

BRUNO DE OLIVEIRA PINHEIRO

VOZ SOBRE IP UTILIZANDO ASTERISK

Monografia apresentada ao Departamento de Ciência de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação *Lato Sensu* em Administração de Redes *Linux*, para obtenção do título de especialista em Redes *Linux*.

Orientador

Prof. MSc. Joaquim Quinteiro Uchôa

LAVRAS

Minas Gerais – Brasil

2005

BRUNO DE OLIVEIRA PINHEIRO

VOZ SOBRE IP UTILIZANDO ASTERISK

Monografia apresentada ao Departamento de Ciência de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação *Lato Sensu* em Administração de Redes *Linux*, para obtenção do título de especialista em Redes *Linux*.

APROVADA em 11 de setembro de 2005.

Prof.

Prof.

Prof. MSc. Joaquim Quinteiro Uchôa
(Orientador)

LAVRAS
Minas Gerais – Brasil
2005

DEDICATÓRIA

Aos meus pais Sylvio e Vilma pela ausente presença que norteia, pela força e garra diante da vida. IN MEMORIAM.

À minha esposa Tatiane pelos momentos que a privei de minha companhia para dedicar-me ao curso e a esse projeto.

AGRADECIMENTOS

Agradeço a Deus por está sempre presente ao meu lado me dando forças para enfrentar qualquer obstáculo. A todos os professores do ARL que colaboraram com a realização deste projeto, transmitindo-me os seus conhecimentos, em especial ao professor Joaquim Quinteiro Uchôa pela orientação, sua dedicação e paciência. Ao Capitão Especialista em Comunicações Marcelo de Souza Freitas pela brilhante sugestão do tema que se concretizou neste projeto e pela compreensão nos momentos que tive que me dedicar à monografia, deixando de lado a labuta do dia-a-dia. Aos meus colegas de turma que juntos caminhamos durante este longo período de curso, em especial a Francisco Saito pelo apoio durante todo o curso e durante a confecção deste trabalho.

RESUMO

Este Trabalho trata do *software* livre Asterisk, uma Central Telefônica Privada IP ou PBX IP que suporta inúmeros protocolos e CODECs da tecnologia de Voz sobre IP. Abordando a sua instalação, configuração, compatibilidade com *hardware* e instalação e configuração dos mesmos. O texto trás ainda, no início, alguns conceitos básicos de comunicação de voz em uma rede IP, mostrando as necessidades para a convergência de dados, bem como os protocolos que permitiram o tráfego de um dado analógico e em tempo real em uma rede de transmissão de dados digitais com comutação de pacotes. Por fim é apresentado um projeto de implementação do Asterisk em um órgão federal, mostrando seus pontos fortes e fracos e, diga-se de passagem, mais fortes do que fraco.

Sumário

Lista de Figuras	8
Lista de Tabelas	9
Introdução.....	10
2 Conceitos Básicos.....	13
2.1 TCP/IP	13
2.1.1 Histórico.....	13
2.1.2 Funcionamento do TCP/IP	14
2.1.3 Modelo de Referência TCP/IP	15
2.2 QoS	15
2.3 Codificação, Compressão de Áudio;.....	17
2.4 CODEC.....	19
2.5 Centrais Telefônicas.....	21
3 VoIP.....	23
3.1 Tecnologia <i>Voice over IP</i>	23
3.2 Obstáculos para o VoIP	24
3.3 Protocolos	26
3.3.1 Processo de Sinalização de Chamadas	26
3.3.2 Processo de Controle do Transporte e Transporte da Mídia 33	
3.4 Equipamentos VoIP	35
3.5 Clientes.....	36
4 Asterisk	38
4.1 Arquitetura do Asterisk.....	40
4.2 Aplicações do Asterisk	42
4.3 Hardware de Telefonia para o Asterisk.....	43
4.3.1 Placa XP100P	44

4.3.2	Placa TDM400P	45
4.3.3	Placa TE110P.....	46
4.4	Instalação do Driver do Hardware de Telefonia.....	46
4.4.1	Configuração dos Hardware de Telefonia	49
4.5	Instalando o Asterisk	50
4.5.1	Hardware Mínimo.....	50
4.5.2	Requisitos Necessários do Sistema Operacional.....	50
4.5.3	Processo de Instalação.....	51
4.6	Configuração do Asterisk.....	52
4.6.1	O Arquivo zapata.conf.....	53
4.6.2	O Arquivo sip.conf	55
4.6.3	O Arquivo extension.conf	57
5	Projeto de Implementação do Asterisk.....	63
5.1	O DECEA	63
5.2	O Projeto	64
6	Considerações Finais	67
	Referências Bibliográficas.....	68
	Apêndice A	71

Lista de Figuras

Figura 1 – Protocolo TCP/IP	15
Figura 2 – Processo de Digitalização da Voz	19
Figura 3 – Pilha de protocolos utilizados por terminais H.323	28
Figura 4 – Janela do X-Lite	37
Figura 5 – Janela do NetMeeting	37
Figura 6 – Arquitetura do Asterisk	40
Figura 7 – Placa XP100P	45
Figura 8 – Placa TDM400P	45
Figura 9 – Placa TE110P	46

Lista de Tabelas

Tabela 1: Valores MOS de alguns CODECs (CISCO-CODEC, 2005)	20
Tabela 2: Valores MOS (DASPL)	21
Tabela 3 – Módulos das Placas Digium (DIGIUM)	48
Tabela 4 – Comparação da RTCaer e VoIP com Asterisk	65

Introdução

A tecnologia de transmissão de voz sobre uma rede IP (VoIP) teve uma grande explosão de utilização nos últimos anos. Muitas companhias convergiram à voz nas redes de dados existentes que interligavam a sede e as filiais da empresa. O maior atrativo para o sucesso da tecnologia era a economia que as empresas obtiam aproveitando seus *links* de dados existentes e já pagos para agora também se comunicar através do telefone com lugares distantes onde antes era necessária uma ligação DDD ou até DDI.

Aproximadamente cinco anos atrás, a migração do sistema convencional para o sistema VoIP despendia um investimento inicial elevado, onde somente grandes empresas eram capazes de alcançar. Esse custo elevado ainda era alimentado pelas soluções proprietárias e tomavam conta do mercado.

Com o Trabalho de *Mark Spencer* e *Jin Dixon*, surgiu um *software* que roda em Linux, tem licença GPL que é capaz de realizar funções de um PBX IP e analógico e outras funções a mais. O *software* criado foi o Asterisk. O Asterisk junto com os *hardware* criados por *Jin Dixon*, o custo da tecnologia VoIP ficou bastante acessível a pequenas e médias empresas e ao usuário residencial.

O objetivo primário deste Trabalho é apresentar o *Software* Asterisk como solução para implementação da Tecnologia VoIP, mostrando todas as suas funcionalidades e particularidades que o tornam uma excelente escolha.

O Asterisk como a tecnologia de Voz sobre a rede IP está se difundido cada vez mais, sendo bem aceito em todas as empresas de pequenas a grandes e também nas residências. Como a tecnologia VoIP hoje tornou-se uma realidade, ou sem medo de exagerar uma necessidade, o Asterisk surge como um catalisador para essa expansão graças às suas inúmeras funcionalidades associadas ao seu baixo custo de implementação.

Outra vantagem do Asterisk é o fato dele ser um *software* livre e de ter sido criado para “rodar” no sistema operacional GNU/Linux, também *software* livre. Com isso além de mais uma economia na implementação, o GNU/Linux conta com uma comunidade internacional que trabalha no desenvolvimento do sistema em si e de *software* compatíveis a ele. Fazendo com que o Asterisk herde uma comunidade de desenvolvedores para que ele evolua cada vez mais com a colaboração da comunidade.

Com todas essas vantagens foi imperativa a escolha do Asterisk para compor os PBX IP que irão substituir a atual Rede Telefônica do Comando da Aeronáutica. O autor atualmente é militar da Aeronáutica e é o analista responsável pelos testes em laboratório, implementação, treinamento e posterior manutenção da nova rede. O autor logo no primeiro contato quando testemunhou a qualidade do *software*, e com o atual e relevante assunto de convergência de voz na rede IP, percebeu que o Asterisk seria um ótimo assunto para a confecção do trabalho de conclusão de curso.

Por todos os motivos expostos esse trabalho foi confeccionado procurando trazer as informações necessárias a um Administrador de Redes Linux para implementar a tecnologia VoIP com o Asterisk. Para atingir esse objetivo o trabalho foi escrito em cinco capítulos organizados da seguinte forma: o Capítulo 2 está centrado nos conceitos básicos para o entendimento do processo da transmissão de voz, um sinal analógico, em uma rede IP. O Capítulo 3 vem com a explicação da tecnologia de voz sobre a rede IP, mostrando os protocolos de rede necessários, os obstáculos para o transporte da voz em uma rede de dados, alguns equipamentos proprietários que hoje estão no mercado e clientes VoIP.

Já no Capítulo 4 o enfoque é o *software* Asterisk, mostrando sua arquitetura e suas funcionalidades e os *hardware* compatíveis. O Capítulo 4 ainda mostra o procedimento de instalação e configuração do programa. O

Capítulo 5 é dedicado à demonstração do projeto de implementação do Asterisk no Departamento de Controle do Espaço Aéreo no Comando da Aeronáutica. Ao final do texto encontra-se o Apêndice A onde são apresentados exemplos de arquivos de configuração.

2 Conceitos Básicos

2.1 TCP/IP

2.1.1 Histórico

A arquitetura TCP/IP surgiu com a criação de uma rede patrocinada pelo Departamento de Defesa do governo dos Estados Unidos da América (DoD - *Department of Defense*). Uma das tarefas essenciais dessa rede seria manter a comunicação, mesmo que apenas em uma parte de órgãos do governo e universidades, numa ocorrência de guerras ou catástrofes que afetassem os meios de comunicação daquele país. Dessa necessidade, surgiu a ARPANET, uma rede que permaneceria intacta caso um dos servidores perdesse a conexão (Sousa, 2001).

A ARPANET necessitava então de um modelo de protocolos que assegurasse tal funcionalidade esperada, mostrando-se confiável, flexível e de fácil implementação. É então desenvolvida a arquitetura TCP/IP, que se torna um padrão de fato.

A ARPANET cresceu e tornou-se a rede mundial de computadores - Internet. A utilização e facilidades do padrão TCP/IP utilizado pelos fabricantes de outras redes, com a finalidade da conectividade com a Internet.

A normalização do TCP/IP chegou após a sua utilização em massa. Hoje, quando se menciona TCP/IP, vem imediata a associação com a Internet, ocorrendo de modo idêntico o inverso: a Internet está diretamente relacionada à arquitetura TCP/IP.

2.1.2 Funcionamento do TCP/IP

O protocolo TCP/IP é, na verdade, um grupo de protocolos que trabalham conjuntamente, com o objetivo de estabelecer a comunicação e a transferência de dados entre dois ou mais computadores ligados em rede.

O TCP (*Transmission Control Protocol*), como o próprio nome diz, controla a transmissão dos dados, cuidando para que os dados enviados por um computador cheguem integralmente ao destino correto.

O TCP nada mais é que uma biblioteca de rotinas instaladas nos computadores origem e destino (ou seja, todos os computadores que utilizem o protocolo TCP/IP para se comunicar) que as aplicações como Apache¹, Postfix², e outras, utilizam quando precisam executar o transporte de dados entre equipamentos.

O TCP é orientado a conexão, portanto, fornece uma comunicação confiável entre as máquinas. Com o TCP ainda é possível o controle de fluxo e recuperação de erros.

Enquanto o TCP cuida da segurança do envio e recebimento dos datagramas o IP é responsável pela transmissão em si, fazendo o serviço de roteamento, ou seja, conduzindo os dados para os endereços corretos. Na verdade, os dois protocolos se completam: enquanto o IP identifica os endereços e cuida para que os dados sejam enviados pelo meio físico, o TCP verifica se estes dados enviados foram transmitidos corretamente.

1 <http://www.apache.org>

2 <http://www.postfix.org>

2.1.3 Modelo de Referência TCP/IP

O modelo padrão para protocolos é o modelo ISO/OSI que acabou se tornando apenas uma base para praticamente todos os protocolos desenvolvidos pela indústria. Para mais detalhes sobre o modelo ISO/OSI consulte (Martins & Correia, 2002). Apresenta-se na Figura 1 o modelo TCP/IP. A Tabela da esquerda apresenta os principais protocolos distribuídos pelas diversas camadas, enquanto que na Tabela da direita as funções são o destaque.

TELNET	NFS	Aplicações Processos
FTP	SNMP	
SMTP	DNS	
TCP	UDP	
IP		Transporte
Enlace		Rede
Física		Física

Figura 1 – Protocolo TCP/IP

2.2 QoS

A camada de transporte tem como sua principal função melhorar a qualidade de serviço (QoS) oferecida pela camada de rede. Se o serviço de rede for perfeito, o Trabalho da camada de transporte será fácil. No entanto, se o serviço de rede não for perfeito, a camada de transporte terá que servir de ponte

para cobrir a distância entre o que os usuários de transporte desejam e o que a camada de rede oferece. Ainda que à primeira vista o conceito de qualidade de serviço seja vago (fazer com que todos concordem sobre o que significa um serviço “bom” não é uma tarefa simples), a QoS pode ser definida por um número específico de parâmetros. Os parâmetros de rede são definidos na recomendação I350 (ITU I350, 1993) como “a habilidade da rede ou dos componentes da rede para prover funções relacionadas à comunicação entre dois usuários”. O serviço de transporte pode permitir ao usuário determinar os valores preferenciais, os valores aceitáveis e os valores mínimos para vários parâmetros de serviço no momento em que uma conexão é estabelecida. Alguns parâmetros também podem ser usados no transporte sem conexão. É tarefa da camada de transporte examinar esses parâmetros e, dependendo do(s) tipo(s) de serviço(s) de rede disponível(eis), determinar se é possível realizar o serviço solicitado. Os parâmetros típicos para a qualidade de serviço da camada de transporte são resumidos em:

- Retardo no estabelecimento da conexão
- Probabilidade de falha no estabelecimento da conexão
- *Throughput*
- Taxa de erros residuais
- Proteção
- Prioridade
- Resiliência

Observe que poucas redes ou protocolos oferecem todos esses parâmetros. Muitas apenas tentam reduzir a taxa de erros da melhor maneira possível. Outras têm arquiteturas de QoS mais elaboradas.

A descrição de QoS é tomada da recomendação do ITU-T E.800 (ITU E800, 1994), onde se define a qualidade de serviço como sendo o efeito

coletivo provocado pelas características de desempenho de um serviço, determinando o grau de satisfação do usuário. Tal definição engloba, originalmente, vários aspectos de diversas áreas de atuação, incluindo o nível de satisfação do usuário.

A QoS é muito importante em uma rede convergente com voz e dados porque o comportamento de cada tipo de pacote na rede pode trazer mais ou menos desconforto ao usuário. Um pacote de dados, por exemplo, se chegar atrasado ou se houver uma perda não irá influenciar muito no resultado final. Mas um pacote de voz se chegar atrasado ou se for perdido trará desconforto para os interlocutores, por isso se definiu que o tráfego de voz tem que ser em tempo real. Os problemas com os pacotes de voz na rede IP serão abordados no Capítulo 3.

Em vista disso, através da QoS na rede TCP/IP é necessário criar regras para priorizar o tráfego de voz em relação ao tráfego de outros dados na rede, utilizando várias técnicas, sendo aplicada em todos os componentes da rede, pois se em um componente não for aplicada técnicas de QoS, a qualidade não poderá ser mais garantida naquela rede.

2.3 Codificação, Compressão de Áudio;

Digitalização da voz foi projetada pelo *Bell System* na década de 50. O projeto original da digitalização de voz foi o empacotamento de dois ou mais canais de áudio para ser transmitido em um par de fios.

A conversão de voz analógica para sinal digital, passa por três passos:

Amostragem – A amostragem do sinal analógico é feito por intervalos

de tempo periódicos. Quanto maior a quantidade de amostras, mais fiel será a reprodução do sinal analógico na decodificação, por outro lado a maior quantidade de amostras exigirá uma maior largura de banda para a transmissão do sinal codificado. Para chegar a uma quantidade de amostras que não degradasse a qualidade da voz e nem usasse muita largura de banda, foi usado o Teorema de Nyquist. O Teorema especifica que um sinal arbitrário que é transmitido, para ser reconstruído pelo receptor, a quantidade de amostras por segundo tem que ser o dobro da maior frequência do sinal transmitido (Soares; Lemos & Colcher, 1995).

Apesar da voz humana ter o espectro de frequência de 20Hz a 20KHz, para uma boa reprodução em canais telefônicos foi admitido uma faixa de 300 a 3400Hz. Arredondando para cima a maior frequência do sinal de voz para canal telefônico então é de 4000Hz, o que corresponde a 8000 amostras por segundo (CISCO - VOICE, 2004).

A saída da amostragem é um sinal modulado em amplitude de sinal.

Quantização – É a classificação das amplitudes das amostras em uma escala de valores para que eles sejam transmitidos. Há duas formas de fazer a quantização. A primeira é a forma linear onde toda a escala tem valores e tamanhos iguais. A outra maneira é a forma logarítmica onde a escala tem valores diferentes, comportando-se de forma semelhante ao ouvido humano.

Codificação – É a transformação dos valores decimais criados pela quantização em valores binários para serem transmitidos em um canal digital.

Um quarto processo opcional é a compressão que pode ser utilizado para diminuir a quantidade de banda utilizada e propiciar a transmissão de mais de um canal de voz em um canal digital. A Figura 2 ilustra essas etapas de

digitalização da voz.

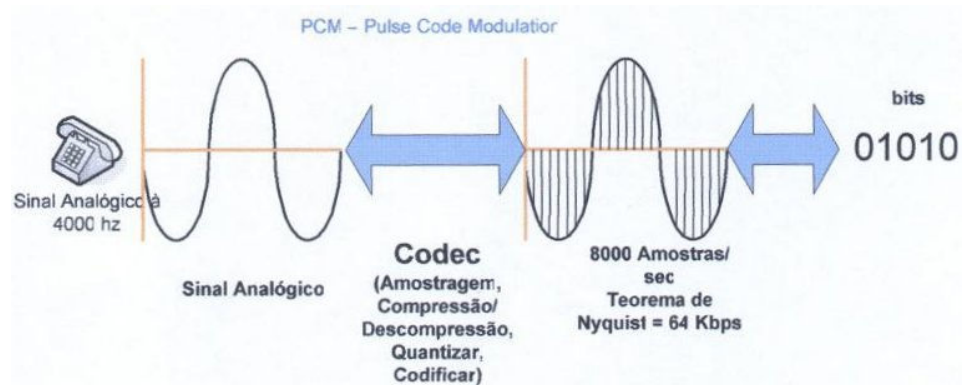


Figura 2 – Processo de Digitalização da Voz

2.4 CODEC

Um CODEC é um programa ou dispositivo com algoritmos de compactação e descompactação específicos para um determinado formato de arquivo. O CODEC converte o vídeo e som analógico em sinais digitais e depois os comprime, para diminuir o tamanho dos arquivos.

Cada CODEC provê certa qualidade de voz, gasta uma quantidade de largura de banda e apresenta um *delay* durante o processamento. O balanço de todas essas características levará ao administrador escolher o CODEC com a melhor relação custo/benefício. Dos parâmetros anunciados a medida de qualidade da voz transmitida é uma resposta subjetiva de um ouvinte. Uma medida comum usada para determinar a qualidade do som produzido pelos CODECs específicos é o MOS (*Mean Opinion Score*). Com o uso do MOS, uma ampla quantidade de ouvintes julgam a qualidade de uma amostra de voz

(correspondendo a um CODEC particular) numa escala de 1 a 5. A partir desses resultados, é calculada a média dos valores para atribuir o MOS para aquela amostra (CISCO-CODEC, 2005).

Na Tabela 1 são apresentados alguns valores dos parâmetros, citados anteriormente, dos CODECs mais usados. Na Tabela 2 são descritos os *valores* MOS.

Hoje o CODEC mais utilizado é o G.729 que possui uma boa qualidade de voz associada a um uso baixo de largura de banda. Porém o G.729 tem uma desvantagem que é o grande uso de processamento, mas mesmo assim o G.729 hoje é a melhor opção para compressão de voz.

Tabela 1: Valores MOS de alguns CODECs (CISCO-CODEC, 2005)

Método de Compressão	Largura de banda (kbit/s)	Valor MOS	<i>Delay</i> (ms)
G.711	64	4.1	0.75
G.726	32	3.85	1
G.728	16	3.61	3 a 5
G.729	8	3.92	10
G.729a	8	3.7	10
G.723.1	6.3	3.9	30
G.723.1	5.3	3.65	30

Tabela 2: Valores MOS (DASPL)

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

2.5 Centrais Telefônicas

O telefone foi criado em 1876 por Graham Bell (FERRARI, 2001). A telefonia trouxe grandes facilidades e avanço para a sociedade e se tornou uma necessidade no mundo de hoje dentro das residências e dentro de grandes e pequenas empresas.

Os componentes básicos do sistema telefônico são:

Telefones Analógicos: são os mais comuns em residências e pequenas empresas. Podem ser conectados diretamente a rede pública ou em interfaces terminais das centrais. Telefones proprietários são constantemente utilizados com as centrais privadas. Esses telefones provêm funções adicionais como viva-voz, controle de volume mensagem de espera do PBX e toques personalizados.

Local Loop: é a interface com a rede pública de telefone. Tipicamente é constituída por um par de fios em que pode trafegar uma ligação. Uma pequena empresa pode ter ainda múltiplos local loop.

Centrais telefônicas: são responsáveis pela sinalização, plano de numeração, encaminhamento de chamada e configuração das chamadas.

Troncos: são conexões entre duas centrais telefônicas públicas, entre uma central pública e uma central privada ou entre duas centrais privadas.

Da sua criação para os dias atuais algumas mudanças foram implementadas no funcionamento dos telefones. Antes a comutação dos circuitos era feita por uma telefonista que sentava a frente de um painel trocava as ligações dos cabos dos assinantes para que eles falassem entre si. Após vários avanços chegou-se hoje a centrais telefônicas automáticas.

As centrais telefônicas automáticas são capazes de comutar os circuitos automaticamente sem o auxílio de uma pessoa. Depois com a evolução ainda maior, com os baixos custos dos equipamentos e a crescente necessidade de telefones nas empresas foram criadas as Centrais Telefônicas Privada, os PBX.

O PBX hoje em dia é um equipamento essencial em qualquer empresa. Existem vários tamanhos de PBX com 20 a 20000 ramais e com diferentes funções que podem ser incorporadas de acordo com as necessidades de cada empresa.

3 VoIP

3.1 Tecnologia *Voice over IP*

VoIP é acrônimo de *Voice over IP*. É uma tecnologia que permite realizar chamadas telefônicas sobre uma rede de dados IP como se tivesse utilizando a rede STFC (Sistema de Telefonia Fixa Comutada), transmitindo a voz digitalizada em uma rede IP. Essa nova tecnologia está redefinindo a arquitetura de um PBX, estagnada por longos anos. Muitos dos componentes, citados a seguir, são distribuídos ao longo da rede para transmitir informações de voz e controle de ligação.

Controle de processo – é um servidor que executa uma aplicação que tem a função de controlar todas as chamadas como, por exemplo: o custo (bilhetagem) e qualidade delas e autenticação de usuários;

Dispositivos de ponta (*endpoints*) – são os telefones IPs ou *softphones* que se conectam diretamente na rede IP ao invés de interligar nos cartões de interfaces dedicadas aos módulos que existem nos PBX. Esses equipamentos precisam de um endereço IP e podem ser atualizados (*firmware*) através de um servidor TFTP com novas funcionalidades. Diferente do PBX tradicional, dois telefones IPs conversam diretamente, sem utilizar os recursos do servidor; e:

Gateway: são interfaces ou equipamentos que convertem a sinalização e o canal de voz para rede IP, fazendo a integração com a rede STFC. Permitem utilizar os telefones analógicos ou digitais existentes, reduzindo os custos da migração para a nova arquitetura.

3.2 Obstáculos para o VoIP

A convergência de dados e voz e até mesmo vídeo em uma única rede traz grandes economias hoje em dia, mas a inserção da voz na rede de dados precisa de cuidados especiais para que a solução não se torne um problema.

Como citado anteriormente, a rede IP tem como o princípio o melhor esforço, onde é usado de todos os esforços para a tentativa de entrega dos pacotes ao destino, mas não garantem até mesmo se eles vão chegar (pacote perdido), se vão chegar na ordem, se vai haver atraso na entrega (*delay*), ou se vai haver diferentes atrasos na entrega dos pacotes (*jitter*). Todos esses problemas são aceitáveis para o tráfego de dados na rede IP, mas não são aceitáveis no tráfego de voz que precisa da transmissão em tempo real. Abaixo estão descritos os principais problemas e as conseqüências para a transmissão de voz sobre a rede IP:

Perda de pacotes – a rede IP pode perder pacotes de voz se a qualidade da rede for ruim, se houver congestionamento na rede ou se o *delay* for muito variado. Alguns CODECs são capazes de corrigir pequenos defeitos ocasionados pela perda de pacotes, mas grandes perdas podem causar picotamento na voz. A maior causa de perda de pacotes é o congestionamento da rede que pode ser resolvida com o uso adequado de QoS.

Um exemplo de conseqüência na voz de perda de pacotes na rede é mostrado a seguir:

Mensagem original: Bom dia, como você está?

Mensagem no destino: Bom d, como vo está?

Delay – é o tempo que o pacote de voz leva para chegar da origem para o destino. O *delay* tem dois componentes:

Delay Fixo: causado pelos processos de codificação e decodificação da voz analógica, pela propagação no meio. O ITU-T através do padrão G.114 define que em uma via da transmissão o tempo de delay aceitável é de no máximo 150ms.

Delay Variável: causado pelo congestionamento da rede e pela serialização dos pacotes nas interfaces de rede. A serialização é uma função dependente da velocidade do *link* e do tamanho do pacote. Por isso esse tipo é considerável variável, pois depende do tamanho do pacote que chega a interface para ser serializado.

Um exemplo de consequência do *delay* é mostrado abaixo:

Mensagem original: Bom dia, como você está?

Mensagem no destino:Bom dia, como você está?

Jitter – é a variação do *delay* da chegada dos pacotes no destino. Esse problema pode ser solucionado utilizando os *buffers* dos terminais de forma que iguale todos os *delays* dos pacotes. Mas se um pacote chegar muito atrasado em relação aos outros, os *buffers* não serão suficientes para corrigir o erro.

Um exemplo de consequência do *jitter* na rede é mostrado abaixo:

Mensagem original: Bom dia, como você está?

Mensagem no destino: Bom dia, como você está?

Na transmissão de voz sobre IP são necessários CODECs para transformação da voz analógica em dados digitais mostrado em tópicos

anteriores e protocolos para a sinalização e envio dos pacotes de voz pela a rede IP apresentados a seguir.

3.3 Protocolos

Para a comunicação de dois terminais na telefonia IP duas etapas distintas ocorrem . A primeira é o processo de sinalização e controle da chamada e após a chamada estabelecida há o controle do transporte e o transporte da mídia. Como cada etapa pode utilizar vários protocolos distintos, para fins didáticos, o autor decidiu separar essas etapas em duas seções.

3.3.1 Processo de Sinalização de Chamadas

O processo de sinalização é composto de três sub-processos:

Estabelecimento da chamada: ocorre entre 2 ou mais terminais e envolve um ou mais *gateway* de voz para obtenção da informação dos terminais de uma mesma zona ou de zonas distintas. Estabelecida a chamada, são criados canais virtuais de controle entre todos os equipamentos envolvidos ou somente entre os equipamentos terminais.

Acompanhamento da chamada: é feito através dos sinais de controle no decorrer da chamada para identificar perda de conexão e outros eventos relevantes e dependentes dos serviços adicionais permitidos pelos terminais, quais sejam: atendimento simultâneo, chamada em espera e etc.

Finalização da chamada: libera os terminais e outros equipamentos envolvidos, libera os canais de controle e atualiza o estado dos terminais juntos aos equipamentos de rede.

Os protocolos mais utilizados para esses processos são: H.323, SIP, MGCP e MEGACO, que serão objeto de estudo a seguir.

O Padrão H.323

O padrão H.323 (ITU H323, 2000) é um conjunto de protocolos verticalizados para a sinalização e controle da comunicação entre terminais que suportam aplicações de áudio (Voz), vídeo ou comunicação de dados de multimídia. A Figura 3 mostra a pilha de protocolos implementados por terminais que utilizam o padrão H.323 em redes VoIP.

O H. 323 é uma recomendação do ITU-T³ que define padrões para comunicação multimídia através de redes que não oferecem Qualidade de Serviço (QoS) garantida, como é o caso de redes IP. Além disso, estabelece padrões para codificação e decodificação de fluxos de dados de áudio e vídeo, garantindo que produtos baseados no padrão H.323 de um fabricante seja compatível com produtos H.323 de outros fabricantes.

3 <http://www.itut.int>

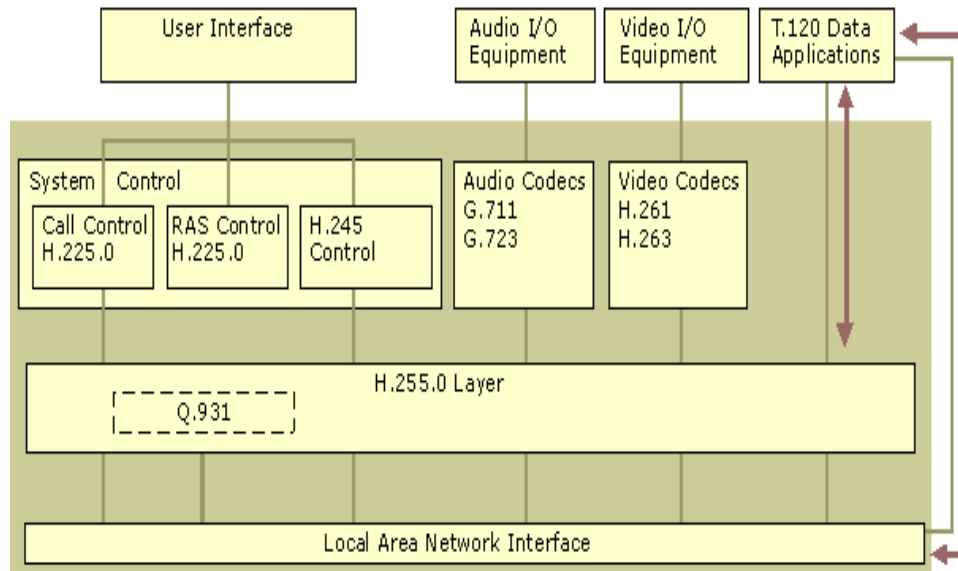


Figura 3 – Pilha de protocolos utilizados por terminais H.323

O padrão é completamente independente dos aspectos relacionados à rede. Dessa forma, podem ser utilizadas quaisquer tecnologias de enlace, podendo-se escolher livremente entre as que dominam o mercado atual como *Ethernet*, *Fast Ethernet*, *FDDI*, ou *Token Ring*. Ele define o formato do fluxo de dados e dos protocolos que os terminais usam para se comunicar uns com outro. Ele também define o gerenciamento e o controle dos protocolos usados entre terminais, *gateways* e MCU.

O H.323 usa tanto comunicação orientada a conexão (*Transport Control Protocol* - TCP) quanto não orientada (*User Datagram Protocol* - UDP) simultaneamente. Para os sinais de controle e dados que requerem transporte onde nenhum pacote pode ser perdido o TCP é utilizado. Por outro lado, o fluxo de áudio e vídeo perde sua qualidade com o tempo, sendo mais eficiente quando enviado por comunicação não orientada a conexão, pois a retransmissão de pacotes aqui não é bem vinda.

O H.323 é um padrão recente. Sua primeira versão foi divulgada em 1996, provendo um serviço sem garantia de QoS para ser utilizado em LANs e sistemas multimídia audiovisuais. Devido à necessidade de um padrão para voz sobre IP, o H.323 foi revisado e surgiu a versão 2. Na versão 3, foi adicionado suporte à comunicação *gatekeeper-gatekeeper*, ao fax sobre redes de pacotes e aos mecanismos de conexão rápidos. E, a mais recente, a versão 4, teve como foco importantes áreas, tais como confiabilidade, escalabilidade e flexibilidade. Através de esforços contínuos do grupo de estudo ITU-T, a recomendação H.323 continua evoluindo e adaptando-se a novas situações. O endereçamento coordenado globalmente e a consistência de QoS são duas áreas onde se esperam ver grandes melhoramentos no futuro.

A complexidade e a flexibilidade existentes no padrão H.323 dificulta sua implementação, pois vendedores de produtos e serviços H.323 freqüentemente escolhem implementar um subconjunto do mesmo que atenda seus requerimentos imediatos (Leopoldino & Medeiros, 2001). Existe uma alternativa ao H.323 que é o *Session Initiation Protocol* (SIP), que será assunto do próximo tópico.

SIP (Session Initiation Protocol)

O protocolo SIP definido através da recomendação RFC 2543 (Handley Et Al, 1999), estabelece o padrão de sinalização e controle para as chamadas entre terminais que não utilizam o padrão H.323, e possui os seus próprios mecanismos de segurança e confiabilidade.

O SIP estabelece recomendações para serviços adicionais tais como transferência, redirecionamento, identificação e autenticação de chamadas, conferência entre outros. É um protocolo baseado em conexões ponto-a-ponto e

utiliza para o transporte das suas mensagens o protocolo UDP.

O protocolo SIP tem dois tipos de agentes. Eles são os agentes usuários e os servidores de rede. Os agentes usuários podem ainda ser divididos em agentes usuários clientes e agente usuários servidores.

Usuários clientes são aqueles que fazem uma chamada SIP a outro terminal e usuários servidores são aqueles que recebem a ligação de um outro terminal. Os usuários SIP podem exercer as duas funções, sendo essa classificação somente didática.

Como exemplo de agentes usuários têm-se Telefones IP, *Softphones* e *Gateways*.

Já os servidores de rede SIPs podem ser quatro tipos:

Proxy Server: faz a intermediação das chamadas recebendo as requisições do usuário cliente e encaminhando-a ao usuário servidor. Com isso o usuário cliente não estabelece comunicação com o usuário servidor diretamente. Pode ser usado para garantir a segurança dos envolvidos na conexão, pois um cliente não saberá o endereço IP do outro e não precisará ter uma conexão direta. O servidor Proxy provê ainda funções como autenticação, autorização, controle de acesso à rede, roteamento, e segurança.

Redirect Server: provê o usuário cliente de informações sobre o usuário servidor. Diferente do *Proxy Server* que faz a requisição ao servidor, o *Redirect Server* somente fornece informações ao cliente sobre o endereço IP do servidor, sendo o cliente responsável por todo o estabelecimento da chamada, continuação e término da chamada.

Register Server: registra todos os clientes SIP que solicitam autenticação. Guardam informações como o endereço IP e identificação do cliente. Essas informações são armazenadas no Servidor de Banco de Dados.

Servidor de Banco de Dados: esse serviço não representa um servidor propriamente dito. É só um local para armazenar as informações dos clientes, como ramal, usuário, senha, endereço IP. Pode ser um banco de dados como LDAP, SQL, ou mecanismo do próprio sistema operacional em que o serviço esteja sendo executado.

As mensagens SIP diferentes das mensagens H.323 são bem fáceis de serem interpretadas pelo analista que deseja visualizar algum problema de conexão, por exemplo. Têm formato ASCII e são padronizados pela RFC 822 (Crocker, 1982), padrões de mensagens texto da Internet e RFC 2068 (Fielding Et Al, 1997), que é o padrão para mensagens HTTP. Abaixo o exemplo de algumas mensagens SIP usadas na sinalização:

INVITE – através dessa mensagem o cliente que deseja realizar a chamada convida outro cliente para fazer a conexão;

ACK – um cliente manda ACK para indicar que recebeu a mensagem de resposta do convite de conexão;

BYE – mensagem que indica que o cliente deseja desconectar;

CANCEL – um cliente envia para interromper qualquer requisição que esteja em processo. Não pode ser utilizado para terminar conexões ativas.

O formato de endereçamento SIP também é bem simples e podem ser das seguintes formas:

Fully qualified domain names

sip:arl@ginix.comp.ufla.br

Endereço E.164: padrão de endereçamento de telefone.
sip:24408765@ginix.comp.ufla.br

MGCP (*Media Gateway Control Protocol*)

O protocolo MGCP, definido através de recomendação RFC 2705 (Arando Et Al, 1999), é usado para controlar as conexões (chamadas) dos *gateways* presentes no sistema VoIP. O MGCP implementa uma interface de controle usando um conjunto de transações do tipo comando – resposta que cria, controla e audita as conexões no *gateways*. Essas mensagens usam como suporte o protocolo UDP da rede IP, e são trocadas entre *gateways* para o estabelecimento, acompanhamento e finalização das chamadas. Diferente dos protocolos SIP e H.323 onde não é necessário à intermediação de um servidor para a execução das chamadas, no MGCP é necessário um servidor. Este texto não irá detalhar o MGCP por não ser muito comum ainda seu uso com o Asterisk. Para mais detalhes ver (Arando Et Al, 1999).

MEGACO (*Media Gateway Control Protocol*)

O protocolo MEGACO é resultado de um esforço conjunto do IETF⁴ e do ITU-T (grupo de estudo 16). O texto da definição do protocolo é o mesmo para o IETF (Cuervo Et Al, 2000) e a recomendação H.248 (ITU H248, 2002), e representa uma alternativa ao MGCP e outros protocolos similares.

4 <http://www.ietf.org>

Este protocolo foi concebido para ser utilizado para controlar os *Gateways* monolíticos (um único equipamento) ou distribuídos (vários equipamentos). Possui interface de sinalização para diversos sistemas de telefonia, tanto fixa como móvel.

Após o estabelecimento da chamada com a sinalização, vem o transporte dos pacotes de voz propriamente dito, que também tem duas etapas distintas. Este texto não irá detalhar o MEGACO por seu uso não ser muito comum com o Asterisk. Para mais detalhes ver (Cuervo Et Al, 2000) ou (ITU H248, 2002) .

3.3.2 Processo de Controle do Transporte e Transporte da Mídia

Os processos de controle do transporte e o do transporte de mídia são descritos abaixo:

Controle de transporte de mídia: estabelecida à chamada, os terminais e *gateways* de voz iniciam um processo de definição do mecanismo de transporte de voz, identifica o tipo de mídia a ser transportada, nesse caso voz e são criados os canais virtuais de controle e de mídia.

Transporte de mídia: inicia-se o transporte bidirecional em tempo real de mídia (Voz) entre os terminais envolvidos através dos canais virtuais criados na fase anterior. São usados recursos do protocolo UDP para minimizar o *overhad* do protocolo, otimizando a rede.

A telefonia IP utiliza o protocolo UDP da rede como infra-estrutura para

o transporte dos pacotes mencionados a cima. O protocolo UDP foi escolhido por apresentar algumas características interessantes para a transmissão em tempo real. No protocolo UDP o cabeçalho introduzido é menor que o do protocolo TCP, trazendo um menor *overhad*. Isso se dá porque o pacote de voz é pequeno, em torno de 20 *bytes*, e o cabeçalho TCP que tem o tamanho de pelo menos 20 *bytes* seria do tamanho ou maior que o pacote de voz. Também algumas propriedades do TCP interessantes para dados, não são para voz. Por exemplo, no caso da perda de algum pacote de voz, não há o interesse em retransmissão do mesmo, pois a voz precisa de tráfego em tempo real e com certeza esse pacote chegaria depois de outros que foram enviados anteriormente.

Mas com o UDP os pacotes de voz perderiam uma propriedade interessante do TCP que o UDP não possui que é o ordenamento dos pacotes que chegam por ventura fora de ordem. Para isso foram criados os protocolos RTP e RTCP. O primeiro além de outras funções, tem a capacidade de fazer o reordenamento dos pacotes. Já o RTCP é utilizado para transmitir pacotes de controle periódicos sobre a transmissão de voz. O RTP e RTCP então são usados para todo o tráfego de voz pela rede utilizando o UDP, incluindo os protocolos H.323 e SIP. A seguir têm-se mais detalhes desses protocolos usados na transmissão de pacotes de dados.

RTP (*Real-Time Protocol*)

Definido através da recomendação RFC1889 (Schulzrinne Et Al, 1996), o RTP é o principal protocolo utilizado pelos terminais em conjunto com o RTCP, para o transporte fim-a-fim em tempo real de pacotes de mídias (Voz) através de redes de pacotes. Pode fornecer serviços *multicast* ou *unicast*.

O RTP não reserva recursos de rede e nem garante qualidade de serviço para tempo real. O transporte de dados é incrementado através do RTCP (protocolo de controle) que monitora a entrega dos dados e provê funções mínimas de controle e identificação. As funções do protocolo RTP são: identificar o tipo de dado transmitido (voz ou vídeo), seqüência numérica, *time stamping* e monitoração de entrega.

O RTP é um componente crítico do VoIP, porque permite que o destino organize os pacotes que chegam, antes de ser transformado em voz e apresentado ao usuário. Utiliza números seqüências somente para ordenar os pacotes, não há retransmissão em caso de perda de pacote.

RTCP (*Real-Time Control Protocol*)

Definido também através da RFC 1889 (Schulzrinne Et Al, 1996), o RTCP é baseado no envio periódico de pacotes de controle a todos os participantes da conexão, usando o mesmo mecanismo de distribuição dos pacotes de mídia. Desta forma, com um controle mínimo é feito à transmissão de dados em tempo real usando o suporte dos pacotes UDP (para Voz e controle) da rede.

3.4 Equipamentos VoIP

A empresa que mais investiu e a mais especializada em transmissão de Voz sobre o protocolo IP atualmente é a CISCO *Networks*®. A CISCO possui *hardware* e *software* especiais para a transmissão de voz na rede IP, se tornando

líder no mercado, apesar dos elevados preços dos equipamentos e programas.

Roteadores CISCO podem ter conexão com centrais PBX, telefônicos analógicos, fazer conexão com o STFC, funcionarem com PBX, URA, dependendo das propriedades dos equipamentos, placas instaladas e configurações.

As vantagens dos equipamentos CISCO é que a maioria é utilizada para topologia da rede, podendo acumular função também no uso da tecnologia VoIP, o alto investimento em pesquisa e melhoria da qualidade dos equipamentos e programas, a grande quantidade de documentação e cursos disponíveis para o profissional que deseja se especializar em VoIP e a compatibilidade com a maioria dos protocolos hoje disponíveis para a tecnologia VoIP.

A principal desvantagem é o alto custo. Módulos de Voz podem chegar a custar US\$10.000,00, *software* de gerenciamento custam em torno de US\$50.000,00, e cursos oficiais custam em torno de R\$5.000,00, o que impossibilita o acesso da tecnologia a pequenas e médias empresas e particulares.

3.5 Clientes

Atualmente há uma grande quantidade de clientes VoIP. Eles podem ser *hardphones* e *softphones*. Os primeiros são telefones IP que já são capazes de transformar a voz analógica em pacotes para trafegarem na rede IP. São conectados diretamente na rede, possuem configuração estática e dinâmica de endereço IP. Possuem recursos de conferência, siga-me, transferência e são compatíveis com todos os protocolos de sinalização disponível no mercado. Há vários fabricantes com diferentes características proprietárias ou não como, por

exemplo, CISCO e 3COM.

Os *softphones* são *software* que rodam em máquinas PC, que utilizam os recursos operacionais para digitalizar a voz e enviar para a rede IP. Possuem interface gráfica bem intuitiva com teclados para discagem dos números, botões para atendimento de chamadas. São capazes de fazer conferência, vídeo-conferência e transferência de chamadas. A maioria dos *softphones* é gratuita e há versões para os principais sistemas operacionais. Alguns exemplos são: X-Lite (cliente SIP) ilustrado na Figura 4 e o Netmeeting (cliente H.323) ilustrado na Figura 5.



Figura 4 – Janela do X-Lite



Figura 5 – Janela do NetMeeting

4 Asterisk

Asterisk⁵ é um *software* GPL⁶ que pode transformar um simples PC em um poderoso e completo PBX. Ele roda em Linux e provê todas as funcionalidades que o leitor poderia esperar de um PBX e muito mais. Foi criado por Mark Spencer da Digium⁷ que apóia o projeto e comercializa o *hardware* de telefonia. Asterisk faz Voz sobre IP em diferentes protocolos e pode se integrar à maioria dos padrões de telefonia utilizando *hardware* de baixo custo e aberto, desenvolvido por Jim Dixon (Gonçalves, 2005).

Seu nome vem do símbolo, *, o qual em Unix e Linux representa um *wildcard* que equivale a qualquer caractere, qualquer arquivo. Similarmente, o Asterisk é destinado a integrar qualquer peça de telefonia, seja *hardware* ou *software*, a aplicações. Para ilustrar essa propriedade do Asterisk abaixo estão listados os protocolos e CODEC que há compatibilidade.

- Protocolos Abertos:
- IAX™ (Inter-Asterisk *Exchange*);
- H.323 ;
- SIP (Session Initiation Protocol) ;
- MGCP (Media Gateway Control Protocol); e
- SCCP (Cisco® Skinny®).

5 <http://www.asterisk.org>

6 <http://www.gnu.org>

7 [Http://ww.digium.com](http://ww.digium.com)

Protocolos proprietários:

- 4ESS;
- BRI (ISDN4Linux);
- DMS100;
- EuroISDN;
- Lucent 5E;
- National ISDN2; e
- NFAS

Codecs:

- ADPCM;
- G.711;
- G.723.1;
- G.726;
- G.729;
- GSM;
- iLBC;
- Linear;
- LPC-10; e
- Speex.

4.1 Arquitetura do Asterisk

O Asterisk foi desenvolvido para ter o máximo de flexibilidade. Algumas APIs especiais foram definidas em torno do núcleo. Dessa maneira o Asterisk fica transparente a protocolos, CODECs e *hardware*, podendo ser compatível com qualquer tecnologia existente ou que venha a ser lançada, sem que sejam necessárias mudanças no núcleo do programa. Carregar os módulos separadamente também permite maior flexibilidade ao administrador, permitindo que ele escolha a melhor e mais enxuta configuração que o atenda. Abaixo a Figura 6 representa a arquitetura básica do Asterisk.

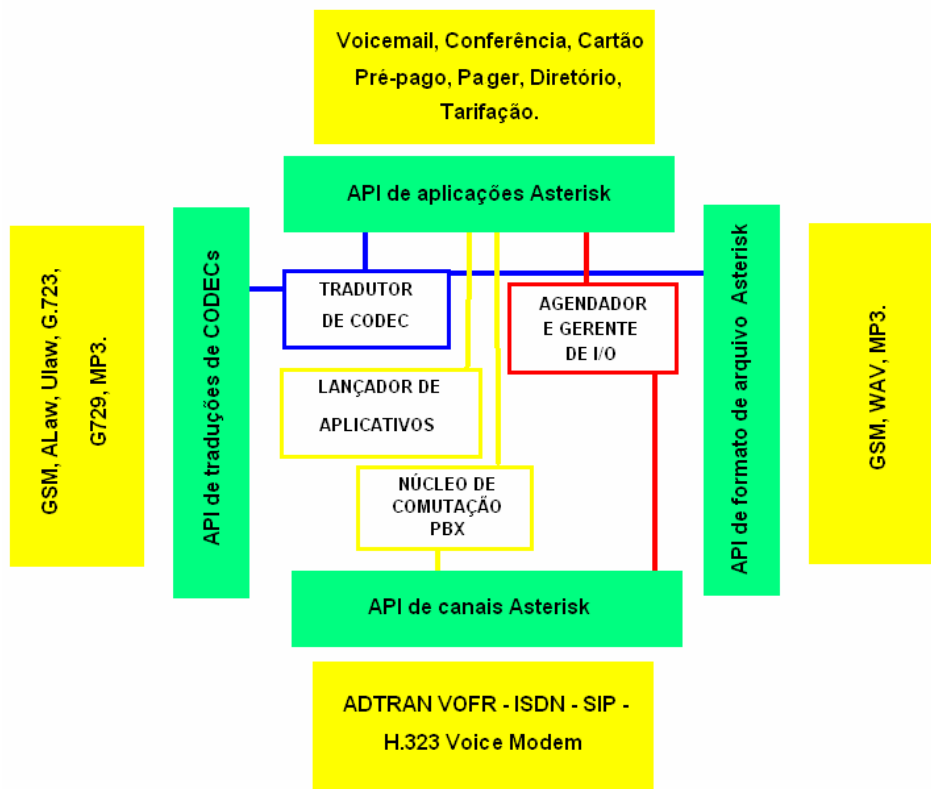


Figura 6 – Arquitetura do Asterisk

O núcleo do programa é dividido em quatro módulos principais:

Módulo PBX: é o módulo principal. Ele é responsável pela conexão das chamadas de vários usuários e tarefas automáticas. Para o módulo são transparentes os diferentes *hardware* ou diferentes *software* que fazem as chamadas.

Lançador de Aplicativos: esse módulo é responsável pela execução dos serviços para os usuários como *voicemail* e a lista de diretórios.

Tradutor de CODECs: esse módulo é responsável pela a tradução dos CODECs suportados pelo Asterisk.

Módulo de Agendamento e Gerenciamento de I/O: esse módulo, como já indica o nome é responsável pelo agendamento de tarefas de baixo nível do programa e gerencia o sistema para uma melhor performance.

As APIs podem ser divididas em quatro tipos principais:

APIs de Canal: esse tipo de API é responsável de fazer a compatibilidade do núcleo com os diversos tipos de conexões onde as chamadas podem ser originadas, como uma conexão VoIP, ISDN e outros tipos a mais. Os módulos são carregados para cada tipo de conexão para fazer compatibilidade com as camadas mais baixas.

APIs aplicativos: essas APIs permite que várias funcionalidades possam ser executadas como: Conferência, Listagem de Diretório e *Voicemail*.

APIs de Tradutor de CODECs: esses módulos são carregados para suportar vários codificadores e decodificadores de áudio com GSM, G.711 e etc.

APIs de formato de arquivo: permite ao Asterisk ler e escrever em vários formatos de arquivos para armazenar no sistema de arquivos.

4.2 Aplicações do Asterisk

O Asterisk tem uma gama de aplicações muito grande podendo substituir qualquer PBX tradicional com periféricos adicionais, reunindo-as todas em uma só máquina, em só programa.

As aplicações mais usuais são:

Correio de voz: permite que quando o usuário que não recebeu a ligação não atenda ao telefone, o usuário chamador recebe um sinal solicitando que deixe uma mensagem da caixa postal. É semelhante a uma secretária eletrônica.

Distribuidor automático de chamadas e fila de atendimento: é o sistema utilizado em *Call-Centers*. Ele atende a ligação efetuada pelo usuário e caso não tenha nenhum operador disponível automaticamente uma mensagem ou uma música toca para que o usuário aguarde. No primeiro atendente que é liberado, o distribuidor encaminha a ligação.

Sistema de mensagens unificadas: esse sistema direciona todas as mensagens dos usuários em um único lugar como, por exemplo, a caixa de correio eletrônico do usuário. As mensagens de e-mail, mensagens de correio de voz e fax são encaminhadas para a caixa postal.

Servidor de música de espera: esse sistema é responsável pela música de espera. A maioria das centrais é necessário um CD ou um rádio acoplado. No Asterisk a música pode estar em arquivo no formato MP3.

Discador automático: utilizado em *telemarketing*, o programa discar para um número e distribui para os operadores.

Sala de conferência: permite que 2 ou mais usuários falem em conjunto. Pode

ser configurada uma senha para acesso a sala.

Media Gateway: converte as ligações em telefonia analógica para telefonia Voz sobre o IP.

Unidade de resposta automática: como nome já sugere, essa funcionalidade permite o Asterisk a criar um sistema de resposta automática, muito usado hoje em dia, como por exemplo, em atendimentos telefônicos bancários.

4.3 Hardware de Telefonia para o Asterisk

O Asterisk suporta inúmeros tipos de *hardware* de telefonia, de inúmeros fabricantes, mas principalmente o *hardware* fabricado pela Digium⁸. Abaixo a lista de *hardware* de telefonia compatíveis.

- Wildcard TE411P
- Wildcard TE410P
- Wildcard TE406P
- Wildcard TE405P
- Wildcard TE210P
- Wildcard TE205P
- Wildcard TE110P
- Wildcard TDM400P
- S100I (IAXy)

8 <http://www.digium.com>

- Wildcard X100P
- Wildcard T100P
- Wildcard E100P
- Generic X100P
- OpenLine4
- OpenSwitch6/12

Este Trabalho mostrará as placas mais utilizadas no Brasil que são: a XP100P, a Wildcard TDM400P e a Wildcard TE110P. Para mais detalhes sobre a instalação e configuração das outras placas recomenda-se visita ao site do Asterisk na URL: <http://www.asterisk.org> .

4.3.1 Placa XP100P

A placa XP100P é uma das placas mais simples com uma porta FXO, que pode ser ligada à rede pública ou a uma interface de ramal de um PBX. Uma placa de fax/modem com *chipset* Intel 537 ou MD3200 pode ser usada como uma XP100P.



Figura 7 – Placa XP100P

4.3.2 Placa TDM400P

A placa TDM400P é uma placa analógica que pode conter até quatro canais. Os canais podem ser FXO (pode se ligar à rede pública ou ramal de um PBX) ou FXS (pode se ligar um assinante, como um telefone analógico) dependendo da configuração escolhida.



Figura 8 – Placa TDM400P

4.3.3 Placa TE110P

A placa TE110P é uma placa com 30 canais digitais no padrão E1-ISDN. Com esta placa o Asterisk pode conectar-se à rede pública ou a central telefônica.



Figura 9 – Placa TE110P

4.4 Instalação do Driver do Hardware de Telefonia

O procedimento de instalação descrito neste Trabalho refere-se às placas Digium por serem mais comuns hoje em dia, por serem líder de mercado. O código fonte dos *drivers* pode ser obtido em <ftp://ftp.digium.com> ou a última versão dos *drivers* pode ser baixada do servidor CVS: <cvs.digium.com> . Para baixar os códigos fonte do servidor CVS basta executar os comandos:

```
# cd /usr/src
# export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
# cvs login          (a senha é “anoncvs”)
# cvs checkout zaptel libpri
```

Com esses comandos os diretórios *zaptel* e *libpri* serão criados. Para instalar é só entrar em cada um dos diretórios e executar os comandos:

```
# cd zaptel
# make clean; make install
# cd ../libpri
# make clean; make install
```

Para qualquer distribuição que utilize o *kernel2.6*, para a instalação do *driver* *zaptel* é necessário executar o comando *make linux26*, ficando assim a seqüência de comandos:

```
# cd zaptel
# make clean; make linux26; make install
# cd ../libpri
# make clean; make install
```

Uma vez instalado os *drivers* é necessário carregar os módulos das placas, mas antes de carregar qualquer módulo é necessário carregar o módulo *zaptel* executando o comando:

```
# modprobe zaptel
```

Os módulos das placas são carregados também com o comando *modprobe*. Cada placa instalada há um módulo diferente que deve ser carregado, como mostra a Tabela 3:

Tabela 3 – Módulos das Placas Digium (DIGIUM)

PLACA	DRIVER
T100P/E100P	wct1xxp
TE405P/TE410P	wct4xxp
TE110P	wctel1xp
Placas TDM	wctdm
XP100P	wcfxo

4.4.1 Configuração dos Hardware de Telefonia

Para a configuração dos parâmetros regionais, como: toque de chamada, desocupado, etc e sinalização dos *hardware* de telefonia o arquivo *zaptel.conf* deve ser editado. Esse arquivo contém muitas informações que não serão tratadas neste Trabalho, para, mas informações consultar em <http://www.asterisk.com>.

Os parâmetros principais a serem configurados são:

loadzone: permite configurar o tipo de tons que a interface irá utilizar. A lista disponível de tons está definida no arquivo `/usr/src/zaptel/zonedata.c`.

fxsks: configura o tipo de sinalização na porta do *hardware*. Neste caso a porta é configurada como *station* (pode ser conectada a um telefone analógico).

fxoks: configura a sinalização como *office* (pode ser conectada à rede pública ou a um PBX).

Um exemplo do arquivo *zaptel.conf* usado para configurar uma placa TDM400P ficaria assim:

```
#  
# Arquivo de configuração zaptel  
#  
fxoks = 1      ; configura a porta 1 da TDM400P com sinalização fxo  
fxsks = 4      ; configura a porta 4 da TDM400P com sinalização fxs  
loadzone = Br  ; configura o padrão de toque brasileiro
```

Após editar o arquivo e carregar o módulo da placa, para configurar basta executar o comando com o modo de visualização para verificar algum erro:

```
# ztcfg -v
```

4.5 Instalando o Asterisk

4.5.1 Hardware Mínimo

O Asterisk faz um uso intensivo de CPU, pois a utiliza para o processamento dos sinais digitais. Para a construção de um primeiro PBX IP para teste um processador de 300MHz com 256MB de RAM é o suficiente. Já o uso de disco rígido não é elevado, ele só requer 100MB para o programa compilado. Mas a inserção de código fonte, *voicemail*, *prompts* customizados requerem mais espaço de acordo com a quantidade de usuários. Para uso de outras funções do Asterisk, algumas placas PCI são necessárias, mas elas serão abordadas em outro tópico.

4.5.2 Requisitos Necessários do Sistema Operacional

O Asterisk foi feito para funcionar em Linux, apesar de rodar em BSD e MAC OS X. Já quanto a pacotes, não há nenhum requisito especial, somente as

bibliotecas *zlib*⁹ e *zlib-devel*¹⁰ e o pacote *bison*¹¹.

4.5.3 Processo de Instalação

O código fonte do Asterisk pode ser baixado do site oficial: <http://www.asterisk.org>, do ftp da Digium: <ftp://ftp.digium.com> ou no servidor CVS: <cvs.digium.com>.

Neste Trabalho será usado o servidor CVS pelo fato deste autor acreditar ser a melhor opção. Para baixar o Asterisk os seguintes comandos devem ser executados:

```
# cd /usr/src
# export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
# cvs login          (a senha é "anoncvs")
# cvs checkout asterisk asterisk-sounds asterisk-addons
```

Após baixar o código fonte do Asterisk, deve-se compilá-lo executando os comandos:

9 <http://www.zlib.net>

10 <http://www.zlib.net>

11 <http://www.gnu.org>

```
# cd /usr/src/asterisk
# make clean
# make
# make install
# make samples
```

4.6 Configuração do Asterisk

A configuração do Asterisk é feita através de arquivos de configuração localizados no diretório */etc/asterisk* . O formato dos arquivos é o ASCII dividido em seções com o nome da seção entre colchetes ([]). Em seguida os pares de chave, valores separados por um sinal de igual (=) ou por um sinal seguido pelo sinal de maior que (=>). O caractere de comentário é o ponto e vírgula e linhas em branco são ignoradas. Um arquivo de exemplo é mostrado abaixo:

```
;
; Comentário
;
[sessao1]
chave = valor ; Designação de variável
[sessao2]
objeto => valor ; Declaração de objeto
```

A configuração básica do Asterisk é bastante simples e envolve três arquivos de configuração. O arquivo de configuração de canais é o *zapata.conf*.

O protocolo adotado neste Trabalho foi o protocolo padrão do Asterisk, o SIP, que é configurado no arquivo: *sip.conf*. O motivo de escolha desse protocolo não foi somente o fato dele ser padrão no Asterisk, mas também de possuir mais clientes. E por último o arquivo de configuração do plano de numeração é o *extensions.conf*. O Asterisk possui outros arquivos de configuração que não serão abordados em detalhes neste Trabalho, para maiores informações no site: <http://www.asterisk.org> ou (Gonçalves, 2005) .

4.6.1 O Arquivo *zapata.conf*

Este arquivo contém parâmetros relacionados aos canais TDM fornecidos pela interface *zaptel* que o Asterisk utilizará. As seguintes palavras-chave compõem o arquivo (ALLISON & SPENCER, 2002) :

context: define o grupo de chamadas para os canais. O valor do **context** será referenciado mais tarde no arquivo *extensions.conf* . Esse parâmetro será usado para prover diferentes aplicações e extensões para diferentes usuários do sistema. Exemplo:

```
context = default
```

channel: indica uma lista de canais, que poderão receber configurações mais adiante no arquivo. Aqui os canais podem ser especificados individualmente, separados por vírgulas ou como uma faixa separada por hífen. Exemplo:

```
channel=>1-15  
channel=>16  
channel=>17,18
```

callerid: configura a identificação da chamada. A identificação é transmitida para outros telefones conectados ao Asterisk ou a telefones da rede pública. Exemplo

```
callerid= "Bruno Pinheiro" <(21)2445-1234>
```

group: configura um grupo para propósito de discagem. Esse grupo criado aqui pode ser usado com referência no arquivo *extensions.conf*. Qualquer chamada para o grupo, o primeiro canal disponível será utilizado. Se os canais forem telefones todos receberão a chamada ao mesmo tempo. Exemplo:

```
group=2  
group=1,3
```

Algumas palavras-chave podem ser utilizadas para habilitar ou desabilitar algumas funções em canais ou grupo de canais. Essas configurações são efetuadas com *yes* ou *no*. A seguir algumas das palavras mais utilizadas são mostradas, para mais detalhes ver em (ALLISON & SPENCER, 2002) .

usecallerid: habilita a transmissão do identificador de chamadas;

hidecallerid: configura se o identificador de chamadas será ocultado;

calleridcallwaiting: configura se a identificação de chamadas será recebida durante uma chamada em espera;

threewaycalling: permite a conferência a três no canal;

transfer: permite a transferência de chamadas no canal;

cancallforward: habilita o siga-me de chamadas.

4.6.2 O Arquivo sip.conf

Este arquivo configura o modo global do funcionamento do Asterisk com o protocolo SIP e os clientes que irão conectar no servidor. Esse arquivo possui duas seções, a primeira, a seção geral [general] onde são configuradas todas as opções do uso do SIP pelo Asterisk. A segunda seção configura os clientes que irão conectar ao Asterisk através do protocolo SIP.

A seção geral do SIP possui as seguintes palavras-chave (GONÇALVES, 2005) :

allow = <codec> : configura os CODECs que serão utilizados em conexões SIP;

disallow = all : desabilita o uso de todos os CODECs;

bindaddr = 0.0.0.0 : endereço IP do servidor Asterisk;

externip = <endereço ip> : endereço IP que será colocado nas mensagens SIP, utilizado se o servidor Asterisk estiver “atrás” de um dispositivo NAT;

localnet = 10.32.48.0/255.255.252.0 : configura o endereço local e máscara da rede onde o servidor está instalado;

maxexpirey = <tempo em segundos> : configura o tempo máximo de registro que o Asterisk irá permitir;

port = <portno> : porta TCP qual a qual os clientes SIP irão se conectar;

e

videosupport = yes|no : habilita ou não o suporte a vídeo do SIP.

A seção que configura usuários é separada por blocos um para cada usuário, como mostra o exemplo abaixo:

```
[xxx]
type = yyy
parâmetro1 = valor1
parâmetro2 = valor2
```

Na seção de usuários podem ser usadas as seguintes palavras-chave:

accountcode = <código> : associa o usuário a um código de contabilização de chamadas;

callgroup = <grupo> : associa o usuário a um grupo criado anteriormente;

host = configura o endereço IP ou nome do *host* do cliente. Se usado à opção *dynamic* o cliente terá que se registrar;

defaultip = <endereço IP> : configura o endereço IP padrão do cliente, se especificado *host=dynamic*. Utilizado se o cliente não se registrou em outro endereço IP;

fromuser = <nome do usuário> : configura o nome do usuário que será transmitido para a identificação da chamada. Importante: esse parâmetro sobrescreve o *callerid* configurado anteriormente;

fromdomain = <nome do domínio> : configura o nome do domínio que será transmitido para a identificação de chamadas;

restrictid = yes|no : não permite a identificação de chamadas, oculta o *callerid*;

port = <número da porta> : configura a porta que o cliente SIP irá se

conectar ao servidor Asterisk;

rtptimeout = <tempo em segundos> : configura o tempo em que a chamada será cancelada caso não haja atividade RTP, quando não estiver em espera. Importante: muito útil para desbloquear canais que possam ter ficado bloqueados por algum problema na desconexão;

username = <nome do usuário> : configura o nome do usuário; e

secret = <senha> : configura a senha do usuário.

Para mais detalhes sobre a configuração do protocolo SIP no Asterisk o autor recomenda a leitura de (GONÇALVES, 2005) .

4.6.3 O Arquivo *extension.conf*

No arquivo *extension.conf* que é configurado o plano de discagem, a parte mais importante do Asterisk. Ele controla como todas as chamadas de entrada e saída são encaminhadas e manuseadas. É nesse arquivo que há o controle de todas as conexões através do PBX.

O arquivo *extension.conf* é dividido em três seções: geral [general] , global [globals] e a última seção com vários blocos onde é definido o plano de discagem.

A seção global possui os seguintes parâmetros:

static = yes|no

writeprotect = yes|no

Esses dois parâmetros afetam o comando *save dialplan* . Esse comando salva o plano de numeração atual alterando o arquivo *extension.conf*, não

fazendo nenhuma cópia do arquivo anterior. Como *default* esses parâmetros são: **static** =yes e **writeprotect** = no, que é uma combinação perigosa porque permite que o comando *save dialplan* altere o *extension.conf*.

Na parte global do arquivo são definidos as variáveis globais e seus valores. Alguns exemplos de variáveis que podem ser definidas na seção global são:

Definir que extensões devem tocar quando uma chamada chegar:

```
INCOMING = Zap/3&Zap/4
```

Quantas vezes o telefone deve tocar antes de ir para o *voicemail*:

```
TOQUES = 3
```

Que arquivo de música deve ser tocado como anúncio que o usuário foi transferido para o *voicemail*:

```
ARQUIVOMP3 = musicas/voicemail.mp3
```

Um exemplo do uso de variável global é mostrado a seguir:

```
exten=>811,1,Dial($INCOMING, $TOQUES)  
exten=>811,2,Voicemail(u811),$ARQUIVOMP3  
exten=>811,102,Voicemail(b811),$ARQUIVOMP3
```

A última seção do arquivo *extension.conf* é aonde será configurado o plano de discagem propriamente dito. O plano de discagem é formado por um conjunto de contextos, que por sua vez são formados por um conjunto de extensões.

Os contextos e extensões criam regras de comportamento para o Asterisk quando uma chamada chega a ele. Essa chamada pode receber vários tratamentos como: encaminhamento a outra extensão, um aplicativo, menu e etc. Essas regras têm muita versatilidade e muitas opções, implementado inúmeros recursos, incluindo:

Segurança: permitir ligações de longa à distância somente de certos telefones;

Roteamento: rotear chamadas baseadas nas extensões;

Auto-atendente: receber a chamada e encaminhar para uma extensão de acordo com a escolha do usuário;

Horários: permitir tratamentos diferentes das chamadas de acordo com a hora; e

Macros: utilizado para criar scripts para implementar funções normalmente usadas.

Todos os recursos implementados pelas extensões não serão abordados por neste Trabalho, para mais detalhes recomenda-se à leitura de (Gonçalves, 2005).

Um exemplo de configuração de contextos e extensões:

Contexto Padrão

Extensão	Descrição
1111	Tatiane
1112	Alexandre
1113	Verificar Correio
1114	Sala de Conferência
0	Telefonista

Neste exemplo os dois primeiros ramais são utilizados por telefones comuns de usuários. O quarto e quinto ramal são usados para encaminhar a chamada a um aplicativo, e o sexto ramal é associado a um operador.

Os contextos são criados definindo o nome entre colchetes seguidos da definição das extensões. Já as extensões têm o seguinte formato:

exten => [ramal],[prioridade],[aplicação],[parâmetros]

[ramal] – número do ramal discado;

[prioridade] – define a prioridade da extensão, a ordem de execução das linhas;

[aplicação] – define a aplicação que a extensão vá executar, se for o caso; e

[parâmetros] – dependem da aplicação. Algumas aplicações não têm parâmetros.

Alguns exemplos de contextos e extensões criadas são mostrados abaixo.

Contexto simples para ligação entre ramais:

```
[default]
exten=>1,1,Dial(Zap/1)
exten=>2,1,Dial(Zap/2)
exten=>3,1,Dial(Zap/3)
exten=>4,1,Dial(Zap/4)
exten=>5,1,Dial(Zap/5)
```

Nesse exemplo foi criado o contexto *default*. Todas as chamadas encaminhadas para esse contexto receberão o tratamento descrito por essa regra. Se o usuário discar o ramal 1, a chamada será enviada para o canal 1, se for discada o ramal 2, será enviada ao 2 e assim sucessivamente.

Contexto com menu de auto-atendimento e encaminhamento de chamada:

```
[entrada]
exten=>s,1,Background(bem_vindo)
exten=>1,1,Dial(Zap/1)
exten=>2,1,Dial(Zap/2)
exten=>3,1,Dial(Zap/3)
exten=>4,1,Dial(Zap/4)
exten=>5,1,Dial(Zap/5)
```

Nesse agora foi acrescentado um menu de auto-atendimento que ficará aguardando o usuário discar um número para encaminhar a chamada. Enquanto aguarda a digitação do número, o Asterisk tocará o arquivo *bem_vindo.gsm*.

Contexto com encaminhamento para o *voicemail*:

```
exten=>811,1,Dial(Zap/1,20)
exten=>811,2,Voicemail(u811)
exten=>811,102,Voicemail(b811)
```

No exemplo o Asterisk encaminhará a chamada para a interface Zap/1 por 20 segundos. Caso expire o tempo à chamada será encaminhada para o *voicemail* com uma mensagem de “não atende” (*unavailable*). Se o ramal estiver ocupado, a chamada também será enviada ao *voicemail*, mas desta vez com a mensagem de ocupado (*busy*).

Contexto com roteamento de acordo com o originador da chamada:

```
exten=>811/36520583,1,Congestion
exten=>811,2,Dial(Zap/1,20)
exten=>811,3,Voicemail(u811)
exten=>811,103,Voicemail(b811)
```

Neste exemplo, feito com base no anterior, se o originador de chamada for o número indicado logo após a barra, o Asterisk retornará o tom de ocupado. Qualquer outro número seguirá a regra do exemplo anterior.

Muitas outras configurações são possíveis de serem criadas com contextos e extensões, mesclando todas essas opções mostradas e outras não citadas.

Exemplos mais completos de arquivos *zapata.conf*, *sip.conf* e *extension.conf* são mostrados no Apêndice.

5 Projeto de Implementação do Asterisk

O projeto de implementação do Asterisk tem cenário no Departamento de Controle do Espaço Aéreo¹² (DECEA) no Comando da Aeronáutica¹³. O VoIP do Asterisk substituirá a Rede Telefônica do Comando da Aeronáutica.

5.1 O DECEA

O Departamento de Controle do Espaço Aéreo no Comando da Aeronáutica (DECEA) tem como missão o controle de todo o espaço aéreo brasileiro, controlando todos os vôos civis e militares e de defesa aérea, vigiando os céus do Brasil. Tem como missão também manter a operacionalidade de toda a Telecomunicação no âmbito do Comando da Aeronáutica.

Um dos componentes do complexo sistema de telecomunicações é a Rede Telefônica do Comando da Aeronáutica (RTC aer) que interliga todos os comandantes de todas as unidades da Força Aérea Brasileira. Essa rede possui uma central exclusiva no Rio de Janeiro e ramais remotos espalhados por todo o território nacional.

Essa rede é de grande importância no dia-a-dia aeronáutico porque provê a comunicação de todos os comandantes, de forma privativa, onde os mesmos trocam informações administrativas e operacionais.

12 <http://www.decea.gov.br>

13 <http://www.fab.mil.br>

5.2 O Projeto

A RTCaer hoje está obsoleta, com uma central telefônica muito antiga e com poucas peças de reposição. Outros fatores desfavoráveis são:

- inúmeros *links* de voz contratados nas operadoras interligando todos os ramais remotos à central no Rio de Janeiro;

- a falta de segurança das informações, pois os dados trafegam nos canais de forma analógica e sem criptografia; e

- a operacionalidade da rede está centralizada no PBX do Rio de Janeiro, em caso de problemas nesse PBX toda a rede fica inoperante.

De acordo com o Plano de Trabalho do Sub-departamento de Tecnologia da Informação do DECEA, a rede com o Asterisk substituirá a RTCaer no prazo de um ano. Os primeiros seis meses, as duas redes ficaram operando juntas, para a rede com o Asterisk ser homologada. O projeto de implementação terá os seguintes passos:

- instalação e configuração de cinco PBX IP com Asterisk, localizados nos grandes centros: Brasília, Curitiba, Rio de Janeiro, Recife e Manaus;

- aquisição e instalação de telefones IP em todas as unidades da Força Aérea Brasileira;

O protocolo utilizado será o SIP, o protocolo mais utilizado atualmente com grande compatibilidade com redes IP. A rede utilizada para a transmissão dos dados será a Rede Metropolitana Intraer (*Intranet* da Aeronáutica) já existente que interliga todas as redes locais das unidades.

Será utilizado o processamento distribuído, onde cada telefone IP se registrará no centro mais próximo, mas também será capaz de registrar em outro centro caso haja algum problema com seu centro principal. Os pacotes de voz

irão trafegar através de um VPN, onde serão criptografados para proteger os dados das comunicações.

Na Tabela 4 é feita uma comparação com a rede RTCaer e rede com o Asterisk para comprovar os inúmeros benefícios tragos com a implementação última, apesar de algumas desvantagens.

Tabela 4 – Comparação da RTCaer e VoIP com Asterisk

<i>RTCaer</i>	<i>Rede VoIP com Asterisk</i>
Obsoleta.	Última tecnologia em transmissão de voz.
Custo de contrato de <i>links</i> de voz junto às operadoras de telecomunicações	Uso dos links já existentes que compõe a rede metropolitana da FAB, a Intraer.
Falta de segurança das comunicações por ser transmitida em meio analógico sem criptografia.	Maior segurança já que a transmissão será feita em forma digital e criptografada.
Processamento centralizado.	Processamento distribuído
Transmissão perfeita da voz no meio analógico sem a necessidade de implementação de QoS	Necessária implementação e gerenciamento de QoS para a perfeita transmissão da voz na rede de dados.
Custo do telefone analógico de aproximadamente trinta reais.	Custo do telefone IP de aproximadamente quinhentos reais.

Com todas essas características a favor, apesar de algumas desvantagens a implementação do VoIP tornará realidade no Comando da Aeronáutica nos

próximos meses.

Até o fechamento deste Trabalho, o projeto encontra-se em fase de processo licitatório das placas Digium e dos telefones VoIP, somente aguardando descentralização de crédito por parte do Comando da Aeronáutica para a compra ser efetuada. Um laboratório foi montado no DECEA para efetuar testes de configuração e para treinamento dos técnicos que irão trabalhar com o Asterisk.

6 Considerações Finais

Desde a época primitiva a comunicação fascinou o homem, proporcionando-o a divulgação do conhecimento e o relacionamento entre as pessoas.

O avanço das comunicações, impulsionado pela invenção do telefone proporcionou ao homem conhecer o mundo todo, se comunicar com os quatro cantos do mundo sem precisar sair de dentro de casa.

Neste Trabalho apresentou-se o que há de mais moderno em comunicações que é a convergência de vários tipos de dados em apenas uma rede, mais especificamente da voz em uma rede IP.

Como toda nova tecnologia, o custo da implementação de Voz sobre IP é elevado, sendo acessível apenas a grandes incorporações, mas com a criação do *software* livre Asterisk, tema deste Trabalho, esse custo inicial baixou absurdamente, tornando o VoIP acessível a todos.

Referências Bibliográficas

SOUSA L., Redes de Computadores: Dados, Voz e Imagem, 1999.

MARTINS R. & CORREIA L., Redes de Computadores. Lavras: UFLA/FAEPE, 2002. (Curso de Pós Graduação “Lato Sensu” (Especialização) a Distância em Administração em Redes Linux).

ITU-T Rec. I350, *International Telecommunication Union Telecommunication Standardization Sector ITU-T, General aspects of quality of service and network performance in digital networks, including, ISDNs*, Março de 1993. URL: <http://www.itut.int/>.

ITU-T Rec. E800, *International Telecommunication Union Telecommunication Standardization Sector ITU-T, Terms and definitions related to quality of service and network performance including dependability*, Agosto de 1994. URL: <http://www.itut.int/>.

SOARES L.; LEMOS G. & COLCHER S., Redes de Computadores: Das LANs, MANs e WANs às Redes ATM, 1995.

CISCO Systems, *CISCO Voice over IP – Volume 1*, 2004. URL: <http://www.cisco.com/>.

CISCO Systems, *Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation*, Julho de 2005. URL: <http://www.cisco.com/>.

FERRARI A., Telecomunicações Evolução e Revolução, 2001.

DASPL - *Digital Audio Signal Processing Lab, Department of Electrical and Computer Engineering (EEN), University of Miami, Online Mean Opinion Score Test.* URL: <http://www.umsis.miami.edu/>.

ITU-T Rec. H323, *International Telecommunication Union Telecommunication Standardization Sector ITU-T, Packet Based Multimedia Communications Systems*, Novembro de 2000. URL: <http://www.itut.int/>.

LEOPOLDINO G. & MEDEIROS R. H.323: Um padrão para sistemas de comunicação multimídia baseado em pacotes, Dezembro 2001. URL: <http://www.rnp.br/>.

HANDLEY M. Et Al. *SIP – Session Initiation Protocol. Internet Engineering Task Force (IETF)*, Março 1999. (Request for Comments: 2543). URL: <http://www.ietf.org/>.

CROCKER D. *Standard for the format of ARPA Internet text messages. Internet Engineering Task Force (IETF)*, Agosto 1982. (Request for Comments: 822). URL: <http://www.ietf.org/>.

FIELDING R. Et Al. *Hypertext Transfer Protocol -- HTTP/1.1. Internet Engineering Task Force (IETF)*, Janeiro 1997. (Request for Comments: 2068). URL: <http://www.ietf.org/>.

ARANGO M. Et Al. *Media Gateway Control Protocol (MGCP). Internet Engineering Task Force (IETF)*, Janeiro 1997. (Request for Comments: 2705). URL: <http://www.ietf.org/>.

CUERVO F. Et Al. *Megaco Protocol Version 1.0 (MEGACO)*. *Internet Engineering Task Force (IETF)*, Novembro 2000. (Request for Comments: 3015). URL: <http://www.ietf.org/>.

ITU-T Rec. H248, *International Telecommunication Union Telecommunication Standardization Sector ITU-T, Gateway Control Protocol*, Março de 2004. URL: <http://www.itut.int/>.

SCHULZRINNE H. Et Al. *RTP: A Transport Protocol for Real-Time Applications*. *Internet Engineering Task Force (IETF)*, Janeiro 1996. (Request for Comments: 1889). URL: <http://www.ietf.org/>.

GONÇALVES, F. *RTP: Como construir e configurar um PABX com Software Livre*, 2005.

DIGIUM. *Quick Install Guide*. URL: <http://www.digium.com/>.

ALLISON M. & SPENCER M. *The Asterisk Handbook: An Operator's Guide to Configure the Asterisk Private Branch Exchange Server*, 2002. URL: <http://www.asterisk.org/>

Apêndice A

Exemplo do arquivo de configuração *zapata.conf*

```
[channels]
language=en

; Default context
context=inbound-from-pstn
switchtype=national
; Signalling method (default is fxs).  Valid values:
; fxs_ls:  FXS (Loop Start)
; fxs_gs:  FXS (Ground Start)
; fxs_ks:  FXS (Kewl Start)
; fxo_ls:  FXO (Loop Start)
; fxo_gs:  FXO (Ground Start)
; fxo_ks:  FXO (Kewl Start)
; fxs_rx:  Receive audio/COR on an FXS kewlstart interface
(FXO at the channel bank)
; fxs_tx:  Transmit audio/PTT on an FXS loopstart interface
(FXO at the channel bank)
; fxo_rx:  Receive audio/COR on an FXO loopstart interface
(FXS at the channel bank)
; fxo_tx:  Transmit audio/PTT on an FXO groundstart
interface (FXS at the channel bank)

signalling=fxs_ks
;
; A variety of timing parameters can be specified as well
; Including:
;   prewink:      Pre-wink time
;   preflash:     Pre-flash time
;   wink:         Wink time
;   flash:        Flash time
;   start:        Start time
;   rxwink:       Receiver wink time
;   rxflash:      Receiver flashtime
;   debounce:     Debounce timing
;
rxwink=300          ; Atlas seems to use long (250ms)
winks
;
usecallerid=yes
```

```

;
; Whether or not to hide outgoing caller ID
;
hidecallerid=no
;
; Whether or not to enable call waiting on FXO lines
;
callwaiting=no
;
; Whether or not restrict outgoing caller ID (will be sent
as ANI only, not available for the user)
; Mostly use with FXS ports
;
;restrictcid=no
;
; Whether or not use the caller ID presentation for the
outgoing call that the calling switch is sending
;
usecallingpres=yes
;
; Support Caller*ID on Call Waiting
;
callwaitingcallerid=yes
;
; Support three-way calling
;
threewaycalling=no
;
; Support flash-hook call transfer (requires three way
calling)
;
transfer=no
;
; Support call forward variable
;
cancallforward=no
;
; Whether or not to support Call Return (*69/1471,3)
;
callreturn=yes

;
; Enable echo cancellation
; Use either "yes", "no", or a power of two from 32 to 256
if you wish
; to actually set the number of taps of cancellation.
;

```



```

echocancel=yes
;
; Generally, it is not necessary (and in fact undesirable)
to echo cancel
; when the circuit path is entirely TDM.  You may, however,
reverse this
; behavior by enabling the echo cancel during pure TDM
bridging below.
;
echocancelwhenbridged=yes
;
; In some cases, the echo canceller doesn't train quickly
enough and there
; is echo at the beginning of the call.  Enabling echo
training will cause
; asterisk to briefly mute the channel, send an impulse,
and use the impulse
; response to pre-train the echo canceller so it can start
out with a much
; closer idea of the actual echo.
;
echotraining=yes
;
; You may also set the default receive and transmit gains
(in dB)
;
rxgain=0.0
txgain=0.0
;
; Logical groups can be assigned to allow outgoing
rollover.  Groups
; range from 0 to 31, and multiple groups can be specified.
;
group=1
;
; Ring groups (a.k.a. call groups) and pickup groups.  If a
phone is ringing
; and it is a member of a group which is one of your pickup
groups, then
; you can answer it by picking up and dialing *8#.  For
simple offices, just
; make these both the same
;
callgroup=9
pickupgroup=9
;

```

```
; Specify whether the channel should be answered
immediately or
; if the simple switch should provide dialtone, read
digits, etc.
;
immediate=yes
musiconhold=default

; On trunk interfaces (FXS) and E&M interfaces (E&M, Wink,
Feature Group D
; etc, it can be useful to perform busy detection either in
an effort to
; detect hangup or for detecting busies
;
busydetect=no
;
; On trunk interfaces (FXS) it can be useful to attempt to
follow the progress
; of a call through RINGING, BUSY, and ANSWERING. If
turned on, call
; progress attempts to determine answer, busy, and ringing
on phone lines.
; This feature is HIGHLY EXPERIMENTAL and can easily detect
false answers,
; so don't count on it being very accurate. Also, it is
ONLY configured for
; standard U.S. tones. This feature can also easily detect
false hangups.
; The symptoms of this is being disconnected in the middle
of a call for no
; reason.
;
callprogress=no
;

channel => 1
```

Exemplo do arquivo de configuração *extension.conf*

```
[general]
static=yes
writeprotect=no

[globals]
;Using this for outbound calls, atm only one FXO card, but
this will use any in group 1
OUTBOUND=Zap/1
; My extension
WELBY=SIP/6001
; robert's extension
ROBERT=SIP/6002
; ewan's extension
EWAN=SIP/6003
; our conumal Free World Dialup User ID
FWDUSERID=57052
;all extensions
EVERYONE=${WELBY}&${ROBERT}&${EWAN}

[default]
;default context
; here just incase i've forgot to edit anything
include => mainmenu

[unrecognised]
;not used at the moment

[international]
;context for international calls
;at the current price of international
;calls on telewest, they are instantly blocked

;block when someone dials 00
exten => _00.,1,Congestion
;block when someone tries to be sneaky and 141 the number
exten => _14100.,1,Congestion

[premium]
;this the generic premium rate number context
;i refuse to pay > 10p / min for a normal call
;so i block all 09** and
;089* numbers(no longer in use afaik)
;and of course silly 070* personal redirects
```

```

;at like 40p / min
;and of course all 118's are they are
;ludicrously inaccurate and expensive....
; and we have BT's EDQ if needed

exten => _09.,1,Congestion
exten => _118.,1,Congestion
exten => _089.,1,Congestion
exten => _070.,1,Congestion
;same as above, but for people who are 141'ing number
exten => _14109.,1,Congestion
exten => _141118.,1,Congestion
exten => _141089.,1,Congestion
exten => _141070.,1,Congestion

[freephone]
;we don't mind freephones so we allow all
;080*, 0500's
exten => _080.,1,Dial(${OUTBOUND}/${EXTEN},tT)
exten => _080.,2,Congestion
exten => _0500.,1,Dial(${OUTBOUND}/${EXTEN},tT)
exten => _0500.,2,Congestion
;same as above, but for people who are 141'ing number
exten => _141080.,1,Dial(${OUTBOUND}/${EXTEN},tT)
exten => _141080.,2,Congestion
exten => _1410500.,1,Dial(${OUTBOUND}/${EXTEN},tT)
exten => _1410500.,2,Congestion

[localcalls]

exten => _XXXXXXX,1,Dial(${OUTBOUND}/${EXTEN},tT)
exten => _XXXXXXX,2,Congestion
;same as above, but for people who are 141'ing number
exten => _141XXXXXXX,1,Dial(${OUTBOUND}/${EXTEN},tT)
exten => _141XXXXXXX,2,Congestion

[pstnservices]
;there are a few services on the phone line we still want
;1571 for instance as telewest STILL won't remove it fully
;150 to phone them to rant
;and 1471 which isnt really of anyuse now
exten => _91571,1,Dial(${OUTBOUND}/1571)
exten => _91571,2,Congestion
exten => _150,1,Dial(${OUTBOUND}/150,tT)
exten => _150,2,Congestion
exten => _91471,1,Dial(${OUTBOUND}/1471)

```

```

exten => _91471,2,Congestion

[outbound]
;this is so i only have to include one context
;for outgoing calls includes all the relevant contexts
include => pstnoutbound
include => emergency
include => pobk_iax
include => fwd-out
include => nexthopproject
include => iaxtel-out

[inbound-from-local]
exten => i,1,Playback(invalid)
exten => i,2,Hangup
include => outbound
include => internalextensions
include => voicemail
include => conference
include => sillystuff
include => onefoursevenone
include => parkedcalls

[inbound-from-pstn]
;lets bung everyone into the default context
include => default

[inbound-from-sip]
;lets bung everyone into the default context
include => default

[voicemail]
;voicemail, speaks for its self really
exten => 86,1,VoicemailMain
exten => 86,2,Hangup
exten => 1571,1,VoicemailMain
exten => 1571,2,Hangup

[conference]
;our converence rooms
exten => 6101,1,meetme,1
exten => 6102,1,meetme,2
exten => 6103,1,meetme,3
exten => 6104,1,meetme,4

```

```
exten => 6105,1,meetme,5
```

```
[mainmenu]
exten => s,1,Wait,1
exten => s,2,Wait,1
;answer it
;its for yooooooooooooou
exten => s,3,Answer
;play hello.gsm
exten => s,4,Background(hello)
;play end.gsm
exten => s,5,Background(end)
;play the short 890 (20sec)
exten => s,6,Background(890)
;play hello again
exten => s,7,Background(hello)
;and end again
exten => s,8,Background(end)
;play 5 minute 890
exten => s,9,background(890long)
;call them insanse for listening to it for 5minutes
exten => s,10,Background(realend)
;hangup
exten => s,11,Hangup
;press one for welby
exten => 1,1,Macro(stdexten,6001,${WELBY})
exten => welby,1,Goto(6001|1)
;press two for robert
exten => 2,1,Macro(stdexten,6002,${ROBERT})
exten => robert,1,Goto(6002|1)
;press three for ewan
exten => 3,1,Macro(stdexten,6003,${EWAN})
exten => ewan,1,Goto(6003|1)
;press four to anyone
exten => 4,1,Macro(stdexten,6004,${EVERYONE})
;i'm sorry you cant type on a phone
exten => i,1,Playback(invalid)
```

Exemplo do arquivo de configuração *sip.conf*

```
[general]
port = 5060
bindaddr = 0.0.0.0

;bindaddr = 192.168.0.5
externip = 81.41.218.85
localnet = 192.168.0.0
localmask = 255.255.255.0
context=inbound-from-sip
allow=ulaw
allow=alaw
allow=gsm
allow=all

musiconhold=default

; welby's cisco phone
[6001]
type=friend
context=inbound-from-local
secret=password
host=dynamic
canreinvite=no
callgroup=1
pickupgroup=1
mailbox=6001
canreinvite=no
username=6001

;robert ata port 1

[6002]
type=friend
secret=password
auth=md5
nat=no
host=dynamic
reinvite=no
canreinvite=no
qualify=1000
dtmfmode=rfc2833
disallow=all
allow=all
```

```
context=inbound-from-local  
mailbox=6002
```

```
;ewan ata port 2
```

```
[6003]  
type=friend  
secret=password  
auth=md5  
nat=no  
host=dynamic  
reinvite=no  
canreinvite=no  
qualify=1000  
dtmfmode=rfc2833  
disallow=all  
allow=all  
context=inbound-from-local  
mailbox=6003
```