de voz G.729 com um pacote formado por dois quadros de amostragem tem 20 bytes de informação transmitida. Com isso, fica evidente o despropósito na distribuição de bytes de controle. Utilizando a técnica descrita na "Request for Commenst 2508" (RFC), a maioria dos cabeçalhos terão seus tamanhos de 2 ou 4 bytes, dependendo do uso ou não do "checksum" pelo UDP. Sabendo que, pós a transmissão do primeiro pacote descomprimido, outros pacotes subseqüentes poderão ser transmitidos suprindo seus cabeçalhos, sendo remontados no destino.

2.3.5.SUSPRENSSÃO DO SILÊNCIO

Durante uma ligação VoIP há vários períodos de tempo onde não há conversa. Ao utilizar o codec de VoIP, há uma supressão desses períodos de tempo. Essa redução pode chegar a 25% da transmissão total.

2.3.6. ATRASO

Atraso é o tempo entre o envio e a chegada do pacote em seu destino. Esse atraso não pode ultrapassar os valores aceitáveis para esse tipo de transmissão, podendo comprometer à qualidade da voz.

Para que se tenha uma transmissão com pequenos atrasos, deve-se levar em consideração a disponibilização de recursos em que a aplicação está inserida. A interatividade entre usuários da aplicação deve ser considerada ao avaliar uma transmissão.

[11] FERNANDES, Nelson Luiz Leal.**Voz sobre Ip.**

O tempo entre a geração do pacote e a entrega deve estar entre 200 e 300ms. Nas aplicações onde à informação é transmitida em um único sentido, a Norma G.114 do ITU-T coloca que o intervalo é de 150 e 400ms, mas deve ser avaliado o impacto na qualidade da transmissão, sendo que atraso superior a 400ms é inaceitável. [FERNANDES, 2003][12]

Atraso = Tempo de propagação + Tempo de transmissão + Tempo de codificação

2.3.7.JITTER

O "Jitter" é a variação do tempo entre a chegada de pacotes consecutivos. Diferentes tempos de propagação podem ser causados pelos datagramas terem tomados caminhos diferentes na rede, como também, terem sofrido congestionamento. Nas entradas dos equipamentos decodificadores, são usados "buffers" que armazenam em uma fila os pacotes que estão chegando, para poderem compensar maiores atrasos, sempre dentro de um limite determinado pelo tamanho do buffer.

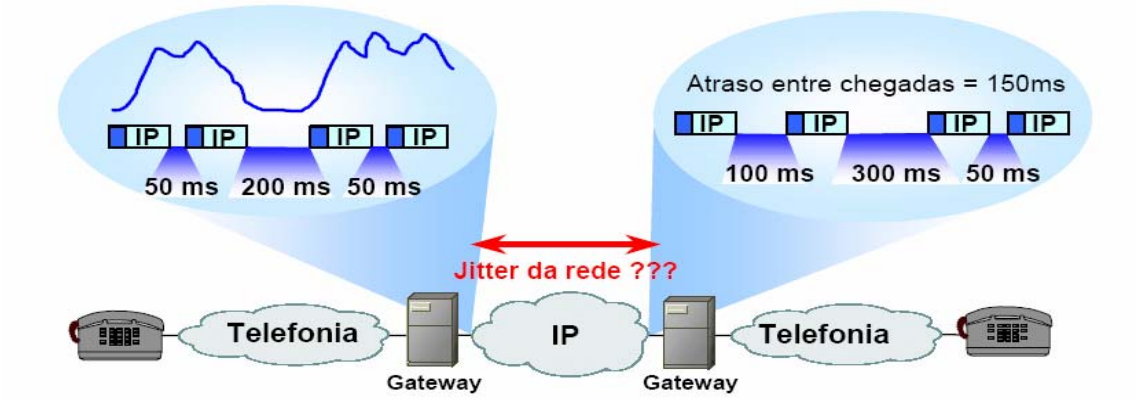

Figura 2-8 – A Figura mostra a introdução de jitter em uma transmissão de voz.

2.3.8.TAXA DE PERDAS E ERROS

Segundo Fernandes, a rede de telefonia IP, tem a transmissão em tempo real, não há como reenviar pacotes perdidos ou com erros para garantir uma boa qualidade na transmissão. Uma alternativa seria o uso de algoritmos "Forward Error Correction" (FEC), onde o mesmo pacote IP conteria vários quadros de voz implicando um uma redundância de quadro, sendo que só se aplica para codificação que geram pouco atraso, já que a formação de um pacote poderia tornar a solução inviável. [FERNANDES, 2003][13]

2.3.9.QUALIDADE DE SERVIÇO (QOS)

Quando se fala em garantia de "Quality of Service" (QoS) para aplicações de voz sobre IP, refere-se à garantia do meio de transmissão com banda suficiente para a transferência do sinal de voz, com um atraso e jitter mínimos. Para que seja possível, um conjunto de mecanismos são implementado junto ao protocolo IP, já que ele em sua concepção não foi desenvolvido com esta finalidade.

---

[12] FERNANDES, Nelson Luiz Leal.**Voz sobre Ip**.
[13] FERNANDES, Nelson Luiz Leal.**Voz sobre Ip**.

Diversas arquiteturas que provêm serviços diferenciados de Internet está sendo pesquisadas nos últimos anos. O Internet Engineering Task Force (IETF) aborda duas formas diferentes o Internet Integrated Services (IIS) e o Differentiated Services (DS).

O IIS tem por característica o foco em um fluxo individual de pacotes entre os pontos de origem e destino. Nesta abordagem, cada fluxo pode requisitar diferentes níveis de serviço definindo a banda mínima necessária para a transmissão e o atraso máximo de tolerância. O IIS é composto por quatro componentes básicos:

a) unidade de controle de admissão, que identifica se a rede pode suprir os níveis mínimos necessários;

b) unidade de classificação, que inspeciona os campos dos pacotes para determinar suas classes e o nível de serviço adotado;

c) unidade de schedule, que aplica um ou mais mecanismos de gerência de tráfego para garantir que o pacote seja transmitido à rede a tempo de satisfazer a banda e atrasos adequados ao tipo de fluxo;

d) o protocolo Resource Reservation Protocol (RSVP), que é o protocolo para reserva recursos.

O RSVP está presente na maioria dos roteadores atuais. Um cliente RSVP pode reservar uma quantidade de banda necessária para prover o tráfego de modo que tenha um baixo atraso para os pacotes de voz. O RSVP é capaz de comunicar a reserva a outros roteadores RSVP. O cliente manda uma mensagem “path” indicando que quer reservar o recurso e o receptor da mensagem manda uma mensagem reservation-request pelo mesmo caminho que a mensagem passou, no entanto esta capacidade tem um impacto direto no desempenho dos roteadores, pois cada roteador tem que manter o estado do fluxo. Como roteadores transportam um número muito elevado de fluxo, o processamento desses estados acarreta na sobrecarga dos roteadores.

Segundo Fernandes, o DS não tem o foco no tipo de fluxo, mas sim nas aplicações que utilizam os níveis de qualidade de serviço semelhante, ou seja, classificam-se os diferentes tipos de tráfegos para determinar que aplicações utilizarão, seguindo o seguinte grupo de classes: [FERNANDES, 2003][14]

a) **Serviço Premium**, para aplicações que necessitam de atrasos e jitter pequenos;

b) **Serviço Garantido**, para aplicações que necessitam um serviço melhor que o “best-effort”.

c) **Serviço Olímpico** (subdividido em três subclasses: ouro, prata, bronze).

O protocolo IP versão 4 (IPv4) implementa esta classificação através do campo “Type of Sevice” (ToS) do cabeçalho IP. Sendo que os três primeiros bits do campo são usados para identificar a importância do pacote: quanto maior o campo maior a prioridade do pacote.

[14] FERNANDES, Nelson Luiz Leal.**Voz sobre**.

2.3.10.PROTOCOLOS

Os protocolo utilizados para o funcionamento do VoIP são os H.323, MGCP, SIP e outros. Eles têm a função de converter sua voz em dados digitais e compactá-la dentro do protocolo TCP/IP. Codificam a voz com G.711, G.723.1, G.729 e a enviam para o destino; chegando lá os dados são descompactados e convertidos para som digital, de modo que você pode estabelecer uma comunicação com outra pessoa em qualquer lugar que possua rede de dados. Esse tópico será bem detalhado durante o capítulo 3.

2.3.11.CODECS

CODEC são protocolos adicionais, cuja função é de controle de qualidade para comunicação. Em outras palavras, são “codificadores de áudio” que convertem e codificam a voz, transformando o pacote de dados em sinal digitalizado. A estrutura do VoIP se baseia nos CODEC G.729, G.723, G.711 e outros, junto com protocolo “Real Time Protocol” (RTP).

Nesse processo encontram-se duas fases: a análise e a sintetização da voz.

A a**nálise da voz** é à parte do processamento da voz que converte o som (voz) para o formato digital, para que seja armazenada apropriadamente nos sistemas de comutação e transmitida em redes digitais ou rede IP. Também chamada de “digitalspeech encoding”.

A s**intetização da voz** é à parte do processamento de voz que converte a voz da forma digital para forma analógica, própria para a audição humana. Também chamada de “speech decoding”.

O CODEC G.729 pode chegar a utilizar apenas 8 Kbps de banda independente do link, desde que exista capacidade de processamento em tempo real para isso, fazendo com que uma ligação VoIP consuma no máximo de 18 a 23Kbps, incluindo tempo de buferização, compressão CRTP e supressão de silêncio.

2.3.12.TCP/IP

A arquitetura “Internet Transmission Control Protocol/Internet Protocol” (TCP/IP) dá uma ênfase toda especial à interligação de diferentes tecnologias de redes. A idéia é que não existe nenhuma tecnologia que atenda aos anseios de todos os usuários, uns querem uma rede de alta velocidade, mas com curto alcance e outros admitem uma rede de baixa velocidade mas com logo alcance, portanto a forma que se pode obter isto é ligar todos os usuários a uma inter-rede.

O “Internet Protocol” (IP) foi projetado para permitir a interconexão de redes de computadores que utilizam a tecnologia de comutação de pacotes. O ambiente das redes consiste em “hosts” conectados entre si e por sua vez conectados a outras redes através de “gateways”. As redes podem variar de redes locais até redes de longa distância.

Segundo Soares (1995), o protocolo IP é um protocolo sem conexão. Tem por função transmitir datagramas de dados de um host origem para um host destino que são localizados através de seus endereços IP. Outro serviço que o IP fornece é a fragmentação e remontagem de pacotes cujo tamanho ultrapassa o máximo permitido para quadros da camada de acesso ao meio. [SOARES, 1995][15]

Como o serviço oferecido pelo IP é sem conexão, os datagramas transmitidos são tratados como unidades independentes, ou seja, cada pacote IP é independente do outro e não é feita nenhuma checagem fim-a-fim ou entre nós intermediários. O único tipo de verificação de erros que é feito é o "Cyclic Redundance Check" (CRC) que garante que as informações estão corretas. [SOARES, 1995][16]

O "Transmission Control Protocol" (TCP) é um protocolo orientado à conexão que fornece um serviço confiável de transferência de dados fim-a-fim.

O TCP interage de um lado com os processos das aplicações e do outro com o protocolo de internet. Tem por função atender as aplicações semelhantes às chamadas que os sistemas operacionais fazem aos processos de aplicação, como abrir e fechar conexões e enviar e receber dados de conexões já estabelecidas.

O TCP é capaz de transmitir uma cadeia de dados nas duas direções e geralmente é ele quem decide a hora de transmitir uma cadeia de dados e também de parar de transmitir. Uma exceção é quando recebe uma ordem explícita da aplicação para transmitir imediatamente os dados que estão nos seus buffers.

Segundo Soares, como o TCP não exige um serviço de rede do protocolo de internet confiável para operar, ele garante a recuperação de dados corrompidos, perdidos, duplicados ou entregues fora de seqüência pelo protocolo de Internet: garante a qualidade de serviço (QoS) da comunicação. [SOARES, 1995][17]

Para que vários processos possam transmitir simultaneamente, o TCP usa o conceito de porta, ou seja, para cada processo de aplicação que está utilizando o TCP atribuí uma porta diferente. No entanto, aos processos de aplicações mais usados, como Protocolo de Transferência de Arquivos (FTP) e Telnet, são atribuídos portas fixas. [SOARES, 1995][18]

Na Figura 2-9, pode-se observar o posicionamento do TCP na arquitetura internet TCP/IP.

---

[15] SOARES, Luiz Fernando G; LEMOS, Guido; COLHER, Sérgio. **Redes de computadores**

[16] SOARES, Luiz Fernando G; LEMOS, Guido; COLHER, Sérgio. **Redes de computadores**.

[17] SOARES, Luiz Fernando G; LEMOS, Guido; COLHER, Sérgio. **Redes de computadores**.

[18] SOARES, Luiz Fernando G; LEMOS, Guido; COLHER, Sérgio. **Redes de computadores**.

Figura 2-9 - Camadas de protocolos da arquitetura Internet TCP/IP

Pode-se considerar o TCP/IP como sendo um conjunto de protocolos de comunicação, utilizado tanto em redes locais como em redes externas.

Segundo Soares, outra opção de protocolo é o "User Datagram Protocol" (UDP) - uma extensão do IP - que fornece um serviço de datagrama não-confiável. O UDP recebe os pedidos de transmissão de uma estação- origem e os entregar ao IP responsável pela transmissão. Também ocorre o processo inverso: recebe mensagens do IP e as entrega aos processos das aplicações. [SOARES, 1995][19]

A arquitetura TCP/IP está dividida em quatro níveis:

a) **nível de acesso ao meio**: possui os protocolos de nível 1 e 2 do modelo "Open System Interconnect" (OSI), que carregam a informação entre pontos de uma rede;

b) **nível de Internet**: tem-se o roteamento dos dados na rede efetuado pelo protocolo IP;

c) **nível de transporte**: onde atuam os protocolos TCP e UDP que pegam os dados roteados pelo protocolo IP no nível anterior e transmitem para o nível superio;

d) **nível de aplicação**: nesse nível se encontram os protocolos de aplicação.

Entre os principais protocolos de aplicação, se destacam:

I) **File Transfer Protocol** (FTP) - protocolo que faz a transferência de arquivos entre computadores;

II) **Simple Mail Transfer Protocol** (SMTP) - protocolo de correio eletrônico;

III) **Simple Network Management** (SNMP) - protocolo de gerenciamento da rede;

IV) **Terminal Emulation** (TELNET) - emula terminais para acesso a sistemas remotos.

[19] SOARES, Luiz Fernando G; LEMOS, Guido; COLHER, Sérgio. **Redes de computadores**.

V) **Hipertext Transfer Protocol** (HTTP) - protocolo de navegação da Internet.

2.4. SERVIÇOS OFERECIDOS

O fato da arquitetura do VoIP oferecer um grande numero de serviços, a torna muito procurada. As redes telefônicas em crescimento estão se deparando com as alterações nas exigências de comunicação e com expectativas cada vez maiores.

2.4.1. INTEGRAÇÃO

A tecnologia VoIP permite integrar os principais meios de comunicação, tais como: FAX, E-Mail e Telefone

2.4.2. FACILIDADE DE IMPLEMENTAÇÃO

Para interligar novas filiais, basta contar com uma rede de dados, não necessitando montar uma nova estrutura de telefonia ou ramais.

2.4.3.INFRA-ESTRUTURA ÚNICA

As empresas e os usuários podem reduzir o custo total de propriedade da rede, migrando a infra-estrutura de dados e de voz para uma rede IP. Com uma rede convergente, as empresas têm apenas uma rede para gerenciar e um único sistema a ser utilizado pelos funcionários técnicos e usuários finais.

2.4.4.DESVIO DE TARIFAS

O desvio de tarifas permite um grande volume de chamadas de longa distância sem grandes alterações tarifárias.

2.4.5.MOBILIDADE DO FUNCIONÁRIO

A comunicação IP permite que os funcionários trabalhem independentemente do escritório ou da localização onde se encontram. Todas as chamadas recebidas e todos os recursos (discagem rápida, toque, preferências de volume e correio de voz) ficam imediatamente disponíveis, assim que um usuário se conecta a rede.

2.4.6. GERENCIAMENTO FLEXÍVEL

Permite uma administração totalmente remota possibilitando unificar a área de suporte a rede. A CISCO enumerou 700 vantagens de se migrar para uma solução VoIP. [Cristiane, 2003][20]

2.4.7. SEGURANÇA

Todas as chamadas podem ser Criptografadas.

[20] CRISTIANE Ligabue, **Voz IP Revolução na Telefonia**

2.5. DIFICULDADE DE IMPLANTAR UM AMBIENTE VOIP

Apesar do rápido crescimento da telefonia IP, ela tem encontrado vários obstáculos para a sua implementação, tais como:

a) custo ainda elevado dos telefones IP;

b) dificuldades para implementação de QOS;

c) maiores cuidados com a performance da rede; e

d) disponibilidade do ambiente (Foco deste trabalho).

2.6. CRESCIMENTO DA REDE VOIP NO BRASIL

No Brasil, a tecnologia VoIP ainda não foi homologada. O VoIP ainda está pouco difundido por motivos não comentados pela Anatel, mas são recebidas ligações VoIP em telefone fixos e celulares.

Em outros países, o VoIP é uma realidade e causa muitos prejuízos para as empresas de telecomunicações. O custo de uma ligação VoIP para um telefone fixo ou celular é muito mais barato do que o da telefonia comum, popularizando a utilização do VoIP e forçando as empresas de telefonia a reduzirem taxas e promoverem bônus, de modo a incentivar a utilização da telefonia comum e reter essa fatia do mercado.

Um dos motivos da disseminação de provedores VoIP é o baixo custo de montagem de uma empresa de telefonia VoIP em comparação com os da de telefonia convencional.

As ligações a custo zero ou reduzido tem atraído a atenção não só das empresas, mas também de usuários domésticos. Esse fato tem pressionado a ANATEL em regulamentar o sistema de telefonia VoIP no Brasil.

Existem vários processos em andamento na Anatel para homologação do VoIP. É uma questão delicada, pois a homologação afetará as empresas de telefonia do Brasil.

Capítulo 3 - Protocolos

Para se realizar chamadas telefônicas em redes VoIP, é necessario utilizar um protocolo para telefonia IP. Hoje, existem quatro padrões que dominam o cenário de telefonia IP. sendo que Estes protocolos são responsáveis pelas tarefas de controle e sinalização (localização de usuário, notificação de chamada, notificação de aceite de chamada, início e fim da transmissão e desconexão).

Quatro padrões de garantia da interoperabilidade entre os equipamentos se destacam: o padrão H.323, proposto pela ITU-T; o “Inter- Asterisk Exchange Protocol” (IAX), proprietário do Asterisk; o “Media Gateway Protocol Control” (MGCP), ainda pouco adotado e o padrão proposto pela IETF, o SIP.

Neste capitulo será dado maior enfoque no padrão SIP, que foi implementado nesse projeto.

3.1. MGPC (MEDIA GATEWAY PROTOCOL CONTROL)

Este protocolo foi desenvolvido para ser utilizado em conjunto com o H.323, SIP e IAX. Sua grande vantagem é a escalabilidade. Toda a inteligência é implementada no “Call Agent” e, não nos Gateways, simplificando a configuração da rede VoIP.

3.2. IAX (INTER- ASTERISK EXCHANGE PROTOCOL)

O IAX é o protocolo proprietário do Asterisk. Fornece o controle e a transmissão da voz sobre redes IP. O IAX foi desenvolvido principalmente para voz, mas pode ser usado como qualquer tipo de mídia (voz e vídeo).

Os objetivos do projeto IAX derivam da experiência com os protocolos de voz sobre IP, como o SIP e o MGPC, para controle e o RTP, para o fluxo-multimída. Os objetivos do IAX são:

a) minimizar o uso de banda passante;

b) prover transparência à NAT (Network Address Translation);

c) ter a possibilidade de transmitir informações sobre o plano de discagem; e

d) suportar a implantação eficiente de recursos de “paging” e intercomunicação.

A pequena utilização do IAX no mercado dificulta a implementação de novos serviços com base nesse protocolo.

3.3. H.323

Este padrão - definido pela ITU-T - é largamente utilizado em voz sobre IP. Ele prevê a implementação de algoritmos que garantem a compatibilidade entre codificadores conhecidos como CODECS e VOCODECS e, também, gerencia os protocolos para controle de chamadas, para estabelecimento de canais de comunicação, negociação de qualidade de serviço, interoperabilidade com outros terminais de telefonia convencional, de "Integrated Services Digital Network" (ISDN) e de voz sobre ATM.

Entre as várias implementações que utilizam H.323, pode-se citar o NetMeeting da Microsoft.

O H.323 compõe uma família de diversas funcionalidades especificadas pelo ITU-T, que inclui: o H.245 para controle, o H.225 para conexão, o H.332 para conferências, o H.335 para segurança, o H.246 para interoperabilidade com o "Real-Time Control Protocol" (RTCP) e a série H.450.x para serviços suplementares.

O H.323 é essencial na conectividade com projetos mais antigos, utilizando roteadores CISCO ou gateways de voz. O H.323 ainda é um padrão para fornecedores de PABX e roteadores, muito embora eles estão começando a adotar o padrão SIP.

3.3.1. ARQUITETURA DA REDE H.323

Para que uma comunicação seja possível em um ambiente de rede H.323, alguns elementos devem ser definidos previamente. No entanto, se a comunicação é somente entre dois computadores, faz-se necessário somente os terminais H.323.

A aplicação de VoIP está implementada no terminal H.323, que atua como um terminal de voz. Já o elemento que fica entre a rede IP e outra rede de telefonia convencional é chamada de Gateway H.323. Para que se possa localizar um usuário na rede VoIP utiliza-se o "Gatekeeper" que provê o controle de acesso e o mapeamento de endereços.

A Unidade de Controle Multiponto (MCU) provê facilidades para três ou mais usuários participarem de um grupo de conferência multiponto.

3.4. SIP

Segundo Silva, o protocolo SIP originou-se em meados dos anos 90 como um método bastante simples de convidar pessoas para visualizar sessões de "multicast", como o lançamento de uma nave espacial ou um programa ao vivo como um show multimídia pela Internet. [SILVA, 2002][21]

Em razão da sua simplicidade, da aplicabilidade e do seu potencial, o SIP foi rapidamente adotado para ser utilizado em outros propósitos pelo IETF, mais notadamente, como um padrão para VoIP. Os provedores de serviços de comunicação viram no VoIP uma maneira de agrupar suas demandas de voz e dados na mesma rede.

[21] SILVA, Arlindo Maia da; RAMOS, Luís.**Serviços de Multimídia.**

Baseado nessa concepção, o protocolo SIP foi modelado à imagem do protocolo de transferência de hipertexto (HTTP), que foi desenhado para trabalhar sobre redes IP. Tal como o HTTP, o SIP não encontrou barreiras para o desenvolvimento e implementação de inovadores e fantásticos serviços, levando o controle das aplicações para os dispositivos terminais. Um dos mais poderosos conceitos da Internet é o fato de que aplicações podem operar entre um servidor Web e um "browser" sem nenhuma dependência e conhecimento da rede IP.

A mesma afirmação pode ser feita para as sessões baseadas em SIP. Um servidor SIP e um dispositivo cliente terão o total controle de suas sessões (voz, vídeo, mensagens, mensagens instantâneas). Essa característica é oposta ao modelo de serviços das operadoras de telecomunicações com circuitos comutados, onde os terminais (telefones) têm apenas a capacidade de receber a chamada e todo o restante dos serviços são controlados por elementos de uma central de operações.

Os padrões para conferências e telefonia na Internet foram desenvolvidos pelo grupo do MMUSIC do IETF, que teve o seu primeiro Standard ratificado e que é o mais usado hoje em dia conhecido como padrão Session Initiation Protocol (SIP) e teve sua publicação na RFC 3261 sendo que posteriormente a IETF passou o desenvolvimento do SIP para um grupo independente para haver uma maior dedicação no trabalho começado.[SILVA, 2002][22]

O SIP abrangeu a telefonia IP de uma forma diferente do padrão H.323 sendo que a sua aplicação se tornou mais simples e mais eficientes, pois incluiu os serviços adicionais de transferência de chamada e chamada em espera. [SILVA, 2002] [23]

Uma das características do SIP é a indefinição dos aspectos de comunicação multimídia e de sinalização, sendo que ele reutiliza algumas características de outros protocolos, como os cabeçalhos, erros e regras de codificação do HTTP.

3.4.1.ARQUITETURA DA REDE SIP

Existem três tipos de servidores espalhados pela rede de VoIP: [OLIVEIRA, 2001][24]

a) **servidores de registros** - recebem requisições sobre a localização corrente de cada usuário;

b) **servidores Proxy ou "next-hop"** - recebem requisições e as enviam para outros servidores ou para os clientes;

c) **servidores de redirecionamento** - recebem requisições e redirecionam os servidores, retornando o endereço do servidor para onde a requisição deve ser encaminhada.

As requisições SIP são geradas por um cliente e enviadas ao servidor que processa a requisão e manda uma resposta ao cliente. Um cliente SIP é formado por dois

[22] SILVA, Arlindo Maia da; RAMOS, Luís.**Serviços de Multimídia.**
[23] SILVA, Arlindo Maia da; RAMOS, Luís.**Serviços de Multimídia.**
[24] OLIVEIRA, Sérgio. **Telefonia IP para ambientes móveis usáveis.**

módulos obrigatórios, um chamado de “User Agent Client” (UAC) que gera as requisições e outro chamado de “User Agent Server” (UAS) que é responsável por responder as chamadas.

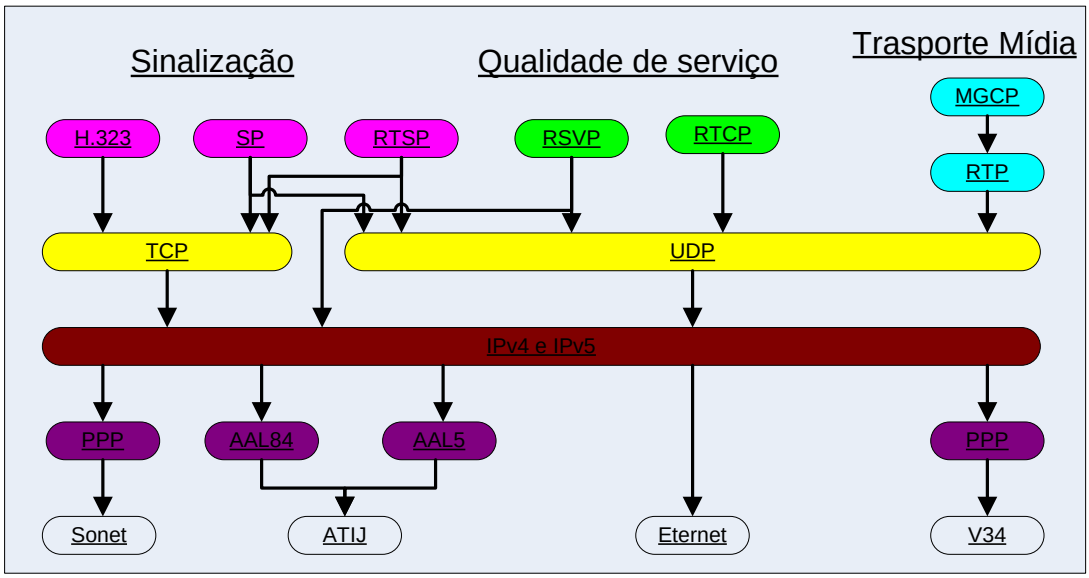

Figura 3-1 - Arquitetura dos protocolos.

3.4.1.1. MENSAGENS SIP

As mensagens SIP são codificadas usando a sintaxe de mensagem do HTTP, sendo que o conjunto de caracteres é o ISO 10646 com codificação de 8 bits e as linhas são terminadas com “Carriage Return, Line Feed” (CRLF). [HERSENT, 2002] [25]

As mensagens SIP trafegam por padrão pela porta 5060, sendo que o usuário pode escolher a porta em que deseja receber as mensagens.

Os dois tipos de mensagens SIP, requests (pedidos) e responses (respostas), compartilham um mesmo formato.

As respostas possuem alguns campos de cabeçalhos que estão tanto nos pedidos como nas respostas tais como:

a) Call-ID: serve para coincidir os pedidos com as respostas correspondentes como nos pedidos REGISTER e OPTIONS. Para os pedidos de INVITE e REGISTER ajuda a detectar cópias duplicadas. Para cada nova chamada, deve-se gerar um novo Call-ID sendo que a primeira parte do Call-ID deve usar um nome único para cada computador e, a segunda, um endereço IP para tornar a máquina globalmente única;

b) Cseq: cada pedido deve ter um campo Cseq que deve ser um campo composto por um número sem sinal. Para cada chamada o número do campo Cseq é incrementado com a exceção do caso da chamada ser uma retransmissão de outra anterior, onde o servidor deve copiar o valor de Cseq para o pedido;

c) From: este campo deve estar em todos os pedidos e respostas sendo que ele contém um nome opcional a ser mostrado e o endereço do originador do pedido;

[25] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

d) To: este campo indica o destino pretendido de um pedido. Ele é simplesmente copiado nas respostas. Um “tag” (etiqueta) pode ser usado quando houver mais de um destinatário (caso de um helpdesk) para distinguir as respostas de pontos finais diferentes.

e) Via: este campo é usado para armazenar o caminho de um pedido SIP para permitir que os servidores intermediários possam retransmitir as respostas para o mesmo caminho. Cada Proxy adiciona seu endereço no campo Via;

f) Encrypton: este campo indica que parte do cabeçalho e o corpo da mensagem podem estar criptografados;

g) Content-Type: este campo indica o tipo de mídia que está sendo utilizado no corpo da mensagem;

h) Content-lenght: contém o número de bytes do corpo da mensagem; e

i) Contact: indica no caso de registro onde o cliente deseja ser contatado.

3.4.1.2. PEDIDOS SIP

Os pedidos SIP são gerados pelo UAC e enviados para o UAS. Esses pedidos podem ser:

a) ACK: um pedido ACK é enviado pelo cliente para confirmar que ele recebeu uma resposta do servidor;

b) BYE: um pedido BYE é enviado para um agente de origem ou de um agente de destino para interromper uma chamada;

c) CANCEL: esse pedido deve ser enviado quando se quer interromper a transmissão antes de receber uma resposta do servido;

d) INVITE: é usado para iniciar uma chamada;

e) OPTIONS: é um pedido enviado a um servidor para saber as capacidades, sendo que o servidor pode enviar de volta uma lista de métodos e até em alguns casos pode enviar as capacidades de algum usuário requisitado;

f) REGISTER: um cliente pode registrar um ou mais endereços de sua localização.

Os pedidos SIP possuem alguns campos adicionais para especificar características específicas adicionais, tais como:

a) Accept: este cabeçalho opcional indica quais tipos de mídia são aceitáveis na resposta;

b) Accept-Language: indica as linguagens preferidas pelo originador da chamada;

c) Expires: para uma mensagem REGISTER esse campo indica quanto tempo o registro será válido. Para uma mensagem INVITE pode ser usado para limitar a duração de buscas;

e) **Priority:** indica a prioridade do pedido;

f) Record-Route: é usado por alguns proxies para adicionar ou atualizar o campo de cabeçalho se quiser fazer parte do caminho de todas as mensagens de sinalização; e

f) **Subject:** campo de texto livre que deve fornecer alguma informação sobre a natureza da chamada.

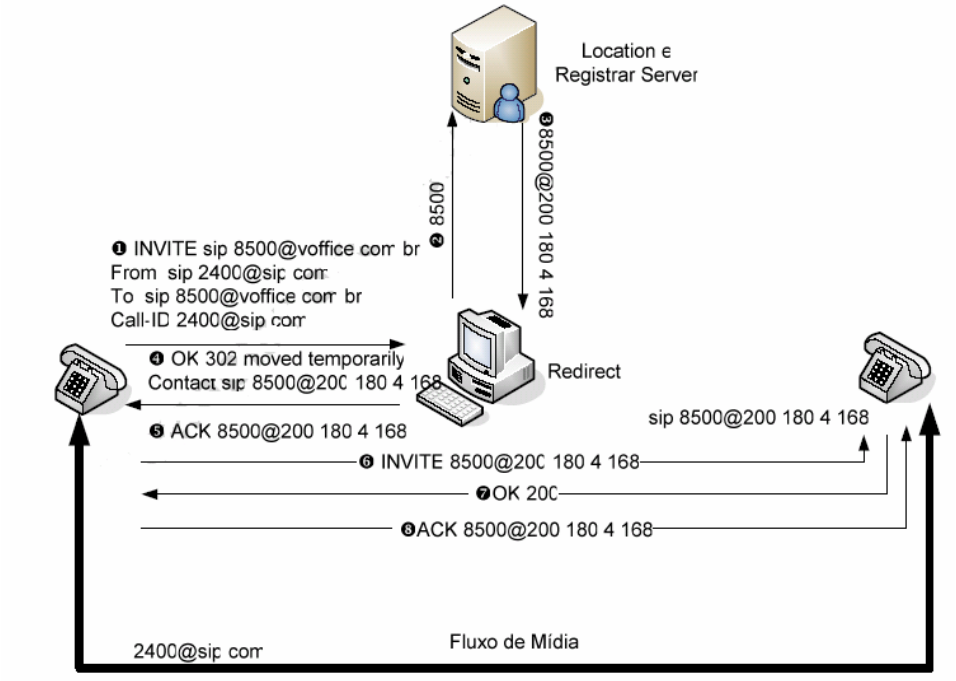

Figura 3-2 - SIP

3.4.1.3. RESPOSTAS SIP

As respostas SIP da primeira linha sempre contêm um código de status e uma frase de justificativa inteligível. A maior parte do cabeçalho é copiada da mensagem de pedido.

Foram definidas seis categorias de código de status para pedidos e respostas SIP, que dependem do primeiro dígito mostrado na Tabela abaixo. Esta classificação torna mais fácil a identificação dos códigos.

1xx – Informativo		Pedido continuando a processar o pedido
	100	Tentando
	180	Chamando
	181	A chamada está sendo retransmitida
	182	Colocado na fila
2xx – Sucesso		A ação foi recebida, atendida aceita com sucesso
	200	OK

3xx – Redirecionamneto		Uma ação adicional deve ser tomada para completar o pedido
	300	Múltiplas escolhas
	301	Movido permanentemente
	302	Movido temporariamente
	380	Serviço alternativo
4xx – Erro de Cliente		O pedido contém sintaxe inválida ou não pose ser efetuado neste servidor
	400	Pedido inválido
	401	Não autorizado
	402	Necessário pagamento
	403	Proibido
	404	Não encontrado
	405	Método não permitido
	406	Não aceitável
	407	Necessária autenticação do *proxy*
	408	Tempo para o pedido esgotado
	409	Conflito
	410	Não mais presente
	411	Necessário fornecer o comprimento
	413	Corpo da mensagem de pedido muito grande
	414	URI do pedido muito grande
	415	Tipo de mídia não suportado
	420	Extensão inválida
	480	Temporariamente não disponível
	481	Transação ou *leg* de chamada não existe
	482	*Loop* (laço) detectado
	483	Excesso de *hops* (segmento)
	484	Endereço incompleto
	485	Ambíguo
5xx		Erro de servidor
	500	Erro interno ao servidor
	501	Não implementado
	502	*Gateway* inválido
	503	Serviço não disponível
	504	Tempo esgotado no *gateway*
	505	Versão SIP não suportada
6xx		Falha global
	600	Ocupado em todos os lugares
	603	Declínio
	604	Não existe em lugar nenhum
	606	Não aceitável

Tabela 3-1 - Categorias de códigos de status

3.4.1.4. ESTABELECENDO UMA CHAMADA

Para estabelecer uma chamada, um cliente SIP chama um outro ponto final SIP enviando uma mensagem de pedido INVITE. Essa mensagem possui informações que o ponto de origem pode suportar para que o ponto de destino estabeleça a conexão de mídia solicitada e também o endereço onde o ponto de origem deseja receber esses dados de mídia. A codificação escolhida aparece como parte do cabeçalho RTP.

Para o ponto de destino indicar que está aceitando uma chamada ele responde com uma mensagem OK e o ponto de origem indica que recebeu a mensagem com uma mensagem ACK.

Isso mostra a simplicidade do SIP para estabelecer uma chamada. É possível estabelecer um canal de comunicação com uma ida e uma volta de mensagens e mais uma ida e uma volta de negociação do canal de mídia.

3.4.1.5. NEGOCIAÇÃO DO CODEC

Para a negociação de CODEC, um terminal de origem envia uma mensagem de INVITE passando um codificador de áudio. Se o receptor não suportar essa codificação de áudio, talvez por não ter a largura de banda necessária ou por não ter o codificador que é requisitado, ele envia uma mensagem "606 Not Acceptable" e seleciona uma lista de CODECS que ele suporta e por sua vez o terminal de origem envia novamente uma mensagem INVITE com um codificador que o terminal de destino pode aceitar.

O SIP não define nenhum tipo de CODEC que deve ser usado para estabelecer uma chamada. [HERSENT, 2002] [26]

3.4.1.6. FINALIZANDO UMA CHAMADA

Se em algum momento da sessão uma das partes quiser finalizar a chamada, ela envia um pedido do tipo BYE e inverte os campos TO e FROM do cabeçalho. Mesmo que os fluxos de mídia não sejam mostrados, as mensagens incluem os cabeçalhos obrigatórios.

3.4.1.7. REJEITANDO UMA CHAMADA

Se um cliente não pode atender à chamada, seja porque ele não deseja atender ou por que está em outra conversação, existem mensagens que expressam essa condição. O originador de uma mensagem pode tentar localizar um destino em outros lugares como, por exemplo, se um usuário não se encontra no terminal no momento, ele pode indicar onde ele está e as mensagens que chegam para ele podem ser enviadas para outro destino. Se o usuário não desejar ser encontrado em nenhum lugar ele pode responder a uma requisição de mensagem com uma resposta do tipo "600 Ocupado em todos os lugares".

3.4.1.8. SDP

O SIP utiliza o protocolo "Session Description Protocol" (SDP) para definir uma sintaxe padrão para o tipo de transmissão de áudio que será usada. Este protocolo inclui as seguintes funções:

a) Fluxo de mídia: o SDP leva informações sobre o número e o tipo de cada fluxo de mídia, já que em uma sessão pode haver vários fluxos;

b) Endereços: para garantir a independência para cada fluxo, é indicado o endereço do destinatário, seja unicast ou multicast;

c) Portas: para cada fluxo, a porta UDP para recepção e/ou envio é indicada;

d) Tipo de conteúdo: define o formato de mídia que pode ser usado na sessão; e

d) Origem: para poder contatar o responsável pela chamada nas sessões do tipo "broadcast".

[26] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

O protocolo SDP consiste em várias linhas “<type>=<value>” que podem ser legíveis aos usuários sendo que os nomes dos campos e os atributos usam caracteres ASCII facilitando assim a programação e a depuração.

v = 0
o = sergiool 87728 8772 IN IP4
10.61.217.1
s = Ola, senhores!
u = http://www.voip.uniceub.br/~sergiool
e = flavio@voip.uniceub.br
c = IN IP4 10.61.217.1
b = CT:64
t = 3086272736 0
k = clear:manhole cover
m = audio 3456 RTP/AVP 96
a = rtpmap:96 VDVI/8000/1
m = video 3458 RTP/AVP 31
m = application 32416 udp wb
a = orient:portrait

Tabela 3-2 - Exemplo de uso dos campos SDP

A tabela 3-2 identifica os campos do protocolo SDP no qual o “v” indica a versão da sessão, a linha “o” apresenta o conjunto de valores para identificar a sessão. que inclui endereço IP, as portas utilizadas e o usuário, o endereço de e-mail e a URL. O campo “u” fornece mais informações sobre a sessão. O endereço para a sessão é indicado no campo “c”. A linha “b” indica a largura de banda. A linha “t”, o tempo de início e de fim. A linha “k” traz a chave de criptografia para a sessão. A linha “m” indica o tipo de fluxo de mídia, o número de porta para o fluxo, o protocolo e a lista de tipos de conteúdo. A linha “a” representa um conteúdo.

3.4.1.9. VISÃO GERAL DO RTP E O RTCP

O receptor tem que levar em conta o jitter numa transmissão de dados em tempo real que usa multiplexação estatística. [HERSENT, 2002] [27]

Para que os receptores pudessem compensar a perda de jitter e a perda de seqüência dos pacotes introduzidos na rede IP, foi projetado o “Real-Time Transport Protocol” (RTP) que pode ser usado para qualquer fluxo de dados em tempo real, definindo o modo de formatar pacotes IP e inclui informação sobre o tipo de dados transportado, “timestamps” e número de seqüências.

Junto com o RTP, pode-se usar o “Real-Time Control Protocol” (RTCP) que é um protocolo de controle de transmissão em tempo real e é usado para transportar algum retorno sobre a qualidade da transmissão e algumas informações a respeito da identidade dos participantes.

[27] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

O comportamento da rede IP não é influenciado pelo uso do RTP e RTCP que não controlam a qualidade de serviço, sendo que a rede pode perder o serviço, inserir atraso ou perder os pacotes RTP. Este permite ao receptor compensar o jitter por meio de controle de buffer e do seqüenciamento dos pacotes, tendo mais informações da rede para poder tomar medidas de correção.

O RTP e RTCP podem ser usados acima de qualquer protocolo, mas eles são usados em cima do UDP, uma vez que os dados precisam ser transportados com uma latência muito baixa. O RTP costuma ser associado a uma porta do UDP de número par e o RTCP é associado à próxima porta ímpar do UDP.

3.4.1.9.1. RTP

O RTP é usado em uma rede que introduz jitter e pode tirar a seqüência dos pacotes para transmitir dados de áudio e vídeo.

Para que o cliente possa gerenciar as chegadas dos pacotes de uma forma correta, o RTP usa o número de seqüência, que, em uma aplicação que esteja reproduzindo o áudio, pode definir um buffer para armazenar os pacotes que chegam em uma ordem correta antes de reproduzir. Se, na hora de reproduzir o áudio o pacote não tiver chegado, a aplicação pode optar por copiar o último pacote que foi reproduzido e repeti-lo até chegar à vez do próximo ou usar algum esquema de interpolação definido pelo CODEC de áudio.

O timestamp que é o campo usado para guardar informações sobre o tipo de dados é usado em uma aplicação de vídeo que permite, por exemplo, deduzir qual parte de tela é descrita pelo pacote IP, no entanto, devido a problemas de seqüenciamento, usa-se o pacote para construir a parte da imagem que ele descreve.

Como cada formato do pacote RTP de informação em tempo real é livre, usa-se o “payload type” (tipo de payload) no cabeçalho de cada pacote RTP para distinguir um formato em particular, sem que seja necessário analisar o conteúdo do payload.

Uma associação de participantes em que cada participante usa dois endereços define uma sessão RTP. No caso do UDP, são duas portas para cada sessão sendo uma para o fluxo RTP e uma para o RTCP.

A fonte de sincronização, “Synchronization Source” (SSRC), é uma fonte de fluxo RTP identificada por 32 bits no cabeçalho RTP. Uma mesma referência de tempo e de seqüenciamento é usada com um SSRC nos pacotes RTP.

A Fonte Contribuinte, “Contributing Source” (CSRC), é usada quando o fluxo RTP é resultado de uma combinação de fluxos feita por um mixer (misturador) RTP, sendo que cada fluxo possui um CSRC, que é adicionado à lista de CSRC, formando um SSRC de todo o fluxo.

O formato “Network Time Protocol” (NTP) é uma maneira padrão de formatar um timestamp escrevendo o número de segundos passados desde 01/01/1900 com 32bits para a parte inteira e 32 bits para a parte decimal.

A ilustração abaixo mostra a estrutura de um pacote RTP. Dois bits são reservados para a versão do RTP, o "bit de padding" (P) indica se o payload sofreu enchimento para fins de alinhamento e, se tiver sofrido enchimento, o último octeto guarda a quantidade que foi acrescentada; e, um bit de extensão X que indica a presença de extensões após eventuais CSRC do cabeçalho fixo. Há o contador de CSRC (CC), que indica a quantidade de CSRC e um bit de marcador (M), que é definido pelo perfil do RTP.

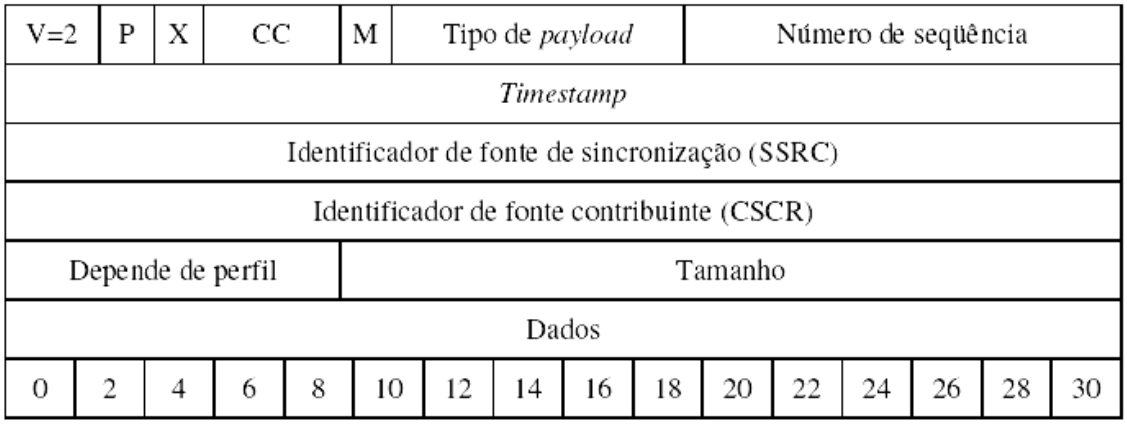

Figura 3-3 - Pacote RTP

3.4.1.9.2. RTCP

Pacotes de controle relativos a uma sessão RTP são transmitidos de tempos em tempos para os participantes de uma transmissão, sendo que esses pacotes incluem informações a respeito dos participantes em suas fontes de fluxo individuais. Uma forma de transmitir essas informações é através do uso de "sender reports" usados pelos transmissores e "receiver reports" usados pelos receptores.

Um problema com a transmissão freqüente desses pacotes é que, a medida que aumenta o número de participantes de uma transmissão "multicast", aumenta o número de pacotes RTCP trafegando na rede. Esse problema não ocorre com o RTP porque os participantes não falam todos ao mesmo tempo.

Como, em uma transmissão, o número de participantes é conhecido por todos, pode-se controlar a emissão de pacotes RTCP, limitando-os ao estritamente necessário. No entanto, o RTP define que 5% da largura de banda da sessão pode ser usado para pacotes RTCP, dos quais a quarta parte se destina aos transmissores que enviam informações destinadas à sincronização.

Há vários tipos de pacotes RTCP para cada tipo de informação, tais como: [HERSENT, 2002][28]

a**) Sender Reports (SR) -** possui informações de transmissão e recepção para transmissores ativos;

b**) Receiver Reports (RR) -**contém informações de recepção para ouvintes que não sejam também transmissores ativos;

[28] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

c) Source Description (SDES) - descrevem vários parâmetros de fonte;

d) BYE - utilizado para um participante abandonar uma conferência; e

g) **APP** - utilizado pela aplicação para funções específicas.

Um pacote UDP pode conter vários pacotes RTCP que possuem informações suficientes para serem decodificados. Esse empacotamento diminui o "overhead" gasto.

Cada pacote SR contém três seções obrigatórias. A primeira contém o contador de relatórios de recepção (RC), com tamanho de 5 bits para determinar o número de relatórios; tipo de payload (PT). Essa seção destina-se a evitar que se misture pacotes SR e RTP. Para representar o tamanho do SR utiliza-se um campo de 16 bits e o SSRC do originador. [HERSENT, 2002] [29]

A Segunda Seção contém as informações a respeito do fluxo RTP enviado para o transmissor, definido um "timestamp" NTP, que se dá no instante do envio do relatório. O campo para a contagem de pacotes do transmissor no início da transmissão possui um tamanho de 32 bits e é do mesmo tamanho do campo para a contagem de octetos do payload do transmissor.

A terceira seção possui um conjunto de blocos de relatórios de recepção para cada fonte que o transmissor teve conhecimento. O SSRC_n é o identificador da fonte de referência. A fração perdida tem um tamanho de 8 bits e é dada pelo cálculo dos pacotes recebidos dividido pelos pacotes esperados vezes 256. O campo que acumula o número de pacotes perdidos desde o início da transmissão tem 32 bits. Esse campo armazena o valor estendido do número de seqüência que se divide nos 16 bits mais significativos, armazenando a quantidade de vezes que os ciclos atingiram o número máximo e, os outros 16 bits, contêm o último número da seqüência. Para fazer a estimativa da variância do tempo entre chegadas utiliza-se o campo jitter entre chegadas. [HERSENT, 2002] [30]

Um pacote "Receiver Report" (RR) difere-se apenas no campo "Payload Type" (PT) e na segunda seção relativa ao transmissor que não está presente.

3.4.1.10. ENTIDADES SIP

As entidades SIP são componentes de apoio à rede de voz sobre IP, que tem por função o mapeamento de usuários.

3.4.1.10.1. REGISTRAR

O Registrar é um servidor que aceita os pedidos REGISTER, sendo que ele também pode implementar outras funções como a de um Proxy. [HERSENT, 2002] [31]

---

[29] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**
[30] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**
[31] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

Para que a localização de um usuário seja conhecida por todos os usuários da rede, utilizamos o Registrar. O IP do usuário pode mudar devido a várias circunstâncias como: o usuário usar uma conexão discada, que fornece endereços dinâmicos; participar de uma rede que utiliza o serviço de DHCP, que também fornece número IP dinamicamente; ou se tratar de um usuário móvel. É necessário manter o mapeamento dos endereços SIP e IP.

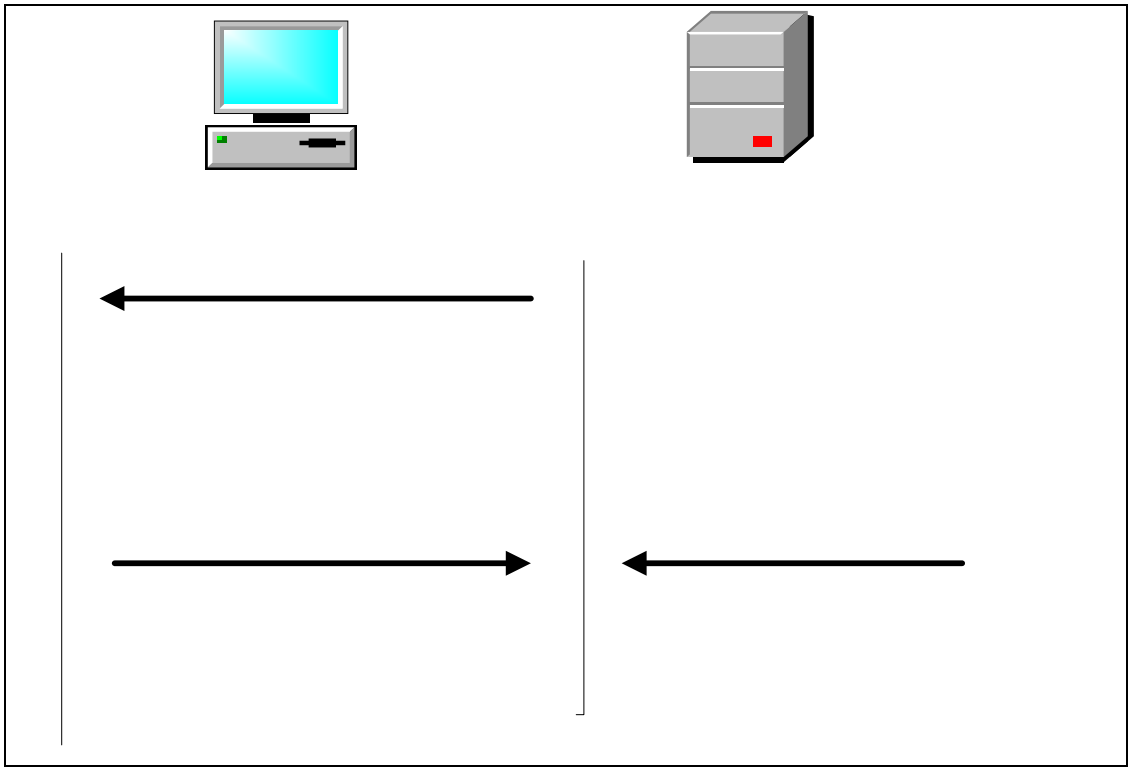

Figura 3-4 - Troca de mensagens SIP com Registrar

A figura 3-4 demonstra a troca de mensagens entre um cliente SIP e o Registrar, mostrando a mensagem REGISTER enviada ao Registrar para processar o registro. Quando da chegada de uma mensagem ao Registrar, ela é processada e mandada para o seu destinatário.

Um cliente SIP não necessita fazer configurações manuais, pois o SIP mantém um endereço conhecido chamado "All SIP Server", possibilitando ao cliente registrar o seu endereço IP através de uma mensagem REGISTER multicast.

Os clientes SIP não podem identificar o endereço de um servidor SIP ou saber se algum servidor Registrar aceitou o registro, pois eles não são capazes de responder a mensagem REGISTER MULTICAST. Esta uma limitação da própria definição do SIP.

Se o endereço do Registrar for conhecido, o cliente tem a opção de contatar o servidor através de mensagem REGISTER UNICAST.

Um usuário deve mandar mensagem REGISTER periodicamente para atualizar seu estado, pois o registro tem um valor padrão de uma hora, sendo que esse valor pode ser definido no campo de cabeçalho "Expires".

3.4.1.10.2. PROXY

Um servidor Proxy pode passar uma mensagem adiante sem alterar ou pode alterar alguns campos do cabeçalho, pois ele atua de um lado como servidor, recebendo mensagens, e, por outro lado, como cliente, enviando mensagens. Ele pode gerar uma resposta localmente e enviar para o transmissor. [HERSENT, 2002] [32]

Se uma mensagem deve seguir o mesmo caminho da ida na volta, talvez por motivo de tarifa ou controle de firewall, os proxys colocam informações nos cabeçalhos para que a mensagem saiba onde é a origem.

Em uma transação SIP, se for utilizado TCP, ele próprio faz o controle para onde deve ir a resposta. No entanto, se for utilizado o UDP, deve incluir informações adicionais no cabeçalho para permitir que o receptor saiba para onde mandar a resposta.

Os cabeçalhos "Via" do SIP servem para controle dos pontos de passagem de uma mensagem, ajudando a evitar loops (laços) de roteamento. Em cada Proxy que a mensagem passa é verificado se o endereço dele está na lista "Via". Caso o endereço não esteja na lista do Proxy SIP, ele o adiciona e quando o Proxy repassa uma resposta ele faz o serviço inverso, retirando o endereço da lista.

O Proxy deve rotear todas as mensagens, os pedidos e a respostas. Caso o cabeçalho Via não seja suficiente, os proxies fazem uso do cabeçalho "Record Route". Isto ocorre porque os clientes SIP adicionam um campo no cabeçalho que permite que os servidores respondam diretamente para os clientes. Fazendo uso do campo "Record Route" os proxies incluem seus endereços SIP na primeira posição do cabeçalho fazendo com que eles estejam no caminho de todas as mensagens.

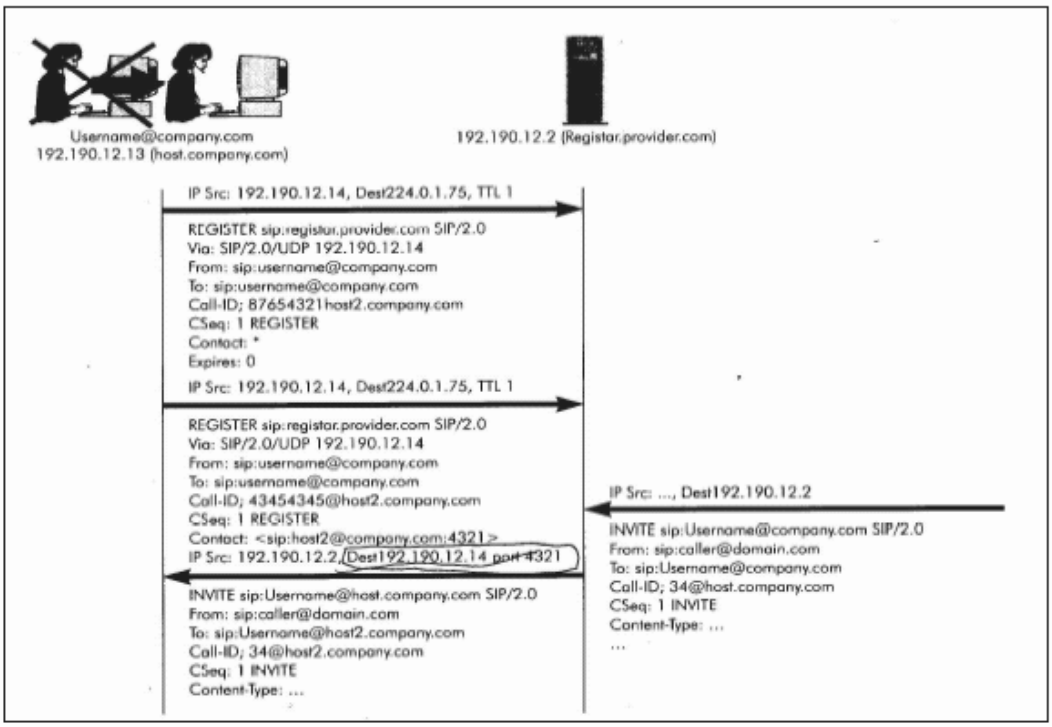

Figura 3-5 - Registro do cliente e o encaminhamento da mensagem

[32] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

A Figura 3-5 demonstra o Registro do cliente e o encaminhamento da mensagem para o cliente.

As mensagens roteadas pelos proxies SIP contém informações suficientes para permitir que os proxies sejam “stateless” (sem-estado), ou seja, não é necessário que eles monitorem todo o fluxo que passa por eles.

3.4.1.10.3. SERVIDOR DE REDIRECIONAMENTO

Um servidor de redirecionamento é uma ferramenta útil para melhorar o escalonamento de servidores de distribuição de chamadas ou servidores agentes de chamadas, pois pode distribuir chamadas entre grupos de servidores secundários, permitindo um equilíbrio de carga. Por exemplo, uma chamada feita para um determinado endereço pode ser atendida por um outro servidor da rede. Isto se faz possível através do campo “Contact” onde ele identifica a partir do domínio do endereço um servidor secundário da rede. [HERSENT, 2002] [33]

Um pedido INVITE pode originar no servidor de redirecionamento algumas respostas, tais como:

a) Múltiplas escolhas (300) - indica que o cliente pode ser encontrado em vários lugares, sendo que a resposta indicaria os endereços;

b) Movido permanentemente (301) indica que o cliente não pode ser encontrado no endereço pedido e que se deve tentar contatar em outra localização possível passando no campo “Contact” da resposta possíveis destinos;

c) Movido temporariamente (302) - indica que para um cliente que esta em outra localização mas por tempo limitado; e

d) Serviço alternativo (380) - além de dizer uma nova localização do usuário, também adiciona as capacidades que o usuário tem para transmitir. Assim, o cliente, ao gerar um pedido para o usuário, adiciona as capacidades que o usuário suporta, evitando uma possível retransmissão.

A Figura 3-6 mostra a troca de mensagem entre o Registrar e o cliente, na qual o cliente manda uma mensagem para o Registrar, que processa e identifica que o cliente esta em outro lugar, enviando a mensagem para o cliente “Moved” indicando o domínio do cliente e o endereço IP.

Para direcionar as chamadas para a localização atual do usuário, o servidor de redirecionamento pode ser usado em conjunto com o Registrar, que também pode atuar como um sistema de distribuição de chamada.

[33] HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP.**

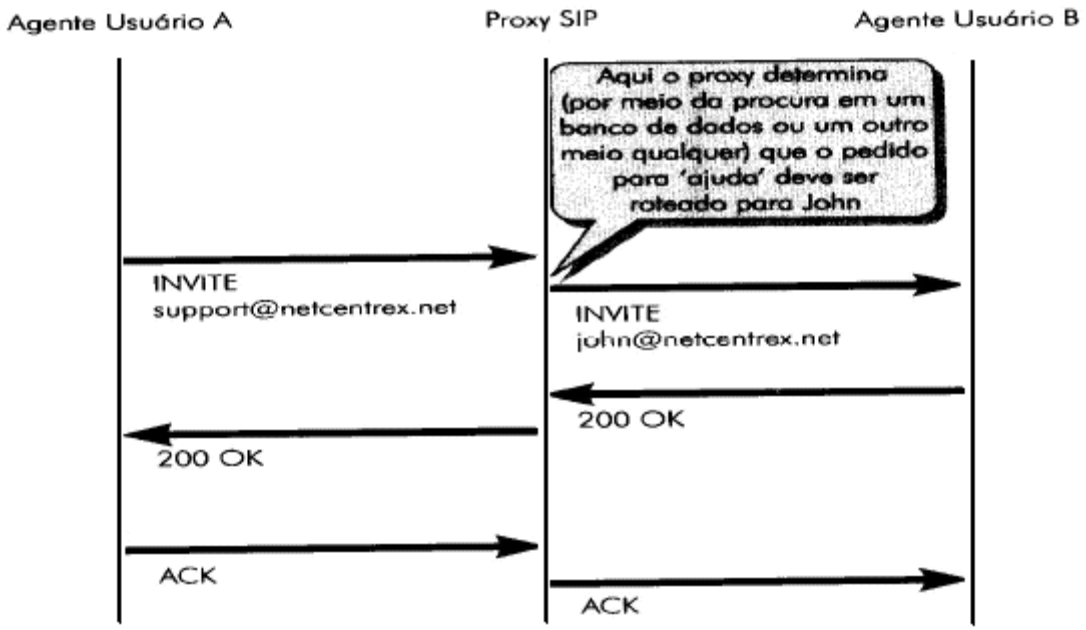

Figura 3-6 - Encaminhamento de mensagem entre o cliente e o Proxy

3.5. COMPARAÇÃO ENTRE O SIP E H.323 (SIP X H.323)

Existem dois padrões principais para possibilitar o tráfego de voz e multimídia sobre IP: o H.323 e o SIP. O primeiro foi oficializado em 1996 pelo ITU-T (International Communications Union) e o SIP nasceu três anos depois na IETF (Internet Engineering Tasf Force).

É importante lembrar que, ao contrário do MGPC (Média Gateway Control Protocol), baseado no modelo centralizado, o H.323 e SIP utilizam a inteligência dos equipamentos. Mas, as similaridades entre os dois param por aí. Aliás, há uma competição saudável dos protocolos.

O H.323 é mais antigo e é utilizado por grande parte dos players tradicionais do mercado VoIP. No entanto, o SIP vem ganhando cada vez mais espaço no setor, principalmente nos EUA e Europa, pela facilidade de permitir o desenvolvimento de aplicações. O SIP é mais flexível e pode rodar nos padrões da Internet.

Um dos problemas sentidos nos projetos corporativos é a dificuldade de fazer com que equipamentos de marcas diferentes conversem entre si. Nesse sentido, o H.323 tende a perder espaço com o tempo, por ser um protocolo fechado.

A tendência é que as empresas partam para o SIP, pois o protocolo é aberto e permite a criação de novas aplicações corporativas. O SIP é uma evolução natural do mundo de VoIP, pois permite uma interoperabilidade entre os dois protocolos.

Os fornecedores não ignoram tal tendência e vêm lançando produtos compatíveis com o SIP. A principal justificativa desses novos produtos é ter que lidar com um legado de equipamentos antigos, que rodam H.323 nativo, mas que tendem a adotar ambientes compatíveis com aplicações mais avançadas, rodando SIP. Portanto,

considera-se que os padrões são complementares. Gradativamente, o H.323 cederá o espaço para que o SIP seja implementado por todos os players como o novo padrão de mercado.

Uma das formas de definir o padrão mais adequado para o ambiente de cada empresa é avaliar, do ponto-de-vista técnico, os benefícios e as carências dos dois protocolos. Na comparação de escalabilidade, por exemplo, em ambientes com alto tráfego de chamadas, o SIP exige menos ciclos de CPU para gerar a sinalização de mensagens. Com isso, o servidor tem condições de, teoricamente, manusear mais transações do que o H.323, que usa mensagens definidas no H.225 para ajustar o Gatekeeper a executar o balanceamento de carga.

Para as aplicações de videoconferência, o H.323 suporta conferência de vídeo e dados. Os procedimentos estão alocados para fornecer controle para as reuniões virtuais, assim como a sincronização de áudio e vídeo. Uma vantagem em relação ao SIP, que apresenta suporte limitado para imagens e não conta com protocolos de conferência de dados, como o T.120.

O SIP é mais seguro, pois utiliza autenticação por HTTP (Hypertext Transfer Protocol), SSL (Secure Sockets Layer) e PGP (Pretty Good Privacy), sendo bastante escalável. O H.323 usa apenas o H.235.

O H.323 é baseado em protocolos do ITU-T, já existentes, e tem uma abordagem voltada aos equipamentos terminais.

O SIP é similar ao HTTP e tem uma abordagem voltada aos usuários e serviços integrados na Internet.

A seguir, comparar-se-á os aspectos de complexidade de implementação ou funcionamento, expansão funcional (facilidade de inclusão de novas funcionalidades), escalabilidade, que é a facilidade para aumento da quantidade de elementos interligados e os serviços oferecidos: [FERNANDES, 2003][34]:

a) COMPLEXIDADE DE IMPLEMENTAÇÃO

A maior complexidade de implementação do H.323 em relação ao SIP se deve ao fato da documentação do H.323 ter 736 páginas contra apenas 128 do SIP, que leva o desenvolvedor dedicar muito tempo para entender o funcionamento do H.323.

O SIP trabalha com apenas 37 tipos de cabeçalhos enquanto que o H.323 tem centenas de tipos.

O H.323 trabalha com vários protocolos sem uma separação clara, ou seja, esses protocolos são usados por vários serviços.

[34]FERNANDES, Nelson Luiz Leal.**Voz sobre Ip.**

b**) EXPANSÃO FUNCIONAL**

A inclusão de novas caracterísitcas é mais simples no SIP, que é compatível com as versões anteriores e permite adicionar parâmetros novos em qualquer parte da mensagem.

No SIP, existem campos predefinidos para essas novas inclusões. Se um novo "CODEC" é registrado em um órgão competente, é possível ser suportado pelo SIP. No H.323, há uma dificuldade na inclusão de novos "CODECS" porque eles devem ser padronizados pelo ITU-T.

c) ESCALABILIDADE

Os servidores ou gateways SIP podem trabalhar nos modos "stateful" ou "stateless". No segundo caso, os servidores recebem e encaminham as requisições não mantendo nenhum tipo de informação, pois as mensagens possuem informações suficientes para garantir o correto endereçamento.

O H.323 é "stateful", ou seja, ele mantém todo o controle do estado da chamada durante toda a duração. No caso de haver muitas chamadas simultâneas, haverá problemas de performance.

d) SERVIÇOS

Os dois protocolos oferecem serviços bastante parecidos. As facilidades de transferência, conferências e encaminhamento de chamadas são entendidos como serviços.

Concluindo, a tabela 3-3 apresenta as medidas de simulações de dois processos realizadas em diversos locais dos Estados Unidos. Estão indicados os percentuais de dias onde ocorreram mais de 1% de rejeição de chamadas.

		Boston	Chicago	West coast	Washington	Colorado
New York	SIP	20,3	77,2	32,3	9,1	15,4
	H.323	28,2	94,7	40.0	20,0	18,5
Boston	SIP		1,6	31,5	0,0	5,4
	H.323		1,6	31,5	0,0	10,8
Chicago	SIP			34,3	5,2	28,6
	H.323			34,3	6,9	61,4
West coast	SIP				33,6	45,3
	H.323				36,7	57,3
Washington	SIP					6,6
	H.323					6,6

Tabela 3-3 Tabela de comparação da satisfação com os protocolos

Capítulo 4 - Asterisk

4.1. ASTERISK

O Asterisk é um software de PABX que usa o conceito de software livre "**General Public License**" (GLP), criado pela Digium Inc. A Digium Inc investe tanto no desenvolvimento do código fonte do Asterisk como no hardware conectado à rede pública de telefonia, PSTN (Public Service Telephony Network). O Asterisk permite conectividade em tempo real entre as redes PSTN e as redes Voip.

O Asterisk é muito mais que um PABX IP. Com o Asterisk é possivel criar novas aplicações em telefonia como:

a. conectar empregados trabalhando de casa para o PABX do escritório sobre conexões de banda larga;

b. conectar escritórios em vários estados sobre IP, o que pode ser feito pela Internet ou por uma rede IP privada;

c. proporcinar correio de voz integrado com a "web" e e-mail aos funcionários;

d. implantar aplicações de resposta automática por voz para os sistema de pedidos ou outras aplicações internas;

e. proporcionar acesso ao PABX da companhia para usuários em viagem, conectando sobre VPN de acordo com as possibilidades do aeroporto, hotel ou qualquer outro local com conectividade à Internet

f. O Asterisk inclui muitos recursos que só eram encontrados em sistemas de mensagem unificada "topo de linha", tais como:

I) música em espera para clientes, suportando "streaming" de media e música em MP3;

II) filas de chamada com agentes que atendem chamadas e monitoram a as filas de forma contínua;

III) integração para sintetização da fala (texto speech);

IV) registro detalhado de chamadas (call-detail-records) para integração com sistema de tarifação;

V) integração com reconhecimento de voz (tal como o software de código aberto para reconhecimento de voz); e

VI) a habilidade de interfacear com linhas telefonicas normais, ISDN em acesso básico (2D+D) e primário (30B+D).

4.2. QUAL É O PAPEL DA DIGIUM?

A Digium, sediada em Huntsville – Alabama, é a criadora e desenvolvedora primária do Asterisk, o primeiro PABX de código aberto da indústria.

Usado em conjunto com as placas de telefonia PCI, o Asterisk oferece uma abordagem estratégica com excelente relação custo/benefício para o transporte de voz e dados sobre arquiteturas TDM, comutadas e redes Ethernet.

A Digium é o principal patrocinador do Asterisk e um dos líderes na indústria do PABX em código aberto.

4.3. PORQUE O ASTERISK?

O Asterisk está para a telefonia como o Apache está para a Web. Ele apresenta as seguintes vantagens e desvantagens:

4.3.1.REDUÇÃO DE CUSTOS

Ao adicionar recursos avançados - tais como o VoIP, Unidade de Resposta Automática (URA) e Direcionador Automático de Chamadas (DAC), o Asterisk proporciona uma economia da ordem de noventa por cento nos custos. Para dar um exemplo, uma única porta de URA - com acesso a um mainframe – está cotada em US$ 1700,00. [FLÁVIO, 2005] [35]

4.3.2.TER CONTROLE DO SEU SISTEMA DE TELEFONIA

Este é um dos maiores benefícios. O Cliente independe do fornecedor para configurar o PABX IP. Com o Asterisk, o próprio cliente, se possuir um pequeno conhecimento de LINUX, pode configurá-lo.

4.3.3.AMBIENTE DE DESENVOLVIMENTO FÁCIL E RÁPIDO

O Asterisk pode ser programado em C, com as “Application Program Interface” (API) nativas ou em qualquer outra linguagem usando “adjusted gross income” (AGI).

4.3.4.RICO E ABRANGENTE EM RECURSOS

Como tem sido afimado ao longo do presente trabalhão, poucos são os recursos encontrados em equipamentos PABX vendidos no mercado que não possam ser encontrados ou criados no Asterisk.

4.3.5. É POSSÍVEL PROVER CONTEÚDO DINÂMICO

Com o Asterisk é possível programar novas aplicações em C ou outras linguagens de domínio da maioria dos programadores. Com isso, as possibilidades de prover conteúdo dinâmico por telefone são ilimitadas.

[35] FLÁVIO Eduardo de Alechandre Gonçalves, 2005, **Asterisk PBX**

4.3.6.PLANO DE DISCAGEM FLEXÍVEL E PODEROSO

Neste ponto, o Asterisk supera a grande maioria das soluções de PABX. A maioria das centrais não possui nem mesmo a rota de menor custo. Com Asterisk este processo é simples e prático.

4.3.7.RODA NO LINUX E É DE CÓDIGO ABERTO

Umas das coisas mais fantásticas do Linux é a comunidade de software livre. Ao acessar o Wiki, ou fóruns de software em código aberto, é fácil perceber que a adoação de usuárias é muito mais rápida. Milhares de questões e de relatos são enviados todos os dias. O Asterisk é provavelmente um dos softwares que mais pessoas têm disponíveis para testes e avanços. Isto torna o código estável e permite a rápida resolução dos problemas.

4.4. LIMITAÇÕES DE ACESSO NO BRASIL

Ainda falta no Asterisk um driver para acesso à R2 Brasil, com código aberto. Já existem algumas implementações no Brasil, mas o código por enquanto está fechado. Isto limita o acesso à rede pública.

A conexão à rede pública com “Foreign eXchange Office” (FXO) pode ser feita para linhas analógicas. Esta restrição se aplica apenas aos acessos digitais.

4.5. LIMITAÇÕES DA ARQUITETURA DO ASTERISK

O Asterisk usa a CPU do servidor para processar os canais de voz, ao invés de ter um DSP (processador de sinais digitais) dedicado a cada canal.

Isto permitiu a redução de custo das placas E1/ T1, mas deixou o sistema dependente da performance da CPU.

É recomendado preservar ao máximo a CPU do Asterisk, rodá-lo sempre em uma máquina dedicada e testar o dimensionamento antes de implantar. O Asterisk, como qualquer outra solução de VoIP, deve ser sempre implementado em uma rede local virtual (VLan) específica para VoIP. Qualquer tempestade de broadcast causada por loops ou vírus pode comprometer o seu funcionamento, devido ao uso das placas de rede na CPU.

4.6. CENÁRIO DE USO DO ASTERISK

A seguir, mostrar-se-ão alguns cenários de uso do Asterisk e como ele se encaixa no modelo atual de telefonia.

4.6.1.VISÃO GERAL

Dentro de uma visão geral, o Asterisk é um PABX híbrido que integra tecnologias como TDM e telefonia IP com funcionalidade, unidade de resposta automática e distribuição automática de chamadas.

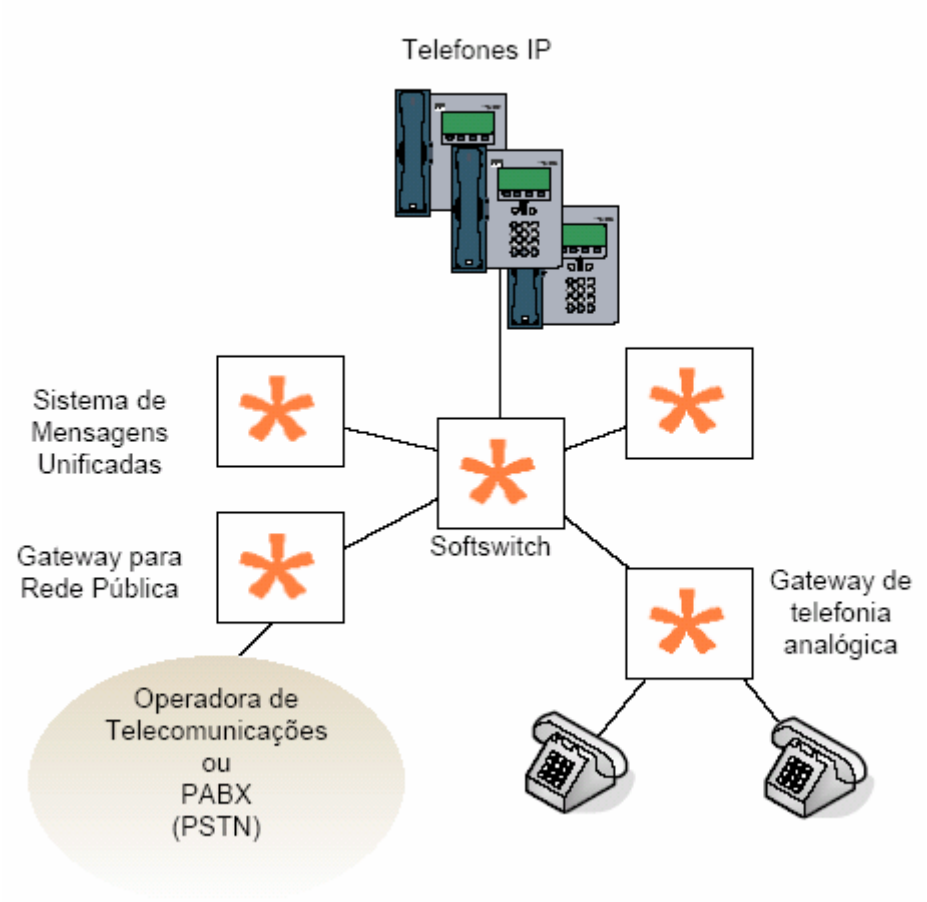

Figura 4-1 – Serviços agregados ao Asterisk [FLÁVIO, 2005][36]

Na figura 4-1, podemos ver que o Asterisk pode se conectar a uma operadora de telecomunicações ou um PABX, usando interfaces analógicas ou digitais. Pode se comportar como um servidor de conferência, correio de voz, unidade de resposta automática, distribuidor automático de chamadas e servidor de músicas em espera. Os telefones podem ser IP, analógicos ou ADSI que é um telefone analógico com display digital.

4.6.2.VAMOS CONCEITUAR DE FORMA MAIS DETALHADA:

4.6.2.1. O CORREIO DE VOZ

Permite que, quando o usuário não atender ao telefone por estar ocupado ou ausente, receba um “prompt”, solicitando que deixe uma mensagem na caixa postal. É semelhante a uma secretária eletrônica ou caixa de mensagens do telefone celular. O Asterisk apresenta esta funcionalidade, sem custo adicional.

[36] FLÁVIO Eduardo de Alechandre Gonçalves, 2005, **Asterisk PBX**

4.6.2.2. SISTEMA DE MENSAGENS UNIFICADAS

É um sistema onde todas as mensagens são direcionadas para um único lugar, por exemplo, a caixa de correio eletrônico do usuário. Neste caso, as mensagens de e-mail, junto com as mensagens de correio de voz e fax, são encaminhadas para a caixa postal do usuário.

4.6.2.3. DISTRIBUIDOR AUTOMÁTICO DE CHAMADAS

Este é um dos conceitos menos óbvios da telefonia. Em um DAC (ACD em inglês, Automatic Call Distribution) as pessoas normalmente se autenticam em uma fila de atendimento para receber as chamadas, o distribuidor verifica se o número está com o telefone livre antes de passar a chamada. Se nenhum operador estiver com o telefone livre, ele segura a chamada, seguida de música ou mensagens de voz. Ao primeiro atendente liberado, o DAC passa a ligação. O DAC é fundamental em qualquer sistema de atendimento e qualquer Call Center receptivo. O sistema de roteamento pode ser muito sofisticado. O DAC custa uma pequena fortuna na maioria das plataformas convencionais, no Asterisk é integrado ao sistema, sem custo.

4.6.2.4. SERVIDOR DE MÚSICAS EM ESPERA.

Na maioria das centrais telefônicas, é preciso colocar um aparelho de CD ligado a um ou vários ramais para que o usuário possa ouvir uma música enquanto aguarda a transferência da sua ligação. Com o uso do Asterisk, isso não é mais necessário ele suporta vários formatos de música, inclusive MP3.

4.6.2.5. DISCADOR AUTOMÁTICO

É uma aplicação muito útil em telemarketing. Pode se programar o sistema para discar automático e distribuir numa fila. Mais uma tecnologia que é vendida separada em outro PABX. No Asterisk, pode se programar a discagem e existem diversos exemplos de discador disponíveis na Internet.

4.6.2.6. SALA DE CONFERÊNCIAS

Permite que vários usuários falem em conjunto. É implementado como sala de conferência. Basta escolhe um ramal para ser a sala de conferência e todos os que discarem para lá estarão automaticamente conectados. Há várias opções de configurações que podem ser implementadas, como senhas, por exemplo.

Estas são algumas das funcionalidades atuais do Asterisk. Novas aplicações estão surgindo a cada dia, com a contribuição de muitas pessoas ao redor do mundo.

4.6.2.7. PABX- SOFTSWITCH NO MODELO CONVENCIONAL

É comum nos dias de hoje, o uso de softswitches que são PCs que comutam circuitos de hardware na forma de interfaces padrão de telefonia. Entretanto, a forma de

comercialização destes equipamentos segue, muitas vezes, a lógica mostrada na figura 4 - 1. Todos os componentes são separados e, muitas vezes, de diferentes fabricantes. Em muitos casos, mesmo a tarifação é feita por um servidor separado, Os custos da aquisição de cada um destes componentes é elevado e a integração muitas vezes é difícil.

Durante muitos anos o mercado de telefonia foi ligado a equipamentos proprietários fabricados por grandes companhias multinacionais. Apesar de termos equipamentos de baixo custo nestas arquiteturas eles também apresentam baixa funcionalidade. Com a entrada do Asterisk, mais e mais empresas vão poder experimentar recursos como URA - unidade de resposta audível, DAC – distribuição automática de chamadas, mobilidade, correio de voz, e conferência, antes restritos às grandes companhias devido ao alto custo.

A telefonia IP quando atingir massa crítica fará com que o PABX de qualquer empresa possa falar com o PABX de qualquer outra através da Internet. O protocolo DUNDI é um primeiro ensaio nesta área. Na hora de avaliar os benefícios do Asterisk é preciso enxergar este horizonte futuro que são operadoras IP como a VONAGE, GVT, FreeWorldDialup e interligação automática com outros PABX. A economia em DDD e DDI é só uma das vantagens deste tipo de telefonia.

4.7. Arquitetura do Asterisk

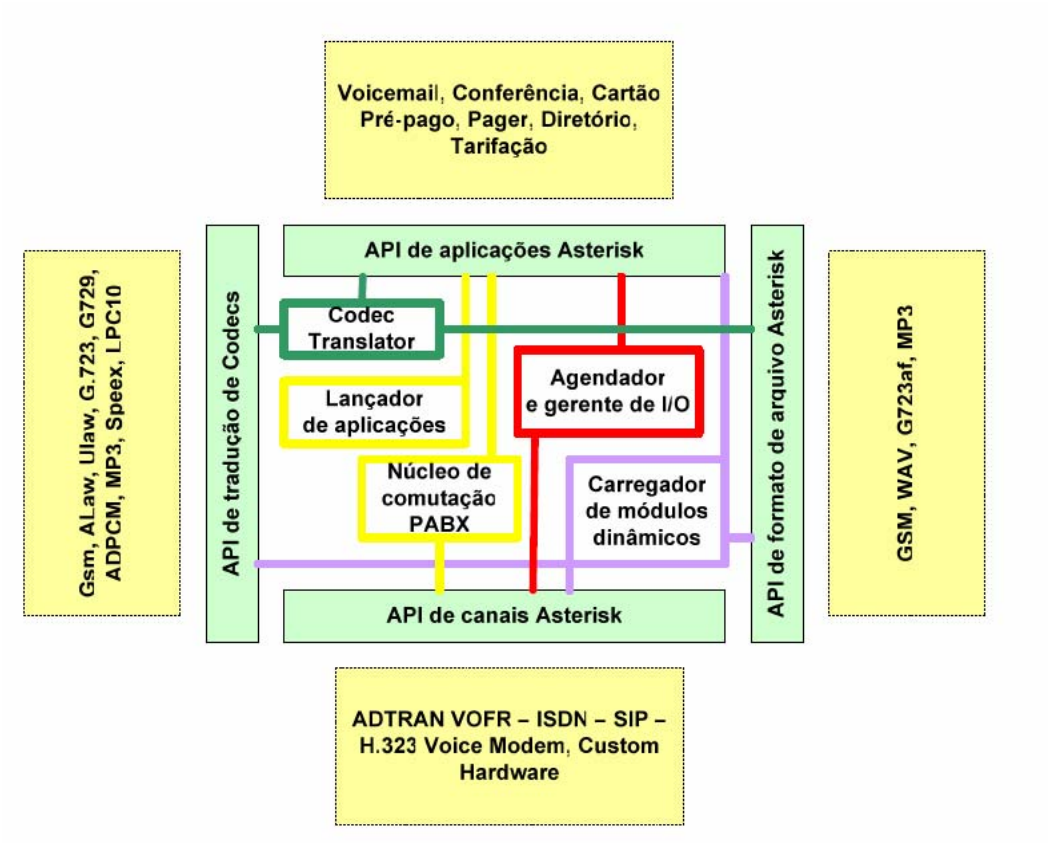

Figura 4-2 – Arquitetura do Asterisk [FLÁVIO, 2005][37]

[37] FLÁVIO Eduardo de Alechandre Gonçalves, 2005, **Asterisk PBX**

A figura acima mostra a arquitetura básica do Asterisk. Vamos explicar abaixo os conceitos relacionados à figura acima como canais, CODECS e aplicações.

4.7.1.CANAIS

Um canal é o equivalente à uma linha telefônica na forma de um circuito de voz digital. Ele geralmente consiste de um sinal analógico em um sistema POTS1 ou alguma combinação de CODEC e protocolo de sinalização (GM com SIP, Ulaw com IAX). No início as conexões de telefonia eram sempre analógicas e por isso mais suscetíveis aos ruídos e ao eco. Mais recentemente, boa parte da telefonia passou para o sistema digital, onde o sinal analógico é codificado na forma digital usando normalmente PCM (Pulse Code Modulation). Isto permite que um canal de voz seja codificado em 64 Kilobits/segundo sem compactação.

Alguns dos hardwares que o Asterisk suporta:

• Zaptel – Wildcard T410P – Placa E1/T1 com quatro portas (PCI 3.3 volts);
• Zaptel – Wildcard T405P – Placa E1/T1 com quatro portas (PCI 5 volts);
• Zaptel – TDM400P – Placa com quatro portas analógicas e ADSI,
• Zaptel - TE110P – Placa com E1/T1 com uma porta, meio comprimento.
• Quicknet, - as placas quicknet, tanto PhoneJack quanto LineJack;
• ISDN4Linux –driver antigo para placas ISDN BRI, acesso básico;
• ISDN CAPI – É a outra forma de suportar as placas ISDN BRI no Linux;
• Voicetronix: possui placas com maior densidade de canais FXS e FXO que as da Digium.

O Asterisk suporta os seguintes canais:

• Agent: Um canal de agente DAC;
• Console: Cliente de console Linux, driver para placas de som (OSS ou ALSA);
• H323: Um dos protocolos mais antigos de VoIP;
• IAX e IAX2: Inter-Asterisk Exchange protocol, o protocolo do Asterisk;
• MGCP: Media Gateway Control Protocol, outro protocolo de VOIP.
• Modem: Usado para linhas ISDN e não modems.
• NBS: Usado para broadcast de som.
• Phone: Canal de telefonia do Linux.
• SIP: Session Initiation Protocol, o protocolo de VoIP mais comum.
• Skinny: Um driver para o protocolo dos telephones IP da Cisco.
• VOFR: voz sobre frame-relay.
• VPB: Linhas telefônicas para placas da Voicetronix.
• ZAP: Para conectar telephones e linhas com placas da Digium. Também usado para TDMoE (TDM sobre Ethernet) e para o Asterisk zphfc (ISDN em modo NT).

Drivers que podem ser instalados, dentre outros:

• Bluetooth: Permite o uso de dispositivos Bluetooth para mudar o roteamento.
• CAPI: canal ISDN CAPI
• mISDN: canal mISDN channel
• SCCP: Um driver alternativo para o Skinny.

4.7.2. CODECS END CONVERSÕES DE CODEC

Pode ser maximazado o número de chamadas em uma rede de dados com a codificação em forma que use menos banda passante.

Este é o papel do CODEC (COder/DECoder), alguns CODECs como o g.729 permitem codificar à 8 Kilobits por segundo, uma compressão de 8 para 1. Outros exemplos são os ulaw, alaw, gsm, ilbc e g729.

O Asterisk suporta os seguintes CODECs:

• G.711 ulaw - (usado nos EUA) – (64 Kbps).
• G.711 alaw - (usado na Europa e no Brasil) – (64 Kbps).
• G.723.1 – Precisa de licenciamento (5.3-6 Kbps)
• G.726 – 32 kbps no Asterisk 1.0.3, 16/24/32/40 kbps no CVS HEAD.
• G.729 – Precisa de licença, a menos que esteja usando o modo passthru.Versão gratuita disponível para uso em países sem patentes ou para uso educacional. (8Kbps)
• GSM – (12-13 Kbps)
• iLBC – (15 Kbps)
• LPC10 - (2.5 Kbps)
• Speex - (2.15-44.2 Kbps)

4.7.3.PROTOCOLOS DE SINALIZAÇÃO

Enviar dados entre telefones seria fácil se os dados encontrassem seu próprio caminho para o destinatário. Infelizmente isto não acontece, é preciso um protocolo de sinalização para estabelecer as conexões, determinar o ponto de destino, e também questões relacionadas à sinalização de telefonia como campainha, identificador da chamada, desconexão etc. Hoje é comum o uso do SIP (Session Initiated Protocol), muito embora outros protocolos também sejam expressivos no mercado como o H.323, o MGCP e recentemente, o IAX que é excepcional quando se trata de trunking e NAT (Network Address Translation). O Asterisk suporta:

• SIP
• H323
• IAXv1 e v2
• MGCP
• SCCP (Cisco Skinny).

Conforme visto no capítulo passado o Protocolo SIP possui os três tipos de servidores espalhados pela rede de VoIP (servidor de registros, servidor Proxy e servidor de redirecionamento). Todos esses 3 servidores são implementados em um único Asterisk.

4.7.4. APLICAÇÕES

Para conectar as chamadas de entrada com as chamadas de saída ou outros usuários do Asterisk são usadas diversas aplicações como o Dial, por exemplo.

A maior parte das funcionalidades do Asterisk são criadas na forma de aplicações como o VoiceMail, correio de voz, o Meetme, conferência, entre outras.

4.8. INSTALANDO O SISTEMA OPERACIONAL E O ASTERISK

O Asterisk funciona em muitas plataformas e sistemas operacionais, mas para esse trabalho escolhemos a plataforma Linux Suse 9.2. As instruções abaixo podem funcionar com outras distribuições Linux, mas isso não foi testado. O Asterisk é conhecido por funcionar na maioria das distribuições.

4.8.1.HARDWARE MÍNIMO

O Asterisk pode fazer uso intensivo do processador, pois ele usa o próprio processador da maquina para fazer o processamento dos sinais digitais. Para construir um PABX para 30 ramais é necessário no minimo um processador de 300Mhz com 256 MB RAM. O Asterisk não requer muito espaço em disco, cerca de 100MB compilados, mais código fonte, voice-mail, prompts customizados e outras aplicações, todos requerem espaço.

4.8.2.INSTALAÇÃO DO SUSE 9.2

Para a instalação do sistema operacional o ideal é fazer a instalação mínima, sem ambiente gráfico e em inglês, com isso teremos uma ambiente mais leve e estável para o Asterisk.

Apos a instalação do sistema operacional é necessário desabilitar o firewall e adicionar os pacotes listados na tabela 4-1:

Pacotes Adicionais
Gcc – GNU C Compiler and support.
Cvs – Concurrent Versions System.
Ncurses – New curses libraries.
Curses-devel – Biblioteca para desenvolvimento com ncurses.
Bison – The GNU parser generator.
Termcap – Termcap library.
Openssl – Secure Sockets and TLS Layer Security.
Openssl-developer – Bibliotecas do openssl.
Zlib-devel

Tabela 4-1 - Pacotes Adicionais

Para concluir a instalação do sistema operacional é necessário configurar a placa de rede para poder atualizar o sistema operacional e baixar o Asterisk. Vale lembrar que por se tratar de um servidor o ideal é configurar um IP fixo.

4.8.3.OBTENDO, COMPILANDO, INSTALANDO E CONFIGURANDO O ASTERISK

Para baixar o Asterisk utilizaremos um servidor de CVS, que é um repositório central que os desenvolvedores usam para controlar o código fonte. Quando uma mudança é feita, ela é enviada para o servidor de CVS onde fica imediatamente disponível para baixar e compilar. Outro beneficio de usar o CVS é que se algo estava funcionando até um ponto, mas uma mudança fez com que parasse de funcionar, a versão de qualquer tipo de arquivo em particular pode ser retornada a certo ponto.

4.8.3.1. BAIXANDO O ASTERISK

- Baixe o arquivo do servidor de CVS para o diretório /usr/src
- Export CVSROOT=:pserver:anoncvs@cvs.digium.com/usr/cvsroot
- Cvs login
- Password is anoncvs

Cvs checkout –r v0-1 Asterisk libpri Asterisk-sounds ASTERISK-ADDONS.

4.8.3.2. COMPILANDO

- Compilar o Asterisk é muito similar aos outros programas para linux.
- cd/usr/erc/Asterisk/
- make clean
- make
- make install
- make samples

4.8.3.3. ARQUIVOS DE INICIALIZAÇÃO DO ASTERISK

Antes de usar o Asterisk, tem que criar os arquivos de configuração. Embora a quantidade de configurações seja muito grande, apenas um pequeno conjunto é necessário de forma a iniciar o Asterisk com sucesso.

Para iniciar o Asterisk em tempo de inicialização utilizamos do anexo1 abaixo:

1 Entre como ROOT
2 Salve o script como /etc/init.d/Asterisk
3 Tem que se criar um link simbólico de /etc/init.d/rc3.d para o script de forma que o Asterisk inicie no boot do sistema. (in –s ../Asterisk S90Asterisk)
4 Tem que se criar um link simbólico de /etc/init.d/rc0.d para o script de forma que o Asterisk seja descarregado na saída do sistema. (in –s ../Asterisk K10Asterisk)

4.8.3.4. CONFIGURANDO O ASTERISK COMO SERVIDOR SIP.

Para o projeto foram configurados 2 ramais utilizando protocolo SIP.

Para criar um ramal é necessário configurar uma entrada no arquivo sip.conf. Segue o arquivo editado para esse projeto.

:Configuração geral do Arquivo

```
 [general]
context=default
port=5060
bindaddr=10.61.217.90
srvlookup=yes


;Entradas dos ramais


[4102]
type=friend
username=4102
canreinvite=yes
disallow=all
callerid=flavio Carneiro
host=dynamic
nat=yes
allow=gsm
allow=ulaw


[4103]
type=friend
username=4103
can reinvite=yes
disallow=all
callerid=rodrigo
host=dynamic
nat=yes
allow=gsm
allow=ulaw
```

O arquivo completo do sip.conf se encontra no anexo 2

A posibilidade de configurações para o arquivo sip.conf é muito grande. Exitem vários parametros que podem ser atribuidos.

4.8.3.5. CONFIGURANDO O PLANO DE DISCAGEM NO ASTERIS

O plano de discagem é a peça mais importante na configuração do Asterisk e ele é configurado no arquivo extensions.conf. Ele controla como todas as chamadas de entrada e saída que são encaminhadas e manuseadas. Editando o arquivo extensions.conf é possível controlar o comportamentos das ligações atraves do PABX.

Para criar o plano de discagem é necessário configurar o arquivo extensions.conf. Segue o arquivo editado para esse projeto.

[default]

exten=>4102,1,Dial(SIP/4102,20)
exten=>4103,1,Dial(SIP/4103,20)

O arquivo completo do extensions.conf se encontra no anexo 3

4.8.3.6. CONFIGURAÇÕES ADICIONAIS.

Para fazer uso somente de VoIP, nenhum outro hardware é necessário.

Serão utilizados softfones da X-TEN (X-Lite versão gratuita) que será utilizado como ramais da rede VoIP. A sua instalação e configuração constam do Anexo 4.

4.9. INICIANDO O ASTERISK

Após a instalação e configuração do Asterisk pdemos verificar no register quais os telefones que já se registraram. Para isso, é necessário entrar via linha de comando do Asterisk.

A figura abaixo traz a tela do register, Através dela é possivel verificar que existem 4 ramais configurados no Asterisk: 3101, 4102, 4103 e 4104; porém nenhum ramal está registrado. Isso se deve ao fato de que não há nenhum telefone ligado ou configurado.

```
linux:/etc/asterisk # asterisk -r status
Asterisk CVS-v1-0-05/20/05-21:29:03, Copyright (C) 1999-2004 Digium.
Written by Mark Spencer <markster@digium.com>
=========================================================================
Connected to Asterisk CVS-v1-0-05/20/05-21:29:03 currently running on linux (pid
 = 5499)
linux*CLI> sip show peers
Name/username    Host            Dyn Nat ACL Mask            Port     Status

4104/cisco       (Unspecified)    D   N      255.255.255.255  0        Unmonitor
ed
4103/4103        (Unspecified)    D   N      255.255.255.255  0        Unmonitor
ed
4102/4102        (Unspecified)    D   N      255.255.255.255  0        Unmonitor
ed
4101/grandstrea  (Unspecified)    D          255.255.255.255  0        Unmonitor
ed
linux*CLI>
```

Figura 4-3 – Registro no Asterisk

Apos ligar um dos telefones que estavam configurados podemos verificar na figura 4-4 que o telefone 4102 foi registrado com o IP 10.61.215.51 na porta 5060 e está pronto para uso. Isso ocorrerá com totos os telefones que foram configurados e ligados.

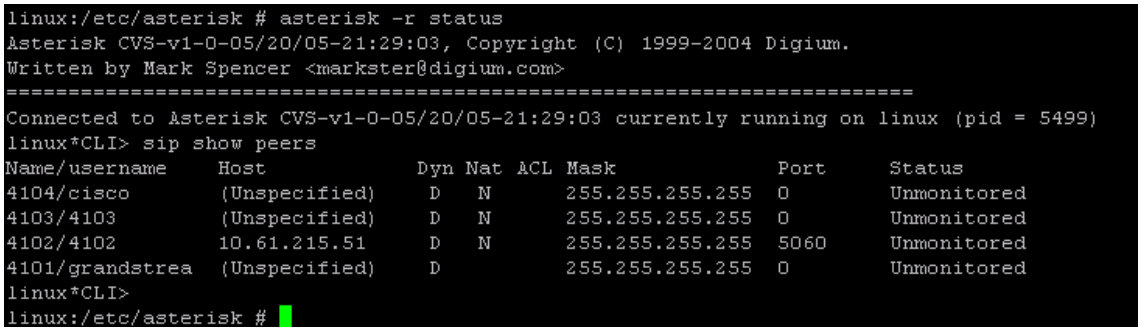

Figura 4-4 – Registro do telefone 4102

Se o telefone configurado não estiver registrado ele não existe para o Asterisk, logo não será possível efetuar e receber ligações neste telefone.

Ao ligar e registrar os 2 telefones já é possivel iniciar os testes com de telefonia IP que foi proposto nete trabalho.

Capítulo 5 - Cluster de Alta-disponibilidade

5.1. PRINCÍPIOS BÁSICOS DOS CLUSTERS

Com a crescente dependência de serviços oferecidos através da Internet, muitos se tornaram críticos, tanto em termos de disponibilidade quanto em termos de desempenho.

Falhas destes tipos de serviços implicam na maioria das vezes em perda de receita. Tais serviços precisam prover mecanismos que os mantenham disponíveis na presença de falhas. É nesse contexto que se encontram os clusters.

Na sua forma mais básica um cluster é um sistema que compreende dois ou mais computadores ou sistemas (denominados nós) no qual trabalham em conjunto para executar aplicações ou realizar outras tarefas, de tal forma que os usuários que os utilizam tenham a impressão que somente um único sistema responde para eles, criando assim uma ilusão de um recurso único (computador virtual). Este conceito é denominado transparência do sistema.

5.2. TIPOS DE CLUSTERS

Existem vários tipos de cluster como:

a) alta-disponibilidade;
b) balanceamento de Carga; e
c) processamento Paralelo.

Porém, esse projeto irá tratar exclusivamente do cluster de alta-disponibilidade que foi utilizado no projeto.

5.2.1. ALTA-DISPONIBILIDADE

Um cluster com alta-disponibilidade é um agrupamento de sistemas tendo redundância suficiente de componentes de software e hardware, que em caso de falha não colocarão em risco a disponibilidade do serviço que se pretende fornecer. [Peter, 1998][38]

Em qualquer negócio dependente de sistemas computacionais verifica-se que:

a) o nível de alta-disponibilidade requerida é determinada pelas necessidades do próprio negócio;

b) os meios de alcançar a alta-disponibilidade afetam todos os aspectos do sistema;

[38] Peter S. Weygant (1998). Clusters for High Availabitity. Prentice Hall PTR, Prentice-Hall, Inc.

c) a probabilidade de falha pode ser reduzida criando uma infra-estrutura que defina procedimentos claros e recorra à manutenção preventiva;

d) todas as falhas possíveis devem ser identificadas; e

e) é muito importante existirem planos de recuperação de falhas.

Por sua vez, as aplicações de software que operam em ambientes críticos devem satisfazer as seguintes características:

a) a taxa de falhas das aplicações deve ser mínima, isto é, o intervalo entre falhas deve ser máximo.

b) as aplicações devem providenciar a auto-recuperação imediata de falhas;

c) o "DownTime" previsto deve ser mínimo;

d) o sistema deve ser configurável sem interrupção total; e

e) as aplicações devem ser geridas facilmente por ferramentas de gestão.

É natural que o custo da alta-disponibilidade dependa do grau de disponibilidade. Assim, o custo da Alta-Disponibilidade (AD) para o negocio está diretamente relacionado com o custo das interrupções do serviço. Quanto maior for este, mais fácil se justifica o custo das soluções de AD.

Esta arquitetara que tenta minimizar os pontos únicos de falha, tem obviamente vantagens em relação ao sistema que se baseie em sistemas únicos. Sendo assim, as vantagens mais relevantes são:

a) "DownTime" controlado;

b) facilidade em re-estruturar o sistema (software, hardware) sem degradação do nível de serviço; e

c) esta solução tem níveis de "DownTime" muito próximos dos sistemas de "Fault Tolerance".

5.2.1.1. CONCEITO

Para que se entenda a Alta-Disponibilidade faz-se necessário, antes de tudo, perceber que a AD não é apenas um produto ou uma aplicação e sim uma característica de um sistema computacional. Existem mecanismos e técnicas, que podem ser utilizados para aumentar a disponibilidade de um sistema.

A Disponibilidade de um sistema computacional, indicada por A(t), é a probabilidade de que este sistema esteja funcionando e pronto para uso em um dado instante de tempo t. Esta disponibilidade pode ser enquadrada em três classes, de acordo com a faixa de valores desta probabilidade. As três classes são: Disponibilidade Básica, Alta-Disponibilidade e Disponibilidade Contínua. De maneira geral a classificação de sistemas por disponibilidade é feita de acordo com a quantidade de "9s" na disponibilidade desses sistemas, conforme tabela 5-1.

Classificação Probabilidade de estar	Probabilidade de estar Disponível.	Tempo Aproximado de Indisponibilidade
Disponibilidade Básica M	Entre 99 e 99,9 %	Menos de 9 horas p/ ano
Alta-Ddisponibilidade	Entre 99,99 e 99,999%	Menos de 1 hora p/ano
Disponibilidade Contínua	Acima de 99,999%	Menos de 5 minutos p/ano

Tabela 5-1 - Classificação de sistemas quanto à sua disponibilidade.

A Disponibilidade Básica é aquela encontrada em máquinas comuns, sem nenhum mecanismo especial, em software ou hardware, que vise de alguma forma mascarar as eventuais falhas destas máquinas. Costuma-se dizer que máquinas nesta classe apresentam uma disponibilidade de 99% a 99,9%. Isto equivale a dizer que em um ano de operação a máquina pode ficar indisponível por um período de 9 horas a quatro dias. Estes dados são empíricos e os tempos não levam em consideração a possibilidade de paradas planejadas, porém são aceitas como o senso comum na literatura da área.

Com a adição de noves se obtém uma disponibilidade cada vez mais próxima de 100%, diminuindo o tempo de inoperância do sistema de forma que este venha a ser desprezível ou mesmo inexistente. A Disponibilidade Contínua significa que todas as paradas planejadas e não planejadas são mascaradas e o sistema está sempre disponível.

O principal objetivo da AD é buscar uma forma de manter os serviços prestados por um sistema a outros elementos, mesmo que o sistema em si venha a se modificar internamente por causa de uma falha. Aí está implícito o conceito de mascaramento de falhas, através de redundância ou replicação.Um determinado serviço, que se quer altamente disponível, é colocado por trás de uma camada de abstração, que permita mudanças em seus mecanismos internos mantendo intacta a interação com elementos externos.

Este é o núcleo da Alta-disponibilidade, uma sub-área da Tolerância a Falhas, que visa manter a disponibilidade dos serviços prestados por um sistema computacional, através da redundância de hardware e reconfiguração de software. Vários computadores juntos agindo como um só, cada um monitorando os outros e assumindo seus serviços caso perceba que algum deles falhou.

Para se entender corretamente do que se está falando quando se discute uma solução de AD, deve-se conhecer os conceitos envolvidos. Não são muitos, porém estes termos são muitas vezes utilizados de forma errônea em literatura não especializada. Deve-se entender o que é falha, erro e defeito. [Milz, 2002][39]

[39] MILZ, Harald**. High Availability**

5.2.1.1.1. FALHA

Uma falha acontece no universo físico, ou seja, no nível mais baixo do hardware. Uma flutuação da fonte de alimentação, por exemplo, é uma falha. Uma interferência eletromagnética também. Estes são dois eventos indesejados, que acontecem no universo físico e afetam o funcionamento de um computador ou de partes dele.

5.2.1.1.2. ERRO

A ocorrência de uma falha pode acarretar um erro, que é a representação da falha no universo informacional. Um computador trabalha com bits, cada um podendo conter 0 ou 1. Uma falha pode fazer com que um (ou mais de um) bit troque de valor inesperadamente, o que certamente afetará o funcionamento normal do computador. Uma falha, portanto, pode gerar um erro em alguma informação.

5.2.1.1.3. DEFEITO

Já esta informação errônea, se não for percebida e tratada, poderá gerar o que se conhece por defeito. O sistema simplesmente trava, mostra uma mensagem de erro, ou ainda perde os dados do usuário sem maiores avisos. Isto é percebido no universo do usuário.

Recapitulando, uma falha no universo físico pode causar um erro no universo informacional, que por sua vez pode causar um defeito percebido no universo do usuário. A Tolerância a Falhas visa exatamente acabar com as falhas, ou tratá-las enquanto ainda são erros. Já a AD permite que máquinas continuem funcionando, mesmo quando existam problemas.

Para que uma máquina assuma o lugar de outra, é necessário que descubra de alguma forma que a outra falhou. Isso é feito através de testes periódicos, cujo período deve ser configurável, nos quais a máquina secundária testa se a outra está ativa. Por serem periódicos, nota-se que existe um intervalo de tempo durante o qual o sistema pode estar indisponível sem que a outra máquina o perceba.

5.2.1.1.4. FAILOVER

O processo no qual uma máquina assume os serviços de outra, quando esta última apresenta falha, é chamado *failover*, que pode ser automático ou manual. Para a AD o failover deve ser do tipo automático.

Algumas aplicações não-críticas podem suportar um tempo maior até a recuperação do serviço e, portanto, podem utilizar *failover* manual. Além do tempo entre a falha e a sua detecção, existe também o tempo entre a detecção e o reestabelecimento do serviço.

Dependendo da natureza do serviço, executar um *failover* significa interromper as transações em andamento, perdendo-as, sendo necessário reiniciá-las após o *failover*. Em outros casos, significa apenas um retardo até que o serviço esteja novamente disponível. Nota-se que o *failover* pode ou não ser um processo transparente, dependendo da aplicação envolvida.

5.2.1.1.5. FAILBACK

Ao ser percebida a falha de um servidor, além do *failover*, é necessária a manutenção desse servidor. Ao ser recuperado de uma falha, este servidor será recolocado em serviço e, então, se tem a opção de realizar um processo inverso ao *failover*, que se chama *failback*, que é o processo de retorno de um determinado serviço de uma outra máquina para sua máquina de origem. Também pode ser automático, manual ou até mesmo indesejado. Em alguns casos, em função da possível nova interrupção na prestação de serviços, o *failback* pode não ser atraente.

5.2.1.1.6. MISSÃO

Quando se calcula a disponibilidade de um sistema, é importante que se observe o conceito de missão. Missão de um sistema é o período de tempo no qual ele deve desempenhar suas funções sem interrupção. Por exemplo, uma empresa que faz uso de um sistema de telefonia IP, que funcione das 8h às 20h, não pode ter seu sistema fora do ar durante este período de tempo. Se este sistema vier a apresentar defeitos fora deste período, ainda que indesejados, estes defeitos não atrapalham em nada o andamento correto do sistema quando ele é necessário. Uma empresa que faz uso de um sistema de telefonia IP 24h obviamente tem uma missão contínua, de forma que qualquer tipo de parada deve ser mascarada.

A AD visa eliminar as paradas não planejadas. Porém, no caso da primeira empresa, as paradas planejadas não devem acontecer dentro do período de missão. Paradas não planejadas decorrem de defeitos, já paradas planejadas são aquelas que se devem a atualizações, manutenção preventiva e atividades correlatas. Desta forma, toda parada dentro do período de missão pode ser considerada uma falha no cálculo da disponibilidade.

Uma aplicação de Alta-Disponibilidade pode ser projetada inclusive para suportar paradas planejadas, o que pode ser importante, por exemplo, para permitir a atualização de programas por problemas de segurança, sem que o serviço deixe de ser prestado.

5.2.1.1.7. MONITORAÇÃO DOS NODOS

Um grupo de recursos consiste em uma entidade lógica composta pela aplicação crítica (serviço), um endereço IP e um dispositivo de armazenamento utilizado pela aplicação, seja ele compartilhado ou não.

Failover é o nome dado ao conjunto de operações realizadas no cluster para a transferência do grupo de recursos do nó falho para um nó backup, mantendo assim a continuidade do serviço. Estas operações devem ser realizadas de tal forma, que o tempo de indisponibilidade do serviço seja o menor possível. Isto porque mover um grupo de recursos de um nó para outro consiste em reiniciar a aplicação em outro nó.

Um grupo de recursos será executado pelo nó ativo até que seja identificada uma falha que o impossibilite de continuar a sua execução. Esta falha é identificada através de mecanismos de monitoração de nós executados no nó ativo, pela emissão de sinais periódicos, através de interfaces seriais ou de rede, para o nó backup indicando que o nó ativo permanece executando o grupo de recursos.

Para realizar o monitoramento mútuo entre dois nós pode-se utilizar algumas soluções como por exemplo: verificar, através do uso do “ping” no IP o estado da outra máquina.

Uma ferramenta que faz o monitoramento dos nós e consegue solucionar os problemas enunciados acima é um programa denominado Heartbeat.

Como o próprio nome sugere, o Heartbeat funciona como um pulso cardíaco que avisa o nó secundário que o primário está vivo, como ilustrado na figura abaixo.

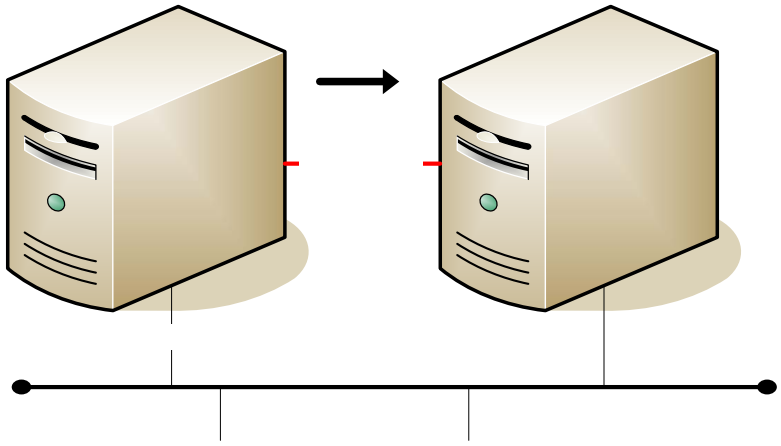

Figura 5-1 - Operação do Heartbeat

Server Primário

O pulso do Heartbeat pode ser transmitido através de uma interface serial, ou ethernet e há possibilidade de se enviar diversos pulsos simultaneamente. De fato é aconselhável ter duas interfaces de pulso, uma serial e uma ethernet.

Ping

A funçaõ do Heartbeat, além de enviar o pulso, é inicializar e desligar os serviços que o cluster estará oferecendo, assim como o IP do cluster.

Claro que neste tipo de configuração não se deseja que as máquinas iniciem os serviços logo na inicialização, porque o nó secundário não deve assumir tais serviços a menos que o nó primário caia.

Heartbeat

Nos arquivos de configuração se diz ao programa quais são os serviços pelos quais ele é responsável e, no nó primário serão inicializados de imediato, mas no nó secundário, só será inicializado o monitor do pulso e apenas em caso de falha de recebimento do pulso após um timeout estipulado os serviços serão assumidos e o IP será tomado.

5.3. IMPLEMENTAÇÃO

IP virtual

Será descrito como cada um dos componentes envolvidos na solução de alta-disponibilidade se relacionam e como são implementados e configurados utilizando Software Livre.

5.3.1. TOPOLOGIA

A idéia dessa implementação é aliar o conceito de alta-disponibilidade reduzindo-se ao máximo os custos. Com isso foi idealizado o cenário com dois computadores que fariam o papel de Call Manager. A figura abaixo resume a topologia da solução.

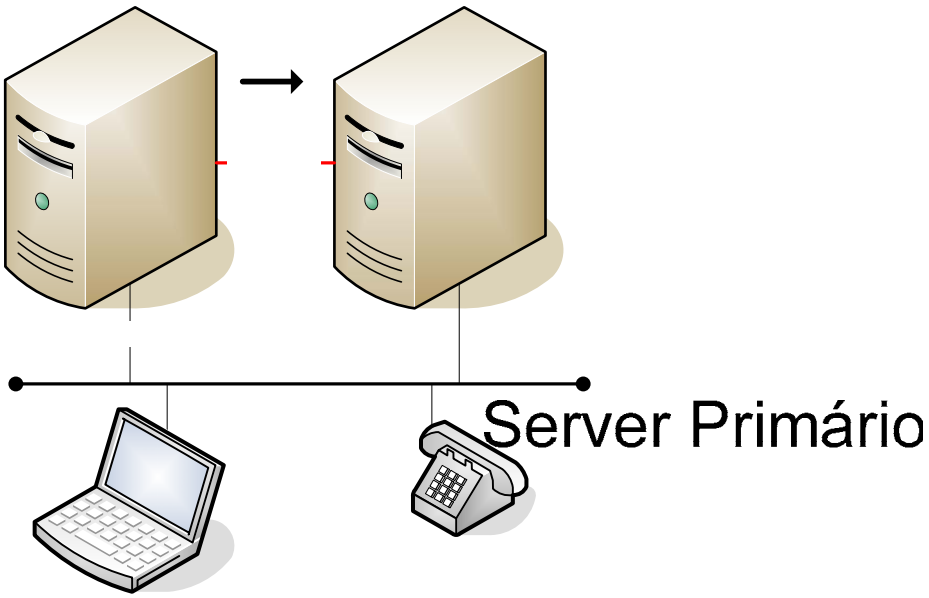

Figura 5-2 - Topologia da Solução implementada

5.3.2.HEARTBEAT

O Heartbeat da Linux-HA é o responsável pela implementação em código aberto do algoritmo de heartbeat é usado para construir clusters de altíssima disponibilidade. Ele pode fazer 2 nós IP com capacidade de assumirem os recursos e serviços de um número ilimitado de interfaces IP. Ele trabalha enviando um 'heartbeat' entre 2 máquinas através de ethernet. Se o heartbeat falhar, a máquina secundária irá detectar que a primária falhou, e assumir os serviços que estavam rodando na maquina primária.

Utilizar-se-ão duas interfaces ethernet para divulgação dos pacotes de heartbeat. A interface disponível aos usuários (eth0) e uma interface dedicada de conexão entre os dois nos do cluster (eth1). O heartbeat será o responsável pela alta-disponibilidade do serviço.

O Heartbeat é mantido pelo Projeto Linux High Availability ou Linux –HA. Sendo que algumas distribuições já o possuem de forma nativa ao sistema.

5.3.3.INSTALANDO

5.3.3.1. BAIXANDO OS PACOTES

- O software do heartbeat pode ser encontrado em http://linux-ha.org/download/.

Sistema Operacional	*Pacotes SUSE 9.1 RPMS for i586*
Suse 9.2	heartbeat-1.2.3-1.i586.rpm
	heartbeat-pils-1.2.3-1.i586.rpm
	heartbeat-stonith-1.2.3-1.i586.rpm
	heartbeat-ldirectord-1.2.3-1.i586.rpm

Tabela 5-2 - Pacotes Utilizados

- Execute os arquivos RMP da tabela 5-2 para fazer as Instalações.

5.3.3.2. ARQUIVOS DE CONFIGURAÇÃO

Assim como quase todos os programas Linux, o heartbeat é composto por arquivos de configuração, estando localizados em */etc/ha.d/*. São eles: ha.cf, haresources e authkeys.

5.3.3.2.1. ha.cf

O ha.cf é o responsável pelas configurações de funcionamento do próprio heartbeat. A tabela abaixo apresenta todas as possíveis opções existentes.

Opção		Definição
debugfile	/var/log/hadebug	Definição dos arquivos e níveis de log
logfile	/var/log/ha-log	
logfacility	local0	
keepalive	2	Intervalo em segundos dos pacotes do heartbeat
deadtime	30	Tempo necessário para indicar que um nodo está indisponível
warntime	10	Tempo necessário para o último aviso de late heartbeat
initdead	120	Tempo máximo de subida do sistema
nice_failback	on	Comportamento do Failback. "On" significa que o recurso não retornará a máquina de origem quando esta restabelecer a comunicação
hopfudge	1	Numero máximo hops menos o numero de nodos
baud	19200	Velocidade da interface serial

serial	/dev/ttyS0	Definição da interface serial para o Heartbeat
udpport	694	Porta UDP utilizada pelo Heartbeat
bcast	eth0 eth1	Interfaces onde serão enviados broadcasts de pacotes heartbeat
mcast	[dev] [mcastgroup] [port] [ttl] [loop]	Configuração para envio de pacotes heartbeat via multicast
ucast	[dev] [peer-ipaddr]	Configuração para envio de pacotes heartbeat via unicast
watchdog	/dev/watchdog	Mecanismo para identificar que o próprio sistema não está enviando pacotes de heartbeat
stonith	baytech/etc/ha.d/conf/stonith.baytech	Configuração do dispositivo de controle utilizando Stonith
node	nodename	Definição dos nodos pertecentes ao cluster
ping	[Endereço IP]	Configuração do pseudo-cluster, utilizado para monitoração
respawn	userid /path/name/to/run	Comandos de start/stop iniciados pelo heartbeat

Tabela 5-3- Opções de configuração do ha.cf

O arquivo completo do ha.cf se encontra no anexo 5

5.3.3.2.2. HARESOURCE

Uma vez que você configurou o ha.cf, você precisa configurar o arquivo haresources. Este arquivo especifica os serviços para o cluster e quem é o proprietário padrão. Nota: Este arquivo deve ser igual nos dois nos!

O arquivo haresource define basicamente a lista de recursos que serão migrados de um nodo a outro. Ele deve ser idêntico em todos os nos pertencentes ao cluster.

Opção		Definição	
node-name	resource1 resource2 ... resourceN	Define a lista de recursos e nodo preferencial	

Tabela 5-4 - Arquivo Haresource

Para nosso exemplo, assumi-se que o serviço é o Asterisk. O endereço IP para o cluster é obrigatório, e não pode configurar o IP do cluster fora do arquivo haresources! O arquivo irá precisar de uma linha:

linuxha1.linux-ha.org 192.168.85.3 asterisk

Então, esta linha diz que ao iniciar, linuxha1 serve o IP 10.61.217.95 e inicia o Asterisk. Ao desligar, o heartbeat irá primeiro parar o Astersik e depois liberar o IP.

Nota: asterisk é o nome do script que inicia o asterisk. Esse escript está no anexo 1. O heartbeat irá procurar por scripts com o mesmo nome nós seguintes lugares:

/etc/ha.d/resource.d
/etc/rc.d/init.d

O nodo preferencial é utilizado para identificar para qual nodo o recurso será migrado em uma configuração com nice_failback desabilitado.

5.3.3.2.3. AUTHKEYS

O terceiro arquivo de configuração é o authkeys. Ele determina a autenticação entre os nós. Os três métodos de autenticação disponíveis são crc, md5 e sha1. Se o heartbeat irá rodar sobre uma rede segura, uma conexão dedicada, o crc é o método mais barato do ponto de vista de recursos. Se a rede é insegura deve-se optar por utilizar o md5 ou sha1 atentando-se ao fato de que o consumo de CPU necessário para esses métodos é mais intenso.

O formato do arquivo é simples:
<number>
<number> <authmethod> [<authkey>]

É necessário configurar as permissões deste arquivo de forma segura, como 600 (direitos de leitura e gravação somente para o usuário privilegiado do sistema e dono do arquivo).

5.3.3.3. INICIALIZANDO E TESTANDO O HEARTBEAT

O heartbeat será inicializado automaticamente pelo sistema através do comando /etc/init.d/heartbeat start.

Pode-se verificar os eu funcionamento observando a configuração dos IPs das interfaces. Na figura 5-3 se nota que nó 1 possui um endereço IP secundário (eth0:0) que é justamente o IP virtual do cluster.

```
[root@node1 root]# ifconfig -a | more
eth0      Link encap:Ethernet  HWaddr 00:0C:29:3C:B8:6D
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:869 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7045 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:74030 (72.2 Kb)  TX bytes:611681 (597.3 Kb)
          Interrupt:10 Base address:0x10c0

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:3C:B8:6D
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:10 Base address:0x10c0
```

Figura 5-3 - Configuração de rede : Nó 1

```
[root@node2 root]# ifconfig -a | more
eth0      Link encap:Ethernet  HWaddr 00:0C:29:CA:E7:F6
          inet addr:192.168.2.3  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1050 errors:0 dropped:0 overruns:0 frame:0
          TX packets:930 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:104863 (102.4 Kb)  TX bytes:72515 (70.8 Kb)
          Interrupt:10 Base address:0x10c0
```

Figura 5-4 - Configuração da rede : Nó 2

Pode-se acompanhar o que está acontecendo pelos logs, que estão em /var/log/ha-log. Em caso de erro, deve-se verificar a causa nesse arquivo.

O heartbeat agora irá cuidar da inicialização dos serviços que foi configurado no haresources. Caso a máquina principal congele, a secundária irá assumir e iniciar os mesmos recursos.

O tempo entre a parada do servidor PABX01 e a entrada do servidor PABX02 é de 60s, porém devido ao tempo de limpeza da tabela ARP o tempo total pode chegar a 2 Minutos. Para retornar o serviço do PABX02 para o PABX01 o tempo é de aproximadamente 2 Minutos.

Capítulo 6 - Conclusão

O inicio deste projeto foi voltado para identificação de soluções que permitissem aumentar a disponibilidade de um sistema de voz sobre IP, levando em conta uma racionalização dos custos envolvidos na área de TI, mas mantendo a qualidade dos serviços prestados.

Depois de passado o primeiro mês de projeto, o trabalho se concentrou na implementação de uma solução de alta-disponibilidade com a utilização de Software Livre, se encaixando, portanto, na idéia inicial de se obter soluções de qualidade á um baixo custo de implementação.

A primeira dificuldade surgiu no momento de se encontrar uma solução em Software Livre para o ambiente VoIP. Praticamente todas as soluções de voz sobre IP implementadas no mercado são proprietárias e muito caras. O Asterisk foi quem possibilitou a redução de custos na implementação do projeto.

Uma limitação encontrada no ambiente foi o fato de que mesmo com os servidores em cluster há uma pequena indisponibilidade, devido ao fato dos telefones terem que se registrar no Register. Com isso, quando há uma queda no ambiente e o segundo servidor tem que assume a aplicação e os telefones se registram novamente. Essa indisponibilidade é muito menor que se tivesse que recuperar o primeiro servidor e coloca-lo em produção, logo o trabalho atingiu o seu objetivo aumentando a disponibilidade do sistema.

Uma outra limitação consiste no fato de que não estamos monitorando todos os possíveis problemas que podem ocorre nesse ambiente. Um bom exemplo para esse tipo de problema e que, se a aplicação ao invés de cair apenas congelar, o monitoramento, não acusará isso como um indisponibilidade do ambiente pois o serviço ainda esta rodando, por isso, o cluster não irá migrar a interface de rede para o outro servidor.

Observando as limitações encontradas no decorrer do trabalho, verifica-se que o próximo passo em ser seguido, em um projeto futuro, seria o estudo de um aprimoramento da solução de alta-disponibilidade e monitoramento, tentado, assim, eliminar essa pequena indisponibilidade, aumentando ainda mais a confiabilidade no serviço.

Ao fim do projeto foi possível fazer de fato uma solução com uma maior disponibilidade utilizando apenas componentes não proprietários, reduzindo assim os custos de implementação.

PROJETOS SEQUENCIAIS

I - Aprimoramento da solução de alta-disponibilidade e monitoramento, para eliminar o tempo de indisponibilidade.

II - Integração do Asterisk ao Sistema de Telefonia Público.

III - Desenvolver uma interface web de administração remota para o Asterisk.

BIBLIOGRÁFIA

ALENCAR, Marcelo Sampaio de. **Telefonia digital**. São Paulo: Érica, 1998

ALVES,Victor Manuel Golçalves.**Apresentaço e análise da *Ip Telephony***, Porto, out 2002. Disponível em: http://www.inescn.pt/~jneves/feup/mrsc-2001/sm/trabalhos.html. Acesso em: 02 jun. 2002.

BURKE, T.et al. **Misson Critical Linux Kimberlite Design Specification**. Disponível em http://oss.missioncriticallinux.com/proje cts/kimberlite/.

FERNANDES, Nelson Luiz Leal.**Voz sobre Ip**:Uma visão geral,Rio de Janeiro, fevereiro, [2003]. Disponível em: http://www.ravel.ufrj.br/publicacoes/tipos.php?IdTipo=4. Acesso em: 01 jun 2003.

Flávio Eduardo de Alechandre Gonçalves (2005) Gia de configuração do Asterisk

FURLAN, Jose Davi. **Modelagem de objetos através da UML-The Unifield Modeling Language**. São Paulo: Makron Books, 1998. 329 p.

HERSENT, Olivier; GURLE, David; PETIE, Jean Pierre. **Telefonia IP**. São Paulo: Addison, 2002.

http://www.linuxha.org/comm/HBdesign.pdf

http://www.linux-ha.org - Linux-HA Project

IPVS Connection Synchronization http://www.linuxvirtualserver.org/docs/sync.html

Jon Hall, the president of open-source organization Linux International.

MACK,Josep L**VS howto** http:// www.linuxvirtualserver.org/Joseph.Mack/HOWTO/index.html.

MACK, Joseph **LVS Mini-Howto** http://www.linuxvirtualserver.org/Joseph.Mack/mini-HOWTO/LVSmini- HOWTO.html.

MILZ, Harald**. High Availability HOWTO**.http://www.ibiblio.org/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html.

OLIVEIRA, Sérgio. Telefonia IP para ambientes móveis usáveis: Simpósio Brasileiro de Redes de Computadores, 19., 2001, Florianópolis. **Anais...** Florianópolis: UFSC, 2001. p. 542-558.

Peter S. Weygant (1998). Clusters for High Availabitity. Prentice Hall PTR, Prentice-Hall, Inc.

PRAMANICK, I. **IEEE Task Force on Cluster Computing - High Availability**. Disponível em: http://www.cs.mu.oz.au/ raj/tfcc/high-availability.html.

ROBERTSON, Alan**. Linux-HA Heartbeat Design.**

Site oficial do Heartbeat. http://www.linux-ha.org

SILVA, Arlindo Maia da; RAMOS, Luís.**Serviços de Multimídia**, Porto, fev 2002. Disponível em: Acesso em: 10 mar. 2002.

SOARES, Luiz Fernando G; LEMOS, Guido; COLHER, Sérgio. **Redes de computadores:** das LANs, MANs e WANs às redes ATM. 2. ed. Rio de Janeiro: Campus, 1995.

SOUZA, José Marcio de. Protótipo de um sitema de VoIP(Voz sobre IP). 2001. 62 f. Trabalho de Conclusão de Curso(Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

ZHANG, W. **Linux Virtual Server (LVS)**. Disponível em: http://www.LinuxVirtualServer.org/.

Anexo 1

Script de inicialização do Asterisk

```
#########################################################################
#!/bin/sh
#
# Asterisk       This shell script takes care of starting and stopping Asterisk.
#
# 08.april.2004 - Modified to be used on a SuSE Linux system by Flávio Carneiro.
#
#
# Source function library.
# Shell functions sourced from /etc/rc.status:
#      rc_check         check and set local and overall rc status
#      rc_status        check and set local and overall rc status
#      rc_status -v     ditto but be verbose in local rc status
#      rc_status -v -r  ditto and clear the local rc status
#      rc_failed        set local and overall rc status to failed
#      rc_failed <num>  set local and overall rc status to <num><num>
#      rc_reset         clear local rc status (overall remains)
#      rc_exit          exit appropriate to overall rc status
. /etc/rc.status

ASTERISK_BIN=/usr/sbin/Asterisk
ASTERISK_LOCK=/var/lock/subsys/Asterisk
ASTERISK_PID=/var/run/Asterisk.pid
ASTERISK_OPTS="-nqg"         # -n: Disable console colorization
                             # -q: Quiet mode (supress output)
                             # -g: Dump core in case of a crash

[ -f $ASTERISK_BIN ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
  start)
        # Start daemons.
        echo -n "Starting Asterisk PBX: "
        startproc $ASTERISK_BIN $ASTERISK_OPTS
      RETVAL=$?
        [ $RETVAL -eq 0 ] && touch $ASTERISK_LOCK
      rc_status -v
        ;;
  stop)
        # Stop daemons.
        echo -n "Shutting down Asterisk PBX: "
        killproc Asterisk
      RETVAL=$?
```

```
        rc_status -v
        [ $RETVAL -eq 0 ] && rm -f $ASTERISK_LOCK
        ;;
    status)
        echo -n "Checking for service Asterisk PBX: "
        /sbin/checkproc -p $ASTERISK_PID $ASTERISK_BIN
        rc_status -v
        ;;
    restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
       rc_status -v
        ;;
    *)
        echo "Usage: Asterisk {start|stop|restart|reload|status}"
        exit 1
esac

exit $RETVAL

#########################################################################
```

Anexo 2

Script de sip.conf

```
;
; SIP Configuration for Asterisk
;
; Syntax for specifying a SIP device in extensions.conf is
; SIP/devicename where devicename is defined in a section below.
;
; You may also use
; SIP/username@domain to call any SIP user on the Internet
; (Don't forget to enable DNS SRV records if you want to use this)
;
; If you define a SIP proxy as a peer below, you may call
; SIP/proxyhostname/user or SIP/user@proxyhostname
; where the proxyhostname is defined in a section below
;
; Useful CLI commands to check peers/users:
;   sip show peers            Show all SIP peers (including friends)
;   sip show users            Show all SIP users (including friends)
;   sip show registry         Show status of hosts we register with
;
;   sip debug                 Show all SIP messages
;
;   reload chan_sip.so        Reload configuration file
;                         Active SIP peers will not be reconfigured
;

[general]
context=default               ; Default context for incoming calls
;recordhistory=yes            ; Record SIP history by default
                          ; (see sip history / sip no history)
;realm=mydomain.tld           ; Realm for digest authentication
                          ; defaults to "Asterisk"
                          ; Realms MUST be globally unique according to RFC 3261
                          ; Set this to your host name or domain name
port=5060                     ; UDP Port to bind to (SIP standard port is 5060)
bindaddr=10.61.217.90         ; IP address to bind to (0.0.0.0 binds to all)
srvlookup=yes                 ; Enable DNS SRV lookups on outbound calls
                          ; Note: Asterisk only uses the first host
                          ; in SRV records
                          ; Disabling DNS SRV lookups disables the
                          ; ability to place SIP calls based on domain
                          ; names to some other SIP users on the Internet

;pedantic=yes                ; Enable slow, pedantic checking for Pingtel
                          ; and multiline formatted headers for strict
                          ; SIP compatibility (defaults to "no")
;tos=184                     ; Set IP QoS to either a keyword or numeric val
```

```
;tos=lowdelay                ; lowdelay,throughput,reliability,mincost,none
;maxexpirey=3600             ; Max length of incoming registration we allow
;defaultexpirey=120          ; Default length of incoming/outoing registration
;notifymimetype=text/plain   ; Allow overriding of mime type in MWI NOTIFY
;videosupport=yes            ; Turn on support for SIP video

;disallow=all                ; First disallow all codecs
allow=all                    ; Allow codecs in order of preference
;allow=ilbc                  ; Note: codec order is respected only in [general]
;musicclass=default          ; Sets the default music on hold class for all SIP calls
                             ; This may also be set for individual users/peers
;language=en                 ; Default language setting for all users/peers
                             ; This may also be set for individual users/peers
;relaxdtmf=yes               ; Relax dtmf handling
;rtptimeout=60               ; Terminate call if 60 seconds of no RTP activity
                             ; when we're not on hold
;rtpholdtimeout=300          ; Terminate call if 300 seconds of no RTP activity
                             ; when we're on hold (must be > rtptimeout)
;trustrpid = no              ; If Remote-Party-ID should be trusted
;progressinband=no           ; If we should generate in-band ringing always
;useragent=Asterisk PBX      ; Allows you to change the user agent string
;nat=no                      ; NAT settings
                             ; yes = Always ignore info and assume NAT
                             ; no = Use NAT mode only according to RFC3581
                             ; never = Never attempt NAT mode or RFC3581 support
                             ; route = Assume NAT, don't send rport (work around more
UNIDEN bugs)
;promiscredir = no           ; If yes, allows 302 or REDIR to non-local SIP address
;                            ; Note that promiscredir when redirects are made to the
;                            ; local system will cause loops since SIP is incapable
;                            ; of performing a "hairpin" call.
;
; If regcontext is specified, Asterisk will dynamically
; create and destroy a NoOp priority 1 extension for a given
; peer who registers or unregisters with us.  The actual extension
; is the 'regexten' parameter of the registering peer or its
; name if 'regexten' is not provided.  More than one regexten may be supplied
; if they are separated by '&'.  Patterns may be used in regexten.
;
;regcontext=iaxregistrations
;
; Asterisk can register as a SIP user agent to a SIP proxy (provider)
; Format for the register statement is:
;      register => user[:secret[:authuser]]@host[:port][/extension]
;
; If no extension is given, the 's' extension is used. The extension
; needs to be defined in extensions.conf to be able to accept calls
; from this SIP proxy (provider)
;
; host is either a host name defined in DNS or the name of a
; section defined below.
```

```
;
; Examples:
;
;register => 1234:password@mysipprovider.com
;
;     This will pass incoming calls to the 's' extension
;
;
;register => 2345:password@sip_proxy/1234
;
;    Register 2345 at sip provider 'sip_proxy'.  Calls from this provider connect to local
;    extension 1234 in extensions.conf default context, unless you define
;    unless you configure a [sip_proxy] section below, and configure a context.
;        Tip 1: Avoid assigning hostname to a sip.conf section like [provider.com]
;        Tip 2: Use separate type=peer and type=user sections for SIP providers
;                    (instead of type=friend) if you have calls in both directions


;externip = 200.201.202.203        ; Address that we're going to put in outbound SIP
messages
                        ; if we're behind a NAT

                        ; The externip and localnet is used
                        ; when registering and communicating with other proxies
                        ; that we're registered with
                        ; You may add multiple local networks.  A reasonable set of
defaults
                        ; are:
;localnet=192.168.0.0/255.255.0.0; All RFC 1918 addresses are local networks
;localnet=10.0.0.0/255.0.0.0    ; Also RFC1918
;localnet=172.16.0.0/12         ; Another RFC1918 with CIDR notation
;localnet=169.254.0.0/255.255.0.0 ;Zero conf local network

;---------------------------------------------------------------------------
; Users and peers have different settings available. Friends have all settings,
; since a friend is both a peer and a user
;
; User config options:       Peer configuration:
; --------------------       -------------------
; context                    context
; permit                     permit
; deny                       deny
; secret                     secret
; md5secret                    md5secret
; dtmfmode                     dtmfmode
; canreinvite                  canreinvite
; nat                        nat
; callgroup                    callgroup
; pickupgroup                    pickupgroup
; language                     language
; allow                      allow
```

```
; disallow                disallow
; insecure                insecure
; trustrpid               trustrpid
; progressinband              progressinband
; promiscredir              promiscredir
; callerid
; accountcode
; amaflags
; incominglimit
; restrictcid
;                       mailbox
;                       username
;                       template
;                       fromdomain
;                       regexten
;                       fromuser
;                       host
;                       mask
;                       port
;                       qualify
;                       defaultip
;                       rtptimeout
;                       rtpholdtimeout

;[sip_proxy]
; For incoming calls only. Example: FWD (Free World Dialup)
;type=user
;context=from-fwd
;[sip_proxy-out]

;type=peer                  ; we only want to call out, not be called
;secret=guessit
;username=yourusername         ; Authentication user for outbound proxies
;fromuser=yourusername         ; Many SIP providers require this!
;host=box.provider.com

;[grandstream1]
;type=friend                ; either "friend" (peer+user), "peer" or "user"
;context=from-sip
;fromuser=grandstream1         ; overrides the callerid, e.g. required by FWD
;callerid=John Doe <1234>
;host=192.168.0.23             ; we have a static but private IP address
;nat=no                     ; there is not NAT between phone and Asterisk
;canreinvite=yes              ; allow RTP voice traffic to bypass Asterisk
;dtmfmode=info                ; either RFC2833 or INFO for the BudgeTone
;incominglimit=1              ; permit only 1 outgoing call at a time
                        ; from the phone to Asterisk
;mailbox=1234@default  ; mailbox 1234 in voicemail context "default"
;disallow=all                ; need to disallow=all before we can use allow=
;allow=ulaw                  ; Note: In user sections the order of codecs
                        ; listed with allow= does NOT matter!
```

```
;allow=alaw
;allow=g723.1                ; Asterisk only supports g723.1 pass-thru!
;allow=g729                  ; Pass-thru only unless g729 license obtained


;[xlite1]
;Turn off silence suppression in X-Lite ("Transmit Silence"=YES)!
;Note that Xlite sends NAT keep-alive packets, so qualify=yes is not needed
;type=friend
;regexten=1234               ; When they register, create extension 1234
;username=xlite1
;callerid="Jane Smith" <5678>
;host=dynamic
;nat=yes                     ; X-Lite is behind a NAT router
;canreinvite=no              ; Typically set to NO if behind NAT
;disallow=all
;allow=gsm                   ; GSM consumes far less bandwidth than ulaw
;allow=ulaw
;allow=alaw


;[snom]
;type=friend                 ; Friends place calls and receive calls
;context=from-sip            ; Context for incoming calls from this user
;secret=blah
;language=de                 ; Use German prompts for this user
;host=dynamic                ; This peer register with us
;dtmfmode=inband             ; Choices are inband, rfc2833, or info
;defaultip=192.168.0.59      ; IP used until peer registers
;username=snom               ; Username to use in INVITE until peer registers
;mailbox=1234,2345           ; Mailboxes for message waiting indicator
;restrictcid=yes             ; To have the callerid restriced -> sent as ANI
;disallow=all
;allow=ulaw                  ; dtmfmode=inband only works with ulaw or alaw!
;mailbox=1234@context,2345   ; Mailbox(-es) for message waiting indicator


;[polycom]
;type=friend                 ; Friends place calls and receive calls
;context=from-sip            ; Context for incoming calls from this user
;secret=blahpoly
;host=dynamic                ; This peer register with us
;dtmfmode=rfc2833            ; Choices are inband, rfc2833, or info
;username=polly              ; Username to use in INVITE until peer registers
;disallow=all
;allow=ulaw                  ; dtmfmode=inband only works with ulaw or alaw!
;progressinband=no           ; Polycom phones don't work properly with "never"


;[pingtel]
;type=friend
```

```
;username=pingtel
;secret=blah
;host=dynamic
;insecure=yes                ; To match a peer based by IP address only and not peer
;insecure=very               ; To allow registered hosts to call without re-authenticating
;qualify=1000                ; Consider it down if it's 1 second to reply
                        ; Helps with NAT session
                        ; qualify=yes uses default value
;callgroup=1,3-4             ; We are in caller groups 1,3,4
;pickupgroup=1,3-5           ; We can do call pick-p for call group 1,3,4,5
;defaultip=192.168.0.60      ; IP address to use if peer has not registred

;[cisco1]
;type=friend
;username=cisco1
;secret=blah
;qualify=200                 ; Qualify peer is no more than 200ms away
;nat=yes                     ; This phone may be natted
                        ; Send SIP and RTP to  IP address that packet is
                        ; received from instead of trusting SIP headers
;host=dynamic                ; This device registers with us
;canreinvite=no              ; Asterisk by default tries to redirect the
                        ; RTP media stream (audio) to go directly from
                        ; the caller to the callee.  Some devices do not
                        ; support this (especially if one of them is
                        ; behind a NAT).
;defaultip=192.168.0.4

;[cisco2]
;type=friend
;username=cisco2
;fromuser=markster           ; Specify user to put in "from" instead of callerid
;fromdomain=yourdomain.com   ; Specify domain to put in "from" instead of callerid
                        ; fromuser and fromdomain are used when Asterisk
                        ; places calls to this account.  It is not used for
                        ; calls from this account.
;secret=blah
;host=dynamic
;defaultip=192.168.0.4
;amaflags=default            ; Choices are default, omit, billing, documentation
;accountcode=markster        ; Users may be associated with an accountcode to ease
billing


[4101]
type=friend
context=from-sip
username=grandstream
callerid=flavio Carneiro
host=dynamic
nat=no
```

```
canreinvite=yes
dtmfmode=info
disallow=all
allow=ulaw
allow=g729

[4102]
type=friend
username=4102
canreinvite=yes
disallow=all
callerid=flavio Carneiro
host=dynamic
nat=yes
allow=gsm
allow=ulaw

[4103]
type=friend
username=4103
canreinvite=yes
disallow=all
callerid=rodrigo
host=dynamic
nat=yes
allow=gsm
allow=ulaw

[4104]
type=friend
username=cisco
canreinvite=no
secret=blah
disallow=all
host=dynamic
nat=yes
allow=gsm
allow=ulaw
```

Anexo 3

Script de extensions.conf.

```
;
; Static extension configuration file, used by
; the pbx_config module. This is where you configure all your
; inbound and outbound calls in Asterisk.
;
; This configuration file is reloaded
; - With the "extensions reload" command in the CLI
; - With the "reload" command (that reloads everything) in the CLI

;
; The "General" category is for certain variables.
;
[general]
;
; If static is set to no, or omitted, then the pbx_config will rewrite
; this file when extensions are modified.  Remember that all comments
; made in the file will be lost when that happens.
;
; XXX Not yet implemented XXX
;
static=yes
;
; if static=yes and writeprotect=no, you can save dialplan by
; CLI command 'save dialplan' too
;
writeprotect=no

; You can include other config files, use the #include command (without the ';')
; Note that this is different from the "include" command that includes contexts within
; other contexts. The #include command works in all Asterisk configuration files.
;#include "filename.conf"

; The "Globals" category contains global variables that can be referenced
; in the dialplan with ${VARIABLE} or ${ENV(VARIABLE)} for Environmental variable
; ${${VARIABLE}} or ${text${VARIABLE}} or any hybrid
;
[globals]
CONSOLE=Console/dsp                     ; Console interface for demo
;CONSOLE=Zap/1
;CONSOLE=Phone/phone0
IAXINFO=guest                           ; IAXtel username/password
;IAXINFO=myuser:mypass
TRUNK=Zap/g2                            ; Trunk interface
;
; Note the 'g2' in the TRUNK variable above. It specifies which group (defined
; in zapata.conf) to dial, i.e. group 2, and how to choose a channel to use in
; the specified group. The four possible options are:
```

```
;
; g: select the lowest-numbered non-busy Zap channel (aka. ascending sequential hunt
group).
; G: select the highest-numbered non-busy Zap channel (aka. descending sequential hunt
group).
; r: use a round-robin search, starting at the next highest channel than last time (aka.
ascending rotary hunt group).
; R: use a round-robin search, starting at the next lowest channel than last time (aka.
descending rotary hunt group).
;
TRUNKMSD=1                              ; MSD digits to strip (usually 1 or 0)
;TRUNK=IAX2/user:pass@provider

;
; Any category other than "General" and "Globals" represent
; extension contexts, which are collections of extensions.
;
; Extension names may be numbers, letters, or combinations
; thereof. If an extension name is prefixed by a '_'
; character, it is interpreted as a pattern rather than a
; literal.  In patterns, some characters have special meanings:
;
;   X - any digit from 0-9
;   Z - any digit from 1-9
;   N - any digit from 2-9
;   [1235-9] - any digit in the brackets (in this example, 1,2,3,5,6,7,8,9)
;   . - wildcard, matches anything remaining (e.g. _9011. matches
;       anything starting with 9011 excluding 9011 itself)
;
; For example the extension _NXXXXXX would match normal 7 digit dialings,
; while _1NXXNXXXXXX would represent an area code plus phone number
; preceeded by a one.
;
; Each step of an extension is ordered by priority, which must
; always start with 1 to be considered a valid extension.
;
; Contexts contain several lines, one for each step of each
; extension, which can take one of two forms as listed below,
; with the first form being preferred.  One may include another
; context in the current one as well, optionally with a
; date and time.  Included contexts are included in the order
; they are listed.
;
;[context]
;exten => someexten,priority,application(arg1,arg2,...)
;exten => someexten,priority,application,arg1|arg2...
;
; Timing list for includes is
;
;   <time range>|<days of week>|<days of month>|<months>
;
```

```
;include => daytime|9:00-17:00|mon-fri|*|*
;
; ignorepat can be used to instruct drivers to not cancel dialtone upon
; receipt of a particular pattern.  The most commonly used example is
; of course '9' like this:
;
;ignorepat => 9
;
; so that dialtone remains even after dialing a 9.
;

;
; Here are the entries you need to participate in the IAXTEL
; call routing system.  Most IAXTEL numbers begin with 1-700, but
; there are exceptions.  For more information, and to sign
; up, please go to www.gnophone.com or www.iaxtel.com
;
[iaxtel700]
exten                                                                    =>
_91700XXXXXXX,1,Dial(IAX2/${IAXINFO}@iaxtel.com/${EXTEN:1}@iaxtel)

;
; The SWITCH statement permits a server to share the dialplain with
; another server. Use with care: Reciprocal switch statements are not
; allowed (e.g. both A -> B and B -> A), and the switched server needs
; to be on-line or else dialing can be severly delayed.
;
[iaxprovider]
;switch => IAX2/user:[key]@myserver/mycontext

[trunkint]
;
; International long distance through trunk
;
exten => _9011.,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _9011.,2,Congestion
[trunkld]
;
; Long distance context accessed through trunk
;
exten => _91NXXNXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91NXXNXXXXXX,2,Congestion

[trunklocal]
;
; Local seven-digit dialing accessed through trunk interface
;
exten => _9NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _9NXXXXXX,2,Congestion

[trunktollfree]
```

```
;
; Long distance context accessed through trunk interface
;
exten => _91800NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91800NXXXXXX,2,Congestion
exten => _91888NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91888NXXXXXX,2,Congestion
exten => _91877NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91877NXXXXXX,2,Congestion
exten => _91866NXXXXXX,1,Dial(${TRUNK}/${EXTEN:${TRUNKMSD}})
exten => _91866NXXXXXX,2,Congestion

[international]
;
; Master context for international long distance
;
ignorepat => 9
include => longdistance
include => trunkint

[longdistance]
;
; Master context for long distance
;
ignorepat => 9
include => local
include => trunkld

[local]
;
; Master context for local, toll-free, and iaxtel calls only
;
ignorepat => 9
include => default
include => parkedcalls
include => trunklocal
include => iaxtel700
include => trunktollfree
include => iaxprovider
;
; You can use an alternative switch type as well, to resolve
; extensions that are not known here, for example with remote
; IAX switching you transparently get access to the remote
; Asterisk PBX
;
; switch => IAX2/user:password@bigserver/local

[macro-stdexten];
;
; Standard extension macro:
;   ${ARG1} - Extension  (we could have used ${MACRO_EXTEN} here as well
```

```
;   ${ARG2} - Device(s) to ring
;
exten => s,1,Dial(${ARG2},20)                          ; Ring the interface, 20 seconds
maximum
exten => s,2,Goto(s-${DIALSTATUS},1)                   ; Jump based on status
(NOANSWER,BUSY,CHANUNAVAIL,CONGESTION,ANSWER)

exten => s-NOANSWER,1,Voicemail(u${ARG1})              ; If unavailable, send to
voicemail w/ unavail announce
exten => s-NOANSWER,2,Goto(default,s,1)                ; If they press #, return to start

exten => s-BUSY,1,Voicemail(b${ARG1})                  ; If busy, send to voicemail w/
busy announce
exten => s-BUSY,2,Goto(default,s,1)                    ; If they press #, return to start

exten => _s-.,1,Goto(s-NOANSWER,1)                     ; Treat anything else as no
answer

exten => a,1,VoicemailMain(${ARG1})                    ; If they press *, send the
user into VoicemailMain

[demo]
;
; We start with what to do when a call first comes in.
;
exten => s,1,Wait,1                    ; Wait a second, just for fun
exten => s,2,Answer                    ; Answer the line
exten => s,3,DigitTimeout,5            ; Set Digit Timeout to 5 seconds
exten => s,4,ResponseTimeout,10        ; Set Response Timeout to 10 seconds
exten => s,5,BackGround(demo-congrats)  ; Play a congratulatory message
exten => s,6,BackGround(demo-instruct)  ; Play some instructions

exten => 2,1,BackGround(demo-moreinfo)  ; Give some more information.
exten => 2,2,Goto(s,6)

exten => 3,1,SetLanguage(fr)           ; Set language to french
exten => 3,2,Goto(s,5)                 ; Start with the congratulations

exten => 1000,1,Goto(default,s,1)
;
; We also create an example user, 1234, who is on the console and has
; voicemail, etc.
;
exten => 1234,1,Playback(transfer,skip)        ; "Please hold while..."
                                ; (but skip if channel is not up)
exten => 1234,2,Macro(stdexten,1234,${CONSOLE})

exten => 1235,1,Voicemail(u1234)               ; Right to voicemail

exten => 1236,1,Dial(Console/dsp)              ; Ring forever
exten => 1236,2,Voicemail(u1234)               ; Unless busy
```

```
;
; # for when they're done with the demo
;
exten => #,1,Playback(demo-thanks)            ; "Thanks for trying the demo"
exten => #,2,Hangup                   ; Hang them up.

;
; A timeout and "invalid extension rule"
;
exten => t,1,Goto(#,1)                ; If they take too long, give up
exten => i,1,Playback(invalid)        ; "That's not valid, try again"

;
; Create an extension, 500, for dialing the
; Asterisk demo.
;
exten => 500,1,Playback(demo-abouttotry); Let them know what's going on
exten => 500,2,Dial(IAX2/guest@misery.digium.com/s@default)     ; Call the Asterisk
demo
exten => 500,3,Playback(demo-nogo)     ; Couldn't connect to the demo site
exten => 500,4,Goto(s,6)              ; Return to the start over message.

;
; Create an extension, 600, for evaulating echo latency.
;
exten => 600,1,Playback(demo-echotest)  ; Let them know what's going on
exten => 600,2,Echo                   ; Do the echo test
exten => 600,3,Playback(demo-echodone)  ; Let them know it's over
exten => 600,4,Goto(s,6)              ; Start over

;
; Give voicemail at extension 8500
;
exten => 8500,1,VoicemailMain
exten => 8500,2,Goto(s,6)
;
; Here's what a phone entry would look like (IXJ for example)
;
;exten => 1265,1,Dial(Phone/phone0,15)
;exten => 1265,2,Goto(s,5)

;[mainmenu]
;
; Example "main menu" context with submenu
;
;exten => s,1,Answer
;exten => s,2,Background(thanks)              ; "Thanks for calling press 1 for sales, 2 for
support, ..."
;exten => 1,1,Goto(submenu,s,1)
;exten => 2,1,Hangup
```

```
;include => default
;
;[submenu]
;exten => s,1,Ringing                    ; Make them comfortable with 2 seconds of
ringback
;exten => s,2,Wait,2
;exten => s,3,Background(submenuopts)    ; "Thanks for calling the sales department.
Press 1 for steve, 2 for..."
;exten => 1,1,Goto(default,steve,1)
;exten => 2,1,Goto(default,mark,2)

[default]
;
; By default we include the demo.  In a production system, you
; probably don't want to have the demo there.
;
include => demo
;
; Extensions like the two below can be used for FWD, Nikotel, sipgate etc.
; Note that you must have a [sipprovider] section in sip.conf whereas
; the otherprovider.net example does not require such a peer definition
;
;exten => _41X.,1,Dial(SIP/${EXTEN:2}@sipprovider,,r)
;exten => _42X.,1,Dial(SIP/user:passwd@${EXTEN:2}@otherprovider.net,30,rT)

; Real extensions would go here. Generally you want real extensions to be 4 or 5
; digits long (although there is no such requirement) and start with a single
; digit that is fairly large (like 6 or 7) so that you have plenty of room to
; overlap extensions and menu options without conflict.  You can alias them with
; names, too and use global variables

;exten => 6245,hint,SIP/Grandstream1&SIP/Xlite1 ; Channel hints for presence
;exten => 6245,1,Dial(SIP/Grandstream1,20,rt)   ; permit transfer
;exten => 6245,1,Dial(${HINT},20,rtT)           ; Use hint as listed
;exten => 6361,1,Dial(IAX2/JaneDoe,,rm)         ; ring without time limit
;exten => 6389,1,Dial(MGCP/aaln/1@192.168.0.14)
;exten => 6394,1,Dial(Local/6275/n)             ; this will dial ${MARK}
exten=>4102,1,Dial(SIP/4102,20)
exten=>4103,1,Dial(SIP/4103,20)



;exten => 6275,1,Macro(stdexten,6275,${MARK})       ; assuming ${MARK} is
something like Zap/2
;exten => mark,1,Goto(6275|1)                  ; alias mark to 6275
;exten => 6536,1,Macro(stdexten,6236,${WIL})    ; Ditto for wil
;exten => wil,1,Goto(6236|1)
;
; Some other handy things are an extension for checking voicemail via
; voicemailmain
;
```

```
;exten => 8500,1,VoicemailMain
;exten => 8500,2,Hangup
;
; Or a conference room (you'll need to edit meetme.conf to enable this room)
;
;exten => 8600,1,Meetme(1234)
;
; Or playing an announcement to the called party, as soon it answers
;
;exten = 8700,1,Dial(${MARK},30,A(/path/to/my/announcemsg))
;
;
; For more information on applications, just type "show applications" at your
; friendly Asterisk CLI prompt.
;
; 'show application <command>' will show details of how you
; use that particular application in this file, the dial plan.
```

Anexo 4

Configuração Softfones da X-Ten(X-Lite versão gratuita)

Será utilizado o sotfphone X-Lite, por ser um freeware e um excelente cliente para voip no protocolo SIP. Faça o download da ultima versão do X-Lite www.xten.com.

Após efetuar o download do software para sua máquina, execute o arquivo baixado e vá seguindo as instruções que vão surgindo na tela para efetuar a instalação. Depois de instalado, abra o X-Lite, e siga as instruções abaixo:

1. Clique no ícone que fica entre o “CLEAR” e o símbolo verde de um telefone fora do gancho.

2. Clique em System Settings.

3. Clique em SIP Proxy.

4. Clique em Default.

5. Agora, configure como mostra abaixo:

- Enable: YES
- Display Name: Nome
- Username: Seu número voip.
- Autorization User: Seu número Voip.
- Password: Senha utilizada
- Domain/Realm: IP Callmanager
- SIP Proxy: IP Callmanager
- Register: Aways
- Todas as demais opções podem continuar iguais.

6. Deve-se configurar o codec utilizado para transmitir e receber os dados de voz. O codec que obteve o melhor desempenho e menor consumo de banda foi o GSM. No X-Lite clique em cima dos codecs G711U, G711A, ILBC e SPX de modo que eles fiquem desabilitados, e deixe somente marcado o GSM.

7. Agora feche e abra novamente o X-Lite. Se estiver tudo configurado corretamente, você estará conectado ao servidor de VOIP, podendo efetuar ligações a qualquer outro usuário conectado bastando apenas digitar o número voip dele.

Anexo 5

Arquivo /etc/ha.d/ha.cf

debugfile /var/log/ha-debug
logfile /var/log/ha-log Muito úteis para debug de configuração
serial /dev/ttyS0 Coloque a interface serial apropriada, se for utilizar a serial
keepalive 2 Seta o tempo entre os heartbeats
deadtime 10 O nodo será declarado como indisponível depois de 10 segundos
baud 19200 Velocidade da serial (bps)
udpport 694 Porta a ser usada (lembre-se de liberar caso esteja usando firewall)
udp eth0 Interface a ser usada pelo heartbeat (broadcast)
mcast eth0 225.0.0.1 694 1 1 Caso prefira usar multicast, coloque as configurações
nice_failback on Opcional. Em caso de falha, quando o principal retornar a rede ele não tentará retormar
os recursos.
node linuxha1.linux-ha.org Obrigatório, hostname da maquina nodo conforme descrito em `uname -a`
node linuxha2.linux-ha.org Obrigatório, hostname da maquina nodo conforme descrito em `uname -a`