



**UNIVERSIDADE DO EXTREMO SUL CATARINENSE**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**MARCIONÍ SERAFIM**

**UMA SOLUÇÃO DE VOZ SOBRE IP DE BAIXO CUSTO PARA AMBIENTES**  
**CORPORATIVOS**

**CRICIÚMA, JULHO DE 2006**

**MARCIONÍ SERAFIM**

**UMA SOLUÇÃO DE VOZ SOBRE IP DE BAIXO CUSTO PARA AMBIENTES  
CORPORATIVOS**

Trabalho de Conclusão de Curso para a Obtenção  
do Grau de Bacharel em Ciência da Computação  
da Universidade do Extremo Sul Catarinense.

Orientador: Prof. M. Sc. Rogério Casagrande

**CRICIÚMA, JULHO DE 2006**

Marcioní Serafim

**Uma Solução De Voz Sobre IP De Baixo Custo Para Ambientes Corporativos.**

Submetido ao corpo docente do Departamento de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

---

**Prof<sup>a</sup>. M.Sc. Ana Claudia Garcia Barbosa**

Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

---

**Prof. M.Sc. Rogério Antônio Casagrande (UNESC)**

Orientador

---

**Prof. Esp. Vilson Gruber (SATC)**

---

**Prof. M.Sc. Evânio Ramos Nicoleit (UNESC)**

Dedico esse trabalho a todas pessoas que,  
profissionalmente ou não, colaboraram  
no seu desenvolvimento.

## AGRADECIMENTOS

Aos meus pais Santo e Nelí, que me deram educação, honra, honestidade e outros vários valores herdados nos anos de convivência, pelo apoio e incentivo;

a minha esposa e filho pela compreensão por tantas vezes ausente física ou mentalmente;

a empresa que serviu como laboratório e tornou possível o uso do sistema em um ambiente real;

aos professores do curso, destacando-se meu orientador Rogério Antônio Casagrande, que deu a idéia do assunto desse trabalho e muito colaborou no seu desenvolvimento;

e principalmente a Deus por ter-me dado a oportunidade de ter uma vida tranquila.

"Se você deseja saber qual é a pessoa mais importante  
no movimento de software livre, olhe-se no espelho"

John "Mad Dog" Hall

## RESUMO

Muito tempo levou para que as tecnologias convergissem da comunicação falada em um modo de transmissão maduro e ao mesmo tempo não muito oneroso aos cofres e bolsos dos cidadãos, possibilitando o transporte da voz sobre as redes de dados, chamada de voz sobre IP (VOIP). No presente trabalho objetivou-se explorar uma solução de baixo custo de software para VOIP e para que isso fosse possível primeiramente fez-se estudo das tecnologias que a antecederam. O presente estudo demonstra todo esse amadurecimento bem como os protocolos próprios para voz sobre IP possibilitando o uso dessas redes para tráfego de voz. Como o objetivo dessa tecnologia é principalmente a redução de custos, estudar e conhecer uma tecnologia que transforme um simples micro computador em um PABX e que tenha baixo custo, torna-se uma solução atrativa. O Asterisk se encaixa perfeitamente nesse contexto, e foi testado em um ambiente de produção real, onde foi instalado e configurado, possibilitando a conclusão do estudo, que aponta os pontos favoráveis ou não dessa nova tecnologia.

**Palavras-Chave: Comunicação, Redes, VOIP, Sistemas Livres, Asterisk**

## ABSTRACT

It has taken a long time for new technologies dwell on spoken communication in a mature way, and at the same time not to be too expensive to citizens, giving them the possibility to the voice transportation over networks, the so-called IP (VOIP). In this study, the researcher aimed a low budget solution of a software to VOIP and to make it possible, it was firstly made a study of the previous technologies. The study demonstrates all the growing process, as well as the very voice protocols over IP that allow the usage of network to the voice flux. As the main aim of this technology was the cost reduction, to study and to know a technology that transforms a simple micro computer into a PABX, and one that has a low cost, the solution turns out to be attractive. The Asterisk fits perfectly in this context, and it was tested in a real production environment, where it has been installed and configured, permitting the conclusion of the study that indicates the pros and cons of this new technology.

**Key Words: Communication, Network, VOIP, Open Systems, Asterisk**

## LISTA DE ILUSTRAÇÕES

Figura 1. Modelo Genérico de um sistema de Comunicação.....	25
Figura 2. Arquitetura de rede OSI.....	30
Figura 3. O grafo de protocolos da Internet .....	31
Figura 4. O cabeçalho TCP .....	33
Figura 5. O formato do cabeçalho IP .....	34
Figura 6. O formato do cabeçalho UDP .....	35
Figura 7. Pilha de protocolos para aplicações de multimídia usando RTP .....	38
Figura 8. Formato do cabeçalho RTP.....	39
Figura 9. Arquitetura H.323 .....	42
Figura 10. Relacionamento dos protocolos no H.323 .....	44
Figura 11. Fases de uma chamada H.323.....	45
Figura 12. Estabelecimento de chamada SIP .....	48
Figura 13. Inicialização de sessão envolvendo proxies e entidades registradoras <i>SIP</i> ...	50
Figura 14. Comparação H.323 e SIP.....	52
Figura 15. Tipos de <i>vocoders</i> .....	58
Figura 16. Atraso em um sentido de milissegundos.....	58
Figura 17. Tipos de Retardos .....	59
Figura 18. Telefonia IP entre dois computadores.....	62
Figura 19. Telefonia IP entre dois telefones fixos.....	62
Figura 20. Telefonia IP Híbrida .....	63
Figura 21. Soyo G668 .....	64
Figura 22. Tenor A200 .....	65
Figura 23. Visão Geral Arquitetura Asterisk.....	67

Figura 24. Visão Geral da Arquitetura Proposta pela SiPfoundry .....	68
Figura 25. O Laboratório.....	75
Figura 26. Placa TE110P.....	78
Figura 27. Placa X100P.....	78
Figura 28. Asterisk interligando usuários x-lite .....	90
Figura 29. Asterisk interligando telefone fixo e x-lite .....	93
Figura 30. Asterisk interligando telefones fixos .....	94

## LISTA DE TABELAS

Tabela 1 – Definindo dispositivos dinâmicos .....	81
Tabela 2 - Compilando o kernel .....	82
Tabela 3 – Exemplo de nova entrada no Grub .....	82
Tabela 4 - Instalando o pacote zaptel .....	83
Tabela 5 - Instalando o pacote libpri .....	83
Tabela 6 - Instalando o pacote Asterisk .....	84
Tabela 7 - Comandos úteis no console do Asterisk: .....	87
Tabela 8 – Exemplo de cadastro usuário SIP .....	88
Tabela 9 – Exemplo de extensão .....	89
Tabela 10 – Exemplo de contexto reservado.....	90
Tabela 11 – Exemplo de Atendimento automático .....	92
Tabela 12 – Exemplo de cadastro usuário IAX.....	94
Tabela 13 – Configuração do musiconhold.conf.....	95
Tabela 14 – Instalação do mpg123.....	95
Tabela 15 – Exemplo de macro .....	97
Tabela 16 – Exemplo de extensão para gravação de mensagens .....	98
Tabela 17 – Custos Envolvidos .....	101
Tabela 18 – Exemplo de extensão para monitoramento de chamadas .....	108
Tabela 19 – Exemplo de extensão para recebimento ou envio de fax.....	110
Tabela 21 – Instalação da biblioteca spandsp.....	111
Tabela 21 – Re-compilando o Asterisk .....	111

## LISTA DE SIGLAS

ADCPM	Adaptive Defferencial PCM
HTTP	Hyper Text Transport Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU-T	Internation Telecom Union – Telecommunication
ISO	International Standards Organization
LAN	Local Area Networking
MCU	Multipoint Control Unit
OSI	Open Systems Interconnection
PCM	Pulse Code Modulation
PSTN	Public Switched Telephone Network
QoS	Quality Of Service
RAS	Registration, Admission and Status
RTP	Real-Time Transport Protocol
RTCP	Real-Time Transport Control Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VOIP	Voice Over Internet Protocol
WAN	Wide Area Networking

## SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVO GERAL .....	16
1.2 OBJETIVOS ESPECÍFICOS.....	16
1.3 JUSTIFICATIVA.....	17
1.4 ESTRUTURA DO TRABALHO.....	18
2 COMUNICAÇÃO DE DADOS.....	19
2.1 COMUNICAÇÃO DE DADOS NA ERA DA INFORMÁTICA.....	20
2.2 TELEPROCESSAMENTO .....	21
2.2.1 Evolução do Teleprocessamento .....	21
3 REDES DE COMPUTADORES .....	23
3.1 TOPOLOGIA DE REDE .....	23
3.2 COMO FUNCIONA A COMUNICAÇÃO DOS DADOS EM UMA REDE. ....	24
3.3 INTERNET .....	26
3.3.1 Intranet.....	26
3.3.2 Extranet.....	27
4 PROTOCOLOS E CAMADAS .....	28
4.1 SURGIMENTO DOS PROTOCOLOS .....	28
4.2 CARACTERÍSTICAS QUE UM PROTOCOLO DEVE APRESENTAR .....	29
4.3 O MODELO DE REFERÊNCIA OSI .....	29
4.4 ARQUITETURA TCP/IP .....	31
4.4.1 O Protocolo TCP.....	32
4.4.2 O Protocolo IP .....	33
4.4.3 O Protocolo UDP.....	34
5 MULTIMÍDIA NA REDE .....	36
5.1 PROTOCOLOS RTP E RTCP.....	37
5.2 O PADRÃO H.323 .....	41
5.2.1 Arquitetura H.323 .....	42
5.2.2 Protocolos Que Compõem o H.323 .....	43
5.3 O PADRÃO SIP.....	46
5.3.1 Mensagens SIP.....	49
5.3.2 H.323 x SIP.....	52
5.3.3 Protocolo SDP .....	53
5.4 QUALIDADE DE SERVIÇO.....	53
5.5 CONVERSÃO E COMPRESSÃO DE ÁUDIO – CODECS .....	55
5.5.1 Padrões de Digitalização e Compactação de Áudio. ....	57
5.6 RETARDO NA TELEFONIA IP .....	58
6 VOIP.....	60
6.1 AMBIENTES DE APLICAÇÕES VOIP.....	61
6.2 TECNOLOGIAS E SOLUÇÕES DE MERCADO .....	63
6.2.1 Soluções Gratuitas .....	65

6.3 ESTUDOS DE CASOS .....	68
7 TRABALHOS CORRELATOS.....	71
8 UMA SOLUÇÃO DE VOZ SOBRE IP DE BAIXO CUSTO PARA AMBIENTES CORPORATIVOS .....	73
8.1 LEVANTAMENTO BIBLIOGRÁFICO.....	75
8.2 IDENTIFICAÇÃO DAS TECNOLOGIAS NECESSÁRIAS PARA INSTALAÇÃO DO ASTERISK .....	76
8.2.1 Identificação da versão e distribuição Linux a ser usada.....	76
8.2.2 Identificação da configuração do computador necessária .....	77
8.2.3 Identificação de hardwares VOIP para uso no Asterisk. ....	77
8.3 INSTALAÇÃO DO SISTEMA OPERACIONAL.....	79
8.4 INSTALAÇÃO DO ASTERISK .....	80
8.4.1 Instalação dos pacotes extras do Linux Conectiva .....	80
8.4.2 Configuração do <i>udev</i> .....	81
8.4.3 O kernel .....	81
8.4.4 Instalação do pacote ZAPTEL.....	82
8.4.5 Instalação do pacote LIBPRI.....	83
8.4.6 Instalação do pacote ASTERISK.....	84
8.5 CONFIGURAÇÃO DO X-LITE .....	84
8.6 CONFIGURAÇÃO DO ASTERISK.....	85
8.6.1 Iniciar os módulos para placa de telefonia.....	85
8.6.2 Iniciar o Asterisk.....	86
8.6.3 Criação de usuários para uso do x-lite.....	88
8.6.4 Plano de discagem para usuários de <i>x-lite</i> .....	89
8.6.5 Plano de discagem para telefones fixos .....	91
8.6.6 Música de espera.....	94
8.6.7 Correio de voz (voicemail) .....	96
8.6.8 Sala de conferência .....	97
8.6.9 Macros .....	97
8.6.10 Gravação de mensagens de voz .....	98
8.7 RESULTADOS OBTIDOS .....	99
8.8 CUSTOS RELACIONADOS .....	101
CONCLUSÃO.....	102
REFERÊNCIAS .....	104
BIBLIOGRAFIA RECOMENDADA.....	107
APÊNDICE A – Escuta e monitoramento de ligações em andamento .....	108
APÊNDICE B – Recebimento e envio de Fax com redirecionamento para e-mail .....	110
APÊNDICE C – Exemplo dos Arquivos <i>zaptel.conf</i> e <i>zapata.conf</i> .....	112
APÊNDICE D –Exemplo de configuração arquivo <i>extensions.conf</i> laboratório A .....	113
APÊNDICE E – Exemplo de configuração Arquivo <i>iax.conf</i> laboratório A .....	117

APÊNDICE F – Exemplo do Arquivo <i>meetme.conf</i> laboratório A.....	118
APÊNDICE G – Exemplo configuração arquivo <i>musiconhold.conf</i> .....	119
APÊNDICE H – Exemplo de configuração do arquivo <i>sip.conf</i> laboratório A .....	120
APÊNDICE I – Exemplo de arquivo <i>voicemail.conf</i> laboratório A .....	123
APÊNDICE J – Exemplo de configuração arquivo <i>iax.conf</i> laboratório B .....	124
APÊNDICE K – Exemplo de configuração do arquivo <i>sip.conf</i> laboratório B .....	125
APÊNDICE L – Exemplo do arquivo <i>extensions.conf</i> laboratório B .....	126
APÊNDICE M – Dificuldades Encontradas .....	128

## 1 INTRODUÇÃO

Faz parte da essência humana ou de qualquer ser vivo a necessidade de comunicação. A facilidade de comunicar-se está relacionada diretamente com a capacidade intelectual do ser. O homem, é claro, tem se superado neste aspecto, apresentando diversas formas de comunicação com outros da mesma espécie. As comunicações oral, escrita ou por gestos são utilizadas diariamente pelo homem na busca de uma melhor interação. Por isso, a comunicação tornou-se um dos fatores mais importantes da sociedade humana e a todo o momento estabelece-se algum tipo de comunicação.

Nas últimas décadas, várias técnicas de transporte de informações foram criadas. Tradicionalmente desenvolve-se uma nova tecnologia de rede que suporte cada novo tipo de serviço criado. Pode-se citar por exemplo o serviço telefônico, as linhas Frame Relay e X.25 e outras redes de transporte de dados. Atualmente começa-se a ter uma nova visão nesse sentido, pois percebe-se a desvantagem de uma empresa adquirir e manter diferentes redes em um mesmo ambiente ao invés de ter uma única rede que forneça todos os serviços.

Devido à necessidade de comunicação entre pontos distintos, sendo que isso muitas vezes atribui um custo elevado nos modos mais convencionais e devido ao uso da tecnologia de redes de computadores já estar disponível na maioria das empresas e cada vez com melhor desempenho, há uma tendência mundial do uso de voz sobre estas redes IP. A integração dessas duas infra-estruturas distintas, a rede de telefonia pública e a rede de dados já é realidade. Mas para muitas empresas essa migração ainda não está muito clara, pois o cenário atual para uso de Voz sobre IP é muito complexo, existindo muitas tecnologias disponíveis e gerando incertezas.

Existem equipamentos para a integração dessas redes (como por exemplo centrais telefônicas híbridas) onde as empresas teriam que fazer um investimento muito alto. De um outro lado, para resolver esse problema, existem *softwares* que instalados em simples computadores poderiam substituí-los.

Como alternativa de *software* para este fim, existem alguns sistemas livres (e portanto gratuitos) que dispensam o uso de soluções proprietárias. A implantação de um sistema desse, representa o objeto final desta pesquisa.

De acordo com Bessa (2005), em 2004 foram gastos mundialmente US\$ 3 bilhões com *VOIP*, com projeção de US\$ 20 bilhões em 2009. Esses números ajudam a mostrar que vale a pena pesquisar e buscar conhecimentos nessa área.

## 1.1 OBJETIVO GERAL

Explorar uma solução de baixo custo de *software* destinado à Voz Sobre IP para ambientes corporativos.

## 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

- a) compreender a tecnologia Voz Sobre IP;
- b) compreender os protocolos H.323 e SIP;
- c) compreender os codificadores de áudio;
- d) pesquisar e avaliar soluções de baixo custo de Voz Sobre IP;
- e) verificar a viabilidade da utilização de Voz Sobre IP usando uma solução de baixo custo;

- f) implantar uma solução adequada ao ambiente corporativo;
- g) verificar as vantagens e desvantagens no uso desse sistema.

### 1.3 JUSTIFICATIVA

Apesar da importância que a telefonia convencional teve e tem para as empresas de um modo geral, a tecnologia *VOIP* vem chamando a atenção pela possibilidade de redução de custos já que os serviços do sistema atual são caros.

A tecnologia Voz Sobre IP usa padrões normatizados por empresas credenciadas para tal, o que facilita sua implantação. Além disso, traz a vantagem de ser tecnicamente compatível com as companhias telefônicas, tornando-se um sistema híbrido pois as redes IP e a rede pública de telefonia podem ser perfeitamente integradas, já que essas operadoras estão se adaptando à esta nova realidade.

Existem sistemas comerciais disponíveis para se aplicar tal forma de comunicação, como por exemplo, o “Interoffice Voice” da CrystalVoice, porém soluções Open Source e gratuitas para telefonia IP irão gerar muitos negócios no mundo inteiro (HALL, 2005).

Então, como o uso de *software* livre tem se tornado uma tendência, uma solução de baixo custo para o uso de Voz Sobre IP em ambientes corporativos seria muito adequada e vem ao encontro às necessidades das empresas, pois ajuda na redução de custo que é atribuído aos seus produtos finais.

O objeto do presente estudo poderá servir de auxílio às empresas que querem adotar uma solução *VOIP*.

## 1.4 ESTRUTURA DO TRABALHO

A seguir é apresentada uma síntese dos capítulos constantes nesse trabalho.

O primeiro capítulo demonstra uma visão geral do presente trabalho e mostra os objetivos a serem alcançados com ele.

O segundo capítulo relata como a comunicação evoluiu desde tempos mais remotos até a era da informática.

O terceiro capítulo demonstra como uma rede de dados funciona e como os dados por ali trafegam.

O quarto capítulo apresenta definições de camadas, protocolos, suas características e a arquitetura TCP/IP.

O quinto capítulo descreve detalhadamente quais e como funcionam os protocolos específicos para o transporte de multimídia na rede de dados.

O sexto capítulo apresenta particularidades da tecnologia de voz sobre IP e algumas soluções de mercado.

O sétimo capítulo apresenta alguns trabalhos correlatos a este.

O oitavo capítulo demonstra o trabalho desenvolvido desde a instalação até os resultados obtidos referentes ao uso de um software livre em um ambiente corporativo.

## 2 COMUNICAÇÃO DE DADOS

A comunicação de dados é um termo abrangente que significa o uso de técnicas que torna possível a transferência de informações de forma confiável entre pontos distantes (SILVEIRA, 1991).

O dia-a-dia das pessoas seria muito mais difícil sem o uso dos recursos que possibilitam a comunicação de dados. Eles estão presentes em vários aspectos da vida da humanidade. Faz-se isso ao ouvir um rádio, ver um programa de televisão, se conectar na Internet ou mesmo em um simples telefonema. Vive-se em uma era voltada para a comunicação de dados.

Os primeiros tipos de comunicação de dados, entretanto, não usavam aparelhos eletrônicos nem eletricidade, entre alguns que se pode citar estão os sinais de fumaça, usados pelos índios americanos e a reflexão da luz solar em espelhos (HELD, 1999).

Quando o uso da eletricidade se tornou possível, os sistemas de comunicação de dados evoluíram consideravelmente. Vieram então alguns precursores das tecnologias atuais (HELD, 1999).

- a) **o telégrafo:** criado por Samuel F.B. Morse no século XIX que, por meio de um fio, transmite corrente elétrica quando uma tecla era pressionada no ponto de origem até o ponto destino, sendo que cada letra era representada por uma determinada seqüência de sinais recebidos;
- b) **o aparelho de telefone:** o aparelho que transmite a voz, foi inventado pelo escocês naturalizado americano Alexander Graham Bell, mas vários outros inventores também desempenharam papéis importantes

nesse desenvolvimento, sendo que o primeiro aparelho foi construído em 1876. No início a ligação entre os telefones era ponto-a-ponto, ou seja, deveria haver um par de fios metálico exclusivo entre dois aparelhos e a comunicação poderia ser feita apenas entre esses dois;

- c) **o aparelho de telex:** foi o próximo passo e um grande avanço tecnológico que influenciou diretamente as comunicações de dados. Ele foi a espinha dorsal para as empresas quando se falava em comunicação. O telex é um sistema de comunicação à distância em que se conjuga o teletipo, uma central telefônica e um aparelho receptor que imprime automaticamente a mensagem transmitida sem necessidade de um operador para recepção.

## 2.1 COMUNICAÇÃO DE DADOS NA ERA DA INFORMÁTICA

Nos anos 50, com o surgimento dos primeiros computadores eletrônicos, teve-se a oportunidade de melhorar de forma muito mais acentuada as comunicações de dados. Os computadores tinham a capacidade de armazenar grande quantidade de informações e processá-las com muita rapidez.

À medida que o uso de computadores aumentava, surge a necessidade dos computadores estarem interligados uns com outros, onde o maior motivo para isso seria a necessidade de compartilhar dados e periféricos. Houve conseqüentemente o aparecimento do teleprocessamento.

## 2.2 TELEPROCESSAMENTO

Quando há situações em que computadores, ou outros dispositivos, usam recursos de telecomunicação uns com os outros, devido ao fato deles estarem afastados fisicamente, diz-se que há teleprocessamento (SILVEIRA, 1991).

As tecnologias que são usadas para obter o teleprocessamento foram evoluindo e se adaptando conforme as necessidades da troca de informação em sistemas computacionais.

### 2.2.1 Evolução do Teleprocessamento

A evolução do teleprocessamento se deu da seguinte forma (SILVEIRA, 1991):

- a) **terminal de aplicação:** cada programa era criado para um tipo de terminal específico, portanto cada programa criado só funcionava para um tipo de máquina e lidavam com problemas de comunicação separadamente e de forma diversa, sem padronização nenhuma, portanto havia falta de flexibilidade e gasto de processamento com problemas de comunicação;
- b) **terminal do computador:** nessa fase tinha-se Mainframes centralizados e uma rede em forma estrela com diversos terminais “burros” conectados a esse servidor. Não havia processamento no terminal, apenas nesse servidor, daí o adjetivo “burro”. Nesse sistema usavam-se arquiteturas proprietárias;

- c) **terminal de rede:** uso de redes locais. Nesse estágio, surgem padrões visando viabilizar a interoperabilidade entre diferentes sistemas;
- d) **terminal de inter-rede:** é a integração de diferentes redes por meio da utilização de gateways. É a própria Internet.

Os estágios “Terminal de aplicação” e “Terminal do Computador” satisfizeram o mercado durante certo período. Este último, em alguns casos raros, pode até ser encontrado atualmente em algumas empresas. Entretanto, as redes de computadores, como a rede *ethernet*, possuem várias vantagens sobre essas tecnologias.

### 3 REDES DE COMPUTADORES

Quando os computadores foram criados, logo se teve a necessidade de interligá-los de algum modo. Então as redes apareceram possibilitando a melhor utilização de informações e de hardware. Tornou-se possível o compartilhamento arquivos, impressoras, discos, acesso remoto a bancos de dados, backups remotos e troca, preservação e proteção de informações.

Na opinião de Guizzo (2002) uma rede de computadores é uma forma de disponibilizar informações e recursos com a interligação de vários computadores. Essa ligação física pode ser feita por cabos metálicos, fibras ou mesmo pelo ar em ondas de rádio e outros. Elas são chamadas de redes locais ou *Local Area Networks* (LAN).

Um trabalho de suma importância para as LANs direcionado para um grande número de computadores foi criado no final da década de 70 e início dos anos 80. Isso aconteceu quando o Centro de Controle de *Palo Alto da Xerox Corporation* (*Palo Alto Research Center – PARC*) desenvolveu um grupo de padrões e protocolos chamado *Ethernet*. A importância de tal criação foi muito grande a ponto desse padrão ser utilizado fortemente até os dias atuais. Outros padrões também conhecidos são ARCnet e Token-Ring, os quais não tiveram tanta difusão.

#### 3.1 TOPOLOGIA DE REDE

A topologia de uma rede de computadores detalha como é a forma do meio em que se dá o tráfego de informações e também como os dispositivos estão conectados fisicamente a ele. Refere-se a esse relacionamento físico e ao lógico de cada nó da rede (cada ponto de conexão com a rede pode ser chamado de nó, indiferente da função do

equipamento representado por ele), ou seja, a forma como estão dispostos (COMER, 1999).

Tem-se aqui então, uma divisão entre topologia lógica e topologia física (BORDIGNON, 2001):

- a) **topologia lógica:** descreve como será o fluxo das informações ao longo da rede, o formato dos dados, etc. É a forma como os protocolos (conjuntos de regras para que a comunicação funcione corretamente) operam no meio físico;
- b) **topologia física:** diz respeito à disposição do cabeamento e componentes de hardware do meio físico, descrevendo onde cada nó da rede está situado fisicamente em relação aos demais, como é feita a distribuição da mídia de conexão (cabeamento de cobre, fibra óptica, wireless, etc). São exemplos: a topologia em barramento, em estrela e em anel.

### 3.2 COMO FUNCIONA A COMUNICAÇÃO DOS DADOS EM UMA REDE.

Com a popularização das redes, tem-se hoje milhares de redes particulares interligadas entre si ou não. Nesse caso ficando isolada no próprio ambiente ao qual estão. As redes estão em empresas, bancos, escolas, universidades, hospitais, órgãos do governo ou em qualquer outro lugar em que se queira interligar dois ou mais computadores.

Na era dos computadores, tornou-se possível à transmissão de dados contendo vários tipos de informação pelo mesmo meio físico: textos, imagens, áudios,

vídeos entre outros. O processo dá-se com a transferência de bits (informações em estado binários) entre dois ou mais dispositivos por um meio físico. A comunicação entre tais dispositivos, no nível mais baixo (físico), nada mais é do que a codificação dos dados em uma forma de energia e o seu envio pelos meios citados.

Nessa transmissão de dados, a fonte, que pode ser analógica (a voz, por exemplo) é digitalizada, esses dados digitais são divididos em pacotes, transmitidos pelo meio disponível e remontados no destino. A estrutura física básica que torna possível essa comunicação é formada por três componentes: sistema de origem, o sistema de transmissão e o sistema de destino, conforme se pode ver na Figura 1. É claro que em um modelo de comunicação bidirecional, origem e destino podem ter ambas as funções.

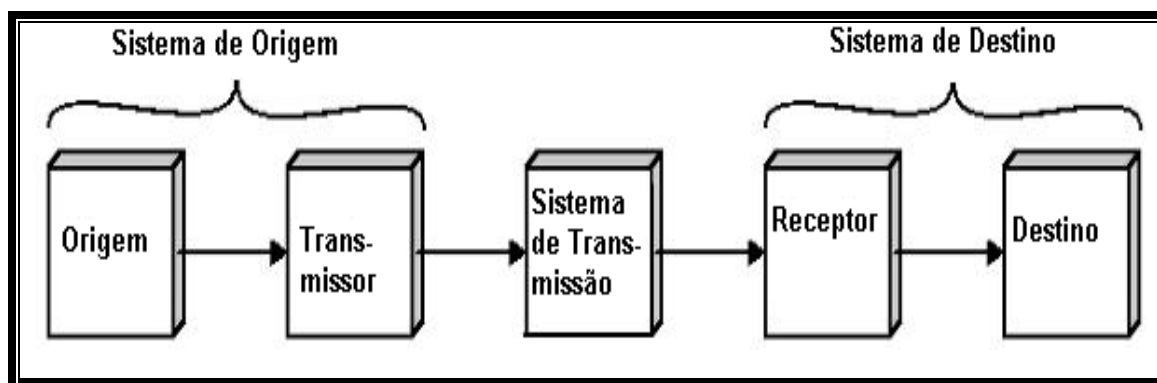


Figura 1. Modelo Genérico de um sistema de Comunicação.  
Fonte: HELD, G. (1999)

Apesar de parecer uma tarefa fácil, são muitas as responsabilidades de um sistema de comunicação, são elas: codificação da informação, gerenciamento da rede, encriptação dos dados, compressão dos dados, endereçamento, roteamento, sincronização, geração de sinal, formatação das mensagens, detecção e correção de erros, controle de fluxo e utilização do sistema de transmissão (HELD, 1999).

Todas essas questões foram tratadas por meio de regras padronizadas e separadas em camadas para menor complexidade. De modo geral, chamam-se essas regras de protocolos.

### 3.3 INTERNET

Segundo Guizzo (2002) Internet é um sistema de comunicação que interliga todos os continentes e é formado por várias redes de computadores interligadas, ou seja, é uma rede de redes.

Apesar de tanto questionamento sobre a segurança na Internet hoje em dia, ela surgiu exatamente pela busca de segurança nas informações militares americanas na década de 1960. Após passagem pelos ambientes acadêmicos, no qual ganhou consolidação, padronização e maturação técnica, chegou aos lares, escolas e empresas por volta do ano 1995 aqui no Brasil. Como protocolos de comunicação, atualmente a Internet utiliza a arquitetura TCP/IP como será visto no capítulo 4.

#### 3.3.1 Intranet

De acordo com Guizzo (2002, p. 38) “Intranets são redes corporativas que usam os mesmos programas e equipamentos usados na Internet (protocolo TCP/IP, servidores Web, browsers, etc.)”, ou seja, as intranets são redes restritas de empresas que aproveitam a infra-estrutura de comunicação de dados usada pela Internet para fixar comunicação interna ou com qualquer outra empresa conectada à Internet. Pode ser considerada também uma intranet, qualquer tipo de rede corporativa de acesso remoto, por exemplo, *LAN* e *WAN*, mesmo que elas não estejam fisicamente ligadas a Internet.

### **3.3.2 Extranet**

Já as extranets formam uma rede geograficamente distribuída (*WAN*). São geralmente utilizadas por empresas com o objetivo de interligar várias filiais e facilitar a comunicação com fornecedores. As extranets são construídas utilizando enlaces de comunicação proprietárias, porém oferecem serviços similares ao da Internet (GUIZZO, 2002).

Os protocolos são essenciais para o funcionamento das redes. As intranets e as extranets, assim como a Internet também usam os protocolos da arquitetura TCP/IP para estabelecerem comunicação.

## 4 PROTOCOLOS E CAMADAS

Para que dois dispositivos se comuniquem perfeitamente, eles precisam seguir algumas regras. De acordo com Derfler e Freed (1998) pode-se chamar um conjunto de regras que definem a interação entre o sistema de origem e o sistema de destino de protocolo. Eles são formados por uma cadeia de símbolos chamados de “conjunto de caracteres”, um conjunto de regras que definem a seqüência e o sincronismo na transmissão e por fim procedimentos que detectam a ocorrência de um erro na transmissão e informações sobre como corrigir um erro.

Para entender melhor porque tais regras são necessárias, pode-se fazer uma analogia entre esse tipo de comunicação com os sistemas de correios. Suponha que um emissor “A” queira enviar uma carta a um receptor “B”. O emissor escreve o texto em um papel, coloca-o em um envelope e neste discrimina o destino, ou seja, o endereço de “B”. Como este irá receber a carta se o endereço não for corretamente indicado? E como “B” poderá entender o conteúdo da carta se ele não estiver em uma linguagem conhecida?

### 4.1 SURGIMENTO DOS PROTOCOLOS

Nos anos 70 o Departamento de Defesa Americano, que possuía grandes e diferentes computadores que não podiam intercomunicar-se, liderou o desenvolvimento de um protocolo que tornaria essa intercomunicação entre essas diferentes arquiteturas possível. Nasceu então, entre outros, o Protocolo de Controle de Transmissão (TCP) e o Protocolo de Internet (IP), que com algum tempo, tornaram-se padrão de comunicação em redes do mundo inteiro.

Nessa mesma década, a IBM começou tornar público os padrões e protocolos que usava, acentuando a tendência para todos fabricantes fazerem o mesmo, assim grande parte das máquinas “falariam a mesma língua” sobre as redes ao qual estavam conectadas. Eventualmente, isso muito contribuiu para a interação e flexibilidade que se desfruta hoje.

#### 4.2 CARACTERÍSTICAS QUE UM PROTOCOLO DEVE APRESENTAR

As principais características que um protocolo deve apresentar são (TANENBAUM, 1997):

- a) deve atender às necessidades de comunicação;
- b) origem e destino devem “entender” o protocolo;
- c) deve responder no mesmo formato;
- d) componentes hierarquizados em camadas;
- e) isolar os detalhes entre as camadas superiores e inferiores;
- f) possibilidade de troca da implementação de uma camada sem a troca do conjunto.

Atendendo os requisitos acima, pode-se dizer que um protocolo desempenhará seu papel de padronização para que a comunicação entre duas entidades seja bem sucedida.

#### 4.3 O MODELO DE REFERÊNCIA OSI

O modelo de comunicação de dados é feito de forma hierárquica, dividido em camadas. O principal objetivo disso é isolar as camadas superiores dos detalhes dos

níveis inferiores e possibilitar a substituição da implementação de uma camada por outra sem interferir nas demais.

Inicialmente a *International Standards Organization* (ISO), em 1983 criou um modelo ideal para que a comunicação em uma rede fosse perfeita, chama-se Modelo *Open Systems Interconnection* (OSI). Ele é dividido em sete camadas hierarquicamente relacionadas: Aplicação, apresentação, sessão, transporte, rede, enlace e físico, sendo que cada uma tem uma função específica (PETERSON; DAVIE, 2004).

O modelo OSI não foi adotado por padrão devido à sua prática ser inviável, pois causaria acúmulo de cabeçalhos agregados às informações puras (que formariam um pacote a ser entregue na rede), sobrecarregando o sistema com um excesso de tráfego.

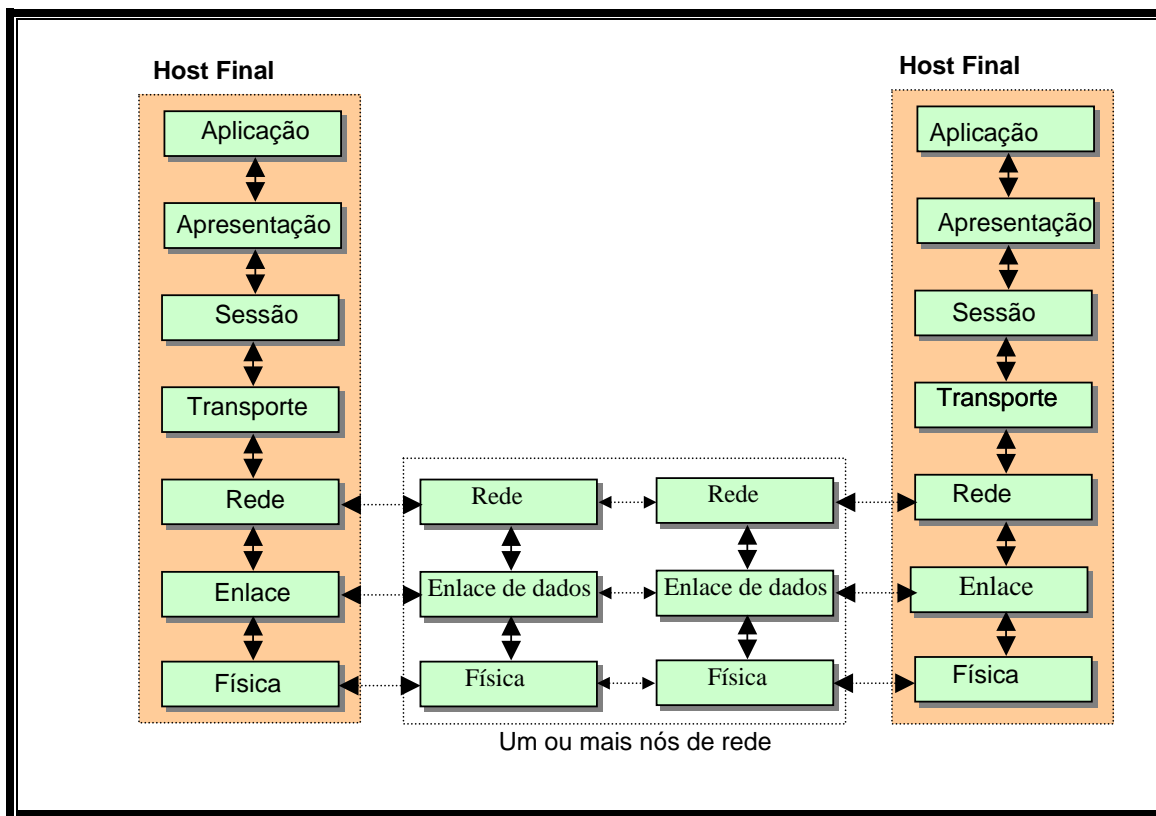


Figura 2. Arquitetura de rede OSI  
Fonte: PETERSON, L. ; DAVIE, B. (2004)

#### 4.4 ARQUITETURA TCP/IP

Já que o modelo OSI não pode ser adotado devido aos motivos indicados anteriormente, o modelo de comunicação de dados adotado para Internet foi a arquitetura TCP/IP, também conhecida como Arquitetura da Internet. É dividida em quatro camadas: aplicação, transporte, Internet e físico.

Na camada superior, a de aplicação, existem vários protocolos, como o FTP, TELNET e o HTTP que interoperam com diversos aplicativos de usuário final. Abaixo dela está a camada de transporte é composta pelos protocolos TCP e UDP ambos com os mesmos objetivos, o transporte é claro, porém com particularidades descritas nas seções a seguir. Na seguinte camada, chamada de camada Internet, o único protocolo que atua é o IP. É ele quem faz a interconexão entre várias tecnologias de redes diferentes em uma só inter-rede lógica. E a última e mais baixa camada é a física, ao qual entra a parte de hardware, placas de redes por exemplo, que interagem por meio de diversos protocolos, entre os quais pode-se citar o protocolo *Ethernet* (KUROSE; ROSS, 2005).

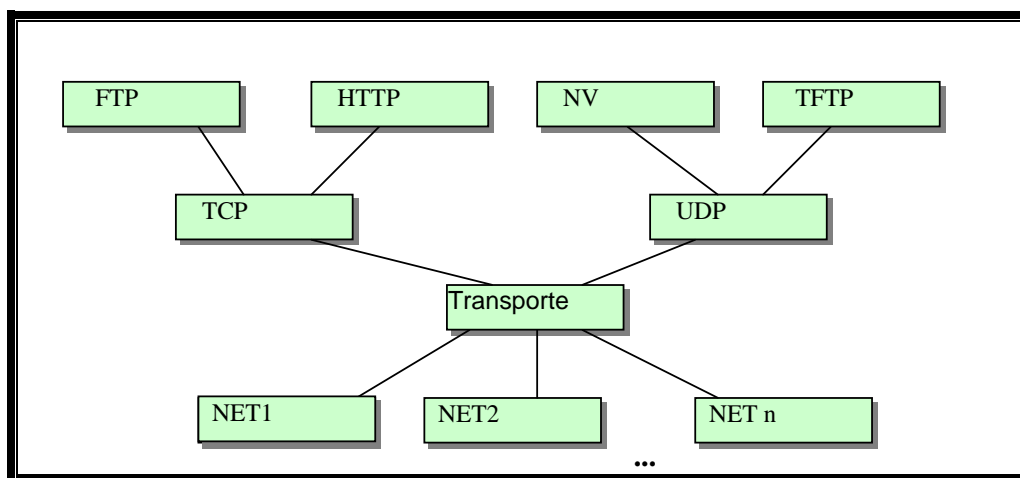


Figura 3. O grafo de protocolos da Internet  
Fonte: PETERSON, L.; DAVIE, B. (2004)

Os dois protocolos principais dessa arquitetura são o TCP e o IP, ao qual formam o seu nome.

#### 4.4.1 O Protocolo TCP

Como normalmente uma informação completa que se deseje transmitir, um arquivo por exemplo, não “caberia” de uma só vez em um único envio pela rede, mesmo porque isso iria congestionar a rede devido o meio físico estar sendo ocupado por uma só máquina durante esse tempo de transmissão, criou-se a idéia da divisão dos dados.

*Transmission Control Protocol*, o TCP, é o protocolo de transporte que divide as informações em “pacotes” (pequenas porções) que serão enviados pela rede por diferentes caminhos ao mesmo tempo para depois de chegarem ao destino serem remontados (MURHAMMER, 2000).

Quando o TCP cria um pacote, ele adiciona um checksum (soma de verificação) baseado nos valores de byte, isto é, uma quantidade precisa de dados no pacote. Quando o pacote chega ao destino, este checksum é verificado: se estiver diferente do valor indicado inicialmente, o TCP descarta o pacote e solicita sua retransmissão (MURHAMMER, 2000).

Pode-se, então, dizer que o TCP tem habilidade de checar os pacotes e determinar se todos foram recebidos. Ele é um protocolo da camada de transporte orientado a conexões que faz transferência de dados confiável fim-a-fim e realiza o serviço de entrega para o protocolo IP, porém uma das desvantagens nesse modo é um possível *overhead*, ou seja, excesso de tráfego na rede devido à muitas solicitações de

retransmissão de pacotes, o que pode ou não acontecer dependendo das condições da rede.

Ao receber o pacote da camada de aplicação, o TCP encapsula-o adicionando mais um cabeçalho com onze campos e envia-o para a camada seguinte, a de Internet, aonde o IP irá destiná-lo. Os campos do cabeçalho do TCP são indicados na Figura 4.

0	Porta de origem		16	Porta de destino		31
Número de seqüência						
Número de reconhecimento						
Hlen	Reservado	Bits de código		Windows		
Total de verificação				Ponteiro urgente		
Opções + Preenchimento						

Figura 4. O cabeçalho TCP  
Fonte: HELD, G. (1999)

#### 4.4.2 O Protocolo IP

Internet Protocol, o IP, segundo Held (1999) é o protocolo da camada de rede que entrega os pacotes no local destino desejado, roteando as mensagens entre as redes origem e destino.

Cada pacote é posto em “embalagens” IP (ou envelopes) separados. Essas embalagens contêm informações que detalham onde exatamente esses dados devem chegar (um endereço IP de 32 bits e uma máscara de rede também de 32 bits estão contidos nessas informações). Todas as embalagens para um certo grupo de dados têm a mesma informação de endereço para que todas possam ser mandadas para o mesmo local para remontagem (HELD, 1999).

Um endereço IP é formado por 4 grupos de 8 bits, no formato "nnn.nnn.nnn.nnn" onde cada "nnn" pode variar de 0 até 255, e é dividido em duas partes: uma que, com o auxílio da máscara, define qual rede ele pertence e outra que indica exatamente qual máquina, dentro daquela rede, deve receber o pacote.

Ao passar por essa camada, o pacote também recebe mais um cabeçalho, no qual os campos a serem preenchidos pelo protocolo, como os endereços IP de origem e destino, dessa seção serão conforme a Figura 5.

0	4	8	16	31
Vers	Hlen	Tipo de Serviço	Comprimento Total	
Identificação			Sinalizadores	Deslocamento de fragmento
Tempo de duração	Protocolo		Cabeçalho	Total de verificação
Endereço IP de origem				
Endereço IP de destino				
Opções + Preenchimento				

Figura 5. O formato do cabeçalho IP  
Fonte: HELD, G. (1999)

#### 4.4.3 O Protocolo UDP

Um outro protocolo da camada de transporte da arquitetura TCP/IP que faz a entrega para o protocolo IP é o *User Data Protocol* (UDP), porém esse não faz a confirmação da entrega do pacote. O UDP faz a entrega de mensagens independentes, designadas por datagramas, entre aplicações ou processos. Ele não proporciona um “serviço confiável” de entrega de dados, pois os datagramas podem ser perdidos ou entregues fora de ordem. Em alguns tipos de aplicações, como de multimídia, onde as

perdas de alguns pacotes que formarão um som ou vídeo pouco seriam notadas e a entrega imediata é mais importante do que a entrega precisa, esse tipo de protocolo é o mais indicado, principalmente por não causar um excesso de retransmissão congestionando a rede (KUROSE; ROSS, 2005).

Ao receber o pacote da camada de aplicação, o UDP encapsula-o adicionando mais um cabeçalho, conforme Figura 6, com quatro campos e envia-o para a camada seguinte, a de Internet, aonde o IP irá destiná-lo.

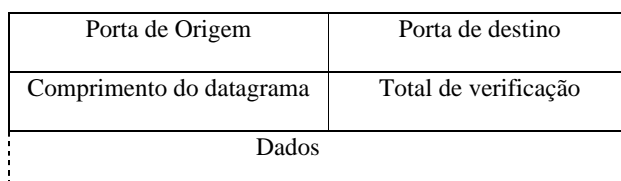


Figura 6. O formato do cabeçalho UDP  
Fonte: HELD, G. (1999)

Com o uso dos protocolos acima e de outros que serão citados adiante, o desenvolvimento de novas aplicações está cada vez mais voltadas para uso em rede, tanto para redes locais como as redes interligadas, a Internet.

Hoje essa “teia” gigante, ao qual é formada pela Internet, intranets e extranets, que interligam computadores no mundo todo, trouxe várias vantagens ao setor comercial, sendo que se destaca nesse trabalho a vantagem da possibilidade de transmitir voz por ela. Porém os protocolos vistos até aqui, sozinhos não atendem os requisitos exclusivos das aplicações multimídia, para suprir tal necessidade um pacote de novos protocolos foi projetado e será visto a seguir.

## 5 MULTIMÍDIA NA REDE

Desde a invenção do telefone em 1876, onde as ligações eram efetuadas somente por meio de uma telefonista, o mundo das telecomunicações tem evoluído continuamente. Pode-se citar, por exemplo, a vinda dos telefones móveis e os celulares que hoje tem participação ativa entre todos os níveis sociais da população.

Nos últimos anos, muitas formas de comunicação têm se convertido da tecnologia analógica para a digital. Exemplo recente é a câmera fotográfica digital, que hoje estão até nos celulares. Mas a conversão ao qual inspirou essa pesquisa foi a conversão da telefonia fixa para redes IP que foi impulsionada pelo grande crescimento da Internet conquistando a atenção do mercado e dos grandes fabricantes com promessas de redução de custo na operação, gerência, provisionamento, manutenção e tarifação.

As aplicações que são voltadas para efetuar o tráfego de voz, assim como vídeos, necessitam de protocolos com características que diferem do TCP e com maiores funcionalidades do que o UDP. A comunicação de multimídia entre terminais usando as redes IP necessita de dois processos simultâneos: a sinalização e controle de chamadas e o processamento da voz em si (ou vídeo). Portanto, pacotes de protocolos precisaram e foram criados para suprir tal necessidade da seguinte forma (KRISHNAMURTHY; REXFORD, 2001):

- a) **transporte de dados:** como a comunicação deve ser feita em tempo real, um fluxo de áudio (e vídeo) necessita ser remetido com propriedades de temporização, na Sessão 5.1 será visto o *RTP* que cuida disso;

- b) **estabelecimento de sessão:** regras para início de uma sessão deveriam ser criadas, como por exemplo: o receptor deve mostrar interesse na participação dessa sessão. Os protocolos SIP e H.323 foram criados para convidar um usuário à participar essa sessão (levantar o telefone de sua base após o mesmo tocar, pode representar claramente esse interesse);
- c) **descrição de sessão:** parâmetros de configuração da sessão precisam chegar aos participantes dela. O protocolo *SDP*, descrito na Sessão 5.3.3 “leva” esses dados (parâmetros de transporte e codificação dos fluxos, o nome o tempo e a finalidade da sessão);
- d) **descrição da apresentação:** deve ser oferecido um controle de como, onde e quando as sessões de multimídia são mostradas.

Para entendimento da funcionalidade desses protocolos criados em especial para multimídia, destaca-se inicialmente o RTP e seu companheiro *RTCP*, responsáveis pelo transporte desse fluxo.

## 5.1 PROTOCOLOS RTP E RTCP

Trabalhando comumente em cima do UDP, o *Real-time Transport Protocol* - RTP (Protocolo de transporte de tempo real) tem várias funcionalidades de um protocolo da camada de aplicação e a ela pertence (mesmo assim é chamado de protocolo de transporte), pois como já foi dito, foi criado para uso específico de aplicações multimídia e foi projetado para admitir grande quantidade dessas aplicações. O RTP é flexível ao que se refere à novas aplicações, pois possibilita que essas novas aplicações sejam desenvolvidas sem ter-se que revisá-lo (PETERSON; DAVIE, 2004) .

Na Figura 7, a aplicação e o RTP que estariam na mesma camada (camada de Aplicação), encaminhariam uma informação desejada para o transporte UDP, que passaria o pacote ao protocolo IP para endereçamento e o último nível faria o transporte físico dos dados. Ao chegar no destino, o caminho inverso seria feito até chegar na camada mais alta.

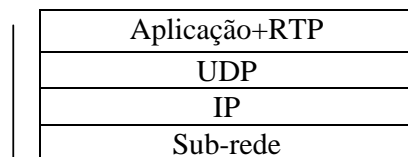


Figura 7. Pilha de protocolos para aplicações de multimídia usando RTP  
 Fonte: PETERSON, L.; DAVIE, S. (2004)

O Real-Time Transport Protocol não contém nenhum tipo de mecanismo para garantir a entrega dos pacotes, nem qualquer outro tipo de qualidade de serviço. O RTP também não evita que os pacotes cheguem fora de ordem nem prevê se a rede é confiável ou não. Porém, ele permite que sejam adicionadas técnicas de garantir isso quando necessário. Os *softwares* desenvolvidos para multimídia devem ser adaptativos para controlar a variação do fluxo de dados em tempo real (BORDIGNON, 2001).

Como um aplicativo entende melhor suas próprias necessidades, o RTP deixa alguns detalhes para serem tratados diretamente no nível de aplicação. Por exemplo, uma aplicação de vídeo MPEG, consegue recuperar de melhor forma os quadros perdidos reagindo de tal forma a fazer ajustes para que a consequência seja a menor possível. Para isso, o RTP define um perfil e um ou mais formatos para cada classe de aplicação (áudio, vídeo...), sendo que essa técnica facilita o entendimento de como os dados que vem após o cabeçalho devem ser tratados pela aplicação (PETERSON; DAVIE, 2004).

V=2	P	X	CC	M	TCU	Número de Seqüência
Timbre de Hora						
ID de Origem de Sincronismo (SSRC)						
IDs de origem de contribuinte (CSRC)						
Cabeçalho de extensão.						
Carga útil RTP.						

Figura 8. Formato do cabeçalho RTP  
 Fonte: PETERSON, L.; DAVIE, S. (2004)

O cabeçalho do RTP tem as seguinte funções (PETERSON; DAVIE, 2004):

- a) **V=2**: versão do RTP;
- b) **P**: preenchimento – indica a adição de bits no conteúdo. Normalmente utilizado por algoritmos de criptografia ou transmissões de pequenos conteúdos;
- c) **X**: indica a presença de um cabeçalho de extensão definido por alguma aplicação específica. Raramente é usado, pois essa informação normalmente consta no campo TCU;
- d) **CC**: contador de quantidade para IDs de origem contribuinte que será vista á seguir;
- e) **M**: bit marcador de indicação de quadros (limites de um quadro – determinado pelo perfil da aplicação);
- f) **TCU**: tipo de carga útil. Determinado pelo perfil da aplicação, indica que tipo de dado de multimídia o pacote contém;
- g) **número de seqüência**: permite que o receptor processe os pacotes recebidos em ordem correta, ajudando nas estatísticas para que este

informe ao emissor pedido de ajuste de taxa de transmissão caso haja perda de pacote. Os pacotes perdidos não são reenviados;

- h) **timbre de hora:** o receptor necessita de informações de temporização para organizar a reprodução do fluxo. O intervalo dos quadros pode ser fixado a partir do cabeçalho RTP mesmo que receptor e emissor não tenham seus relógios sincronizados. Isso permite que o receptor reproduza a mídia com o intervalo conveniente entre as amostras ou quadros seqüenciais;
- i) **ID de origem de sincronismo (SSRC):** número de 32 bits que reconhece de forma única a origem de um fluxo RTP. Utilizado, por exemplo, em teleconferência onde as origens dos pacotes são diversas;
- j) **IDs de origem contribuinte (CSRC):** esse campo é usado quando uma série de fluxos de diferentes origens (exemplo: teleconferência) é misturados por um equipamento de redução de requisitos de largura de banda, o mixer. É ele que aparece como origem de sincronismo (campo anterior) mas as origens reais aparecem na lista dos contribuintes;
- k) **cabeçalho de extensão:** usado apenas se o campo “x” for marcado.

O RTP tem o auxílio de um outro protocolo para controle de fluxo, o *Real-Time Transport Control Protocol* (RTCP), que em português significa protocolo de controle de transporte de tempo real. Enquanto o RTP transporta os dados propriamente ditos entre um caminho origem/destino, o RTCP envia de tempos em tempos, informações de controle associadas à esses dados (BORDIGNON, 2001).

As funções de controle que o RTCP realiza são três (KRISHNAMURTHY; REXFORD, 2001):

- a) **feedback:** ajuda a identificar deficiências na recepção dos pacotes. Nesse caso o *RTP* pode detectar problemas adaptando a taxa de transmissão de acordo com o necessário;
- b) **identificação:** agrega todos os identificadores de origem dos pacotes *RTP* enviados sob um único nome, possibilitando que o receptor associe um grupo de fluxos a um único emissor;
- c) **sincronismo:** cria uma associação de tempo (a hora específica do relógio da *BIOS* da máquina, por exemplo) às marcas de hora nos pacotes *RTP*, permitindo que o receptor sincronize a reprodução de um grupo de fluxos de dados, sabendo assim que eles pertencem à mesma sessão.

Durante uma conversação a entrega do fluxo de voz é coordenada pelos protocolos *RTP* e *RTCP*, que foram criados especificamente para a comunicação multimídia sobre redes *IP*. Porém, o *RTP* sozinho não atenderia os padrões da Telefonia *IP*, pois itens como sinalização de uma chamada iniciada não são tratados por ele. Para esse controle, os padrões *H.323* e *SIP* foram criados.

## 5.2 O PADRÃO H.323

Definido em 1996 pela *ITU-T*, pertencente a *International Telecommunications Union*, órgão mundial que define padrões técnicos para todas as áreas de comunicação, o padrão *H.323* garante a interoperabilidade entre várias tecnologias necessárias para conversão da telefonia fixa para as redes *IP* e também videoconferência (KUROSE; ROSS, 2005).

Ele contém um conjunto de protocolos verticalizados para sinalização e controle da comunicação realizada entre equipamentos que suportam aplicações de áudio, dados de multimídia e vídeo.

O padrão H.323 usa esse grupo de protocolos para tornar possível desde um telefonema sobre redes IP até uma vídeo conferência com uma grande quantidade de participantes, já que protocolos padrões devem ser adotados em todos os pontos da comunicação para resolver questões como qual codificação de áudio que se está usando e qual porta do protocolo UDP esta sendo usada.

### 5.2.1 Arquitetura H.323

A Figura 9 mostra uma rede típica usando o padrão H.323, no qual se pode imaginar como fossem dois escritórios distantes, chamados individualmente como “Zona”, mas ligados por uma rede IP onde tráfego de voz poderá passar.

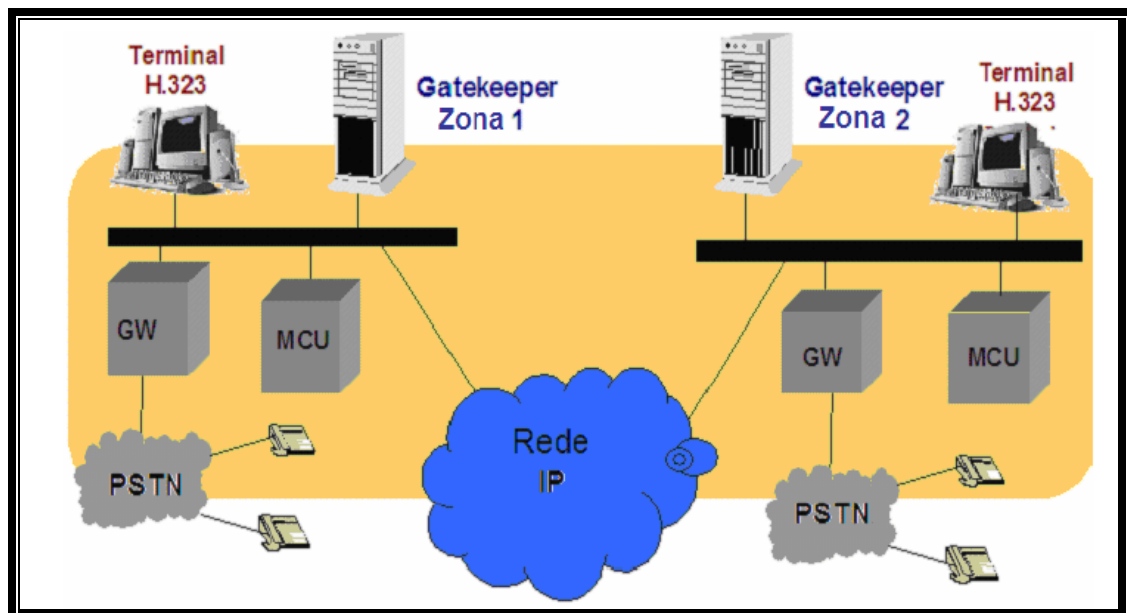


Figura 9. Arquitetura H.323  
Fonte: MAGRO, J. (2005)

Cada elemento nessa rede tem as seguintes funcionalidades (MAGRO, 2005):

- a) **Gatekeeper:** realiza funções como: tradução entre os vários formatos de endereços existentes em ligações telefônicas, controle de várias chamadas no mesmo instante, mantém o registro dos clientes e acha um cliente registrado em outro gateway. Cada zona tem apenas um *Gatekeeper* fazendo o seu gerenciamento;
- b) **Terminal H.323:** poder ser uma estação de trabalho ou um telefone IP que possui porta *ethernet* para conexão direto com a porta de um *hub* ou *switch*. Esses terminais podem estabelecer comunicação com outro terminal diretamente ou por meio de um *Gatekeeper*;
- c) **Gateway (GW):** componente da rede que provê comunicação em tempo real entre os terminais da rede IP e terminais da rede de telefonia pública (*PSTN*);
- d) **MCU:** são as unidades de controle multiponto que provêm controle em uma conferência em vários pontos.

### 5.2.2 Protocolos Que Compõem o H.323

São os seguintes protocolos que compõem o padrão H.323(MAGRO, 2005):

- a) **Registration, Admission and Status (RAS):** descobre a localização de um *Gatekeeper* para que um terminal ou Gateway possa se registrar. Ainda solicita permissão para uma chamada, alocação de banda e a desconexão. É usado somente quando há um *Gatekeeper* presente;

- b) **Q.931:** sinaliza o estabelecimento e término de uma chamada entre dois terminais;
- c) **H.245:** controla as conexões dando a liberdade para que dois terminais negociem a capacidade de processamento da rede;
- d) **H.225:** estabelece conexão entre os pontos terminais (fim a fim);
- e) **Real Time Transport Protocol (RTP):** transporta os pacotes de voz no padrão H.323;
- f) **H.332:** utilizado em grandes conferências;
- g) **H.450:** utilizado em serviços suplementares.

A Figura 10 mostra a relação entre protocolos do H.323. Note-se que alguns estão sobre o TCP, enquanto outros estão sobre o UDP. G7XX e H.26X são *codecs* usados na codificação de áudio e vídeos respectivamente.

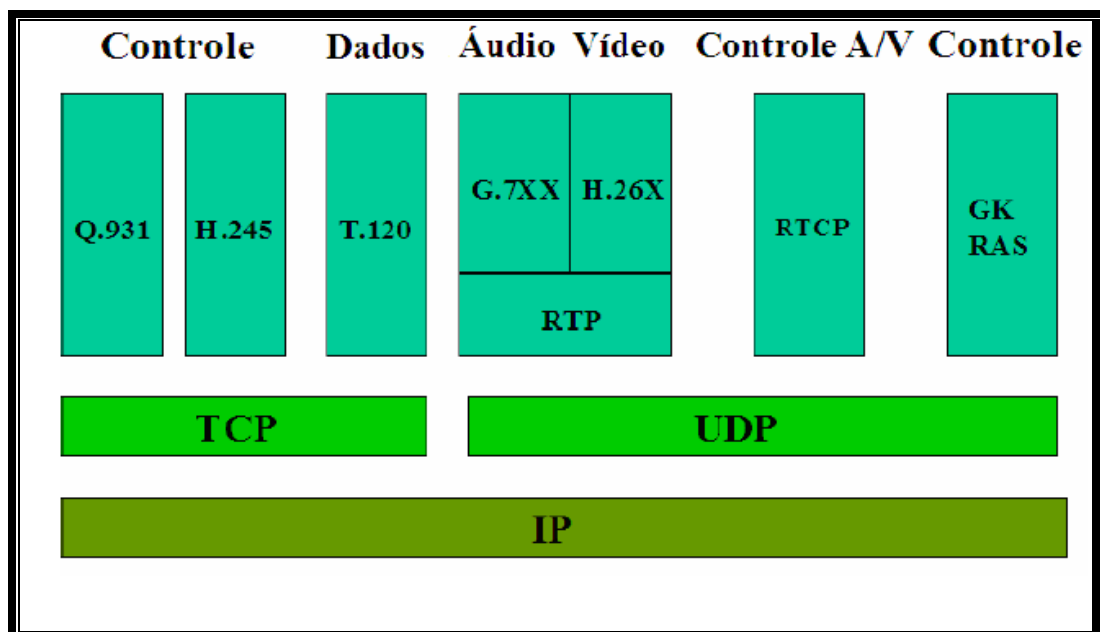


Figura 10. Relacionamento dos protocolos no H.323  
 Fonte: BORDIGNON, M. (2001)

A Figura 11 mostra, de forma genérica, como uma chamada é realizada no padrão H.323 quando há a presença de um Gatekeeper. Se ele não estiver presente, as fases 1 e 7 não existirão.

FASES		PROTOCOLO	FUNÇÃO
1	Permissão para a Chamada	RASA	Solicita permissão ao GK para fazer/receber chamadas. Ao final desta fase o elemento chamador recebe o endereço Q.931 (end. IP) do elemento chamado
2	Estabelecimento da Chamada	Q.931	Estabelece a chamada entre dois elementos, o elemento chamador recebe o endereço H.245 do elemento chamado.
3	Negociação Das Capacidades dos elementos e estabelecimentos de canal lógico.	H.245	Negocia as facilidades dos elementos. Determina o relacionamento mestre/escravo. Abre os canais lógicos entre os elementos. Ao final desta fase os elementos conhecem os endereços RTP/RTCP um do outro.
4	Conversação	RTP	As duas partes conversam.
5	Fechamento dos canais	H.245	Fechamento dos canais lógicos.
6	Desligamento da chamada	Q.931	Desligamento da chamada
7	Liberação da chamada	RAS	Libera os recursos utilizados na chamada.

Figura 11. Fases de uma chamada H.323  
Fonte: MAGRO, J. (2005)

O padrão H.323 foi o ponto de partida inicial para o padrão *VOIP*. Quando sua primeira versão foi criada, incluía algoritmos para transmissão, compressão dos dados além de protocolos como o RTP. Suas versões posteriores tornaram-se melhores e mais eficientes, porém novos padrões para telefonia IP foram criados para assegurar melhor qualidade no processamento de uma comunicação desse tipo. Entre esses padrões estão o MGCP, MEGACO e o SIP. Este último, de acordo com Magro (2005), é o que vem sendo mais usado devido sua menor complexidade.

### 5.3 O PADRÃO SIP

Apesar de ter os mesmos objetivos do H.323, o protocolo SIP da camada de aplicação, que foi criado pela IETF - *Internet Engineering Task Force* em 1996, destaca-se por ser mais flexível, extensível e de fácil implementação do que o seu concorrente (MAGRO, 2005).

O padrão SIP dá assistência a aplicações como aquelas voltadas à telefonia IP, em que o receptor recebe um convite para participar de uma sessão de comunicação. Nesse exemplo, o emissor (aquele que deseja iniciar uma chamada) pode usar o SIP para localizar uma ou mais “pessoas” e convidá-las a se juntarem à uma sessão.

O SIP é baseado em texto e no modelo cliente-servidor (solicitação-resposta) que proporciona o estabelecimento de sessões fim-a-fim entre duas ou mais entidades. Por tais características possui alguma semelhança com o protocolo HTTP além do fato de usar na sua sintaxe a mesma codificação e incorporar endereços *Uniform Resource Locators (URL)*. O SIP, porém, foi projetado para aplicações diferentes do HTTP e possui as seguintes capacidades (PETERSON; DAVIE, 2004):

- a) alcançar o local correto onde se encontra o usuário desejado independentemente do local e equipamento em que este esteja usando no momento;
- b) verificar se o usuário chamado pode e deseja participar de uma conversação;
- c) determinar certas tecnologias escolhidas como a codificação de voz escolhida (*codec*) e escolha de mídia (telefone IP, micro entre outros);
- d) estabelecer parâmetros configurados para a sessão, como por exemplo, números de portas usadas entre os elementos da conversação;

- e) gerenciar várias funções referentes à sessão, como encaminhamento de chamadas e modificações de parâmetros da sessão em andamento.

Apesar de serem relativamente fáceis, a primeira capacidade apontada é mais complexa que as outras devido ao fato do usuário a ser chamado poder estar em sua mesa usando um telefone IP ou levar seu desktop portátil em uma viagem e nele utilizar um softphone (*software* que substitui o Telefone IP), por exemplo. Uma das vantagens do SIP em relação ao H.323 é que ele apresenta a possibilidade de mobilidade por meio de servidores proxy e de redirecionamento que encaminham as requisições para a localidade atual do usuário (PETERSON; DAVIE, 2004).

Apesar disso, nenhum dos dois padrões: SIP e H.323, contemplam a capacidade de handoff<sup>1</sup> durante uma ligação, ou seja, uma comunicação se perderia ao o usuário mudar de sub-rede (suponha que o mesmo usuário acima, com seu notebook, esteja se deslocando durante uma conversaçãõ).

Os autores Kurose e Ross (2005) descrevem o seguinte exemplo prático de conversaçãõ, sendo que a comunicaçãõ se dá pelos usuários Alice (que conhece o endereço IP de Bob) e Bob. Ambos usuários de uma aplicaçãõ que utiliza o protocolo SIP:

---

<sup>1</sup> Handoff: nome dado a tecnologia que torna capaz um aparelho celular poder trocar de zona durante uma conversaçãõ.

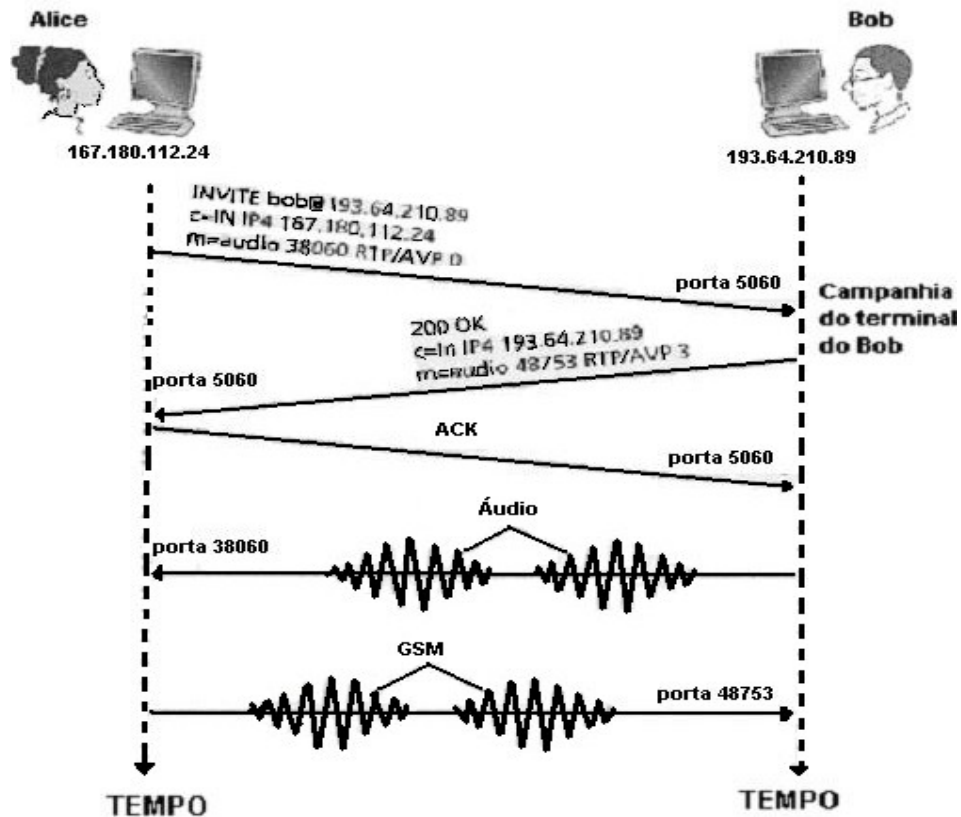


Figura 12. Estabelecimento de chamada SIP  
 Fonte: KUROSE, J; ROSS, K (2005)

- Alice manda, via UDP e porta 5060 (porta do SIP), uma requisição (*invite*) ao Bob. Essa mensagem contém um identificador de destino bob@193.64.210.89, IP atual de Alice, indicação que Alice deseja receber áudio (no formato desejado e encapsulado em RTP) e indicação que as respostas de Bob devem vir pelo RTP porta 38060;
- Bob responde com uma mensagem SIP pela porta 5060. A resposta contém um 200 OK (apto a estabelecer sessão), indicação do seu endereço IP, o código de empacotamento que deseja para recepção e seu número de porta que os pacotes de áudios devem ser enviados (48753);

- c) Alice envia uma mensagem SIP de reconhecimento (*ACK*) e após isso a comunicação é iniciada. Por definição das aplicações em uso, cada um vai codificar o áudio por meio de um *codec* diferente, entretanto reconhecido pelo *software* alheio.

Vale a pena ressaltar que Alice poderia não conhecer o IP do seu destinatário, pois Bob pode ter acesso *VOIP* de vários dispositivos (em casa, no trabalho) além de poder ter adquirido seu IP dinamicamente.

### 5.3.1 Mensagens SIP

Entre os vários tipos de mensagens *SIP* existentes, será exemplificado a mensagem *SIP INVITE*, usando o mesmo exemplo anterior, porém simulando que Alice conheça apenas o endereço SIP de Bob (*bob@domain.com*) e não mais o seu IP como anteriormente, a mensagem seria algo como (KUROSE; ROSS, 2005):

*INVITE sip:bob@domain.com SIP/2.0*

*Via: SIP/2.0/UDP 167.180.112.24*

*From: sip:Alice@hereway.com*

*To: sip:bob@domain.com*

*Call-ID: ale3a@pigeon.hereway.com*

*Content-Type: application/sdp*

*C=IN IP4 167.180.112.24*

*M=audio 38060 RTP/AVP 0*

A primeira linha contém o endereço *SIP* e sua versão. Sempre que uma mensagem *SIP* passa por um dispositivo *SIP* (inclusive o dispositivo que originou a mensagem), um cabeçalho *Via* (segunda linha da mensagem) é anexado contendo o

endereço do dispositivo. Linhas *From*, *To* e *Call-ID* também são adicionadas semelhantes á uma mensagem de e-mail. As próximas duas linhas contém respectivamente o tipo de conteúdo usado na mensagem e o seu tamanho em bytes. As duas últimas linhas definem o endereço de IP de Alice e como ela quer receber o áudio (KUROSE; ROSS, 2005).

Nesse último exemplo, como o emissor não conhece o endereço de *IP* do destinatário, um *Proxy* que o determina com o auxílio de uma entidade registradora *SIP* que todo usuário tem associado.

Segundo KUROSE e ROSS (2005), o IP de um destinatário pode ser determinado da seguinte forma:

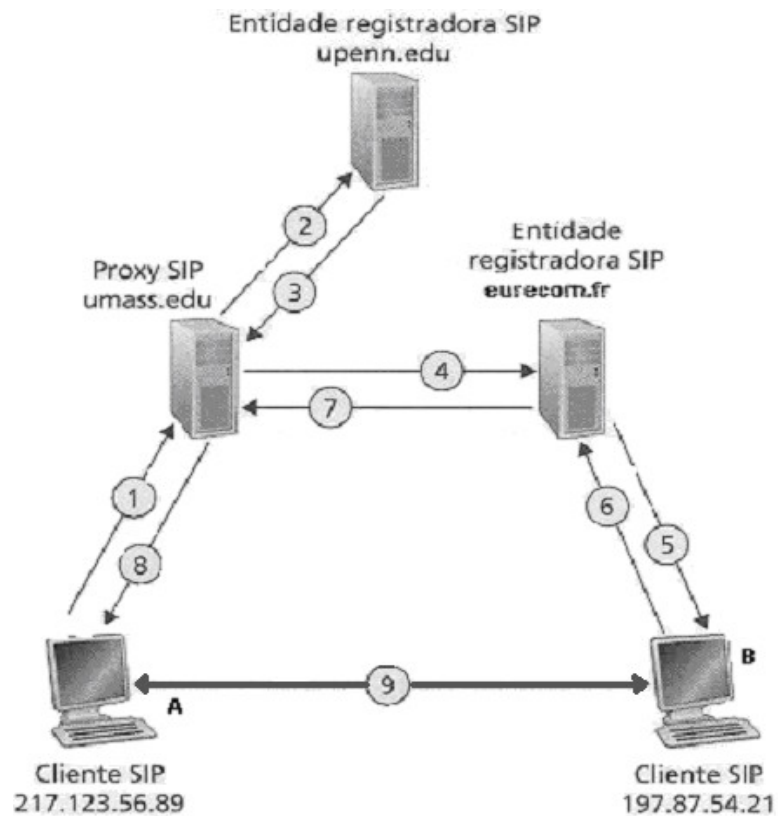


Figura 13. Inicialização de sessão envolvendo proxies e entidades registradoras *SIP*  
 Fonte: KUROSE, J; ROSS, K (2005)

- a) **rota 1:** “A” envia uma mensagem *INVITE* ao proxy *SIP* de “umass”;

- b) **rota 2:** o proxy realiza uma consulta DNS para a entidade registradora *SIP* de upenn.Edu;
- c) **rota 3:** como “B” não está mais inscrito na entidade registradora de upenn, esta envia uma resposta de redirecionamento, indicando que é preciso tentar eurecom.fr;
- d) **rota 4:** o proxy de umass envia uma mensagem *INVITE* à registradora *SIP* de eurecom;
- e) **rota 5:** essa registradora conhece o endereço de *IP* de “B” e repassa a mensagem *INVITE* ao destino 197.87.54.21 que está executando o cliente *SIP* de “B”.

As rotas 6,7 e 8 são o retorno ao emissor com informação de que o destino (*IP*) foi encontrado, então “A” já pode falar com “B” diretamente.

De acordo com Ribeiro, Aguiar e Marcondes (2001), além do *INVITE* existem as seguintes mensagens *SIP*:

- a) **ACK:** sinalização recebida como fechamento a um *INVITE*;
- b) **OPTIONS:** questiona quais métodos e extensões que são suportados pelo servidor e pelo usuário descrito no campo de cabeçalho (To:);
- c) **BYE:** usado para liberar os recursos associados a uma ligação e forçar a desconexão da mesma;
- d) **CANCEL:** cancela uma requisição que ainda esteja pendente (em andamento). Para que uma requisição seja considerada pendente, ela não deve receber uma resposta final;
- e) **REGISTER:** mensagem que faz o registro dos clientes de uma zona em uma registradora *SIP*.

### 5.3.2 H.323 x SIP

Apesar dos Protocolos H.323 e SIP resolverem praticamente os mesmos problemas, as seguintes diferenças destacam-se entre eles (KUROSE; ROSS, 2005):

- a) o H.323 é formado por vários protocolos que, integrados verticalmente, resolvem as questões de sinalização, registro, controle de admissão, transporte e *codecs* em uma conferência multimídia;
- b) o SIP trata apenas da inicialização e gerenciamento de sessão. Ele é um elemento separado e pode ser combinado com outros protocolos e serviços;
- c) o H.323 é original da ITU (órgão ligado à telefonia) enquanto o SIP vem da IETF e agrega conceitos da Web, DNS e do e-mail da Internet;
- d) O H.323 é grande e complexo.

De acordo com Magro (2005), o H.323 leva vantagem de possuir muito investimento ao seu favor e poder coexistir junto com o SIP, entretanto ele faz a seguinte comparação entre os dois:

H.323	SIP
Protocolo complexo	Comparativamente Simples
Representação binária para as mensagens	Representação Textual
Requer total compatibilidade com versões anteriores	Não requer total compatibilidade com versões anteriores
Pouco modular	Muito modular
Pouco escalável	Altamente escalável
Sinalização complexa	Sinalização simples
Primeiro a ser lançado no mercado	Orientado pelo IETF
Centenas de elementos	Somente 37 cabeçalhos
A detecção de loop é difícil	A detecção de loop é comparativamente fácil

Figura 14. Comparação H.323 e SIP  
Fonte: MAGRO, J. (2005)

O protocolo SIP necessita do RTP e também do TCP ou UDP para transporte, porém a sintaxe das mensagens independe de qual deles foi usado. Outro “auxiliar” do SIP é o protocolo SDP.

### 5.3.3 Protocolo SDP

Protocolo necessário para cuidar da negociação e identificação das sessões em multimídia, o SDP – *Session Description Protocol*, trabalha em conjunto com o SIP transportando as seguintes informações (PETERSON; DAVIE, 2004):

- a) o nome do emitente e finalidade da sessão;
- b) horário inicial e final para a sessão;
- c) espécie de mídia a ser utilizada (como áudio e vídeo);
- d) configurações estabelecidas para a presente sessão: protocolo de transporte a ser usado, números de porta e tecnologias de codificação.

## 5.4 QUALIDADE DE SERVIÇO

A migração da voz da rede pública de telefonia para a rede de computadores agrega muita complexidade. Um dos problemas a serem analisados e resolvidos é a qualidade de serviço, ao qual conhece-se como QoS. Ela torna possível o fluxo desses pacotes com retardo mínimo, evitando assim atrasos e cortes na transmissão. QoS refere-se a um conjunto de parâmetros que asseguram o desempenho em relação à disponibilidade e qualidade do serviço na rede. Esses parâmetros fixam valores de referência para características de rede como retardo, *jitter*, largura de banda e prioridade de tráfego (RACHID, 2004).

As redes *Ethernet* não foram originalmente projetadas para dar qualidade de serviço (QoS), por isso as aplicações em tempo real podem ser afetadas com atraso de pacotes prejudicando o entendimento da voz. Porém, o *Institute of Electrical and Electronics Engineers* (IEEE), instituto fundado nos Estados Unidos que desenvolve padrões para a indústria de computadores e eletro-eletrônicos, tem se esforçado adotando padrões que melhorem a qualidade de serviço sobre as redes IP (MAGRO, 2005).

As questões que determinam a qualidade de serviço são as seguintes (BORDIGNON, 2001):

- a) **confiabilidade do enlace:** correção de erros (em um fluxo de multimídia os erros não devem ser tratados, pois além de aumentar o *overhead* provoca atraso), serviço confirmado ou não, taxa máxima de erros e perda de pacotes;
- b) **retardo e variação do retardo de transmissão:** as características do meio físico utilizado, a distância percorrida e o congestionamento da rede são os fatores que podem ocasionar o retardo em uma transmissão;
- c) **natureza do modelo:** levam em conta se a rede é orientada a circuito (PSTN) ou pacote (Redes IP);
- d) **tipo de serviço de rede:** no tipo orientado à conexão, um circuito virtual é estabelecido entre as entidades envolvidas e para obter *QoS*, transmissor e receptor negociam os parâmetros ao estabelecer a conexão. Há uma grande vantagem nesse tipo de serviço, já que com a utilização de uma mesma rota física para transferência dos pacotes, recursos são reservados exclusivamente para essa conexão, garantindo a largura de banda e a prioridade no tráfego. Já no serviço não orientado a

conexão, como trabalha o protocolo *RTP*, cada pacote enviado deve conter os endereços de origem e destino para que os roteadores possam efetuar a entrega do pacote pelo melhor caminho no momento, mas sem garantia de QoS;

- e) **compartilhamento do meio físico:** aspectos de garantia de banda e tecnologias de acesso ao meio.

Para um bom desempenho, as aplicações de multimídia nas redes IP, requerem um atraso mínimo (ou nenhum) para entrega dos pacotes, mais que qualquer outro tipo de aplicação. As medidas de qualidade de serviço para a aplicação (adotadas a partir das questões que determinam a qualidade de serviço) dependem essencialmente dos *codecs*, não dependem da rede (BORDIGNON, 2001).

De nada adianta garantir medidas para a qualidade de serviço e aumentar a banda suficientemente a ponto de não mais necessitar da compactação efetuada pelos *codecs*, se são eles os responsáveis pela digitalização, codificação e decodificação dos fluxos de multimídia.

## 5.5 CONVERSÃO E COMPRESSÃO DE ÁUDIO – *CODECS*

Para que um som analógico (frequentemente a fala humana) possa ser usado na telefonia IP chegando até um destinatário, as seguintes etapas são necessárias (KRISHNAMURTHY; REXFORD, 2001) :

- a) **captura:** o som deve ser capturado por um dispositivo análogo, como por exemplo um microfone e transformado em um formato digital;
- b) **codificação:** a cada quadro criado, um codificador converte-o para um formato específico de áudio;

- c) **armazenamento:** um servidor pode armazenar o fluxo que foi codificado para uso futuro;
- d) **remessa:** no caso de uma comunicação em tempo real, o fluxo é transmitido imediatamente após sua captura e codificação;
- e) **decodificação:** o fluxo é decodificado no destinatário e um áudio é reproduzido.

O procedimento de codificação e decodificação é efetuado por um *codec*.

Um sinal de áudio depois de digitalizado, pode ser tratado como um arquivo e nele ser aplicado qualquer tipo de tratamento computacional disponível. Com o objetivo de transmiti-lo pela rede, um dos tratamentos mais sensatos a serem feitos é a compressão para a melhor economia possível da banda de rede. No caso do uso de *VOIP*, essa técnica de codificação e compressão de dos dados, deve ocorrer em tempo real para que a compressão e interação na sessão não sejam prejudicadas. Muitos *codecs* efetuam essa compressão. Isso pode variar de acordo com as condições da rede. Se a rede tem largura de banda suficiente para enviar os pacotes sem a compressão e sem causar congestionamento, a compressão pode ser evitada.

Um sinal de áudio perceptível aos ouvidos humanos deve estar aproximadamente na faixa de frequência 20 e 20.000 Hz. Essa delimitação pode ser usada pelos codificadores que retiram sinais abaixo de 20 Hz e acima de 20.000 Hz diminuindo o tamanho dos dados de forma que a onda resultante pode não ser igual a original, porém a diferença é imperceptível aos ouvidos humanos.

Segundo Bordignon (2001), a compressão é a forma de reduzir o tamanho da informação sem perder a informação original. A técnica de compressão, como já foi dito, é realizada por algoritmos de codificação, mas para que o destinatário (no caso, o

receptor na outra ponta da sessão de comunicação) possa entender as informações, elas devem ser descompactadas por um algoritmo de decodificação compatível.

### **5.5.1 Padrões de Digitalização e Compactação de Áudio.**

Existe uma variedade de padrões de codificadores, onde os mais utilizados são (BORDIGNON, 2001):

- a) *PCM (Pulse Code Modulation)*: é o padrão de codificação usado por empresas prestadoras de telecomunicações. Baseado na equação de Nyquist que mostra que um sinal analógico, dentro dos limites inferior e superior (no sistema telefônico essa frequência está entre 0 e 4.000 Hz) e com frequência  $H$ , pode ser digitalizado, transmitido e reconstruído sem perdas a partir de  $2 \times H$  amostras por segundo;
- b) *ADPCM (Adaptive Delta Pulse Code Modulation)*: uma adaptação do padrão anterior com uso de técnicas de compactação. As principais versões do ADPCM são: G.729, G.723, G.722 e GSM.

A padronização na utilização dos algoritmos de áudio é essencial para a interoperabilidade do sistema. A ITU e outros órgãos internacionais padronizaram os principais algoritmos para esse fim (que também são conhecidos como *vocoders*) (NÓBREGA, 2001).

Vocoders	Técnica
G.711	PCM
G.723.1	MPL-MLQ
G.726	ADPCM
G.728	LD-CELP
G.729 <sup>a</sup>	CS-ACELP

Figura 15. Tipos de *vocoders*  
 Fonte: NÓBREGA, O. (2001)

## 5.6 RETARDO NA TELEFONIA IP

De modo geral, muitos fatores contribuem para os atrasos da entrega de pacotes nas redes, desde a qualidade e problemas na parte física até os algoritmos utilizados pelos computadores, roteadores e outros.

A recomendação da ITU-T define os valores que caracterizam os atrasos em milissegundos na telefonia IP, conforme Figura 16 (NÓBREGA, 2001):

Excelente	Bom	Pobre	Inaceitável
0	150	300	450

Figura 16. Atraso em um sentido de milissegundos  
 Fonte: NÓBREGA, O. (2001)

Ainda segundo Nóbrega (2001), o atraso percebido em uma sessão *VOIP* são formados por dois tipos:

- a) **atraso na formação do pacote:** é o tempo requerido para que um pacote a ser entregue seja preenchido pelo *vocoder* com uma pequena amostra de voz do emitente ;

- b) **atraso gerado pela rede:** é o tempo gasto pela rede física (placas de redes, roteadores e outros) para que o pacote seja entregue em seu destino.

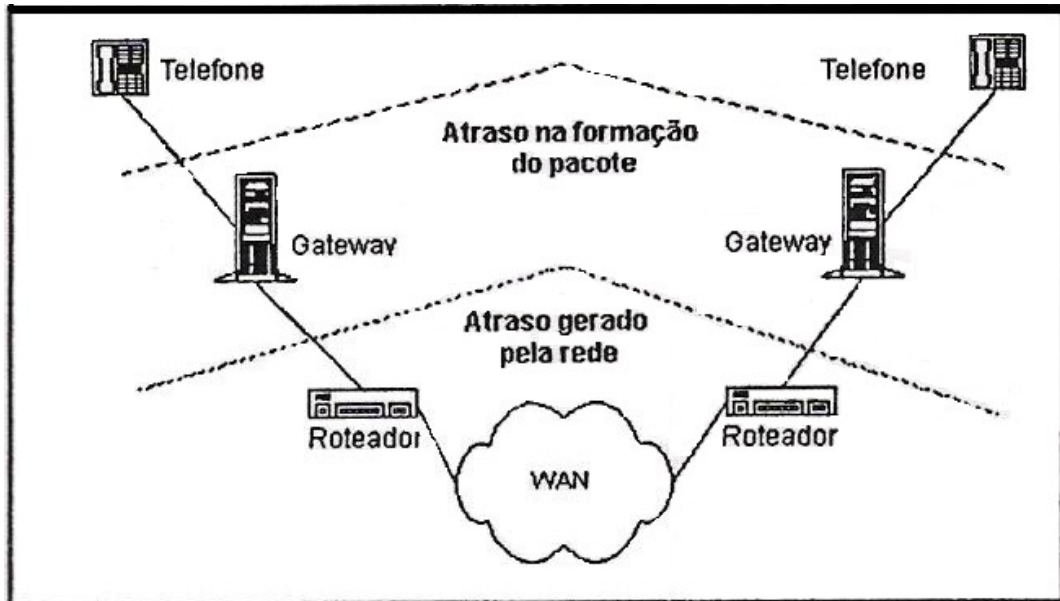


Figura 17. Tipos de Retardos  
Fonte: NÓBREGA, O. (2001)

Após entender todas essas tecnologias citadas aqui nesse capítulo, que tornam possível o uso da voz sobre as redes IP, pode-se estudar o que realmente é *VOIP*, onde está sendo usado e quais suas vantagens.

## 6 VOIP

Segundo Grecco (2004), *VOIP* e Telefonia IP podem ser facilmente confundidas, mas não são a mesma coisa. De acordo com ele, a primeira utiliza apenas a rede de computadores, enquanto a segunda interliga essa à rede de telefonia pública

Voz sobre IP (*VOIP* que em inglês significa *Voice over Internet Protocol*) é uma tecnologia, que por meio do uso de uma rede de dados (Internet, intranet e extranet), transmite voz. O conceito *VOIP* nasceu da ligação de duas redes distintas até então: as redes de telefonia pública com tráfego entre centrais telefônicas e Internet onde toda a base de fluxo é feita sobre o protocolo TCP/IP, podendo essa junção trazer aumento de eficiência em comunicações e a redução de custos em telefonia. O fluxo de áudio enviado nas redes IP, “disputa” o meio físico com o fluxo de dados que por ali trafega mas com a melhoria de desempenho das conexões das redes em geral, esse recurso vem se tornando cada vez mais viável. O uso de *VOIP* aparece como uma alternativa para o desenvolvimento de uma nova série de serviços que ajudam na redução dos custos (RACHID, 2004).

Os benefícios da *VOIP* interligada à rede pública de telefonia são (GRECCO, 2004):

- a) **integração:** economia no investimento inicial, já que voz e dados usarão a mesma infra-estrutura;
- b) **aplicações:** telefones IP podem ser usados como computadores, já que alguns deles já dispõem de navegadores e outros recursos;
- c) **mobilidade:** possibilidade de ativar um ramal em qualquer lugar do mundo como se estivesse no local da empresa;

- d) **convergência:** com as melhorias das redes IP, vários tipos de aplicativos, como videoconferência, estão se adaptando à elas;
- e) **período de transição:** possibilidade de migrar passo a passo antes da conversão completa já que é possível usar centrais híbridas.

Com tais benefícios, além da redução de custos, conclui-se que a adoção da voz sobre IP a médio prazo é inevitável tanto nas empresas como em residências.

O dispositivo final que um usuário deve usar para fazer uso de *VOIP* pode ser um telefone convencional, um telefone especial chamado Telefone IP ou até mesmo um computador desde que um gateway converta os pacotes de áudio para o formato que os participantes da sessão possam ouvir. O uso de tal dispositivo é dependente do tipo de cenário usado.

## 6.1 AMBIENTES DE APLICAÇÕES *VOIP*

Apesar dos diversos cenários possíveis para *VOIP*, o estabelecimento de telefonia sobre as redes IP pode ser dado basicamente de três formas (CALVACHE, 2005):

- a) **computador à computador:** é o caso mais simples. Quando conversa-se por meio de um pc, usando seu sistema de multimídia (microfone, caixas e placa de som) e uma rede qualquer (internet, intranet ou extranet) pratica-se *VOIP*. Atualmente existem várias aplicações (ver Sessão 6.2), gratuitas ou não que proporcionam esse tipo de comunicação. A rede de telefonia pública não é usada;

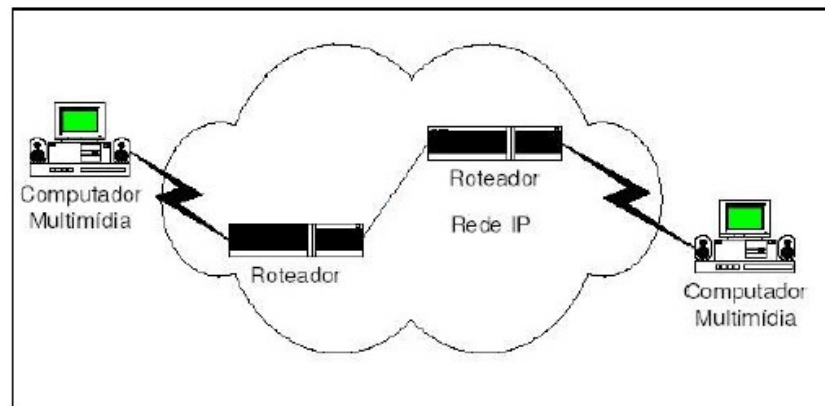


Figura 18. Telefonia IP entre dois computadores  
 Fonte: CALVACHE, H. (2005)

- b) **telefone convencional à telefone convencional:** nesse tipo de comunicação, a operadora de telefonia, ou um gateway, fornece um “link” entre a rede convencional e a rede IP (Internet), possibilitando telefonia IP entre dois telefones convencional;

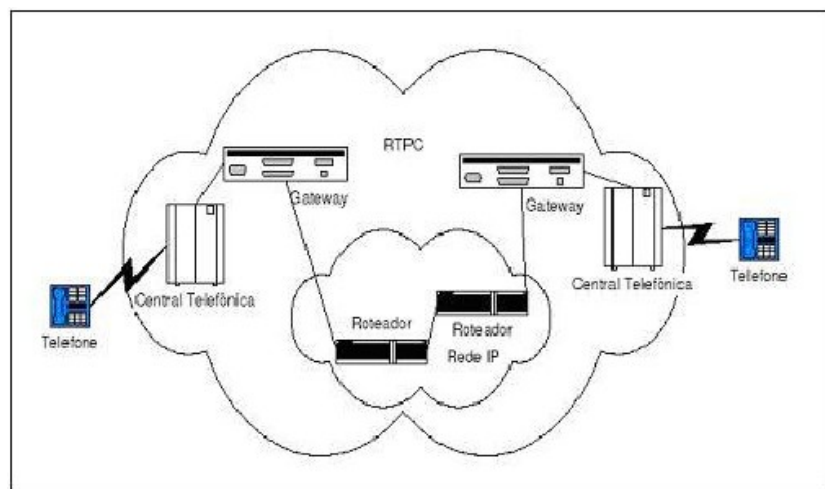


Figura 19. Telefonia IP entre dois telefones fixos  
 Fonte: CALVACHE, H. (2005)

- c) **telefone convencional à um computador (rede híbrida):** nesse tipo de telefonia, existe uma interface (gateway) que “mescla” as duas redes

distintas permitindo a comunicação entre elas. Ela também converte a voz de analógico para digital e associa um identificador (ID) ao endereço IP do micro.

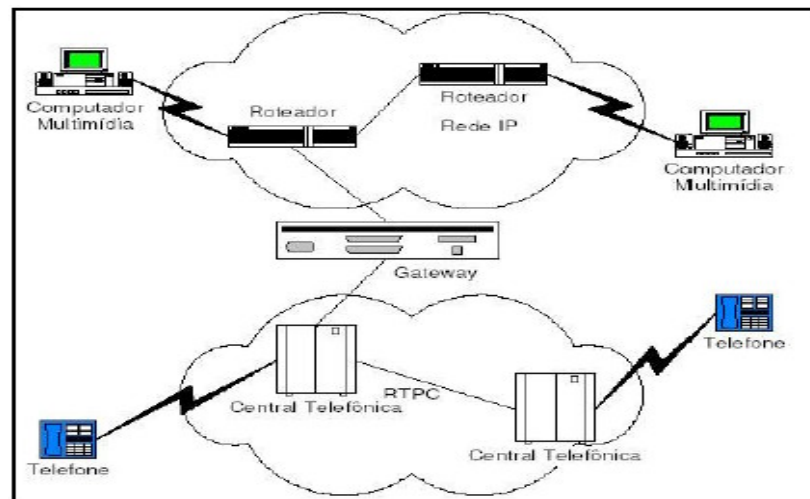


Figura 20. Telefonia IP Híbrida  
Fonte: CALVACHE, H. (2005)

## 6.2 TECNOLOGIAS E SOLUÇÕES DE MERCADO

Algumas aplicações e tecnologias que já fazem parte do dia-a-dia de muitas empresas (e até mesmo em residências) e disponibilizam ao usuário final uma forma de usar *VOIP*, são:

- a) **Soyo G668 Ethernet IP Phone:** telefone para *VOIP* que funciona de modo autônomo, ou seja, sem presença de um computador. Basta ter um serviço de banda larga como ADSL ou Internet a cabo e conectá-lo ao modem desse serviço. Dependendo da configuração de rede em que se encontra, o aparelho obtém um endereço IP dinamicamente (via *DHCP*) ou pelo seu display pode-se configurar um IP fixo. Ele funciona

como um telefone celular pré-pago, compra-se os créditos do site da própria Soyo (www.soyo.com). Uma ligação para um telefone fixo de São Paulo e Rio de Janeiro custa R\$ 0,05 por minuto, para as outras cidades do Brasil R\$ 0,08 e para um celular brasileiro R\$ 0,23. Preço do aparelho: R\$ 480,00 (MACHADO, 2004).



Figura 21. Soyo G668  
Fonte: MACHADO, C. (2004)

Esse telefone também pode ser usado de duas outras formas: a primeira permite que ele substitua um micro que use um *software* de comunicação computador à computador, e a segunda por meio do uso de um equipamento PABX IP que realiza a conversão de endereço IP para um número de ramal comum de uma empresa;

- b) gateway Tenor A200:** dois aparelhos desses conectam dois escritórios e possibilitam duas ligações simultâneas de *VOIP* (apenas entre esses dois pontos). Para que todos os telefones da empresa tenham acesso às linhas IP, o aparelho deve ser integrado ao PABX da empresa. Preço de dois aparelhos R\$ 5.760,00.

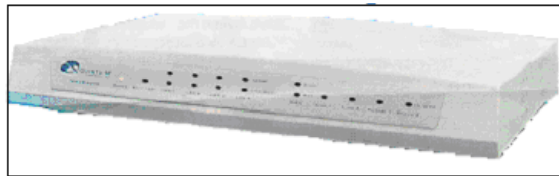


Figura 22. Tenor A200  
Fonte: YURI, F. (2004)

- c) **gateway da Vonage:** é um aparelho de pequeno porte que conectado a uma rede de banda larga de qualquer lugar do mundo, fornece o número de uma linha telefônica de qualquer cidade dos Estados Unidos e o usuário pode fazer ligações, com custo de uma ligação local, da cidade em que se encontra. Custo mensal do serviço US\$ 29,99. (REGGIANI, 2004);
- d) **UOL Fone:** *software* que permite fazer ligações para qualquer telefone fixo e celular do mundo. Também funciona como um telefone celular pré-pago. Paga-se por ligação feita. Uma chamada para os Estados Unidos, por exemplo custa R\$ 0,16 por minuto (INFO ABRIL);
- e) **ATA180:** equipamento da Cisco que possibilita o uso de telefones convencionais em redes IP. O aparelho contém uma interface que conecta essas duas tecnologias por meio de conectores RJ-45 e RJ-11.

### 6.2.1 Soluções Gratuitas

O fascínio pela liberdade é uma característica de todo ser humano. Quando se fala em *software*, o fascínio pela liberdade de seu uso, estudo do funcionamento e a possibilidade de adaptações em seu código, também existe (pelo menos por certa parte

da população). O desejo por essa liberdade apareceu na década de 80, porém acentuou-se no início dos anos 90 com o surgimento do sistema operacional Linux. Nos dias atuais, o *software* livre já é uma realidade e está presente em todo tipo de aplicação (WIECZOREK, 2005).

Algumas aplicações gratuitas de *VOIP* são:

- a) **skype:** *software* de plataforma Windows e Linux gratuito que permite interligar dois computadores em qualquer lugar do mundo. Para falar e escutar por meio das redes IP, o micro deve ser equipado com dispositivos de captura de áudio (microfone) e saída (caixas de som) (INFO ABRIL);
- b) **EyeBall Chat:** *software* de plataforma windows gratuito da empresa EyeBall Networks que permite conversação entre computadores pela rede IP e também integra mensageiros instantâneos como o msn (INFO ABRIL);
- c) **Glophone Blue:** *software* gratuito de plataforma windows desenvolvido pela empresa VoiceGlo que permite tráfego de voz pela Internet. Ele é associado a um serviço que fornece um número de telefone fixo nos Estados Unidos (INFO ABRIL);
- d) **GnoPhone:** *software* gratuito e aberto que roda em Linux e permite fazer ligações para outros usuários GnoPhone ou para a rede de telefonia pública, desde que um gateway Asterisk PBX esteja disponível. Utiliza codec GSM (INFO ABRIL);
- e) **Linphone:** *software* gratuito e aberto que roda em Linux. Faz ligações telefônicas sobre redes IP e pode ser cliente em um sistema com Asterisk (INFO ABRIL);

f) **Asterisk:** é um *software* de PBX completo e gratuito. Ele roda na plataforma Linux (qualquer distribuição) tornando um simples computador em um PABX, provendo todas as funcionalidades que se espera de um desses. Asterisk faz Voz sobre IP em diferentes protocolos (H.323, SIP, IAX) e pode se integrar à maioria dos padrões de telefonia utilizando hardware de baixo custo. Ele dispõe de recursos como: secretária eletrônica, fax, áudio-conferência, correio de voz. (GONÇALVES, 2005).

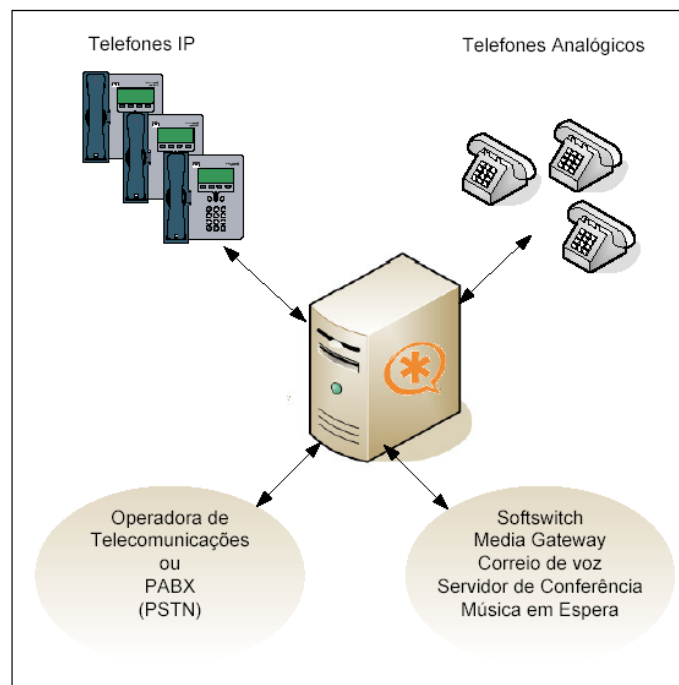


Figura 23. Visão Geral Arquitetura Asterisk  
Fonte: GONÇALVES, F. (2005)

g) **SiPX:** é um *software* de PBX com interface gráfica gratuito para plataforma Linux. Ele é um sistema de telefonia IP formado por um número independente de servidores que se comunicam usando o protocolo SIP. Com ele é possível a criação de uma rede híbrida,

interligando uma rede IP com uma rede de telefonia comum (SIPFOUNDRY, 2004);

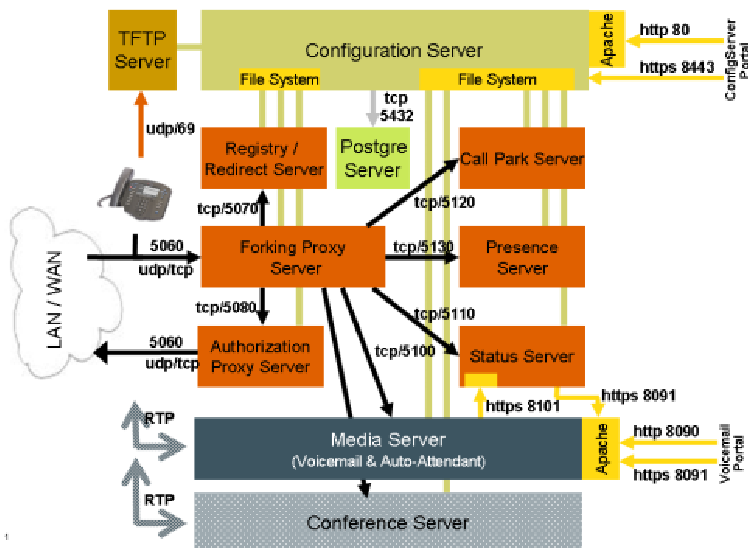


Figura 24. Visão Geral da Arquitetura Proposta pela SiPfoundry  
Fonte: SIPFOUNDRY (2004)

### 6.3 ESTUDOS DE CASOS

As tecnologias que permitem o uso da voz sobre as redes IP evoluíram muito e possibilitaram que diversas empresas as adotassem sem medo de retorno negativo, ou seja, de não suprir suas necessidades de telefonia e de não terem um retorno financeiro significativo. Pelo contrário, serviços que elas nem imaginavam que eram possíveis estão disponíveis após a migração para *VOIP* (REGGIANI, 2004).

Algumas empresas que implantaram sistemas *VOIP* são:

- a) **Indiana Seguros:** investirá um total US\$ 280.000,00 em um sistema telefônico híbrido baseado na internet fornecido pela empresa 3Com. Com intenção de reduzir 30% de custos nas ligações, a Indiana Seguros investiu inicialmente US\$ 160.000,00 em uma infra-estrutura para 280

ramais que mantêm em funcionamento paralelo com os outros ramais convencionais (FUSCO, 2005);

- b) **Construtora Andrade Gutierrez:** sem delay e sem som metálico, após aplicar uma solução *VOIP* a empresa economiza R\$ 11.000,00 mensalmente com telefonia. Foram investidos R\$ 120.000,00 em equipamentos, como o PABX Meridian da empresa canadense Nortel, para interligar 3 filiais permitindo até 32 ligações simultâneas em cada uma delas. Em paralelo, está sendo testado um soft Phone proprietário, também da Nortel, que possibilita que um funcionário da construtora fale de qualquer lugar do mundo com seu notebook como se estivesse no escritório local (YURI, 2005);
- c) **Casas Bahia:** com objetivo de baixar seus custos com ligações telefônicas que atingiam anualmente a casa de 1 milhão de reais, a Casas Bahia implantou telefonia IP em suas 360 lojas pelo Brasil. Após um ano de migração (ficou pronta em 2002), o sistema que usa equipamentos Cisco numa linha Frame Relay, se pagou em seis meses (REGGIANI, 2004);
- d) **Bunge Alimentos:** após aquisição de uma central Siemens, alguns equipamentos *VOIP* e aumento da banda de seu link em 1 MB, funcionários da Bunge podem fazer ligações interurbanas com custo reduzido. Com investimento de R\$ 20.000,00 a empresa reduziu bastante seus gastos em telefonia que eram superior a R\$ 40.000,00 (REGGIANI, 2004);
- e) **Ford:** atualmente 800 mil ligações mensais são efetuadas em cima das redes IP. A empresa migrou de um sistema voz sobre IP de telefonia

conhecido como TDM (implantado em 1995), para um sistema moderno de *VOIP* e garante que conseguiu 15% a mais de reduções nos custos com ligações além do novo sistema ter manutenção bem mais simples (REGGIANI, 2004);

- f) Roche:** fábricas dessa empresa situadas no Brasil, México e Argentina usam *VOIP* após instalação de um backbone regional na América Latina. Mesmo possuindo centrais PABX de marcas diferentes, um equipamento de *VOIP* da Cisco possibilita o uso de telefonia IP para 60% das ligações entre as fábricas com meta de chegar a 100%. Foram investidos até o momento US\$ 35.000,00 mas somente para o Brasil uma economia de US\$ 150.000,00 anual já foi alcançada (REGGIANI, 2004).

Nem tudo foi descoberto quando se fala em *VOIP*. Muitas pesquisas e trabalhos são efetuados na área e podem trazer à tona novas tecnologias melhorando ainda mais seu desempenho e uso.

## 7 TRABALHOS CORRELATOS

Por ser uma área relativamente nova e com os grandes benefícios que ela pode trazer, existem muitos estudos em diferentes linhas relacionadas com telefonia sobre redes IP.

A exemplo de outros sistemas que são mantidos com o código fonte aberto (como o sistema operacional “Linux”) uma comunidade virtual de *software* livre espalhada pelo mundo ajuda constantemente a melhorar o código fonte dos protocolos do padrão H.323 que é organizada pela ITU-T no projeto OpenH323. Códigos fonte e maiores informações podem ser obtidos na página [www.openh323.org](http://www.openh323.org).

Outra comunidade internacional de *software* com código fonte aberto na área de *VOIP* é a SIPfoundry. Seu objetivo é ajudar na aceleração do uso do padrão de protocolos SIP para a área. Maiores detalhes podem ser vistos em [www.sipfoundry.org](http://www.sipfoundry.org).

Na Universidade Estadual de Campinas, Stella Wing Kwan Chung, autora do trabalho “Propostas de mobilidade para os protocolos H.323 e SIP em Telefonia IP, utilizando linguagem de especificação formal SDL orientada à objetos”, concluído em julho de 2004, propõe a mobilidade interdomínio utilizando equipamentos móveis *VOIP*, ou seja, a possibilidade de durante uma sessão pode estar se locomovendo e trocar de domínio de uma rede sem interrupção da comunicação.

Também na Universidade Estadual de Campinas, um trabalho intitulado “Estudo da Qualidade de Voz em Redes IP” foi desenvolvido por Júlio César Magro (MAGRO, 2005) tendo como principal objetivo a caracterização do serviço de voz em redes IP levando em consideração os efeitos dos fatores de rede e gateway no tempo de estabelecimento de uma chamada (conexão) e na qualidade da voz.

Em Blumenau engenheiro de telecomunicações Humberto Maxclioff apresentou em seu trabalho de conclusão de curso “Protótipo de um Sistema de Telefonia IP Criptografado Baseado no Padrão SIP” um protótipo de um sistema de telefonia IP criptografado entre dois computadores utilizando o padrão SIP em um ambiente Windows. Sua motivação foi a crescente troca de informações de multimídia em redes IP.

Na Universidade do Extremo Sul Catarinense o Bacharel Marcelo Bongioiolo, desenvolveu uma aplicação para *VOIP* apresentado no trabalho “Aplicação Experimental Para Comunicação de Voz Sobre IP” que teve como objetivo principal prover comunicação sobre as redes IP de uma empresa de telecomunicações.

O próximo capítulo apresenta o trabalho proposto ao qual objetiva essa pesquisa.

## 8 UMA SOLUÇÃO DE VOZ SOBRE IP DE BAIXO CUSTO PARA AMBIENTES CORPORATIVOS

Asterisk foi originalmente criado por Mark Spencer da Digium (Alabama - USA) que apóia o projeto e comercializa o hardware de telefonia para ser usado com Linux. Hoje o código do Asterisk tem contribuição de toda comunidade mundial de software livre.

Seu nome originou-se do símbolo \*, que no Linux é um *wildcard*, ou seja, equivale a qualquer caractere. O hardware é desenvolvido, principalmente, por Jim Dixom e sua equipe da Digium, tem baixo custo e é aberto também, ou seja, o esquema da construção das placas, com cada componente usado, pode ser baixado para quem quiser fabricar (GONÇALVES, 2005).

Asterisk usa diferentes protocolos, como H.323 e o SIP, já descritos nesse trabalho, mas possui um protocolo próprio, o *Inter-Asterisk eXchange* (IAX) que tem a vantagem de utilizar uma única porta (UDP 4569) para transporte das sinalizações e fluxo de voz (A COMUNIDADE BRASILEIRA DO PBX OPEN SOURCE, 2006).

Além de algumas inovações em telefonia, como capacidade de conectar funcionários trabalhando em casa ao PABX da empresa como se estivessem na própria empresa e capacidade de conectar filiais a sua matriz sobre redes IP, os recursos mais usuais do Asterisk são (GONÇALVES, 2005):

- a) música em espera para clientes esperando em filas (formato mp3);
- b) filas de chamada onde agentes de forma conjunta atendem as chamadas e monitoram as filas;
- c) envio de fax com redirecionamento para e-mail;

- d) registro detalhado de chamadas para integração com outros sistemas que necessitem desse serviço;
- e) habilidade de conexão com linhas telefônicas normais;
- f) correio de voz. Permite que, quando um usuário não atenda a uma ligação, uma mensagem possa ser gravada para posteriormente ser ouvida por ele;
- g) sala de conferência que permite que dois usuários ou mais falem em conjunto;
- h) atendimento automático por meio de unidade de resposta auditiva (URA).

Como foi visto, o Asterisk tem um conjunto de aplicações muito grande e pode, quando devidamente instalado, substituir qualquer PABX tradicional ou trabalhar em conjunto a ele.

Sendo essa última condição adotada para os testes em laboratório, o trabalho desenvolvido compreendeu a instalação, configuração e uso do Asterisk em um ambiente corporativo, ao qual pode-se testar, em um ambiente de produção real, as vantagens e desvantagens no uso desse sistema VOIP de baixo custo.

O laboratório em que o sistema foi testado, trata-se de uma empresa de médio porte que contém uma central, na cidade de Içara, e mais oito filiais espalhadas no estado. Como duas dessas filiais já tinham uma solução VOIP em funcionamento, foram usadas como laboratórios a central e mais seis filiais, às quais serão referenciadas neste trabalho na seguinte forma: central como laboratório *A*, filial de Criciúma como laboratório *B* e as demais filiais como laboratório *C*, *D*, *E*, *F* e *G*.

Nos laboratórios *A* e *B* foram instalados um servidor Asterisk cada, nos quais tornou-se possível o uso de VOIP por meio de telefones fixos ou *soft phones* entre

essas duas localidades. Nos demais laboratórios usou-se apenas *soft phones* para chamadas VOIP. A Figura 25 apresenta esta solução.

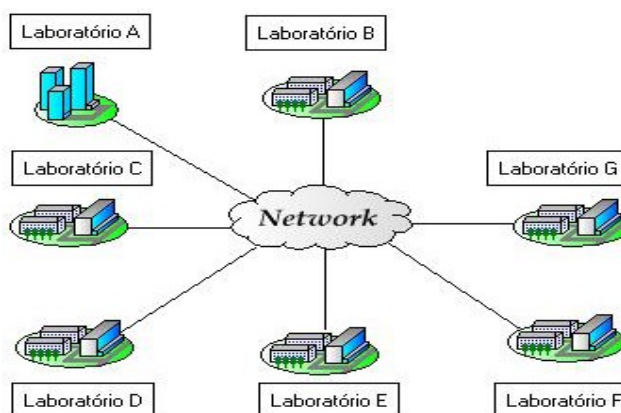


Figura 25. O Laboratório

A metodologia para o desenvolvimento desta pesquisa, compreendeu as etapas de: levantamento bibliográfico, identificação das tecnologias necessárias para instalação do Asterisk, instalação do sistema operacional, instalação do Asterisk, configuração do x-lite, configuração do Asterisk, relato das dificuldades encontradas.

## 8.1 LEVANTAMENTO BIBLIOGRÁFICO

A parte inicial da fundamentação teórica do trabalho, apresentada nos capítulos com assuntos sobre: comunicação de dados, redes de computadores, protocolos e camadas, multimídia na rede (protocolos para VOIP), obteve sua principal fonte em livros que foram encontrados com certa facilidade por serem assuntos já consolidados mundialmente.

Os dois últimos capítulos, que apresentam assuntos sobre: VOIP e trabalhos correlatos, apresentaram maior dificuldade na obtenção de suas fontes. É certo que essa

dificuldade foi encontrada por ser a tecnologia VOIP um assunto relativamente novo. A maioria dos materiais com essa referência foram encontrados em revistas, matérias em sites respeitados, monografias, dissertações e trabalhos de conclusão de curso.

Mas se voz sobre IP é uma tecnologia nova, ainda mais nova é o Asterisk. São muitos os sites que disponibilizam tutoriais para auxílio na instalação e configuração dele, livros porém são poucos. O site brasileiro [www.asteriskbrasil.org](http://www.asteriskbrasil.org) é um dos poucos exemplos de tutoriais na língua portuguesa. A maioria é disponibilizada em inglês, sendo um dos que mais se destaca, a página [www.asterisk.org](http://www.asterisk.org). Entretanto as principais fontes sobre Asterisk foram os livros de Mahler e Gonçalves, este último gentilmente cedido pelo próprio autor á UNESC.

## 8.2 IDENTIFICAÇÃO DAS TECNOLOGIAS NECESSÁRIAS PARA INSTALAÇÃO DO ASTERISK

Para que o Asterisk fosse instalado e colocado em condições de uso no laboratório, precisou-se identificar os itens: versão do linux a ser usado, configuração necessária do computador e hardwares VOIP a serem usados.

### 8.2.1 Identificação da versão e distribuição Linux a ser usada

O Asterisk é desenvolvido para plataforma Linux, portanto qualquer distribuição poderia ser usada. Apesar disso, Mahler (2005) sugere o uso da distribuição Linux MEPIS e de Gonçalves (2005) sugere o da Debian. Mesmo assim, optou-se pela distribuição Linux Conectiva versão 10, devido ao uso de vários anos dessa distribuição pela empresa onde seriam realizados a implantação e os testes.

### **8.2.2 Identificação da configuração do computador necessária**

Gonçalves (2005) afirma que um micro com processador Pentium (ou compatível) de 300 Mhz, 256 memória RAM e 100 MB de espaço em disco é suficiente para testar e analisar o Asterisk.

De acordo com Mahler (2005), para uma empresa de médio porte, uma máquina com processamento de 2 GHz, 512 de memória RAM e 60 GB de *Hard Disk* é o recomendado. Ele ainda diz que o Asterisk usa intensamente o processador da máquina, portanto é desejável que ela seja usada apenas para o uso desse sistema.

Os dois servidores usados para a implantação do sistema para testes em campo foram:

- a) Um Pentium 4 de 1.7 GHz. 256 MB RAM e 20 GB de disco;
- b) Um Celerom 2.3 GHz, 256 MB RAM e 30 GB de disco.

Para evitar possíveis conflitos de irq, é aconselhável desabilitar no Setup da máquina, qualquer dispositivo que não será usado, como portas USB, portas COM1 e 2 e multimídias.

### **8.2.3 Identificação de hardwares VOIP para uso no Asterisk.**

Diversos tipos de placas de telefonia, de vários fabricantes, podem ser usados dentro de um computador que roda o Asterisk. Mas o principal fabricante desses dispositivos é a Digium, que fabrica desde dispositivos completos como a placa TE110P que suporta até 30 canais digitais a placas simples de apenas um canal analógico, como é o caso da X100P.

De acordo com Gonçalves (2005), a placa de Modem da Intel, com chipset Ambient MD3200 é considerada clone da X100P e pode ser usada para fazer integração entre a rede de dados e um PABX, apesar de ficar limitada a apenas uma ligação por vez (quando se trata de telefones fixos).

Duas dessas placas clones foram instaladas e usadas nos testes em laboratório, sendo uma delas instalada em uma unidade de Içara e outra em Criciúma, possibilitando ligações gratuitas entre ramais telefônicos comuns usando a rede de dados entre esses dois pontos remotos.



Figura 26. Placa TE110P  
Fonte: GONÇALVES, F. (2005)



Figura 27. Placa X100P  
Fonte: GONÇALVES, F. (2005)

Outros exemplos de hardwares que podem ser usados na arquitetura Asterisk são os telefones IP e adaptadores ATA, conforme descritos no capítulo 6,

sendo que esse primeiro foi substituído nos testes realizados por programas conhecidos como soft phones, que são simuladores de telefones e devem ser instalados em micros.

Após a identificação das tecnologias necessárias para instalação do Asterisk prossegui-se a instalação. Antes disso deve-se instalar o sistema operacional. Na sessão 8.3 é descrita a instalação do sistema operacional Linux Conectiva 10.

### 8.3 INSTALAÇÃO DO SISTEMA OPERACIONAL

A distribuição Conectiva Linux versão 10, contém três cds principais sendo que o download das imagens dos arquivos que os compõem pode ser feito em <ftp://ftp.conectiva.com.br/pub/conectiva/10/>.

Após conectar no micro a placa de telefonia, deve-se configurar o Setup da máquina para iniciar por meio do cd-rom, reiniciá-la com o cd 1 do conectiva no cd-rom e executar os passos abaixo:

- a) responder a questões simples, como escolha do tipo de mouse e teclado, que não irão intervir no uso do sistema;
- b) tipo de instalação: qualquer tipo deverá funcionar, porém se a máquina for usada apenas para o Asterisk, usar a instalação mínima. Nessa etapa, pode-se escolher "Forçar Particionamento Manual" para definir qual partição deverá ser usada pelo Linux. Se a máquina não contiver nenhum outro sistema, não marcar essa opção, pois o instalador saberá onde instalar. Se ela for marcada, uma tela de particionamento será apresentada. Então uma partição swap deve ser criada com tamanho entre 128 e 256 MB sendo que o restante do HD deverá ser a partição "/" com um sistema de arquivo do tipo ext3;

- c) configuração da rede: deve-se escolher um nome para a máquina e configurar um IP fixo;
- d) as próximas telas serão questões comuns, como escolha do monitor e placa de vídeo. O instalador irá detectar automaticamente, caso isso não ocorra, deve-se escolher de acordo com o hardware possuído.

## 8.4 INSTALAÇÃO DO ASTERISK

Para instalar o Asterisk no Conectiva 10, é necessário: instalar os pacotes extras do sistema, configurar o *udev*, re-compilar o kernel, instalar o pacote *zaptel*, instalar o pacote *libpri* e instalar o pacote *asterisk*.

### 8.4.1 Instalação dos pacotes extras do Linux Conectiva

Para instalação do Asterisk os seguintes pacotes extras do Conectiva, que provavelmente não foram instalados automaticamente, serão necessários: *openssl-devel*, *bison*, *zlib-devel*, *glibc*, *glibc-devel*, *ncurses-devel*, *gcc*, *kernel-source*, *make*, *udev*, *sendmail*, *mutt* e *libtiff* ( estes três últimos necessários apenas para envio de e-mails).

Para instalá-los, deve-se entrar com usuário *root* e digitar o comando abaixo. Dependendo do tipo de instalação feita, algum pacote já poderá estar instalado.

```
apt-get install "nome-do-pacote".
```

### 8.4.2 Configuração do *udev*

O serviço *udev* permite ao Linux ter um diretório */dev* dinâmico e atualizado de acordo com os dispositivos conectados, criando e deletando dispositivos dinamicamente. Isso possibilita que se possa criar suas próprias regras de nomeação de dispositivos.

Para criar esses dispositivos dinâmicos, deve-se usar um editor de textos para inserir o conteúdo da tabela 1 no final do arquivo */etc/udev/udev.rules* e inicializar o serviço. Para isso deve-se logar como *root* no Linux e digitar os comandos *cds* e *./udev start*.

No caso de uso de kernel menor que 2.6, não é necessário a sua instalação.

```
KERNEL="zapctl", NAME="zap/ctl"  
KERNEL="zaptimer", NAME="zap/timer"  
KERNEL="zapchannel", NAME="zap/channel"  
KERNEL="zappseudo", NAME="zap/pseudo"  
KERNEL="zap[0-9]*", NAME="zap/%n"
```

Tabela 1 – Definindo dispositivos dinâmicos

### 8.4.3 O kernel

O próximo passo é re-compilar o kernel para habilitar o carregamento de módulos necessário para o pacote *zaptel*. O usuário deve logar-se como usuário *root*, se posicionar em */usr/src/linux* e digitar os comandos conforme seqüência da tabela 2:

1	make mrproper
2	cp configs/kernel-pentium4.config .config
3*	vi makefile
4	make menuconfig
5	make dep
6	make clean
7	make bzImage
8	cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.5-asterisk
9	cp System.map /boot/System.map-2.6.5-asterisk
10	ln -s /boot/System.map-2.6.5-asterisk /boot/System.map
11	make modules
12	make modules_install
13	cd /lib/modules/-2.6.5-asterisk
14	mkinitrd 2.6.5-asterisk

Tabela 2 - Compilando o kernel

Deve-se observar o seguinte: \*na linha 3, após digitar o comando, deve-se localizar o campo EXTRAVERSION e mudar seu valor para *-asterisk*. Na linha 4, apenas escolha exit e yes, ou pode-se aproveitar para tirar do futuro kernel tudo que não irá usar.

Após digitar todos comandos, um novo kernel será criado, bastando criar uma nova entrada para ele no arquivo menu.lst do diretório /boot/grub. Usando seu editor de texto preferido, adicione as linhas da tabela 3.

title = Conectiva Linux 10 - asterisk
kernel = (hd0,5)/boot/vmlinuz-2.6.5-asterisk root=/dev/hda6 vga=0x314 splash=silent 5
initrd = (hd0,5)/boot/initrd-2.6.5-asterisk

Tabela 3 – Exemplo de nova entrada no Grub

#### 8.4.4 Instalação do pacote ZAPTEL

Sendo uma placa de telefonia usada no servidor ou se desejado usar recursos de música de espera ou sala de conferências, o pacote zaptel deverá ser instalado. Deve-se logar-se como usuário *root* e criar um diretório com o comando `mkdir /usr/src/asterisk`, e em seguida:

- a) descompactar o pacote `zaptel-1.2.0.tar.gz` nesse diretório, com o comando: `tar xzvf zaptel-1.2.0.tar.gz`. Esse arquivo pode ser encontrado em <http://ftp.digium.com/pub/zaptel/releases/zaptel-1.2.0.tar.gz>;
- b) se não possuir uma placa de telefonia e for usar "Sala de Conferência" ou "Música em Espera" descomente a linha `ztdummy` do arquivo `usr/src/asterisk/zaptel-1.2.0/Makefile`;
- c) posicionar-se no diretório `/usr/src/asterisk/zaptel-1.2.0` e compilar o pacote da seguinte forma:

1	<code>cd /usr/src/asterisk/zaptel-1.2.0</code>
2	<code>make clean</code>
3	<code>make linux26</code>
4	<code>make config</code>
5	<code>make install</code>

Tabela 4 - Instalando o pacote `zaptel`

#### 8.4.5 Instalação do pacote **LIBPRI**

Descompactar o pacote `libpri-1.2.1.tar.gz` no diretório `/usr/src/asterisk`, com o comando: `tar xzvf libpri-1.2.1.tar.gz`. Esse arquivo pode ser encontrado em <http://ftp.digium.com/pub/libpri/releases/libpri-1.2.1.tar.gz>. Compilar o pacote da seguinte forma:

1	<code>cd /usr/src/asterisk/libpri-1.2.1</code>
2	<code>make clean</code>
3	<code>make</code>
4	<code>make install</code>

Tabela 5 - Instalando o pacote `libpri`

### 8.4.6 Instalação do pacote ASTERISK

Descompactar o pacote asterisk-1.2.0.tar.gz no diretório /usr/src/asterisk, com o comando: tar xzvf asterisk-1.2.0.tar.gz. Esse arquivo pode ser encontrado em <http://ftp.digium.com/pub/asterisk/releases/asterisk-1.2.0.tar.gz>. Compilar o pacote da seguinte forma:

1	cd /usr/src/asterisk/asterisk-1.2.0
2	make clean
3	make
4	make install
5	make samples(instala exemplos em /etc/asterisk)

Tabela 6 - Instalando o pacote Asterisk

### 8.5 CONFIGURAÇÃO DO X-LITE

Um soft phone é um aplicativo que pode ser instalado em um micro usando seus dispositivos de multimídia (caixas de som e microfone) substituindo um telefone IP ou um telefone convencional. Eles usam um ou mais protocolos para comunicação. O *soft phone*, usado nos terminais em todos os casos nos testes em laboratório, foi o x-lite, que é livre, usa protocolo SIP e pode ser conseguido no site [www.xten.com](http://www.xten.com) para as plataformas Windows e Linux.

A única configuração necessária para uso do x-lite após a sua instalação é um canal sip. Para isso, deve-se abrir a tela de comunicação, escolher a opção "System Settings", "SIP Proxy", "Default" e configurar:

- a) enabled: yes;
- b) display nome: "nome\_do\_usuario";
- c) username: usuário cadastrado no servidor /etc/asterisk/sip.conf;

- d) authorization User: repita o número do usuário;
- e) password: senha cadastrada no servidor /etc/asterisk/sip.conf;
- f) domain: domínio.com.br (domínio local);
- g) SIP Proxy: IP do servidor Asterisk;
- h) out Bound Proxy: IP do servidor Asterisk;
- i) direct Dial IP: NO.

## 8.6 CONFIGURAÇÃO DO ASTERISK

Após a instalação do Asterisk, deve-se configurá-lo de maneira que funcione de acordo com as necessidades desejadas. A medida que as alterações são feitas pode-se testar o resultado delas, para isso é necessário executar o sistema.

### 8.6.1 Iniciar os módulos para placa de telefonia.

Antes de executar o Asterisk propriamente dito, alguns módulos devem ser iniciados para funcionamento do ztdummy ou das placas de telefonia. Nesse último caso, pode-se digitar lspci, no *root*, para verificar se a placa foi reconhecida pelo Linux. Nos testes realizados, a seguinte linha apareceu para a placa X100P (clone) utilizada:  
*"Communication Controller> Tiger Jet Network Inc. Intel 537"*

Para que uma x100p funcione, os arquivos /etc/zaptel.conf e /etc/asterisk/zapata.conf, devem conter as linhas conforme modelos no apêndice C e o módulo zaptel deve estar sendo executado. Para isso, deve-se entrar no usuário *root*, digitar *cds* (irá para o diretório /etc/rc.d/init.d) e *./zaptel start*. Esse processo executará

basicamente os seguintes comando: `modprobe zaptel`, `modprobe ztdummy` (apenas se a linha `ztdummy` foi habilitada) e `modprobe wcfxo` (para placa x100p clone) .

Para ver se os módulos foram carregados digite: `lsmod` e verifique se aparecem linhas para `zaptel`, `ztdummy` e `wcfxo`. Se desejar finalizar esses módulos digite `./zaptel stop`. Também pode-se verificar se a placa esta configurada corretamente com o comando `ztcfg -vvv`.

Nesse ponto, um ramal do PABX deve ser ligado diretamente á placa x100p. Nos testes em laboratório simplesmente foi desconectado o fio de um telefone que estava funcionando e colocado na porta “line” da placa.

### 8.6.2 Iniciar o Asterisk

Para executá-lo manualmente, pode-se digitar no *root* o comando `asterisk -vvvvvgc`, sendo que um console do Asterisk ira aparecer (algo como `CLI>`). Um console é o local onde são mostrados os possíveis erros do Asterisk e onde comandos podem ser tratados por ele.

Para tornar o Asterisk inicializável junto com o Linux, não tendo que digitar o comando `asterisk -vvvvvgc` sempre que a máquina for reiniciada, deve-se copiar o arquivo `rc.redhat.asterisk` do diretório que contém o código fonte do Asterisk (`cp /usr/src/asterisk-1.2.1/contrib/init.d/ /etc/rc.d/init.d`) e fazer um link simbólico para o diretório de nível inicialização com um dos comandos abaixo:

- a) `ln -s /etc/rc.d/init.d/rc.redhat.asterisk /etc/rc.d/rc3.d/S98asterisk;`
- b) `ln -s /etc/rc.d/init.d/rc.redhat.asterisk /etc/rc.d/rc5.d/S98asterisk.`

No caso de dúvida, deve-se digitar `runlevel` no *root* para verificar o nível de inicialização. Se aparecer 3, deve-se usar o primeiro comando, caso contrário o

segundo. Feito isso, o Asterisk será executado automaticamente, junto com o sistema operacional, sempre que o micro for reiniciado. A partir daí, para ver um console (CLI>), deve-se usar o comando `asterisk -r`.

<code>reload</code>	Recarregar o Asterisk após uma alteração
<code>stop now</code>	Finalizar o Asterisk
<code>zap show status</code>	Mostra as placas de telefonia encontradas
<code>help</code>	Mostra uma tela de ajuda
<code>show applications</code>	Mostra todas aplicações do Asterisk
<code>sip show registry</code>	Mostra um Asterisk remoto registrado
<code>sip show peers</code>	Mostra usuários SIP registrados

Tabela 7 - Comandos úteis no console do Asterisk:

Agora que o asterisk já pode ser executado, pode-se configurar serviço à serviço e reiniciá-lo com o comando `reload` para testar se a alteração obteve sucesso.

Antes de começar a configurar o asterisk, é importante conhecer os diretórios usados por ele, onde os principais são:

- a) `/etc`: é onde estará o arquivo `zaptel.conf`;
- b) `/etc/asterisk`: é onde estarão todos os arquivos com extensão “`conf`” e que configuram o Asterisk, destacando-se: `extensions.conf`, `sip.conf`, `musiconhold.conf`, `iax.conf`, `voicemail.conf` e `zapata.conf`. Os arquivos `conf` são internamente divididos em contextos que tem como cabeçalho um nome entre colchetes. Exemplo: `[nome do contexto]`;
- c) `/var/lib/asterisk`: biblioteca Asterisk, que, entre outros, possui o diretório `sounds` contendo mensagens que poderão ser ouvidas (formato `gsm`) e o diretório `mohmp3` que contém as músicas no formato `mp3` para uso no serviço de música de espera;
- d) `/var/spool/asterisk/voicemail`: onde serão armazenadas as mensagens do serviço de correio de voz.

A partir de agora, serão apresentadas tabelas contendo partes dos arquivos de configuração. Tabelas e suas numerações foram incluídas apenas para facilitar as devidas explicações e portanto deverão ser ignoradas ao serem adicionadas nos arquivos de configuração. Os exemplos completos de todos arquivos de configurações alterados em laboratório são mostrados nos apêndices desse trabalho com exceção dos arquivos que ficaram inalterados a partir da instalação do Asterisk

### 8.6.3 Criação de usuários para uso do x-lite

Para que um usuário possa usar o Asterisk por meio de um *soft phone* x-lite, deve-se cadastrá-lo ao final do arquivo sip.conf seguindo o exemplo:

<i>[013402]</i>
<i>type=friend</i> <i>secret=222</i> <i>host=dynamic</i> <i>context=from-sip</i> <i>allow=all</i> <i>qualify=no</i> <i>canreinvite=no</i>

Tabela 8 – Exemplo de cadastro usuário SIP

Nesse caso o usuário terá como número de telefone 013402, senha 222 e acessará apenas o contexto *from-sip* do arquivo extensions.conf. Quem tem acesso a esse contexto, poderá ligar somente para as extensões que ali estiverem cadastradas.

### 8.6.4 Plano de discagem para usuários de *x-lite*

Para que o usuário da tabela 8 possa ser chamado, deve haver uma extensão que referencie ele no contexto *from-sip* do arquivo *extensions.conf*. A sintaxe de uma extensão é: *exten => <número>, <seqüência>, <comando>*. Um exemplo simples para isso seria: *exten => 013402,1,Dial(SIP/013402)*, onde ao chegar uma chamada para o número 013402, o Asterisk executa o comando *Dial* chamando, por meio do protocolo SIP, o próprio número.

	<i>[from-sip]</i>
1	<i>exten =&gt; 013402,1,SetMusicOnHold(default)</i>
2	<i>exten =&gt; 013402,2,Dial(SIP/\${EXTEN},20)</i>
3	<i>exten =&gt; 013402,3,Voicemail(u013402)</i>
4	<i>exten =&gt; 013402,103,Voicemail(b013402)</i>

Tabela 9 – Exemplo de extensão

Um exemplo mais sofisticado para esse mesmo usuário seria como na tabela 9. Que tem o seguinte resultado:

- a) **Linha 1:** uso de música em espera é habilitado (ver Sessão 8.6.6);
- b) **Linha 2:** o ramal, que é vinculado a variável *EXTEN*, é chamado por 20 segundos;
- c) **Linha 3:** caso não seja atendido, a origem ouvirá uma mensagem do correio de voz de usuário indisponível (ver Sessão 8.6.7);
- d) **Linha 4:** caso esteja ocupado, a origem ouvirá uma mensagem do correio de voz de usuário ocupado. Note que a seqüência nesse ponto é 103, pois sempre que o comando *Dial* recebe a resposta de ocupado, ele procura a seqüência *n+101*.

Com o que já se viu até aqui, é possível cadastrar dois usuários SIP utilizando x-lite e planos de discagem para eles dentro de um mesmo contexto [from-sip] fazendo ligações entre micros. A Figura 28 apresenta esta situação.

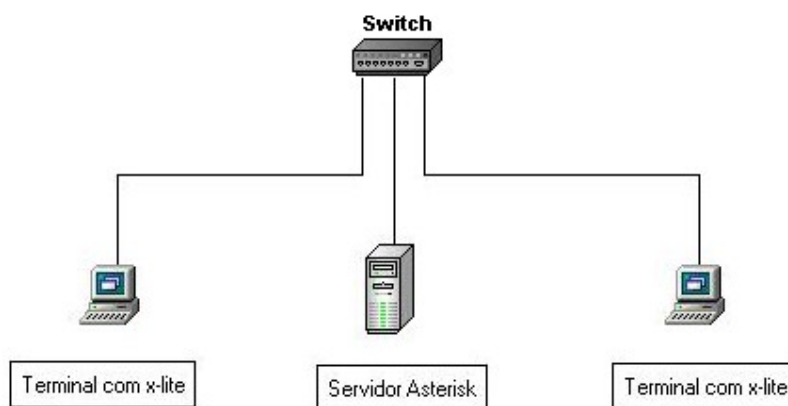


Figura 28. Asterisk interligando usuários x-lite

Pode-se criar um contexto com números que somente um certo grupo de pessoas podem acessar, como por exemplo ligações para fora de empresa pelo x-lite (desde que o ramal usado na placa x100p dê esse acesso). Nos testes realizados, as seguintes configurações foram feitas para resolver essa questão: os usuários com acesso a ligações externas tiveram em seu cadastro no arquivo *sip.conf* o campo *context=elite* e portanto poderiam acessar esse contexto no *extensions.conf* que foi criado conforme tabela 10.

1	[elite]
2	include => from-sip
3	exten => _0XXXXXXX.,1,Dial(Zap/1/{EXTEN})

Tabela 10 – Exemplo de contexto reservado

Explicando o contexto elite:

- a) **Linha 1:** nome do contexto;
- b) **Linha 2:** inclusão do contexto *from-sip* para esses usuários, permitindo a eles o uso de todas as extensões lá cadastradas;
- c) **Linha 3:** quando o usuário liga para um número que inicia com zero (representado por `_0`), seguido de pelo menos sete números (representados por `XXXXXXXX`) e mais qualquer quantidade deles (representado por um ponto), a chamada é encaminhada para o PABX por meio do canal 1 da placa x100p.

### 8.6.5 Plano de discagem para telefones fixos

A placas de telefonia x100p clones utilizadas nos testes, possibilitaram chamadas entre x-lites e telefones fixos, telefones fixos e x-lite e entre telefones fixos.

Para uma chamada vinda de um x-lite ser direcionada para um telefone convencional, um canal Zap deve ser acionado. Em laboratório, adicionou-se a linha `exten => _34XX,1,Dial(Zap/1/${EXTEN},20)` no contexto `[from-sip]` do arquivo `extensions.conf`, onde ao discar para qualquer ramal que comece com 34 e tenha mais duas casas, o canal Zap 1 chama esse ramal do PABX por meio do ramal que foi conectado fisicamente à placa .

Para que ramais fixos possam ligar para soft phones, uma entrada `start (s)`, deve ser configurada no contexto `[default]` do arquivo `extensions.conf`. Quando qualquer ramal fixo ligar para o ramal que esteja conectado na placa de telefonia, o Asterisk receberá a chamada e poderá encaminhar para o x-lite desejado.

1	exten => s,1,Answer
2	exten => s,n,Set(TIMEOUT(response)=10)
3	exten => s,n,Set(TIMEOUT(digit)=5)
4	exten => s,n(restart),BackGround(atendimento)
5	exten => s,n,WaitExten
6	exten => 1,1,Goto(from-sip,013463)
7	exten => 2,1,Goto(from-sip,013403)
8	exten => i,1,Playback(invalid)
9	exten => i,2,Goto(default,s,1)
10	exten => t,1,Hangup

Tabela 11 – Exemplo de Atendimento automático

O que acontece quando um telefone fixo disca para o ramal conectado a placa clone:

- a) **Linha 1:** o asterisk aceita a chamada;
- b) **Linha 2:** determina que o tempo de espera para a escolha do ramal x-lite é de 10 segundos e encaminha para a próxima linha (n=next);
- c) **Linha 3:** determina que o tempo para espera após digitar o ramal x-lite desejado é de 5 segundos;
- d) **Linha 4:** uma mensagem “atendimento” (para gravar sua própria mensagem ver Sessão 8.6.10) é tocada para dar instruções tipo: “disque 1 para João, 2 para Maria..., mas permite que a qualquer momento digite-se o número desejado;
- e) **Linha 5:** o Asterisk fica esperando o usuário discar a extensão desejada, que no exemplo acima pode ser 1 ou 2;
- f) **Linha 6 e 7:** de acordo com a escolha feita, o comando Goto é usado para direcionar para as extensões 013402 e 013403 do contexto from-sip;
- g) **Linha 8 e 9:** se o número discado for inválido (i) toca a mensagem *invalid* e volta (Goto) para a primeira linha;

- h) **Linha 10:** se nada for discado no tempo estabelecido de 10 segundos ( $t=\text{timeout}$ ), a ligação é finalizada.

A Figura 29 mostra esta situação.

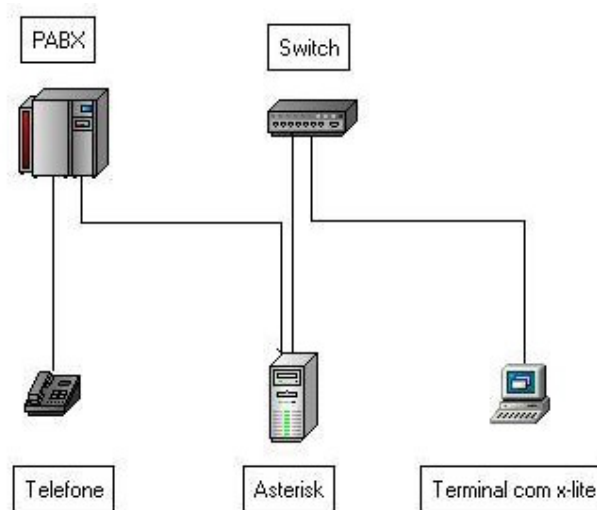


Figura 29. Asterisk interligando telefone fixo e x-lite

Nos testes em laboratório foram realizadas ligações entre telefones fixos de dois pontos remotos mas interligados por uma rede de dados e portanto, gratuitas. Para que isso fosse possível, a instalação no ponto remoto do segundo servidor Asterisk foi necessária sendo que ele também possuía uma placa x100p ao qual foi ligada ao PABX daquele ambiente.

Nos testes realizados, para ligar de um telefone fixo do laboratório A para um telefone fixo do laboratório B, o texto `exten=> _2XX,1,Dial(IAX2/iaxremoto@192.168.10.50/${EXTEN}@from-sip)` foi adicionado entre as linhas 7 e 8 da tabela 11 do local "A". O texto `exten => _2XX,1,Dial(Zap/1/${EXTEN})` também foi inserido no contexto [from-sip] do extensions.conf do local "B", onde também criou-se o usuário iaxremoto no arquivo iax.conf de acordo com a tabela 12.

[iaxremoto]
type=friend
context=from-sip

Tabela 12 – Exemplo de cadastro usuário IAX

A partir dessa configuração efetuada, estando no local A, ao ligar para o ramal local que está conectado á placa x100p, e escolher qualquer ramal que comece com o número 2 e tenha mais dois dígitos, a chamada será direcionado para o Asterisk remoto com IP 192.168.10.50, contexto [from-sip] do extensions.conf , (Figura 30).

Baseado nessas informações, pode-se também configurar para que se possa fazer ligações VOIP entre telefones fixos do local B para A.

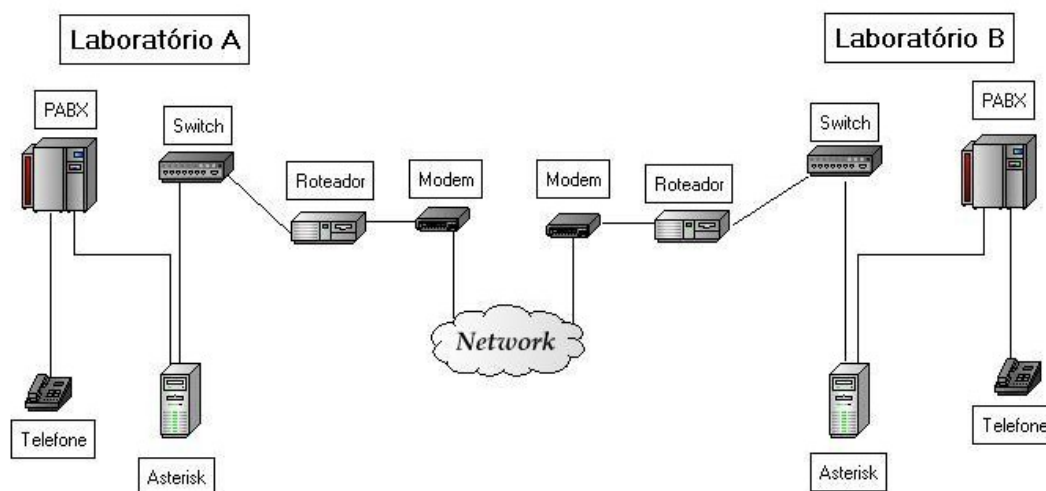


Figura 30. Asterisk interligando telefones fixos

### 8.6.6 Música de espera

Enquanto a maioria dos PABXS precisa estar conectado a um aparelho de som externo para prover música de espera aos seus clientes, o Asterisk dispõe desse recurso usando arquivos no formato mp3. Para configurá-lo, deve-se definir quais os

usuários disponibilizarão música em espera com o comando *SetMusicOnHold* (<contexto>) inserido no arquivo *extensions.conf* conforme mostrado na tabela 9. Deve-se também configurar o arquivo *musiconhold.conf* conforme tabela 13 e colocar as músicas mp3 desejadas no diretório descrito em *directory*.

[default]
mode=mp3 directory=/var/lib/asterisk/mohmp3

Tabela 13 – Configuração do *musiconhold.conf*

Por último, instale o pacote *mpg123-0.59r.tar.gz* (que pode ser obtido em [www.mpg123.de/mpg123/mpg123-0.59r.tar.gz](http://www.mpg123.de/mpg123/mpg123-0.59r.tar.gz) e copiado para */usr/src/asterisk*) como usuário *root* com os comandos da tabela 14. O programa *mpg123* é um player de mp3 utilizado pelo Asterisk para tocar as músicas de espera.

1	<code>cd /usr/src/asterisk</code>
2	<code>tar xzvf mpg123-0.59r.tar.gz</code>
3	<code>cd mpg123-0.59r</code>
4	<code>make linux</code>
5	<code>make install</code>

Tabela 14 – Instalação do *mpg123*

Para testar a música de espera para um usuário que discou para um x-lite, o destinatário poderá pressionar o número da linha que recebeu a ligação (o x-lite tem três linhas para receber ligações).

### 8.6.7 Correio de voz (voicemail)

O correio de voz permite que quando um usuário receba uma chamada e não possa atender ou esteja ocupado, uma mensagem seja gravada pelo usuário que chamou para posterior verificação do receptor. Pode-se definir quais os usuários poderão ter correio de voz pelo comando *Voicemail* no arquivo *extensions.conf*, conforme já foi visto na tabela 9. Além disso o usuário deve estar cadastrado no arquivo *voicemail.conf* conforme sintaxe: <caixaPostal>: <senha>, <nome completo>, <email>. Exemplo: *013402 =>123456789,Fulano de tal, fulano@dominio.com.br*.

Caso se tenha necessidade que o usuário receba um e-mail de aviso quando ele receba uma nova mensagem em seu correio de voz do Asterisk é necessário que o pacote *sendmail* do Asterisk esteja instalado (ver Sessão 8.4.1) e esteja em execução. Para isso entre como *root* no Linux e digite os comandos *cds* e *./sendmail start*. Para que esse serviço seja sempre iniciado com o Linux pode-se executar o utilitário *ntsysv* e marcar o *sendmail*. Caso queira alterar o conteúdo do e-mail, o campo *emailbody* do arquivo *voicemail.conf* pode ser alterado como no apêndice G.

Para acesso ao correio de voz, deve-se configurar uma extensão no contexto *from-sip* do arquivo *extensions.conf*, seguindo o modelo *exten => 9000,1,VoicemailMain()* permitindo que o usuário ligue para esse número, no caso 9000, e manipule suas mensagens recebidas, que são gravadas no diretório */var/spool/asterisk/voicemail/default/*, ouvindo-as e apagando-as e até mesmo podendo gravar as suas próprias mensagens de ocupado e indisponível.

### 8.6.8 Sala de conferência

Sala de conferência do Asterisk é um recurso que possibilita que dois ou mais usuários consigam conversar em conjunto por meio do x-lite ou até mesmo por telefones fixos. Para isso deve-se criar uma extensão dentro do contexto *from-sip* do arquivo *extensions.conf* com uma linha semelhante a *exten => 8600,1,Meetme()*, onde os usuários que ligarem para o número 8600 terão a opção de digitar a sala de conferência no qual querem participar. Para que a sala seja válida, uma entrada do tipo *conf => 1* (onde 1 é uma sala permitida) deve ser cadastrada no contexto *rooms* do arquivo *meetme.conf*.

### 8.6.9 Macros

Um macro é um conjunto de comandos que podem ser criados para facilitar a adição de novos ramais no *extensions.conf*. Nos testes realizados, a macro da tabela 15 foi criada. Assim a cada novo número de ramal criado (um novo usuário de x-lite), apenas uma entrada como *exten => 013402,1,Macro(empresa,SIP/\${EXTEN})* precisaria ser criada no *extensions.conf* (onde 013402 é o novo ramal).

	[macro-empresa]
1	exten => s,1,SetMusicOnHold(default)
2	exten => s,2,Dial(\${ARG1},30)
3	exten => s,3,Voicemail(u\${MACRO_EXTEN})
4	exten => s,103,Voicemail(b\${MACRO_EXTEN})

Tabela 15 – Exemplo de macro

Seguindo o exemplo, quando a extensão 013402 é chamada, o comando *Macro* é executado passando como parâmetro o argumento *SIP/\${EXTEN}* para o macro *empresa* ao qual deve estar cadastrado acima (e separado) do contexto *from-sip* do arquivo *extensions.conf*, já que a própria macro é um contexto. Todos os comandos presentes nesse macro serão executados para o ramal passado como parâmetro.

### 8.6.10 Gravação de mensagens de voz

As mensagens de voz geradas na instalação do Asterisk para uso, como por exemplo, na indicação de escolha inválida de uma opção ou das instruções para uso na verificação do correio de voz, são em inglês. Um pacote contendo essas mesmas mensagens porém em português podem ser encontradas em [http://www.portalasterisk.com.br/download/sounds\\_pt\\_BR.zip](http://www.portalasterisk.com.br/download/sounds_pt_BR.zip). Deve-se usar o seguinte comando no usuário *root* para instalá-lo: `unzip sounds_pt_BR.zip -d /var/lib/asterisk`.

Pode-se ainda gravar suas próprias mensagens para o Asterisk executar automaticamente como por exemplo num atendimento automático (ver Sessão 8.6.5). Para isso, deve-ser incluir um texto como o da tabela 16 no contexto *from-sip* do arquivo *extensions.conf*.

1	<code>exten =&gt; 100,1,Record(/tmp/asterisk-recording:gsm)</code>
2	<code>exten =&gt; 100,2,Wait(2)</code>
3	<code>exten =&gt; 100,3,Playback(/tmp/asterisk-recording)</code>
4	<code>exten =&gt; 100,4,Hangup</code>

Tabela 16 – Exemplo de extensão para gravação de mensagens

Ao ligar de um x-lite para o número 100:

- a) **Linha 1:** após um *beep*, o Asterisk executa o comando *Record* que grava tudo que o usuário falar até pressionar a tecla #;
- b) **Linha 2:** há uma pausa de 2 segundos;
- c) **Linha 3:** o som gravado é reproduzido.
- d) **Linha 4:** a ligação é encerrada.

Após esse procedimento, entre com usuário *root* e copie o arquivo gravado para o diretório *sounds* com o comando `cp /tmp/asterisk-recording.gsm /var/lib/asterisk/sounds/arquivo.gsm`, sendo *arquivo* o nome desejado. Para usá-lo, coloque na extensão desejada o comando *Playback(arquivo)*.

Outros recursos testados em laboratório, foram o monitoramento de chamadas e o uso do Asterisk como servidor de fax. Ambos descritos respectivamente nos apêndices *A* e *B*.

## 8.7 RESULTADOS OBTIDOS

A pesquisa consistiu-se na instalação de um sistema VOIP de baixo custo denominado Asterisk que interligou várias filiais de uma empresa ao qual usou-se como laboratório, servindo para análises de vantagens e desvantagens no uso de um sistema livre em um ambiente corporativo. Pode-se realizar vários testes de desempenho e verificações de funcionalidade desse sistema.

A comunicação de voz sobre a rede de dados mostrou-se viável e até a finalização desse trabalho ainda estava em uso no ambiente de produção, apesar da limitação de apenas uma ligação por vez entre telefones fixos do laboratório A para o B ser possível (ou vice-versa). Essa limitação deu-se devido ao uso de placas não

profissionais (clone). Outra limitação foi a má qualidade da voz nos casos de disputa de tráfego dessa com os dados. Os laboratórios foram interligados com um link Frame-Relay de baixa velocidade (128 Kb/s) e além disso não dispunha de Qualidade de Serviço (QoS), mesmo assim quando havia pouco ou nenhum tráfego de dados, a qualidade da voz era muito boa.

No período em que estava sendo analisado, ou seja, entre abril e junho, o Asterisk proporcionou uma considerável redução do uso das linhas telefônicas convencionais entre os laboratórios, principalmente entre *A* e *B*, locais em que se tornou possível o uso de VOIP por meio de telefones fixos. Isso proveu uma pequena redução no total de gastos com telefonia para a empresa no período e ainda provou que vale a pena o uso da tecnologia, mesmo que ainda tenha alguns obstáculos a vencer.

As ligações entre os laboratórios *A* e *B* podiam ser efetuadas diretamente nos setores dispensando a intermediação da telefonista.

O uso desse sistema proporcionou a utilização de novos recursos como: correio de voz, gravação das ligações em arquivos wav, a possibilidade de enviar fax direcionando-o para um endereço de e-mail e possuir o controle do seu sistema de telefonia: ao invés de esperar alguém configurar o PABX proprietário, o usuário poderá fazê-lo. Por outro lado, o gerenciamento do Asterisk exige conhecimento para sua manutenção, o que poderá implicar em custos para treinamento desses gerenciadores.

Outra vantagem no uso desse sistema, foi baixo custo. As duas máquinas cedidas para uso, citadas na Sessão 8.2.2, eram máquinas modestas com valor de mercado conforme citado na Tabela 17 e as duas placas utilizadas eram placas de modem comuns, de marca Intel e modelo 537.

Os pontos negativos verificados no uso do Asterisk foram: a complexidade encontrada para a configuração do sistema, agravada pelo uso de interface não-gráfica

(toda a configuração é feita pela edição de arquivos textos) e a escassez de manuais e livros em português. Cada empresa tem suas próprias particularidades e estas funcionalidades desejadas poderiam não estar detalhadas nos materiais utilizados.

## 8.8 CUSTOS RELACIONADOS

Estimativa de custos envolvidos para implementação dessa solução.

<b>Equipamento/Serviço</b>	<b>R\$</b>
1 Micro Pentium 4 1.7 GHz	1.500,00
1 Micro Celeron 2.3GHz	1.500,00
2 Placas Modem Intel 537	120,00
Sistema Operacional Conectiva Linux 10	0,00
Software Asterisk	0,00
Software Soft Phone X-Lite	0,00
Capacitação - 20 *Horas	2.000,00
Mão obra para instalação - 10 *Horas	1.000,00
<b>TOTAL</b>	<b>6.120,00</b>

Tabela 17 – Custos envolvidos

\* Custo estimado de um profissional da área.

Obs: não foram considerados custos referentes estrutura de redes, ramais telefônicos disponibilizados e estrutura elétrica (inclusive No-Break) pois o laboratório já possuía estrutura necessária.

## CONCLUSÃO

O transporte de voz sobre rede de dados tem induzido várias empresas a adotarem essa tecnologia, influenciadas pela possibilidade de redução de custos com ligações telefônicas e de novos recursos agregados.

Neste trabalho explorou-se o grande avanço na tecnologia das comunicações, que é a convergência de vários tipos de dados em uma única rede, a rede IP. Entretanto mesmo sendo uma nova tecnologia, provou-se que é possível diminuir boa parte dos gastos usando software livre.

Os objetivos esperados desse trabalho foram atingidos, pois necessitou-se compreender os fundamentos teóricos referentes aos conceitos da voz, tecnologia VOIP, protocolos, padrões e qualidade de serviço, além de arquiteturas de rede. A implantação de um sistema para voz sobre IP de baixo custo foi bem sucedida em um ambiente real onde pode se verificar os benefícios e as dificuldades dessa nova tecnologia.

O Asterisk mostrou-se útil para interligar departamentos e filiais de uma mesma empresa por meio de *soft phones* e telefones fixos proporcionando ligações sem custo, entretanto ligações entre telefones fixos só são possíveis com uso de um servidor Asterisk em cada filial.

Diversos outros tipos de utilização do Asterisk podem ser possíveis, mas como seria difícil abordar tudo isso em uma única pesquisa, indica-se para futuros trabalhos: a integração com outras empresas que também usam Asterisk, disponibilizando ligações também gratuitas. Outra sugestão seria a integração com operadoras telefônicas, disponibilizando ligações VOIP para qualquer parte do mundo a preços mais acessíveis devido ao fato da voz trafegar numa rede de dados entre as operadoras. Por final sugere-se também como trabalho futuro utilizar o gerenciador do

asterisk por uma interface gráfica, que possibilite toda configuração por meio de páginas web. Uma solução dessa já existente é conhecida como ASTBILL.

O estudo efetuado proporcionou aprimoramento de conhecimentos importantes para o crescimento pessoal ampliando desta forma a área de atuação profissional. A vantagem de se trabalhar na área de tecnologia da empresa no qual serviu como estudo de caso foi muito importante, obtendo o apoio necessário para concluir esta pesquisa.

## REFERÊNCIAS

- A COMUNIDADE BRASILEIRA DO PBX OPEN SOURCE. Disponível em: <[www.asteriskbrasil.org](http://www.asteriskbrasil.org)>. Acessado em 16/04/06.
- BESSA, Liesle. **Redução de custos impulsiona adoção de Voip no mercado brasileiro. Operadoras de telefonia devem mudar estratégias para recuperar receita.** 2005. Disponível em: <[http://www.idcbrasil.com.br/brasil/telas/pagina.asp?id\\_area=3&n=181](http://www.idcbrasil.com.br/brasil/telas/pagina.asp?id_area=3&n=181)>. Acesso em: 05/08/2005.
- BORDIGNON, Márcio R. **Vídeo Conferência.** conceitos, tecnologias e uso. Rio de Janeiro: Book Express,2001.
- CALVACHE, Humberto Maxclioff. **Protótipo de um Sistema de Telefonia IP Criptografado Baseado no Padrão SIP.** 2005. Trabalho de Conclusão de Curso (Engenheiro Elétrico) Universidade Regional de Blumenau.
- COMER, Douglas; STEVENS, David L. **Interligação em rede com TCP/IP.** Rio de Janeiro: Campus, 1999.
- DERFLER JÚNIOR; FREED, Les. Frank. **Como funcionam as redes III.** São Paulo: Quark, 1998.
- FUSCO, Camila. **Indiana Seguros Investe em VOIP.** Computerworld. São Paulo. Jun. 2005.
- GONÇALVES, Flávio Eduardo de Andrade. **Asterisk Guia de Configuração: Como construir e configurar um PABX com Software Livre.** São Paulo: Voffice, 2005.
- GRECCO, Fernando. **VOIP e Telefonia IP: Paralelas e Inevitáveis (2004).** Disponível em: <[http://www.timaster.com.br/revista/artigos/main\\_artigo.asp?codigo=934](http://www.timaster.com.br/revista/artigos/main_artigo.asp?codigo=934)> Acesso em: 12/11/2005.
- GRIZZO, Erico. **Internet: o que é, o que oferece, como conectar-se.** São Paulo: Ática, 2002.
- HALL, Jon. **Open-Source VOIP 'will be bigger than Linux'(2005).** Disponível em: <<http://news.zdnet.co.uk/communications/networks/0,39020345,39169076,00.htm>> Acesso em: 20/11/2005.

HELD, Gilbert. **Comunicação de dados**. 6.ed Rio de Janeiro: Campus, 1999.

INFO ABRIL. Seção de Downloads. São Paulo, 2005. Disponível em: <info.abril.com.br/download>. Acessado em 16/11/05.

KUROSE, James F.; ROSS, Keith W. **Rede de Computadores e a Internet: Uma abordagem top-down**. São Paulo: Pearson Addilson Wesley, 2005.

LIMA JÚNIOR, Almir Wirth. **Telecomunicações multimídia**. Rio de Janeiro: Book Express, 2001.

MACHADO, Carlos. Papo de Negócios pela Internet. **Revista Info Exame**. São Paulo, ano 19, n 230, Maio. 2004.

MAGRO, Julio César. **Estudo da Qualidade de Voz em Redes IP**. 2005. Dissertação de Mestrado (Mestre em Engenharia Elétrica) Faculdade de Engenharia Elétrica e de Computação - FEEC/UNICAMP.

MAHLER, Paul. **VOIP Telephony with Asterisk**. San Francisco: Signate, 2005.

MURHAMMER, Martin et al. **TCP/IP: tutorial e técnico**. São Paulo: Makron Books, 2000.

NÓBREGA, Obionor at al. Um Algoritmo Adaptativo de Transmissão para Serviços de Voz Sobre Redes IP in Simpósio Brasileiro de Redes De Computadores, 19, 2001, Recife. **Anais...** Florianópolis: UFSC. 2001. p. 559-573.

PETERSON, Larry L; DAVIE, Bruce S. **Redes de Computadores: uma abordagem de sistemas**. Rio de Janeiro: Elsevier, 2004.

RACHID, Euler Mendes. **Cancelamento de ECO em Telefonia IP**. 2004. Dissertação de Mestrado (Mestre em Engenharia Elétrica) Faculdade de Engenharia Elétrica e de Computação - FEEC/UNICAMP.

REGGIANI, Lucia. **VOIP a próxima chamada**. **Revista Info Exame**. São Paulo, ano 19, n 221, Agosto. 2004.

REXFORD, Jennifer; KRISHNAMURTH, Balachander. **Redes para a web**. Rio de Janeiro: Campus, 2001.

RIBEIRO, Bruno F. M.; AGUIAR, Paulo; MARCONDES, Cesar. Implementação de Gateway de Sinalização entre Protocolos de Telefonia IP SIP/H.323 in Simpósio Brasileiro de Redes De Computadores, 19, 2001, Rio De Janeiro. **Anais...** Florianópolis: UFSC. 2001. p. 574-589.

SILVEIRA, Jorge Luiz D. **Comunicação de Dados e Sistemas de Teleprocessamento**. São Paulo: Makron Books, 1991.

SIPFOUNDRY. Comunidade SIP Internacional. 2004. Disponível em: <[www.sipfoundry.org](http://www.sipfoundry.org)>. Acessado em 19/11/05.

TANENBAUM, Andrew S. **Redes de computadores**. 4.ed. Rio de Janeiro: Ed. Campus, 1997.

WIECZOREK, Stefan Victor. **LIBERTAS QUAE SERA TAMEN. COMPUTERWORLD**. São Paulo. Jun. 2005.

YURI, Flávia. Tijolo, Cimento e Voz Sobre IP. **Revista Info Exame**. São Paulo, ano 19, n 216, Março. 2004.

YURI, Flávia. A Voz Pela Rede. **Revista Info Exame**. São Paulo, ano 19, n 217, Abril. 2004.

**BIBLIOGRAFIA RECOMENDADA**

BIONDI, Rogério; ESPINOSA, Ivan. **Comunicação de dados para microcomputadores**. Rio de Janeiro: Brasport, 1996.

BONGIOLO, Marcelo. **Aplicação Experimental para Comunicação de Voz Sobre IP**. 2004. TCC (Bacharel em Ciência da Computação) Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, 2004.

CASAD, Joe. **Aprenda em 24 horas TCP/IP**. Rio de Janeiro: Campus 1999.

HAYDEN, Matt. **Aprenda em 24 Horas Redes**. 2.ed. Rio de Janeiro: Campus, 1999.

SOUSA, Lindeberg Barros de. **Redes de computadores: dados, voz e imagem**. 4.ed. Rio de Janeiro: Érica, 2001.

TOLEDO, Adalton Pereira de. **Redes de Acesso em Telecomunicações**. São Paulo: Makron Books. 2001.

## APÊNDICE A – Escuta e monitoramento de ligações em andamento

Um outro recurso muito importante para uso em prol de uma empresa e que o Asterisk fornece é a escuta e a gravação de conversas que passam por ele, vindas do x-lite ou de telefones fixos que usam essa solução VOIP.

Para escutar uma chamada que esteja em andamento de um x-lite específico, inclui-se a linha `exten=>_5.,1,ChanSpy(SIP/${EXTEN:1})` em um contexto restrito (como o *elite* descrito na tabela 10) do arquivo `extensions.conf` e com um x-lite que tenha acesso a esse contexto, digita-se o número 5 seguido do ramal desejado.

Da mesma forma, para escutar uma chamada em andamento em um telefone fixo que esteja usando a placa `x100p` clone, inclui-se a linha `exten=>7777,1,ZapBarge(1)` e digite 7777 no x-lite para escutar.

É possível gravar essas chamadas automaticamente com o comando *monitor*. Usando o exemplo da tabela 17, o Asterisk irá gravar as chamadas atendidas pelos ramais iniciados pelo número 34.

1	<code>exten =&gt; _34XX,1,SetVar(GRAVACAO=\${EXTEN}-\${TIMESTAMP})</code>
2	<code>exten =&gt; _34XX,2,Monitor(wav,\${GRAVACAO},m)</code>
3	<code>exten =&gt; _34XX,3,Dial(Zap/1/\${EXTEN},20)</code>

Tabela 18 – Exemplo de extensão para monitoramento de chamadas

- a) **Linha 1:** quando qualquer ramal iniciado pelo número 34 receber uma ligação, uma variável com nome `GRAVACAO` é criada com valor composto pelo número do ramal chamado seguido de data e hora atual.  
Exemplo 3410-20060511-163122;
- b) **Linha 2:** um arquivo do tipo `wav` com nome igual ao valor da variável é criado no diretório `/var/spool/asterisk/monitor`, onde a conversa será

armazenada. Se a opção *m* dessa linha for retirada, um arquivo *in* e outro *out* serão criados contendo separadamente as vozes da origem e destino;

c) **Linha 3:** o ramal é chamado e a gravação será armazenada.

Os recursos de escuta e gravação foram testados mas não aplicados pois não conversou-se com a direção da empresa sobre questões éticas e legais em que isso implicaria.

## APÊNDICE B – Recebimento e envio de Fax com redirecionamento para e-mail

O Asterisk possibilita o recebimento e o envio de fax com redirecionamento para e-mail, ou seja: na recepção de um fax, um arquivo com extensão tif é criado no servidor e enviado como anexo para um e-mail previamente cadastrado, ou ainda: ao enviar um fax (por um aparelho de fac-símile normal) é possível que ele seja transformado em um e-mail e enviado ao seu destinatário ao invés de ser impresso por outro aparelho fac-símile.

Nos testes realizados, o texto da tabela 18 foi adicionado no contexto default do arquivo extensions.conf do laboratório B possibilitando o envio de fax para o laboratório A sendo que ele era transformado em um e-mail e enviado ao destinatário.

<code>exten=&gt;9,1,WaitExten</code>
<code>exten=&gt;27,1,rxfax(/tmp/fax.tif)</code>
<code>Exten=&gt;27,2,system(mutt -a /tmp/fax.tif -s "Vc recebeu um Fax"</code> <code>marcioni.serafim@giassi.com.br&lt;.)</code>

Tabela 19 – Exemplo de extensão para recebimento ou envio de fax

No exemplo da tabela 18, ao ligar para o ramal conectado no Asterisk por meio de um aparelho de fac-símile e escolher a opção 9 seguida do número escolhido (no exemplo da tabela, o número 27), o comando rxfax é acionado recebendo o documento lido e transformando-o no arquivo fax.tif que é gravado no servidor. Após isso, o comando system executa o aplicativo mutt enviando um e-mail com o anexo para o usuário ali definido.

Para que o uso desse serviço seja possível, necessita-se da instalação do aplicativo rxfax pois nativamente ele não faz parte do Asterisk. Para isso, deve-se fazer

os downloads dos arquivos `app_rxfax.c`, `app_txfax.c`, `apps_makefile.patch` e `spandsp-0.0.2pre21.tar.gz` em <http://soft-switch.org/downloads/spandsp/spandsp-0.0.2pre21d/>, copiar os três primeiros para o diretório `/usr/src/asterisk/asterisk-1.2.0/apps` (código fonte do asterisk) e o último para `/usr/src/asterisk`, entrar como usuário *root* e executar os comandos das tabelas 19 e 20.

1	<code>cd /usr/src/asterisk</code>
2	<code>tar xfvz spandsp-0.0.2pre21.tar.gz</code>
3	<code>cd /usr/src/spandsp-0.0.2</code>
4	<code>./configure --prefix=/usr</code>
5	<code>make</code>
6	<code>make install</code>

Tabela 20 – Instalação da biblioteca spandsp

1	<code>cd /usr/src/asterisk/asterisk-1.2.0/apps</code>
2	<code>mv apps_makefile.patch Makefile.pach</code>
3	<code>apt-get install patch</code>
4	<code>patch &lt; Makefile.patch</code>
5	<code>cd /usr/src/asterisk/asterisk-1.2.0</code>
6	<code>Make</code>
7	<code>make install</code>

Tabela 21 – Re-compilando o Asterisk

É necessário ressaltar que para a criação dos arquivos com extensão `tif` o Linux necessita da biblioteca `libtiff` (ver Sessão 8.4.1) e que a reinstalação do arquivo `sounds_pt_BR.zip` pode ser necessária (ver Sessão 8.6.10).

## APÊNDICE C – Exemplo dos Arquivos zaptel.conf e zapata.conf

;Obs: as linhas iniciadas por ; (ponto e vírgula) são usadas para ;fazer comentários portanto não são tratadas pelo Asterisk.  
;ATENÇÃO: nunca usar em início de linha um sinal de menos com um ;comentário (;-) pois ocasiona conflito.

; arquivo /etc/zaptel.conf para placa x100p

**fxsks=1**

**loadzone = us**

**defaultzone=us**

; arquivo /etc/asterisk/zapata.conf

**[channels]**

**language=en**

**usecallerid=yes**

**hidecallerid=no**

**callerid=asreceived**

**callwaiting=no**

**usecallingpres=yes**

**callwaitingcallerid=yes**

**threewaycalling=yes**

**transfer=yes**

**cancallforward=yes**

**callreturn=yes**

**echocancel=yes**

**echocancelwhenbridged=yes**

**immediate=no**

**musiconhold=default**

**group=1**

**relaxdtmf=yes**

**callerid=asreceived**

**context=default**

**busydetect=yes**

**signalling = fxs\_ks ; essencial para a placa x100p.**

**channel => 1 ; define o canal a ser usado.**

## APÊNDICE D – Exemplo de configuração arquivo *extensions.conf* do laboratório A

```

;Esse é o arquivo de configuração de extensões estáticas, usadas pelo
;modulo pbx_config. Aqui se configura todas chamadas para ramais
;portanto pode-se dizer que é principal arquivo do Asterisk
;Este arquivo de configuração é lido ao colocar o asterisk
;no ar ou ao recarregar o asterisk com:"extensions reload" ou "reload"
;no console do Asterisk
[general]

static=yes
writeprotect=no
clearglobalvars=no
priorityjumping=no

; -----
;Detecta automaticamente o fim de uma ligação. Caso use "no" um
;Hangup deve ser usado após cada plano de chamada
; -----
autofallthrough=yes

[globals]

CONSOLE=Console/dsp
IAXINFO=guest
TRUNK=Zap/g2
TRUNKMSD=1

[demo]

; -----
;Ao ligar para 500, toca um demo e executa a próxima linha na
;seqüência (n=next). A seguir, faz chamada internacional para Digium
; -----
exten => 500,1,Playback(demo-abouttotry)
exten => 500,n,Dial(IAX2/guest@misery.digium.com/s@default)

[default]

include =>demo ;Inclui o contexto demo permitindo ter acesso á ele

; -----
;Entrada inicial para telefones fixos. Ao ligar de um telefone fixo
;local para o ramal que está conectado a placa x100p clone (Modem
;Intel 537 - Ambiente MD3200), tem-se um atendimento automático
; -----
exten => s,1,Answer ; Responde a chamada
exten => s,n,Set(TIMEOUT(digit)=5) ; tempo de espera após digitar o
; ramal desejado
exten => s,n,Set(TIMEOUT(response)=10); tempo espera p/digitar ramal.

;Toca a mensagem /var/lib/asterisk/sounds/atendimento.gsm permitindo
;ao usuario digitar o ramal durante o som
exten => s,n(restart),Background (atendimento)
exten => s,n,WaitExten ;Espera por uma digitação (ramal desejado)

;Escolhas Disponiveis e avisadas no "atendimento" automático

```

```

exten => 1,1,Goto(from-sip,013463,1) ;ao discar 1,irá p/extensão"from-
                ;sip",ramal 013463,linha 1)
;Procura usuário iaxremoto no iax.conf do Asterisk 192.168.10.50
;e liga para um ramal 2XX
exten => _2XX,1,Dial(IAX2/iaxremoto@192.168.10.50/${EXTEN}@from-sip)

;Escolhas Disponíveis mas não publicadas no "atendimento" pois são
;para testes

;ligando para um telefone fixo do servidor 192.168.10.50 (usando SIP)
exten => _92XX,1,Dial(SIP/${EXTEN:1}@192.168.10.50)
;grava mensagem de voz de qualquer telefone fixo
exten => 100,1,Goto(from-sip,100,1)
;ouvir seu correio de voz por um telefone fixo
exten => 9000,1,Goto(from-sip,9000,1)
;conferência usando um telefone fixo
exten => 8600,1,Goto(from-sip,8600,1)
exten => i,1,Playback(invalid) ;Se o numero discado for invalido,
exten => i,2,Goto(default,s,1) ;volta a linha 1
exten => t,1,Hangup ; se exeder o tempo de 10 segundos, desliga

;Para ser atendido por esse menu via x-lite, deve-se ligar para 1000
exten => 1000,1,Goto(default,s,1)

[macro-empresa]

; -----
;Note que sempre deve ter a string "macro-" na frente do contexto
;macro criado e ARG1 recebe o valor do parâmetro passado na chamada da
;macro
; -----
exten => s,1,SetMusicOnHold(default) ; habilita a música de espera
exten => s,2,Dial(${ARG1},30) ; chama o ramal por 30 segundos
exten => s,3,Voicemail(u${MACRO_EXTEN});se não atender vai p/caixa voz
exten => s,103,Voicemail(b${MACRO_EXTEN});se ocupado também

[elite] ; Contexto que permite ligações para fora da empresa

include => from-sip ; inclui o contexto from-sip

; -----
; Ligações externas:discando 0 para "pegar" linha externa
; X = qualquer digito de 0 a 9
; . = um ou mais caracteres
; -----
exten => _0XXXXXXX.,1,Dial(Zap/1/${EXTEN})
exten => _0XXXXXXX.,2,Congestion

[from-sip]

include => demo ; inclui o contexto demo
include => default ; inclui o contexto default

; -----
; Para criação dos usuários SIP (número dos ramais para os x-lites)
; adotou-se o seguinte layout: 0+número da filial+número do telefone
; fixo do antigo PABX presente na sala do usuário.
; -----

```

```

; -----
; exemplo de extensão simples:
; -----
exten => 013401,1,Dial(SIP/${EXTEN},20) ; usuário Paulo

; -----
; As extensões abaixo, chamam um macro [macro-empresa](procure acima),
; ao qual é passado como parâmetro: SIP e o número chamado,
; -----
exten => 013427,1,Macro(empresa,SIP/${EXTEN}) ; usuário Nestor
exten => 013428,1,Macro(empresa,SIP/${EXTEN}) ; usuário Claudenir
exten => 013402,1,Macro(empresa,SIP/${EXTEN}) ; usuário Marcioni
exten => 013463,1,Macro(empresa,SIP/${EXTEN}) ; usuário Rodrigo
exten => 0829,1,Macro(empresa,SIP/${EXTEN}) ; usuário CPD loja 8
exten => 0424,1,Macro(empresa,SIP/${EXTEN}) ; usuário CPD loja 4
exten => 01132,1,Macro(empresa,SIP/${EXTEN}) ; usuário CPD loja 11

; -----
; exemplo extensão com música de espera e correio de voz.
; usuário CPD loja 9
; -----
exten => 0927,1,SetMusicOnHold(default) ; habilita música em espera
exten => 0927,2,Dial(SIP/${EXTEN},20) ; chama o ramal 0927
exten => 0927,3,Voicemail(u0927) ; se não atender
exten => 0927,103,Voicemail(b0927) ; ou ocupado cai na caixa voz

; -----
; O x-lite 010213, Labor. B, foi cadastrado no Asterisk instalado lá.
; Motivo: proporcionar uma configuração diferente, fazendo que fosse
; possível ligações entre dois x-lite cadastrados em dois servidores
; Asterisk diferentes.
; -----
exten => 010213,1,Dial(SIP/${EXTEN:1}@192.168.10.50)

; -----
; Para ligar de x-lite para qualquer x-lite iniciado por 013
; -----
exten => _013XXXX,1,Dial(SIP/${EXTEN:1},20)

; -----
; Tocar uma desmonstração
; -----
exten => 30,1,Playback(demo-congrats)

; -----
; Gravação de mensagens
; pressione # ou hangup para PARAR a gravação
; -----
exten => 100,1,Wait(2)
exten => 100,2,Record(/tmp/asterisk-recording:gsm)
exten => 100,3,Wait(2)
exten => 100,4,Playback(/tmp/asterisk-recording)
exten => 100,5,Wait(2)
exten => 100,6,Hangup

; -----
; Tocar músicas OnHold presentes em /var/lib/asterisk/mohmp3
; -----
exten => 2091,1,MusicOnHold,default

```

```

exten => 2091,2,Hangup

; -----
; Ouvindo seu correio de VOZ.
; (Deve-se configurar um Voicemail para o Dial do Usuário
; e também criar uma entrada para ele no voicemail.conf)
; -----
exten => 9000,1,VoicemailMain()
exten => 9000,2,Hangup

; -----
; Chamando Ramal fixo da central por meio de: x-lite ou Ramal fixo
; de um asterisk remoto.
;   _34XX = qualquer número que comece com 34 + 2 digitos
;   ${EXTEN} = é o próprio ramal discado
;   20 = tempo em segundos que vai ficar tocando
; -----
exten => _34XX,1,Dial(Zap/1/${EXTEN},20)

; -----
; Discando de x-lite para ramais Fixos que começam com 13 e 16
; -----
exten => _13XXXX,1,Dial(Zap/1/${EXTEN})
exten => _16XXXX,1,Dial(Zap/1/${EXTEN})

; -----
; Ao discar *21 um x-lite pode "puxar" para si a ligação que estava
; chamando em outro ramal do mesmo setor
; -----
exten => *21,1,Dial(Zap/1/${EXTEN})

; -----
; Sala de conferência, podendo ser entre x-lite e telefone fixo usando
; VOIP. como não foi definida a sala entre (), o usuário devera
; digitar a sala desejada (deve estar cadastrada em meetme.conf)
; -----
exten => 8600,1,Meetme()

; -----
; Ligando de um x-lite para ramais começam com *115
; -----
exten => *_115.,1,Dial(Zap/1/${EXTEN})

; -----
; Se alguém que só tem acesso a esse contexto tentar fazer uma ligação
; externa vai receber um aviso de negação
; -----
exten => _0XXXXXXX.,1,Playback(semDireitoLigacaoExterna)

```

**APÊNDICE E – Exemplo de configuração Arquivo *iax.conf* laboratório A**

```
[general]
bandwidth=low
disallow=lpc10
jitterbuffer=no
forcejitterbuffer=no
tos=lowdelay
autokill=yes

[guest]
type=user
context=default
callerid="Guest IAX User"

[iaxtel]
type=user
context=default
auth=rsa
inkeys=iaxtel

[iaxfwd]
type=user
context=default
auth=rsa
inkeys=freeworlddialup

[demo]
type=peer
username=asterisk
secret=supersecret
host=216.207.245.47

[iaxremoto]
type=friend
context=from-sip
autokill=yes
```

**APÊNDICE F – Exemplo do Arquivo *meetme.conf* laboratório A**

```
[general]
```

```
[rooms]
```

```
;exemplo para configuração completa:
```

```
;conf => numero, senha, senha para administrador
```

```
; Salas de conferências habilitadas
```

```
conf => 1
```

```
conf => 2
```

**APÊNDICE G – Exemplo configuração arquivo *musiconhold.conf***

```
[default]
mode=mp3
directory=/var/lib/asterisk/mohmp3
```

**APÊNDICE H –Exemplo de configuração do arquivo *sip.conf* laboratório A**

```
[general]
context=default
bindport=5060
port=5060
bindaddr=0.0.0.0
srvlookup=yes
rtptimeout=10
rtpholdtimeout=60
;Registrando no Asterisk remoto
;usuário central deve estar cadastrado no sip.conf remoto
register => central@192.168.10.50
[authentication]
extension 1234

[010225]          ;usuário
type=friend       ;tipo de usuário que pode fazer e receber ligações
secret=222        ;senha
host=dynamic      ;onde autenticar
context=from-sip;contexto simples permitido em extensions.conf
allow=all         ;Permite todos codecs.
qualify=no
canreinvite=no   ;tráfego de áudio passa por dentro do

[013427]
type=friend
secret=222
host=dynamic
context=elite
allow=all
qualify=no
canreinvite=no

[013428]
type=friend
secret=222
host=dynamic
context=elite
allow=all
qualify=no
canreinvite=no

[013463]
type=friend
secret=222
host=dynamic
context=elite
allow=all
qualify=no
canreinvite=no

[013401]
type=friend
secret=222
host=dynamic
context=elite
```

```
allow=all
qualify=no
canreinvite=no
```

```
[013402]
type=friend
secret=222
host=dynamic
context=elite
allow=all
qualify=no
canreinvite=no
```

```
[0927]
type=friend
secret=222
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
```

```
[0424]
type=friend
secret=222
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
```

```
[0829]
type=friend
secret=222
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
```

```
[01132]
type=friend
secret=222
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
```

```
[0134005]
type=friend
secret=222
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
```

```
;usuário p/asterisk remoto, ele devera ter uma linha register no seu
;sip.conf
```

```
[asteriskF10]
type=friend
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
insecure=very ;não solicita senha do asterisk remoto
autokill=yes
```

## APÊNDICE I – Exemplo de arquivo voicemail.conf laboratório A

```
[general]
format=wav49|gsm|wav
serveremail=asterisk
attach=no
maxmessage=180
minmessage=3
skipms=3000
maxsilence=10
silencethreshold=128
maxlogins=3
;-----
;Sessão para uso do e-mail enviado avisando do recebimento.
;-----
;Indica emissor do e-mail
fromstring=Asterisk VOIP Giassi

; Variáveis possíveis para o Assunto e corpo da mensagem
;   VM_NAME, VM_DUR, VM_MSGNUM, VM_MAILBOX, VM_CALLERID, VM_CIDNUM,
;   VM_CIDNAME, VM_DATE

;Assunto do e-mail
emailsubject= Mensagem recebida para a caixa ${VM_MAILBOX}

;Corpo do email - limite 512 caracteres
emailbody=Caro ${VM_NAME}:\n\nVocê tem uma nova mensagem de
voz.\nTamanho: ${VM_DUR} Minutos/Segundos \nNumero de sequencia:
${VM_MSGNUM}\nDestinatário: ${VM_MAILBOX}\nOriginador:
${VM_CALLERID}\nEm: ${VM_DATE}\nFavor checar. Grato!\n\n\t\t--
Asterisk\n

emaildateformat=%A, %B %d, %Y at %r
cidinternalcontexts=from-sip
sendvoicemail=yes
[DEFAULT=no]
;-----
; se envelope = yes, ao ouvir uma mensagem, antes
; será informado a hora e dia que foi recebida.
;-----
envelope=no

[zonemessages]
eastern=America/New_York|'vm-received' Q 'digits/at' IMp
central=America/Chicago|'vm-received' Q 'digits/at' IMp
central24=America/Chicago|'vm-received' q 'digits/at' H N 'hours'
military=Zulu|'vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'

[default]
maxmsg=50
013402 =>013402,Marcioni Serafim,marcioni.serafim@giassi.com.br
013427 =>013427,Nestor,nestor.zeferino@giassi.com.br
013428 =>013428,Claudenir,claudenir.serafim@giassi.com.br
013463 =>013463,RodrigoReus,rodrigo.reus@giassi.com.br
10213 =>10213,CpdF10,f10@giassi.com.br
0927 =>0927,CpdF09,f09@giassi.com.br
```

**APÊNDICE J –Exemplo de configuração arquivo *iax.conf* laboratório B**

```
[general]
bandwidth=low
disallow=lpc10
jitterbuffer=no
forcejitterbuffer=no
tos=lowdelay
autokill=yes
[guest]
type=user
context=default
callerid="Guest IAX User"
[iaxtel]
type=user
context=default
auth=rsa
inkeys=iaxtel
[iaxfwd]
type=user
context=default
auth=rsa
inkeys=freeworlddialup
[demo]
type=peer
username=asterisk
secret=supersecret
host=216.207.245.47
;-----
;Usuário para Asterisk remoto
;friend = pode fazer e receber ligações
;from-sip = limita a sessão dentro do extensions.conf
;O uso desse usuário é opcional, para acessar o extensions.conf local
;diretamente,basta suprir o usuario na linha do extensions.conf remoto
;-----
[iaxremoto]
type=friend
context=from-sip
autokill=yes
```

**APÊNDICE K –Exemplo de configuração do arquivo sip.conf laboratório B**

```
[general]
context=default
bindport=5060
port=5060
bindaddr=0.0.0.0
srvlookup=yes
rtpholdtimeout=60
;-----
; Registrando no Asterisk remoto
; asteriskF10 = deve estar cadastrado no sip.conf remoto
;-----
register => asteriskF10@192.168.0.11

[authentication]
;-----
; Usuário Local Soft Phone
;-----
[10213]
type=friend
secret=222
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
;-----
; Usuário para asterisk remoto.
; No sip.conf remoto deve ter linha register => central@192.168.10.50
;-----
[central]
type=friend
host=dynamic
context=from-sip
allow=all
qualify=no
canreinvite=no
insecure=very
```

## APÊNDICE L –Exemplo do arquivo *extensions.conf* laboratório B

```
[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjumping=no
[globals]
CONSOLE=Console/dsp
IAXINFO=guest
TRUNK=Zap/g2
TRUNKMSD=1

[default]

exten => s,1,Answer
exten => s,n,Set(TIMEOUT(digit)=5)
exten => s,n,Set(TIMEOUT(response)=10)
exten => s,n(restart),BackGround(beep2) ;
exten => s,n,WaitExten
; ----> Escolhas Disponiveis
exten => _34XX,1,Dial(IAX2/iaxremoto@192.168.0.11/${EXTEN}@from-sip)
;Digitando 9+ramal,um fax pode ser enviado diretamente para um e-mail
exten => 9,1,WaitExten
exten => 19,1,rxfax(/tmp/fax.tif) ; grava o fax com arquivo .tif
exten => 19,2,system(mutt -x -a /tmp/fax.tif -s "Vc recebeu um Fax da
F10" everaldo.dagostin@giassi.com.br<.)
exten => 25,1,rxfax(/tmp/fax.tif)
exten => 25,2,system(mutt -x -a /tmp/fax.tif -s "Vc recebeu um Fax da
F10" renato.cesino@giassi.com.br<.)
exten => 27,1,rxfax(/tmp/fax.tif)
exten => 27,2,system(mutt -x -a /tmp/fax.tif -s "Vc recebeu um Fax da
F10" marcioni.serafim@giassi.com.br<.)
exten => 28,1,rxfax(/tmp/fax.tif)
exten => 28,2,system(mutt -x -a /tmp/fax.tif -s "Vc recebeu um Fax da
F10" claudenir.serafim@giassi.com.br<.)
exten => i,1,Playback(invalid)
exten => i,2,Goto(default,s,1)
exten => t,1,Hangup

[from-sip]
include => default
exten => 10213,1,SetMusicOnHold(default) ; Ari
exten => 10213,2,Dial(SIP/${EXTEN},30)
exten => 10213,3,Voicemail(u10213)
exten => 10213,103,Voicemail(b10213)
;para acesso ao correio de Voz
;-----
exten => 9000,1,VoicemailMain()
exten => 9000,2,Hangup

;Soft Phone para Soft Phone:
;Encaminhando para outro ASTERISK usando protocolo IAX2
;iaxremoto= deve estar cadastrado no iax.conf do servidor remoto
;from-sip = contexto que esse que podera acessar
;${EXTEN:3} = desconsidera 3 dígitos do ramal digitado
;-----
exten => _0134XXX,1,Dial(IAX2/192.168.0.11/${EXTEN})
```

```
;Um soft Phone Chamando um Ramal local
;ou por Ramal fixo de um asterisk remoto
;-----
exten => _2XX,1,Dial(Zap/1/${EXTEN})

;Soft phone local chamando soft phone no asterisk remoto
;-----
exten => _0134XX,1,Dial(SIP/${EXTEN}@192.168.0.11)
exten => _08XX,1,Dial(SIP/${EXTEN}@192.168.0.11)
exten => _09XX,1,Dial(SIP/${EXTEN}@192.168.0.11)
exten => _04XX,1,Dial(SIP/${EXTEN}@192.168.0.11)
exten => _011XX,1,Dial(SIP/${EXTEN}@192.168.0.11)
```

## APÊNDICE M – DIFICULDADES ENCONTRADAS

Como não podia ser diferente, por sua complexidade e por ser uma tecnologia nova, muitas foram as dificuldades encontradas para colocar o Asterisk em funcionamento. Desde a instalação do sistema operacional até o término da configuração conforme desejava-se, muitos degraus foram alcançados.

Como no laboratório havia uma tendência para o uso do Conectiva 10, optou-se pelo seu uso. Porém verificou-se que a compilação de um novo Kernel nesse sistema era necessária, pois o suporte a módulos vem desabilitado na instalação.

Essa compilação foi uma tarefa complexa, pois apesar de diversas “receitas de bolos” disponíveis na internet, sempre há imprevistos não relatados, tornando a pesquisa mais árdua. Por isso, sugere-se também a instalação do Asterisk na distribuição Linux Fedora Core 4, já que ele dispensa a compilação de um novo Kernel.

O uso do *soft phone* x-lite como solução de voz sobre IP é de suma importância, porém como nem todos os funcionários têm acesso a computadores, é desejável o uso de telefones comuns ou especiais, os telefones IP.

Devido aos custos para aquisição de hardwares que tornassem isso possível, optou-se pelo uso de 2 placas clones que foram encontradas com muita dificuldade por se tratar de um modelo de placa que não há mais fabricação.

Após a instalação do Asterisk, a primeira funcionalidade disponibilizada para uso foi a ligações entre x-lites. Após isso, de acordo com as necessidades sentidas, configurou-se outros serviços e a cada um deles, várias foram as dificuldades encontradas para a sua configuração, como por exemplo: como ligar de um x-lite para um telefone fixo, como ligar de um telefone fixo para uma x-lite, como interligar os dois servidores Asterisk, como configurar o serviço de correio de voz.

This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.