

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

OUTROS TRABALHOS EM:
www.projetoderedes.com.br

IMPLEMENTAÇÃO DE UM PABX IP E GATEWAY VOIP COM ASTERISK

**CURITIBA
2006**

**RAFAEL MORIGGI DE SOUZA
ANDERSON EDUARDO DE LIMA
RAFAEL LEANDRO DOS SANTOS**

IMPLEMENTAÇÃO DE UM PABX IP E GATEWAY VOIP COM ASTERISK

Trabalho de Conclusão de Curso, requisito parcial para obtenção de grau de Tecnólogo, no Curso de Tecnologia em Eletrônica, da Universidade Tecnológica Federal do Paraná.

Orientador: M.Sc. Luis Carlos Vieira

**CURITIBA
2006**

**RAFAEL MORIGGI DE SOUZA
ANDERSON EDUARDO DE LIMA
RAFAEL LEANDRO DOS SANTOS**

IMPLEMENTAÇÃO DE UM PABX IP E GATEWAY VOIP COM ASTERISK

Este Trabalho de Diplomação para Tecnologia foi julgado e aprovado como requisito parcial para obtenção do título de Tecnólogo em Eletrônica pela Universidade Tecnológica Federal do Paraná.

Curitiba, 13 de dezembro de 2006.

Profa. M.Sc. Simone Massulini Acosta
Coordenadora de Curso
Departamento Acadêmico de Eletrônica.

Profa. Dra. Denise Elizabeth Hey David
Coordenadora de Trabalho de Diplomação
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. M. Sc. Vagner Gonçalves Leitão

Prof. M.Sc. Luis Carlos Vieira

Prof. Dr. Jean Carlos Cardozo da Silva

Dedicamos este trabalho às nossas famílias, amigos, professores e todas as outras pessoas que, de uma forma ou de outra, nos fizeram crescer.

AGRADECIMENTOS

Aos nossos pais, irmãos, namoradas e demais parentes, por todo o apoio em todos os sentidos;

Aos amigos e colegas, sejam do curso de graduação ou não, com os quais tivemos a oportunidade de aprender muito;

Aos pesquisadores e professores da Universidade Tecnológica Federal do Paraná, os quais não medem esforços para ajudar e tirar dúvidas de alunos;

Ao departamento acadêmico de eletrônica da UTFPR, que financiou e disponibilizou toda infra-estrutura para pesquisas e desenvolvimento do projeto;

E por fim, agradecimento em especial, ao professor Luis Carlos Vieira que nos apoiou com bravura, dedicação e acreditou em nossa competência e potencial para tornar este projeto uma realidade.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário.”

(Albert Einstein)

SUMÁRIO

LISTA DE ILUSTRAÇÕES.....	8
LISTA DE SIGLAS.....	10
RESUMO.....	11
ABSTRACT.....	12
1 INTRODUÇÃO.....	13
2 OBJETIVOS.....	16
2.1 OBJETIVO GERAL.....	16
2.2 OBJETIVOS ESPECÍFICOS.....	16
3 TELEFONIA CONVENCIONAL.....	17
3.1 INTRODUÇÃO.....	17
3.2 PABX.....	18
4 CONVERGÊNCIA DE VOZ E DADOS.....	20
4.1 PROTOCOLO TCP/IP.....	20
4.2 TECNOLOGIA VOIP.....	23
4.3 CODIFICAÇÃO DE VOZ.....	26
4.4 PROTOCOLOS VOIP.....	27
4.5 QoS.....	29
5 ASTERISK.....	32
5.1 ARQUITETURA DO ASTERISK.....	32
5.2 APLICAÇÕES DO ASTERISK.....	34
5.3 DIRETÓRIOS DE CONFIGURAÇÃO DO ASTERISK.....	35
5.4 HARDWARE MÍNIMO PARA INSTALAÇÃO DO ASTERISK.....	36
5.5 HARDWARE DE TELEFONIA PARA O ASTERISK.....	36
6 PREPARANDO O IPBX ASTERISK.....	38
6.1 INTRODUÇÃO.....	38
6.2 INSTALANDO O LINUX SUSE 10.1.....	38
6.3 CONFIGURANDO O LINUX SUSE.....	41
6.4 OBTENDO E COMPILANDO O ASTERISK.....	45
6.4.1 Instalando e Compilando Hardware TDM400P.....	47
6.5 CONFIGURANDO A PLATAFORMA ASTERISK.....	48
6.5.1 Sip.Conf.....	48

6.5.2 Iax.Conf.....	50
6.5.3 Extension.Conf.....	52
6.5.4 Zapata.Conf.....	53
6.5.5 Zaptel.Conf.....	55
6.5.6 Meetme.Conf.....	55
6.5.7 Musiconhold.Conf.....	56
6.6 SOFTPHONES.....	56
6.6.1 Softphone SIP.....	57
6.6.2 Softphone IAX.....	57
7 INICIANDO A PLATAFORMA ASTERISK.....	59
7.1 VISUALIZANDO USUÁRIOS.....	60
7.2 ENCAMINHAMENTO DAS LIGAÇÕES NO IPBX ASTERISK.....	61
7.2.1 Efetuando Ligações VoIP – VoIP.....	62
7.2.2 Efetuando Ligações PABX – PABX.....	64
7.2.3 Efetuando Ligações Entre o IPBX Asterisk e a Central PABX.....	65
7.2.3.1 Ligações Saintes do IPBX Asterisk.....	65
7.2.3.2 Ligações Entrantes no IPBX Asterisk.....	66
8 DIFICULDADES ENCONTRADAS.....	69
9 CONCLUSÕES.....	70
REFERÊNCIAS BIBLIOGRÁFICAS.....	72
Apêndice A.....	73
Apêndice B.....	76

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama Esquemático do Projeto Proposto.....	15
Figura 2 - Diagrama em Blocos de um PABX.....	18
Figura 3 - Camadas do Modelo de Referência TCP/IP.....	20
Figura 4 - Digitalização da Voz Analógica.....	24
Figura 5 - Logotipo do Asterisk.....	32
Figura 6 - Arquitetura do Asterisk.....	33
Figura 7 - Aplicações do Asterisk.....	35
Figura 8 - Módulo Wildcard XP100P.....	36
Figura 9 - Módulo Wildcard TDM400P.....	37
Figura 10 - Módulo Wildcard TE100P.....	37
Figura 11 - Logotipo do Linux Suse.....	38
Figura 12 - Tela Inicial de Instalação do Sistema Operacional Linux.....	39
Figura 13 - Yast – Centro de Controle.....	40
Figura 14 - Instalando o Pacote GCC.....	41
Figura 15 - Configurando Dispositivo de Rede.....	42
Figura 16 - Configurando Dispositivo de Rede.....	42
Figura 17 - Configurando IP	43
Figura 18 - Configurando DNS.....	43
Figura 19 - Configurando <i>Firewall</i>	44
Figura 20 - Desabilitando <i>Firewall</i>	44
Figura 21 - Diretórios do Asterisk.....	45
Figura 22 - Configuração <i>X-Lite</i>	57
Figura 23 - Configuração <i>laxComm</i>	58
Figura 24 - Acessando o Diretório do Asterisk.....	59
Figura 25 - Iniciando Hardware TDM400P e Plataforma Asterisk.....	59
Figura 26 - Inicialização do Asterisk.....	60
Figura 27 - Visualização dos Clientes Ativos.....	61
Figura 28 - Ligação entre Ramais VoIP.....	63
Figura 29 - Visualização de Chamada <i>Softphone / Softphone</i>	64
Figura 30 - Ligação entre Ramais da Central Saturno 5000.....	64
Figura 31 - Interligação entre o IPBX Asterisk e Central PABX.....	65

Figura 32 - Ligações <i>Softphone</i> / PABX.....	65
Figura 33 - Visualização de Chamada <i>softphone</i> / PABX.....	66
Figura 34 - Ligações PABX / <i>Softphone</i>	67
Figura 35 - Visualização de Chamada PABX / <i>Softphone</i>	68

LISTA DE SIGLAS

ANATEL - Agência Nacional de Telecomunicações
API - *Application Programming Interface*
ATA - *Analog Terminal Adapter*
CPA - Controle por Programa Armazenado
DNS – *Domain Name Service*
EMBRATEL - Empresa Brasileira de Telecomunicações
FTP - *File Transport Protocol*
FXO - *Foreign Exchange Office*
FXS - *Foreign Exchange Station*
HTTP - *HyperText Transfer Protocol*
Hz – Hertz
IAX - *Inter-Asterisk Exchange*
ID - *Identification*
IDC - *Internet Data Center*
IETF - *Internet Engineering Task Force*
IP - *Internet Protocol*
ITU - *International Telecommunications Union*
Kb - Kilo bits
Kbps - Kilo bits por segundo
NAT - *Network Address Translation*
PABX - *Private Automatic Branch Exchange*
PCI - *Peripheral Component Interconnect*
PCM - *Pulse Code Modulation*
QoS - *Quality of Service*
RTP - *Real Time Protocol*
RTCP - *Real Time Control Protocol*
RFC - *Request for Comment*
SER - *Sip Express Router*
SIP - *Session Initiation Protocol*
SDP - *Session Description Protocol*
SNA - *Systems Networking Architecture*
SMTP - *Simple Mail Transfer Protocol*
SNMP - *Simple Network Management Protocol*
TCP - *Transmission Control Protocol*
UA - *User Agent*
UAS - *User Agent Server*
UAC - *User Agent Client*
UDP - *User Datagram Protocol*
URL - *Uniform Resource Locator*
UTP - *Unshielded Twisted Pair*
VAD - *Voice Activity Detection*
VoIP - *Voice over IP*
XML - *Extensible Markup Language*

RESUMO

Este trabalho trata do *software* livre Asterisk, uma central telefônica privada IP (ou PABX IP) que suporta inúmeros protocolos e Codecs da tecnologia de voz sobre IP, abordando a sua instalação, configuração e compatibilidade com hardware de telefonia. O texto traz alguns conceitos básicos de comunicação de voz em uma rede IP, mostrando as necessidades para convergência de voz e dados, bem como todos os protocolos que permitem o tráfego de aplicações em tempo real através de uma rede de transmissão de dados digitais com comutação de pacotes. Por fim, é descrita a implementação do PABX IP no ambiente acadêmico da UTFPR - campus Curitiba, com auxílio da plataforma de programação em software livre (Linux e Asterisk). Descreve-se também, a instalação, configuração e testes de uma interface entre o PABX IP e um PABX de telefonia convencional, possibilitando que a plataforma Asterisk funcione como *gateway* VoIP.

ABSTRACT

This essay deals with free software Asterisk, a telephone exchange private IP (or PABX IP) that it supports innumerable protocols and Codecs of the voice technology on IP, approaching its installation, configuration and compatibility with the telephony hardware. The text brings some basic concepts of communication of voice in a net IP, showing the necessary stuff for voice convergence and data, as well as all the protocols that allow the traffic of applications in real time through a net data-communication digital with commutation of packages. Finally, the implementation of PABX IP in the academic environment of the UTFPR is described - Curitiba campus, with the assist of the platform of programming in free software (Linux and Asterisk). One also describes, the installation, configuration and tests of an interface between PABX IP and a PABX of conventional telephony, making possible Asterisk platform functions as gateway VoIP.

1. INTRODUÇÃO

A telefonia é uma tecnologia antiga e eficiente, e também uma forma de comunicação que atinge grande parte da humanidade. A rede telefônica, desta forma, possui uma base instalada muito estável e confiável.

Entretanto, é inegável que as redes de comutação de pacotes, inicialmente projetadas para a transmissão de dados, vêm sofrendo um grande crescimento no número de usuários, nos dias atuais. Neste contexto, fica evidenciada a importância da Internet, cuja tecnologia se baseia no protocolo IP (*Internet Protocol*).

Existem motivações que podem levar a pensar na possibilidade de trafegar voz nas redes IP. A estrutura básica da tecnologia IP tem um custo relativamente baixo, além do que existe a possibilidade de economia para as redes telefônicas privadas de empresas, evitando-se os custos de ligações interurbanas e internacionais.

Atualmente, a maioria das empresas tem interesse em estratégias que surgem para integrar tecnologias de rede em uma infra-estrutura comum. Esta tendência foi iniciada, há anos atrás, na área de interconexão, em que redes corporativas começaram com a migração das redes de dados SNA (*Systems Networking Architecture*) para uma infra-estrutura de redes locais e protocolo IP. Agora, vem sendo examinado como as redes de dados, voz e vídeo podem ser integrados de forma mais eficiente. Uma das alternativas que temos hoje no mercado é a implantação da tecnologia de Voz sobre IP (VoIP – *Voice over Internet Protocol*). Disponível em: <<http://www.gta.ufrj.br/~gardel/redes/intro.htm>>. Acesso em: 26 Mai. 2006.

Um serviço de Voz sobre IP é um serviço onde os sinais de voz são convertidos em pacotes de dados e transmitidos utilizando-se a tecnologia IP (tecnologia com base na Internet), ao invés da convencional e pouco flexível linha telefônica. A tecnologia VoIP está sendo vista, pelas empresas e fabricantes de equipamentos de rede, como uma atraente solução para a convergência de voz e dados. Isto está relacionado diretamente a pontos técnicos e comerciais, tais como a redução de custos e a integração de serviços. A utilização de VoIP nas redes corporativas faz com que haja uma economia na comunicação entre pontos da

empresa, onde exista até mesmo algum tipo de enlace para tráfego de dados como correio, *web*, *ftp*, etc.

Existem vários cenários de aplicação da tecnologia VoIP nas empresas e, também, em ambiente residencial utilizando diversos tipos de protocolos, dentre eles: H323, SIP (*Session Initiation Protocol*) e IAX (*Inter-Asterisk Exchange*). Há também diversos tipos de terminais VoIP, tais como:

Softphones: São *softwares* que simulam telefones reais e são instalados no computador para utilizar a conexão do microcomputador à Internet ou LAN (*Local Area Network*) e que processam a comunicação e tradução dos pacotes em voz.

Telefones IP: Aparelhos telefônicos reais que são diretamente ligados à Internet ou rede LAN e que processam a comunicação e tradução dos pacotes em voz sem utilizar o computador.

Telefones com Adaptador: É um aparelho telefônico convencional conectado à rede IP através de um adaptador ATA (*Analog Terminal Adapter*).

Há cinco anos atrás, a migração do sistema convencional para o sistema VoIP dispndia um investimento inicial elevado, onde somente grandes empresas eram capazes de ter acesso a esta tecnologia. Esse custo elevado ainda era alimentado pelas soluções proprietárias que tomavam conta do mercado.

Com o trabalho de Mark Spencer e Jim Dixon, surgiu um programa para sistema operacional GNU/Linux que é capaz de realizar funções de um PABX. O *software* desenvolvido foi o Asterisk. O Asterisk junto com os *hardwares* criados por Jim Dixon tornou o custo desta tecnologia bastante acessível a pequenas e médias empresas e ao usuário residencial.

O Asterisk é um *software* livre que vem se popularizando em aplicações VoIP. Ele é um controlador de equipamentos de voz sobre IP e de placas de telefonia (FXS/FXO/E1) que é capaz de tratar todos estes equipamentos como ramais de um PABX (*Private Automatic Branch eXchange*), realizando a tradução entre os mais diversos protocolos e CODECs utilizados na tecnologia VoIP.

O *software* Asterisk roda em servidores dedicados baseados no sistema operacional GNU/Linux, nas mais diversas distribuições e configurações, realizando diversas funções como: PABX, plataforma de comunicação, *gateway*, *switch*, servidor de mídias, servidor de conferência e *call center*.

O projeto proposto trata *do software* livre Asterisk, uma central telefônica privada IP que suporta inúmeros protocolos e codecs da tecnologia de voz sobre IP. Abordando a sua instalação, configuração, compatibilidade com hardware e configuração dos mesmos, desenvolveu-se o projeto de implementação do Asterisk no ambiente acadêmico da UTFPR - campus Curitiba, através da implementação de um PABX IP (VoIP), com auxílio de plataforma de programação em *software* livre (Linux e Asterisk) com conexão a um PABX convencional (Equitel Saturno 5000), conforme a figura 01.

Esta figura apresenta os softphones interligados na rede IP e os ramais convencionais conectados no PABX. A conexão entre o PABX convencional e o PABX IP será implementada utilizando-se uma interface de telefonia FXO (*Foreign Exchange Office*) e uma interface FXS (*Foreign Exchange Station*). Com isso, será possível realizar duas conversações simultâneas entre ramais VoIP e ramais convencionais.

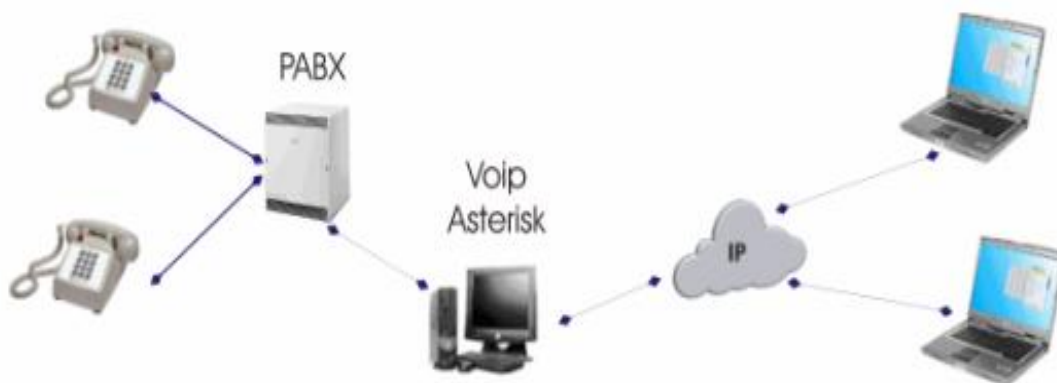


Figura 01 – Diagrama Esquemático do Projeto Proposto
(Fonte: Autoria própria)

2. OBJETIVOS

2.1 OBJETIVO GERAL

Integrar ramais de PABX convencional com ramais VoIP (*Softphones*), possibilitando estabelecer conexões de voz entre eles, no ambiente acadêmico da UTFPR - campus Curitiba, através da implementação de um PABX IP e servidor VoIP, com auxílio de plataforma de programação em *software* livre (Linux e Asterisk).

Doravante, o PABX IP e servidor VoIP (implementado com o *software* Asterisk) serão denominados simplesmente de IPBX Asterisk.

Deverão ser cumpridas várias metas intermediárias, para que o objetivo geral seja atingido. Estas metas estão descritas na sequência como objetivos específicos do trabalho.

2.2 OBJETIVOS ESPECÍFICOS

- Estudar os protocolos utilizados em aplicações VoIP e definir os que serão utilizados na implementação do IPBX Asterisk.
- Definir e configurar o *software* que será utilizado como *softphone* nos ramais VoIP.
- Configurar o IPBX Asterisk com algumas funcionalidades básicas de PABX IP e servidor VoIP.
- Configurar as interfaces FXO/FXS para interconectar o IPBX Asterisk com o PABX convencional.
- Testar o sistema implementado, verificando o funcionamento das conexões entre ramais convencionais e ramais VoIP, bem como as conexões dos ramais VoIP, entre si.

3. TELEFONIA CONVENCIONAL

3.1 INTRODUÇÃO

O telefone foi criado em 1876 por Graham Bell. A telefonia trouxe grandes facilidades e avanços para a sociedade e se tornou uma necessidade no mundo de hoje, tanto nas residências como nas empresas.

Os componentes básicos do sistema telefônico são:

Telefones Analógicos: São os mais comuns em residências e pequenas empresas. Podem ser conectados diretamente à rede pública ou ramais de centrais PABX. Telefones proprietários são constantemente utilizados com as centrais privadas. Esses telefones provêm funções adicionais como viva-voz, controle de volume, mensagem de espera da central telefônica e toques personalizados.

Linha de Assinantes: É a interface com a rede pública de telefonia. Tipicamente é constituída por um par de fios em que pode trafegar uma ligação. Uma pequena empresa pode ter ainda múltiplas linhas de assinantes.

Centrais Telefônicas: São os equipamentos responsáveis pela interconexão das linhas de assinantes entre si e das linhas de assinantes com os troncos. Para tanto, realizam funções tais como: sinalização, encaminhamento de chamadas e tarifação telefônica.

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

Desde sua criação até os dias atuais, várias mudanças foram implementadas no funcionamento das redes de telefonia. Antes, a comutação dos circuitos era feita por uma telefonista que sentava à frente de um painel e conectava as ligações dos cabos dos assinantes para que eles falassem entre si. Após vários avanços tecnológicos, chegou-se às atuais centrais telefônicas automáticas e programáveis via *software*, denominadas centrais CPA (controle por programa armazenado). Estas centrais possibilitam que as operadoras de telefonia forneçam, além da conexão de voz, outros serviços aos usuários tais como: conferência, chamada em espera, identificação de chamada. Disponível em: < <http://www.ginux.ufla.br/documentacao/monografias/mono-BrunoPinheiro.pdf>>. Acesso em: 7 Jun. 2006.

3.2 PABX

A central PABX tradicional utiliza tecnologia proprietária e os usuários ficam sempre limitados ao mesmo fabricante, para adicionar outras funcionalidades. Quaisquer modificações na programação ou conserto, normalmente, dependem de técnicos especializados, o que onera o custo de manutenção e operação.

A arquitetura de um PABX tradicional é composta por: controladores centrais, matrizes de comutação e interfaces de linha/tronco, conforme a figura 2.

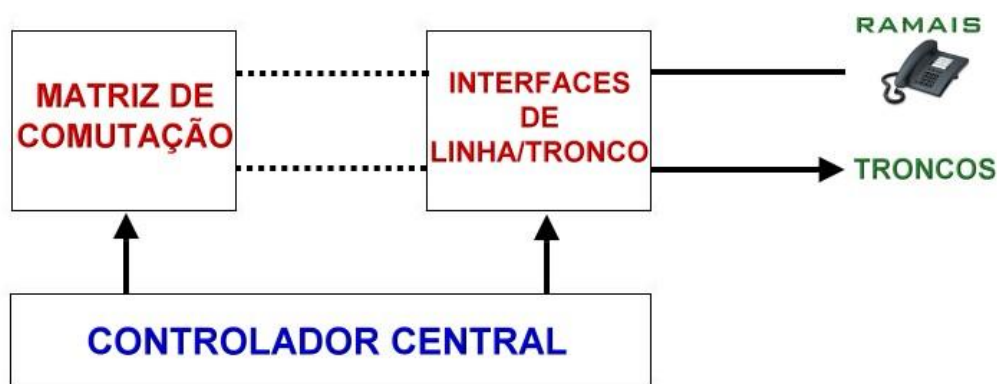


Figura 02 - Diagrama em Blocos de um PABX
(Fonte: Autoria própria)

O controlador central executa o *software* de comunicação, que opera todas as funcionalidades do sistema, atribuindo organização ao PABX. A matriz de comutação é quem vai unir a ligação de origem à rota do destino, identificando se a saída é para outro ramal ou para um tronco. Já as interfaces de linha/tronco são por onde as ligações são completadas, de acordo com a rota criada pela matriz de comutação, podendo ser FXS, FXO e E1. Na sequência, são descrita cada uma dessas interfaces.

FXO (Foreign Exchange Office): Correspondente a uma posição de tronco analógico do PABX. FXO, pode ser conectado numa posição de ramal de um PABX externo. Age como um aparelho telefônico analógico DTMF (*Dual Tone Multi Freqüencial*), fechando circuito e discando em sistemas externos, no caso de chamadas originadas no PABX. É capaz de reagir a correntes de toque e atender às ligações entrantes no PABX.

FXS (*Foreign Exchange Station*): Corresponde à uma posição de ramal do PABX. Um dispositivo FXS fornece alimentação elétrica e tom de discagem. A interface FXS foi projetada para ser diretamente ligada a um aparelho telefônico analógico DTMF ou então a qualquer posição de tronco analógico de um PABX externo. Esta interface reconhece discagens DTMF e gera toque de campainha para aparelho telefônico analógico.

E1: Equivalente Europeu do T1, funciona a 2Mbps. O G.703 é uma recomendação da ITU-T que trata das especificações da interface física a 4 fios e da sinalização digital para transmissão a 2,048 Mbps (E1 = 32 canais digitais de 64 Kbps). Esta interface utiliza codificação de linha HDB3, nível elétrico de Tx em ± 3 V e detecta Rx em ± 30 mV, tem alcance limitado (1,8 km @ diam. = 0,5 mm) e exige repetidores em longas distâncias.

4. CONVERGÊNCIA DE VOZ E DADOS

4.1 PROTOCOLO TCP/IP

“O modelo de Referência TCP/IP, cujo nome vem de seus dois mais conhecidos protocolos (TCP e IP), foi criado com o objetivo de ligar várias redes. Pois houve vários problemas quando surgiram redes de satélite e rádio que deveriam se comunicar com a ARPANET (rede de pesquisa criada pelo departamento de defesa dos Estados Unidos)”. (TANENBAUM, 1997)

Tal modelo é decomposto em quatro camadas, conforme a figura 3.

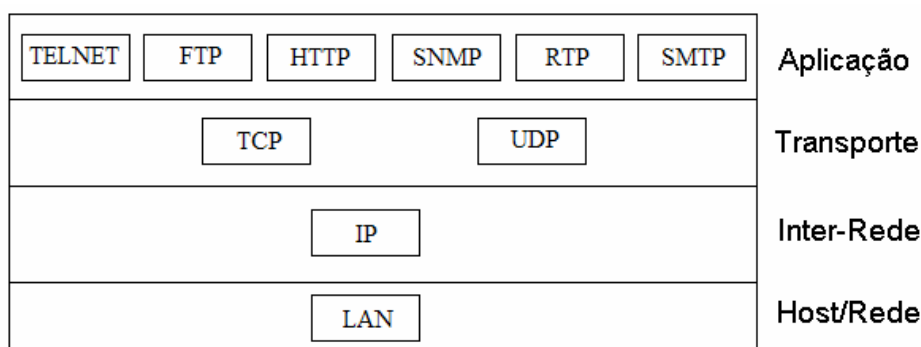


Figura 03 - Camadas do Modelo de Referência TCP/IP
(Fonte: Autoria própria)

Para que os computadores de uma rede possam trocar informações, é necessário que todos adotem as mesmas regras para o envio e o recebimento de informações. Este conjunto de regras é conhecido como Protocolo de Comunicação. Falando de outra maneira, pode-se afirmar que, para que os computadores de uma rede possam trocar informações entre si, é necessário que todos estejam utilizando o mesmo protocolo. No protocolo de comunicação estão definidas todas as regras necessárias para que o computador de destino entenda as informações no formato em que foram enviadas pelo computador de origem. Dois computadores, com protocolos diferentes instalados, não serão capazes de estabelecer uma comunicação e trocar informações.

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

O TCP (*Transmission Control Protocol*), como o próprio nome diz, controla a transmissão dos dados, cuidando para que os dados enviados por um computador cheguem integralmente ao destino correto. O TCP nada mais é que uma biblioteca de rotinas instaladas nos computadores de origem e destino, que suas aplicações utilizam quando precisam executar o transporte de dados entre equipamentos.

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

Enquanto o TCP cuida da segurança do envio e recebimento dos datagramas o IP é responsável pela transmissão em si, fazendo o serviço de roteamento, ou seja, conduzindo os dados para os endereços corretos. Na verdade, os dois protocolos se completam: enquanto o IP identifica os endereços e cuida para que os dados sejam enviados pelo meio físico, o TCP verifica se estes dados enviados foram transmitidos corretamente. Disponível em: < <http://orbita.starmedia.com/rodrigues/tcp/func.html>>. Acesso em: 28 Jun. 2006.

O protocolo UDP (*User Datagram Protocol*) é um protocolo de camada 4 (transporte) do modelo OSI, que se caracteriza por ser mais simples que o TCP, o outro protocolo de camada 4. Enquanto o TCP se preocupa com a conexão e a chegada correta dos dados no destino, o UDP, por ser mais simples, não tem a mesma preocupação; portanto, ele não verifica o recebimento dos dados pelo destino (também não possui o serviço de reenvio); não ordena as mensagens, ou seja, elas vão sendo agrupadas conforme vão chegando; não controla fluxo de informação e não verifica integridade dos dados para o destino. O UDP, como o TCP, utiliza portas para que vários programas utilizem, ao mesmo tempo, o protocolo. Desta forma o sistema operacional sabe para que programa se destina tal pacote. Outra característica do protocolo UDP é que o cabeçalho, que ele inclui nas mensagens, é pequeno se comparado com o TCP, trazendo um menor cabeçalho.

O UDP é utilizado em aplicações que necessitam de tráfego em tempo real, como a voz, onde não há interesse em que um pacote perdido seja retransmitido porque, com certeza, ele chegaria depois de outros que foram enviados anteriormente.

Quando se utiliza o protocolo TCP/IP como protocolo de comunicação em uma rede de computadores, alguns parâmetros devem ser configurados em todos os

equipamentos (computadores, servidores, *hubs*, *switchs*, impressoras de rede, etc) que fazem parte da rede. Cada computador da rede precisa de, pelo menos, dois parâmetros configurados:

- Endereço IP
- Máscara de sub-rede

Os endereços IP identificam cada microcomputador na rede. A regra básica é que cada micro deve ter um endereço IP em sua versão IV diferente e todos devem usar endereços, dentro da mesma faixa. Um endereço IP é composto de uma seqüência de 32 bits, divididos em 4 grupos de 8 bits cada. Cada grupo de 8 bits recebe o nome de octeto. Assim tem-se que 8 bits permitem 256 combinações diferentes. Para facilitar a configuração dos endereços, são usados números de 0 a 255 para representar cada octeto, formando endereços como 220.45.100.222, 131.175.34.7, etc. O endereço IP é dividido em duas partes. A primeira identifica a rede à qual o computador está conectado e a segunda identifica o computador dentro da rede. Como temos apenas 4 octetos, esta divisão limitaria bastante o número de endereços possíveis, o que seria uma grande limitação no caso da Internet, onde são necessários muitos endereços. Se fosse reservado apenas o primeiro octeto do endereço, por exemplo, teríamos um grande número de micros conectados a cada rede, mas em compensação poderíamos ter apenas 256 redes diferentes, o que seria muito complicado considerando a abrangência das redes interligadas mundialmente. Disponível em: < <http://www.guiadohardware.net/termos/endereco-ip-1>>. Acesso em: 28 Jun. 2006.

Para permitir uma gama maior de endereços, os desenvolvedores do TCP/IP dividiram o endereçamento IP em cinco classes, denominadas A, B, C, D, e E, sendo que as classes D e E estão reservadas para expansões futuras. Cada classe reserva um número diferente de octetos para o endereçamento da rede.

A máscara de sub-rede é um método para determinar qual a parte corresponde à identificação da rede e qual a parte do endereço IP corresponde à identificação de *Host* (servidor). Foi criada para substituir o obsoleto conceito de classes de IPs, que perdia muitos endereços de *Host* válidos. Com o tempo a internet foi crescendo e os endereços IP ficaram escassos, e mudanças foram implementadas para evitar maiores problemas. Uma máscara de sub-rede pode ser composta da seguinte seqüência numérica: 255.255.255.0. Como se pode notar, o

valor máximo para cada um dos campos é 255 e o mínimo é 0. Uma máscara de sub-rede, obrigatoriamente, deve ter valores máximos seguidos de valores mínimos. Assim sendo, 0.255.0.255 não é uma máscara de sub-rede válida.

Antigamente, na Internet, existia o conceito de Classes de IPs para que os clientes TCP/IP pudessem determinar qual parte do endereço correspondia à identificação de rede e qual determinava um *Host*. Dado um endereço IP da forma w.x.y.z, as classes eram as seguintes:

Classe A: Reserva o primeiro número dos quatro (w) para endereço de redes e o resto fica para endereços de *Host*, onde "w" pode variar de 1 a 126. Permite 126 redes e 16.777.214 *Hosts* por rede. Por essa definição, o endereço 110.224.16.15 é da classe A. Seria o equivalente a uma máscara de sub-rede 255.0.0.0.

Classe B: Reserva os dois primeiros números para endereço de rede (w.x) e os dois restantes para endereços de *Host*. Neste caso, "w" pode variar de 128 a 191. Permite 16.384 redes e 65.534 *Hosts* por rede. Exemplo de endereço classe B: 135.200.223.5. Equivalente a uma máscara de sub-rede 255.255.0.0.

Classe C: Reserva os três primeiros endereços para identificação de rede (w.x.y) e o último para identificação de *Host*. Os valores de "w" variam entre 192 e 223. Nesse caso, 2.097.152 redes são possíveis, com 254 *Hosts* por rede. Exemplo: 200.248.170.1. Equivalente a uma máscara de sub-rede 255.255.255.0.

Classes D e E: Os endereços da classe D e E são reservados para usos específicos. Disponível em: <<http://www.baboo.com.br/absolutenm/templates/content.asp?articleid=4513&zoneid=24>>. Acesso em: 3 Jun.2006.

4.2 TECNOLOGIA VoIP

Para que a transmissão de voz através da internet seja possível, o VoIP captura a voz, que até então é transmitida de forma analógica e a transforma em pacotes de dados, que podem ser enviados por qualquer rede TCP/IP, exemplificado pela figura 4. Assim, é perfeitamente possível trafegar com esses pacotes pela internet. Quando o destino recebe os pacotes, estes são decodificados em sinais analógicos e enviados a um transdutor sonoro (autofalante ou fone de ouvido). Disponível em: <http://www.internext.com.br/index.php?option=com_content&task=v

iew&id=90&Itemid=2>. Acesso em: 3 Jun. 2006.

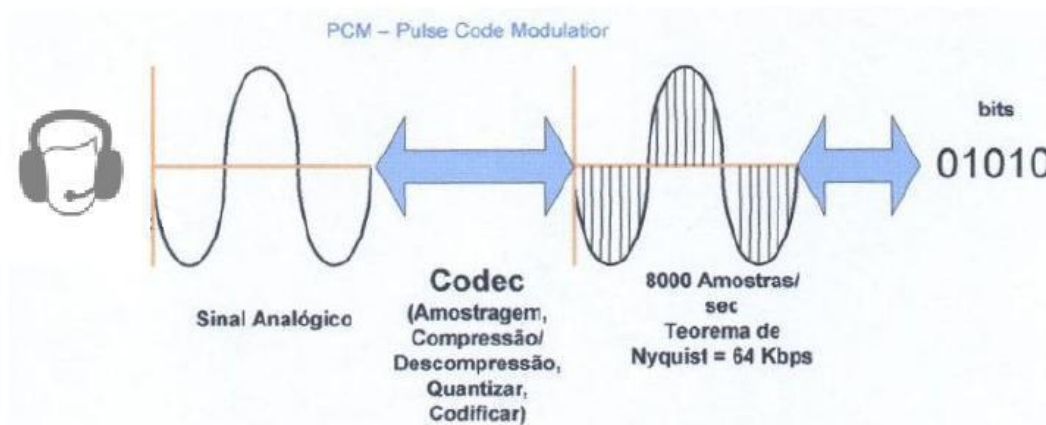


Figura 4 - Digitalização da voz analógica
(Fonte: <http://www.qinux.ufla.br/documentacao/monografias/mono-BrunoPinheiro.pdf>)

Para enviar áudio sobre uma rede de dados, a forma de onda deve ser considerada em bits através de uma operação chamada de “*encoding*” (digitalização). Após isto, esses bits devem ser inseridos no campo de dados do protocolo de transporte. Aliás, isto deve ser feito para o transporte de áudio sobre qualquer rede de dados, até sobre a própria rede de telefonia digital [XAVIER, 2000]

Existem hoje diferentes formas de telefonia por Internet. Alguns sistemas permitem apenas que duas pessoas falem usando um computador com *software* específico. Outros sistemas, além da utilização do microcomputador, permitem que qualquer pessoa, com conexão à internet por banda larga, possa utilizar um telefone IP ou um telefone convencional (recorrendo a um adaptador) para telefonar para qualquer telefone no mundo, seja cliente do mesmo serviço ou não. Disponível em: <http://www.telemoveis.com/voip/aboutvoip.asp>. Acesso em: 15 Jul. 2006.

Além disso, temos a utilização mais antiga do VoIP: a utilização corporativa. Uma empresa pode instalar a sua própria infra-estrutura VoIP utilizando a sua rede privativa de telecomunicações. Quando uma chamada é efetuada dentro da rede privativa, ela não tem custos. Quando é efetuada para fora, pode ser realizada via Internet ou pela rede pública.

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

Como citado anteriormente, a rede IP tem como princípio o melhor esforço, onde são usados todos os esforços para a entrega dos pacotes ao destino, mas a rede não garante, até mesmo, se eles vão chegar (pacotes perdidos), se vão chegar na ordem, se vai haver atraso na entrega (*delay*), ou se ocorrerão diferentes atrasos na entrega dos pacotes (*jitter*). Todos esses problemas são aceitáveis para o tráfego de dados na rede IP, mas não são aceitáveis no tráfego de voz, que precisa da transmissão em tempo real.

Abaixo, estão descritos os principais problemas e as consequências para a transmissão de voz sobre a rede IP:

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

Delay: É o tempo que o pacote de voz leva para chegar da origem para o destino. É ocasionado pelo congestionamento da rede e pela serialização dos pacotes nas interfaces de rede. A velocidade do *link* e tamanho do pacote são os principais fatores que influenciam no tempo do *delay*. O ITU-T através do padrão G.114 define que em uma via da transmissão o tempo de *delay* aceitável é de no máximo 150ms.

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

Na transmissão de voz sobre IP são necessários Codecs para transformação da voz analógica em dados digitais e protocolos para a sinalização e envio dos pacotes de voz pela rede IP, apresentados a seguir.⁵

4.3 CODIFICAÇÃO DE VOZ

Os Codecs são vários modelos matemáticos utilizados para converter o sinal analógico da voz numa versão digital codificada, objetivando atingir um balanço entre eficiência e qualidade. Variam na qualidade de som, largura de banda de que precisam, requisitos de processador, etc. Normalmente cada serviço, programa, telefone, *gateway*, entre outros, suportam múltiplos Codecs e quando falam uns com os outros, negociam o CODEC a utilizar (Exemplo: GIPS, GSM, Ilbc, G.711, etc).

Obviamente é desejado colocar tantas chamadas quanto possíveis em uma rede de dados. Isto pode ser feito codificando em uma forma que use menos banda passante. Este é o papel do CODEC (Coder/Decoder), por exemplo, o G.729 permite codificar a 8 Kbps uma compressão de 8 para 1 comparando com o G.711 que codifica a 64 Kbps.

O Asterisk, plataforma IPBX, é compatível com os Codecs citados no quadro 1.

CODEC	Taxa de bit de dados(Kbps)	Necessidade de licença
G.711	64 Kbps	Não
G.726	16, 24, 32 ou 40 Kbps	Não
G.723.1	5,3 ou 6,3 Kbps	Sim
G.729A	8 Kbps	Sim
GSM	13 Kbps	Não
iLBC	13,3 ou 15,2 Kbps	Não
Speex	Variável (2,15 a 22,4 Kbps)	Não

Quadro 1 – Comparativo dos Codecs

G.711: CODEC fundamental para o PSTN (*Public Switched Telephone Network*). A rede de telefonia estruturada PCM (Modulação por Codificação de Pulsos) utiliza este CODEC com a transmissão de 8 bits a 8 mil vezes por segundo, ou seja, 64 Kbps.

G.726: CODEC conhecido como ADPCM (Modulação Pulso-Código Diferencial Adaptativo), pode operar nas taxas de 16, 24, 32 ou 40 Kbps. A principal vantagem é utilizar a metade da banda e requer baixo desempenho do sistema

computacional comparado com o G.711 com a mesma qualidade. Porém perdeu espaço nos anos 90 devido à inabilidade de transportar sinais de modem e fax.

G.723.1: Projetado para transmitir baixas taxas de bits 5,3 ou 6,3 Kbps. Ele está atualmente coberto por patentes, sendo assim, necessária a utilização de licença para aplicação deste CODEC no Asterisk ou em outros sistemas.

G.729A: Este CODEC fornece excelente qualidade na transmissão de som, levando-se em consideração sua baixa largura de banda e transmissão (8 Kbps). Extremamente popular, pelo fato de suportar diversos sistemas. É patenteado e faz-se necessário a utilização de licença.

GSM: Taxa de transmissão de 13 Kbps. Tem como vantagem não precisar de liberação (Licença) e não utiliza alto nível de processamento do sistema (CPU). De acordo com os estudiosos do Asterisk este é o CODEC mais indicado.

iLBC: CODEC de Baixa Taxa de Bits de Internet (iLBC) tem a vantagem de fornecer baixa largura de banda (15 Kbps) sem necessidade de licença. Desvantajoso devido à incompatibilidade com alguns telefones IP, sistemas VoIP e a utilização de algoritmos complexos para atingir um alto nível de compressão dos dados.

Speex: CODEC com Taxa de Bits Variável (VBR) sendo possível a variação dinâmica da taxa de bits de 2,15 a 22,4 Kbps. O Speex é um CODEC de distribuição totalmente livre.

4.4 PROTOCOLOS VoIP

O protocolo H.323 é um padrão do ITU-T (*International Telecommunications Union-Telecommunication*), muito popular desde o seu aparecimento, principalmente por ser o primeiro conjunto definido de padrões. A maioria dos produtos VoIP existentes são H.323. Trata-se de um protocolo específico, desenhado desde o início para a comunicação multimídia sobre IP. Especifica tudo na comunicação, incluindo o CODEC e a forma de transferir os pacotes RTP (*Real Time Protocol*), trazendo para o protocolo a maior parte da complexidade, assumindo, desta forma, a possibilidade de falha da rede, o que pode ser um esforço desnecessário. As mensagens H.323 são binárias e codificadas em ASN. 1, o que as torna muito complexas e de difícil alteração.

O SIP é o protocolo *Standard* do IETF (*Internet Engineering Task Force*) - RFC 3261 - que define a sinalização para sessões multimídia e VoIP. É também muito flexível, porque as suas mensagens em texto permitem adicionar funcionalidades. É um protocolo da família dos protocolos Internet e por ser um *standard* IETF, está preparado para ser utilizado com as outras normas da Internet (DNS, HTTP,...). Além disso, a sua arquitetura distribuída delega a complexidade aos terminais, à imagem da Internet. Cada vez mais fabricantes estão a adaptar o SIP, o que o torna cada vez mais utilizado hoje em dia. Possui uma importante funcionalidade designada “*forking*”, em que um utilizador pode estar registrado em vários telefones ao mesmo tempo. Ao receber uma chamada, os vários telefones tocam e a chamada é atendida no telefone que atender primeiro. Deixa assim de ser necessário encaminhar as chamadas de uns telefones para os outros, criando uma mobilidade muito maior.

O H.323 e o SIP são protocolos de sinalização, ou seja, são utilizados para o estabelecimento das chamadas em VoIP. No entanto, o transporte de mídia para telefonia baseada em IP é implementado com o RTP, que é o mecanismo padrão de fato, e provê transporte fim a fim para dados em tempo real (áudio e vídeo) e dados para aplicações que não são em tempo real. O RTP requer o uso de um protocolo de sinalização para estabelecer a conexão e negociar os formatos de mídia que serão utilizados. A funcionalidade do RTP é enriquecida com o RTCP (*Real Time Control Protocol*), que provê monitoramento fim a fim da entrega de dados e da qualidade do serviço (QoS).

Já o IAX (*Inter-Asterisk Exchange*) é o protocolo proprietário do Asterisk. Eficiente em banda passante e principalmente pode passar facilmente por *firewalls* com NAT. Se quiser usar SIP com NAT na internet pode se usar o SER (*Sip Express Router*) em conjunto com o Asterisk. Os pontos fortes do IAX são: eficiência em banda passante, segurança e facilidade no NAT. Porém, seu ponto fraco é o fato de ser um protocolo proprietário.

O IAX é um protocolo de controle e transmissão de dados de mídia numa rede ponto a ponto e está atualmente na sua versão 2 (IAX2). Tem a capacidade de transportar qualquer dado de mídia, porém sua principal função é a de sinalização de chamadas. Diferente do H.323 e do SIP, o IAX não utiliza os protocolos RTP/RTCP para a transmissão de dados de mídia, e sim faz multiplexação de

diversos fluxos de mídia em um único datagrama UDP, que também é utilizado para a sinalização de chamadas. Todos os datagramas UDP são transmitidos utilizando a porta padrão 4569, podendo dessa forma unificar duas atividades em um único protocolo e eliminar problemas com o NAT (*Network Address Translation*) existentes no SIP.

4.5 QoS

A camada de transporte tem como sua principal função melhorar a qualidade de serviço (QoS) oferecida pela camada de rede. Se o serviço de rede for perfeito, o trabalho da camada de transporte será fácil. No entanto, se o serviço de rede não for perfeito, a camada de transporte terá que servir de ponte para cobrir a distância entre o que os usuários de transporte desejam e o que a camada de rede oferece. Ainda que, à primeira vista, o conceito de qualidade de serviço seja vago (fazer com que todos concordem sobre o que significa um serviço de boa qualidade não é uma tarefa simples), a QoS pode ser definida por um número específico de parâmetros. Os parâmetros de rede são definidos na recomendação I350 (ITU I350, 1993) como “a habilidade da rede ou dos componentes da rede para prover funções relacionadas à comunicação entre dois usuários”.

O serviço de transporte pode permitir ao usuário determinar os valores preferenciais, os valores aceitáveis e os valores mínimos para vários parâmetros de serviço, no momento em que uma conexão é estabelecida. Alguns parâmetros também podem ser usados no transporte sem conexão. É tarefa da camada de transporte examinar esses parâmetros e, dependendo do(s) tipo(s) de serviço(s) de rede disponível(eis), determinar se é possível realizar o serviço solicitado. Os parâmetros típicos para a qualidade de serviço da camada de transporte são resumidos em:

Retardo no Estabelecimento da Conexão: Tempo entre a solicitação de conexão e o recebimento de sua confirmação (exemplo comando Ping do Windows).

Probabilidade no Estabelecimento da Conexão: A não possibilidade de conexão, devido a um congestionamento na rede, falta de banda passante, etc.

Troughput: Calcula o número de bytes de dados transmitidos por segundo durante um determinado intervalo de tempo. É medido separadamente para cada direção (*download* e *upload*).

Retardo de Trânsito: Calcula o tempo entre o envio de uma mensagem pelo usuário de transporte da máquina de origem até seu recebimento pelo usuário de transporte da máquina de destino.

Taxa de Erros Residuais: Calcula o número de mensagens perdidas ou corrompidas em uma porcentagem do total enviado. Na teoria, a taxa de erros residuais deveria ser zero, pois o trabalho da camada de transporte é esconder os erros da camada de rede.

Prioridade: Examina qual mensagem tem prioridade para envio e recepção.

Resiliência: Finaliza uma conexão espontaneamente devido a problemas internos ou congestionamentos.

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

A descrição de QoS é tomada da recomendação do ITU-T E.800, onde se define a qualidade de serviço como sendo o efeito coletivo provocado pelas características de desempenho de um serviço, determinando o grau de satisfação do usuário. Tal definição engloba, originalmente, vários aspectos de diversas áreas de atuação, incluindo o nível de satisfação do usuário.

A QoS é muito importante em uma rede convergente com voz e dados porque o comportamento de cada tipo de pacote na rede pode trazer mais ou menos desconforto ao usuário. Um pacote de dados, por exemplo, se chegar atrasado ou se houver uma perda, não irá influenciar muito no resultado final.

Mas um pacote de voz se chegar atrasado, ou se for perdido, trará desconforto para os interlocutores, por isso se define que o tráfego de voz tem que ser em tempo real.

Em vista disso, através da QoS na rede TCP/IP é necessário criar regras para priorizar o tráfego de voz em relação ao tráfego de outros dados na rede, utilizando várias técnicas, sendo aplicada em todos os componentes da rede, pois

se em um componente não for aplicada técnicas de QoS, a qualidade não poderá ser mais garantida naquela rede. Disponível em: < <http://www.dimap.ufrn.br/~glaucia/RAV/Monografia.pdf>>. Acesso em: 17 Ago. 2006.

5 ASTERISK

Asterisk é um *software* que pode transformar um simples microcomputador em um poderoso e completo PABX. Ele “roda” em Linux e provê todas as funcionalidades de um PABX e muito mais. Foi criado por Mark Spencer, da Digium, que apóia o projeto e comercializa o hardware de telefonia. Asterisk faz Voz sobre IP em diferentes protocolos e pode se integrar à maioria dos padrões de telefonia utilizando hardware de baixo custo e aberto, desenvolvido por Jim Dixon.

Seu nome vem do símbolo *, representado pela figura 5, o qual em Unix e Linux representa um *wildcard* que equivale a qualquer caractere, qualquer arquivo. Similarmente, o Asterisk é destinado a integrar qualquer módulo de telefonia, seja *hardware* ou *software*, a aplicações.



Figura 05 – Logotipo do Asterisk
(Fonte: <http://www.asteriskbrasil.org>)

5.1 ARQUITETURA DO ASTERISK

O Asterisk foi desenvolvido para ter o máximo de flexibilidade. Algumas APIs (*Application Programming Interface*) especiais foram definidas em torno do núcleo. Dessa maneira, o Asterisk fica transparente a protocolos, Codecs e hardware, podendo ser compatível com qualquer tecnologia existente ou que venha a ser lançada, sem que sejam necessárias mudanças no núcleo do programa. Carregar os módulos separadamente também permite maior flexibilidade ao administrador, permitindo que ele escolha a melhor e mais enxuta configuração que o atenda. Na figura 6, apresenta-se um diagrama em blocos da arquitetura do Asterisk.

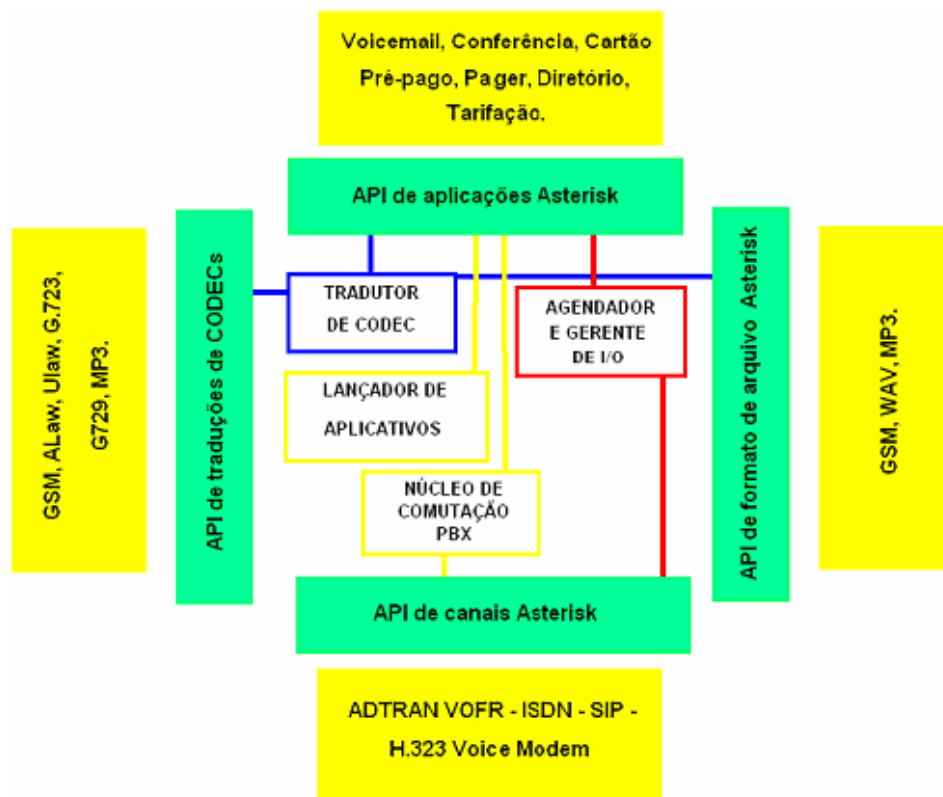


Figura 06 - Arquitetura do Asterisk
(Fonte: <http://www.asterisk.org>)

O núcleo do programa é dividido em quatro módulos:

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

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

Tradutor de Codecs: Esse módulo é responsável pela tradução dos Codecs suportados pelo Asterisk.

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

As APIs podem ser divididas em quatro tipos principais:

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

APIs de Aplicações: Essas APIs permitem que várias funcionalidades possam ser executadas como: Conferência, listagem de diretório e *voicemail*.

APIs de Tradutor de CODECs: Esses módulos são carregados para suportar vários codificadores e decodificadores de áudio como GSM, G.711 e outros.

APIs de Formato de Arquivo: Permitem ao Asterisk ler e escrever em vários formatos de arquivos para armazenar no sistema de arquivos.

5.2 DIRETÓRIOS DE CONFIGURAÇÃO DO ASTERISK

O *software* Asterisk é um gerenciador de equipamentos de voz sobre IP que suporta diversas placas de telefonia convencional (FXS/FXO/E1). É capaz de tratar todos estes equipamentos como um PABX, realizando a tradução entre os mais diversos protocolos e codecs utilizados na tecnologia da Voz sobre IP.

Todas as configurações do servidor Asterisk são realizadas através de arquivos (.conf) que ficam localizados no diretório do sistema operacional Linux de configuração(/etc/asterisk). A configuração do servidor é realizada através de arquivos. É no arquivo *extensions.conf* que são realizadas as configurações do plano de discagem, responsável por realizar o encaminhamento das solicitações (encaminhamento das chamadas), dos ramais, grupos de ramais e para os canais do hardware de comunicação. O arquivo *zaptel.conf*, que fica localizado em /etc/, configura os parâmetros do hardware de telefonia, por exemplo o *drive* da placa TDM400P, utilizado nesse projeto. Já o arquivo *zapata.conf* estabelece a configuração da interface entre os módulos de telefonia e o Asterisk, definindo o tipo de sinalização e habilitando outras funções, tais como: cancelamento de eco, chamada em espera e identificação de chamadas.

5.3 APLICAÇÕES DO ASTERISK

O Asterisk permite conectividade em tempo real entre as redes PSTN e redes VoIP. O Asterisk é muito mais que um PABX padrão. Com este *software* é possível criar diversas aplicações de voz, podendo substituir qualquer sistema de PABX, sendo muito útil para as empresas, conforme a figura 7.

- Conectar empregados trabalhando de casa para o PABX do escritório sobre conexões de banda larga.
- Conectar escritório em vários estados sobre IP. Isto pode ser feito pela Internet ou por uma rede IP privada.
- Fornecer aos funcionários correio de voz, integrado com a *web* e *e-mail*.
- Construir aplicações de resposta automática por voz, que podem conectar um usuário ao sistema de pedidos, por exemplo, ou ainda outras aplicações internas.
- Dar acesso ao PABX da companhia a usuários que viajam, conectando sobre VPN (*Virtual Private Network*) de um aeroporto ou hotel.
- Distribuidor automático de chamadas e fila de atendimento, sistema utilizado em *call centers*.
- Servidor de música de espera, disponível no formato MP3.
- Discador automático, sistema bastante usado em telemarketing.
- Sala de conferência, a qual permite que dois, ou mais usuários, falem em conjunto.

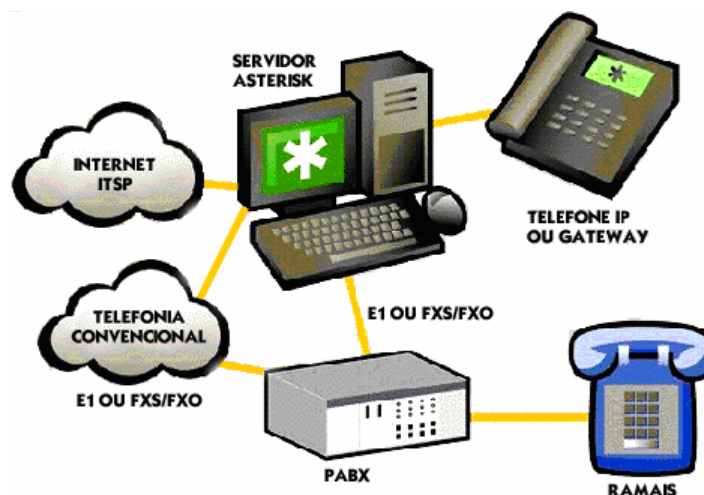


Figura 07 - Aplicações do Asterisk
(Fonte: http://www.ipfone.com.br/info_asterisk.asp)

5.4 HARDWARE MÍNIMO PARA INSTALAÇÃO DO ASTERISK

O Asterisk faz o uso intensivo de CPU, pois utiliza para o processamento dos sinais analógicos (voz) e digitais (pacotes de dados VoIP). Para a construção de um primeiro PABX IP para teste, um processador de 300MHz com 256MB de memória RAM é o suficiente. Já o uso de disco rígido não é elevado, ele requer 100MB para o programa compilado. Mas a inserção de código fonte, *voicemail*, *prompts* customizados requerem mais espaço, de acordo com a quantidade de usuários e placas PCIs.

5.5 HARDWARE DE TELEFONIA PARA O ASTERISK.

O Asterisk suporta inúmeros tipos de hardware de telefonia, de inúmeros fabricantes, mas, principalmente, o hardware fabricado pela Digium.

Segue abaixo o resumo de três tipos diferentes de placas PCIs (*Peripheral Component Interconnect*), fabricadas pela Digium, mais utilizadas no Brasil.

Wildcard XP100P: É uma das placas mais simples com uma interface FXO, que pode ser ligada à rede pública ou a uma posição de ramal analógico de um PABX. Uma placa de fax/modem com chipset Intel 537 ou MD3200 pode ser usada como uma XP100P.

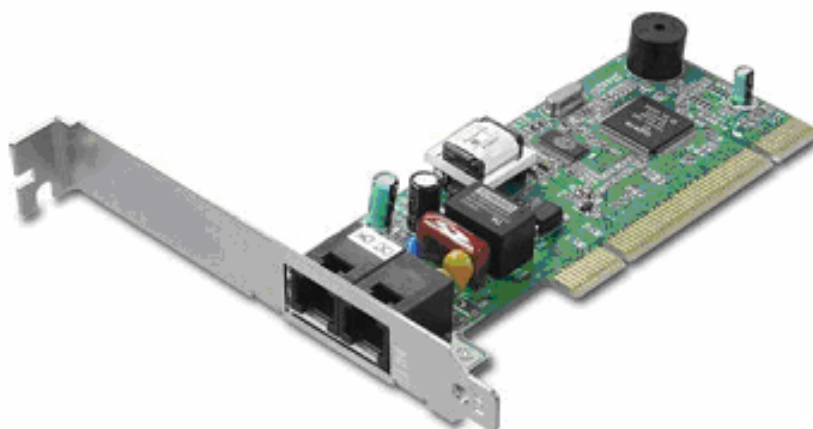


Figura 08 – Módulo Wildcard XP100P
(Fonte: <http://www.asteriskbrasil.org>)

Wildcard TDM400P: É uma placa analógica que pode conter até quatro canais. Os canais podem ser FXO (pode ser ligado à rede pública ou a um ramal de PABX) ou FXS (pode se ligar um assinante, com um telefone analógico) dependendo da configuração escolhida.



Figura 09 – Módulo Wildcard TDM400P
(Fonte: <http://www.asteriskbrasil.org>)

Wildcard TE100P: É uma placa com 30 canais digitais no padrão E1 / ISDN. Com esta placa o Asterisk pode conectar-se à rede pública ou a central telefônica.



Figura 10 – Módulo Wildcard TE100P
(Fonte: <http://www.asteriskbrasil.org>)

6 - PREPARANDO O IPBX ASTERISK

6.1 – INTRODUÇÃO

O Asterisk funciona em muitas distribuições Linux. Foi escolhido manter as coisas simples e ficar em uma única plataforma e distribuição do Linux. No Projeto proposto, foi utilizada a distribuição Linux SUSE 10.1, não somente por haver estudos direcionados para tal, mas também por ser gratuita a aquisição do sistema.

Para implementar nosso primeiro IPBX, o Departamento de Eletrônica da UTFPR disponibilizou um microcomputador com processador Intel Celeron 600MHz com 128MB de memória RAM e com um disco rígido (HD) de 20GB. Devido a problemas de instabilidade do *hardware*, este microcomputador foi substituído, posteriormente, por um microcomputador com processador Sempron 3000+, 512 Mb de memória RAM e disco rígido de 40 Gigabytes, para termos maior precisão nos nossos testes. O Asterisk não requer muito espaço em disco, cerca de 100 MB compilados, mais código fonte e *prompts* customizados.

6.2 – INSTALANDO O LINUX SUSE 10.1



Figura 11 – Logotipo do Linux Suse
(Fonte: <http://en.opensuse.org>)

Consegue-se os arquivos de instalação fazendo o *download* no site [http://en.opensuse.org/PT-BR-Released Version](http://en.opensuse.org/PT-BR-Released-Version) onde obtivemos os 5 CD's para efetuarmos a instalação do Sistema Operacional.

O início da instalação e o decorrer da mesma foram, relativamente, simples. Ligamos o microcomputador já com o CD1 no leitor e selecionamos no *setup* do microcomputador o tipo de *Boot* para *First Boot CD-ROM*. A partir deste ponto,

durante aproximadamente 54 minutos, estava concluída a instalação do Sistema Operacional Linux Suse 10.1. Mais detalhes sobre a instalação do sistema operacional, ver apêndice A.

Iniciado o sistema operacional, conforme a figura 12.

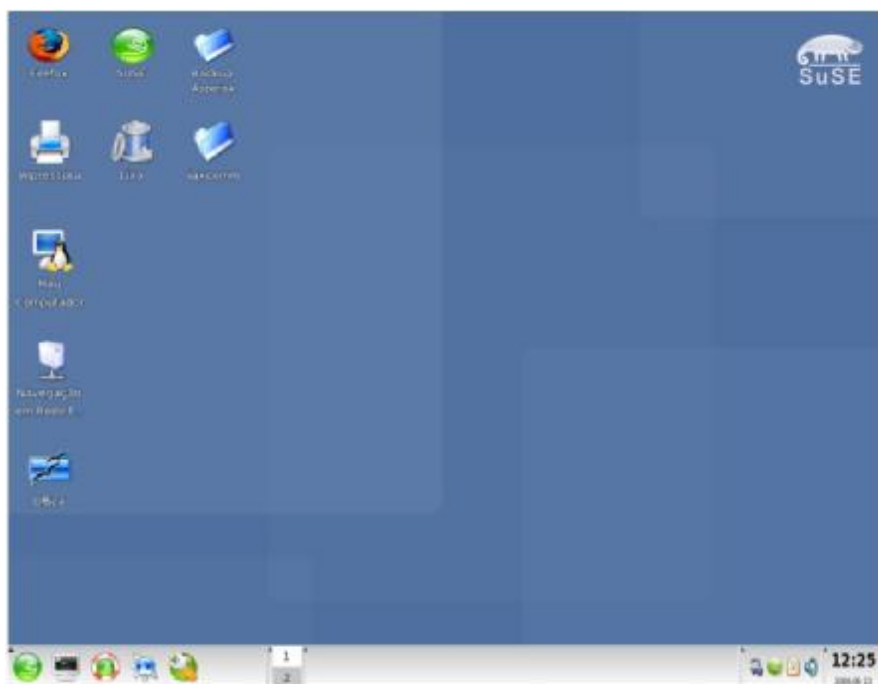


Figura 12 – Tela inicial do Sistema Operacional Linux
(Fonte: Autoria própria)

Ao iniciarmos o Linux, tivemos que realizar a instalação de alguns pacotes para compilação do Asterisk utilizando a ferramenta YAST (Centro de Controle), conforme apresentado na figura 13.

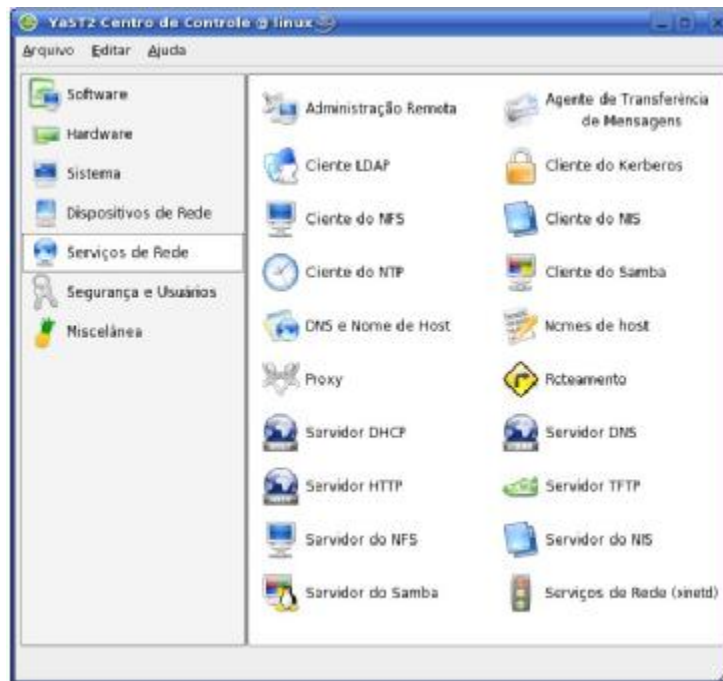


Figura 13 – Yast – Centro de Controle
(Fonte: Autoria própria)

Os pacotes que são necessários instalar são:

- Kernel Sources – v. 2.6.13-15
- gcc – GNU C Compiler and Support Files – v. 4.0.2
- cvs – Concurrent Versions System – v. 1.12.12-4
- ncurses – New curses libraries – v. 5.4-71
- ncurses-devel – Bibliotecas para desenvolvimento com ncurses – v. 5.4-71
- bison – The GNU parser generator – v. 1.875-56
- Termcap – Termcap library – v. 2.0.8-880
- openssl – Secure Sockets and TLS Layer Security – v. 0.9.7g-2
- openssl-devel – Bibliotecas do openssl – v. 0.9.7g-2
- zlib-devel – v. 1.2.3-3

Para esta atualização através do Yast, são necessários os CD's que foram utilizados para instalar o Linux. Abaixo, temos a visão da atualização do pacote GCC no Yast, mostrado pela figura 14.

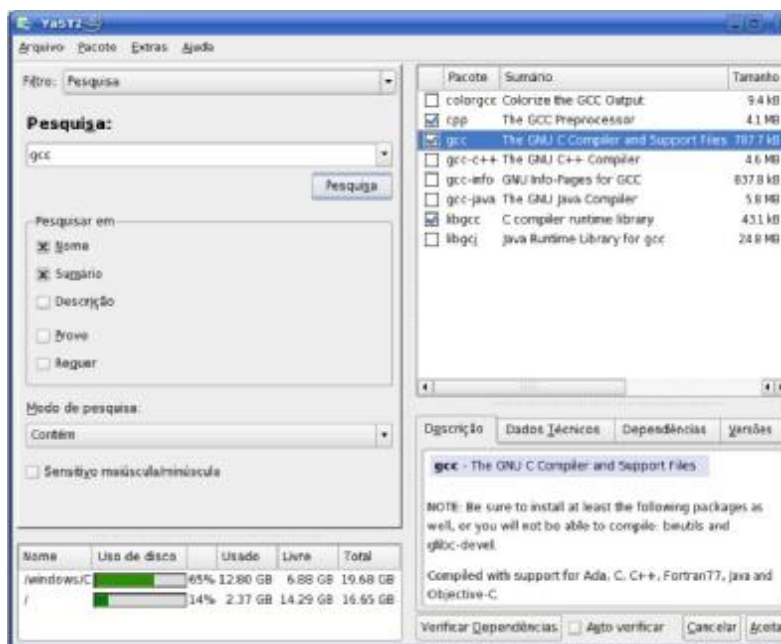


Figura 14 – Instalando o Pacote GCC
(Fonte: Autoria própria)

Depois de concluída a instalação dos pacotes citados acima, tivemos que desabilitar o *firewall* e executar as configurações de rede do servidor VoIP, antes da compilação do Asterisk.

6.3 – CONFIGURANDO O LINUX SUSE

Com o empenho do Departamento de Eletrônica, conseguimos um endereço IP fixo e válido na rede mundial de computadores.

Este endereço IP é único na rede mundial de computadores e poderá ser acessado de qualquer equipamento conectado na Internet, e servirá para os clientes VoIP efetuarem a autenticação no servidor Asterisk, independentemente de onde estes estiverem se conectando.

No projeto desenvolvido, utilizou-se o endereço IP 200.19.73.18, válido e fixo disponibilizado na rede da UTFPR e o configuramos no servidor conforme as figuras 15, 16 e 17.



Figura 15 – Configurando Dispositivo de Rede
(Fonte: Autoria própria)

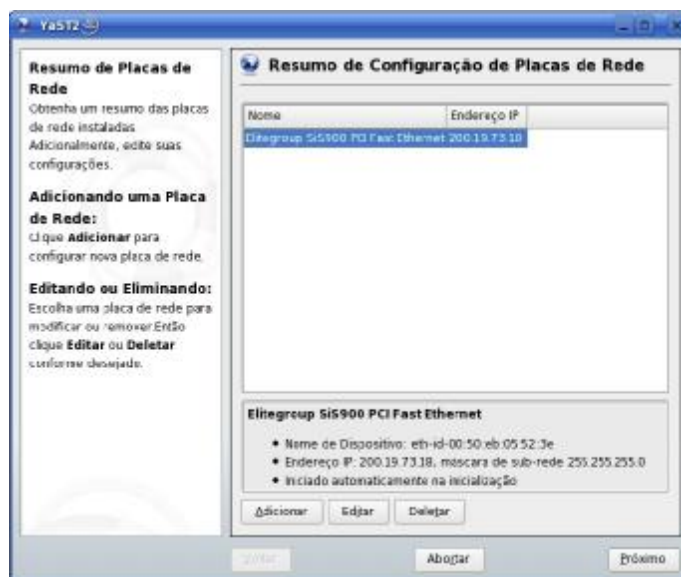


Figura 16 – Configurando Dispositivo de Rede
(Fonte: Autoria própria)

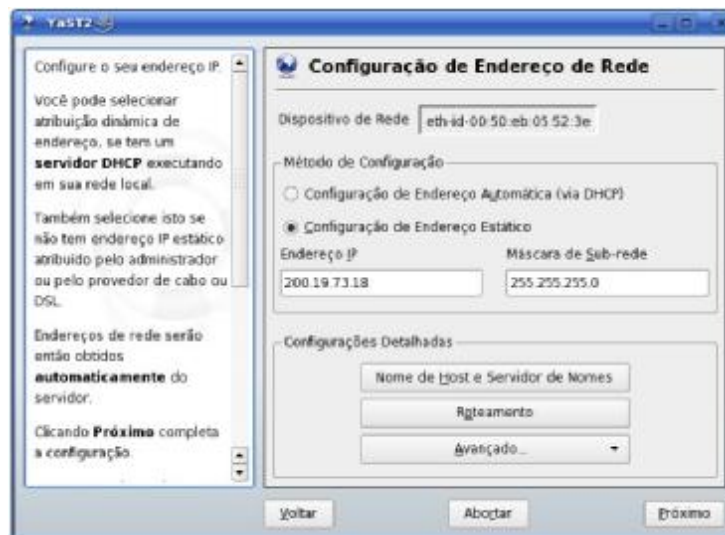


Figura 17 – Configurando IP
(Fonte: Autoria própria)

Outra configuração necessária é a do endereço IP do servidor DNS (*Domain Name Service*), disponibilizado pela UTFPR. Um servidor DNS é utilizado para ajudar a encontrar endereços IP na Internet. Por motivo de segurança na rede da UTFPR, não é divulgado o endereço do DNS. O endereço IP do Servidor DNS foi configurado no sistema, conforme a figura 18.

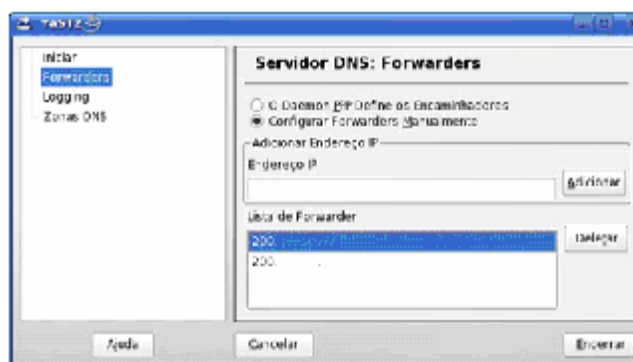


Figura 18 – Configurando DNS
(Fonte: Autoria própria)

Após a configuração do DNS, efetua-se a desabilitação do *Firewall* do sistema operacional Linux. O *Firewall* é um dispositivo que atua como mecanismo de segurança e tem como objetivo apenas receber e transmitir pacotes de dados

autorizados. Para desabilitar o *Firewall* no Yast, têm-se a opção de configuração apresentado nas figuras 19 e 20.

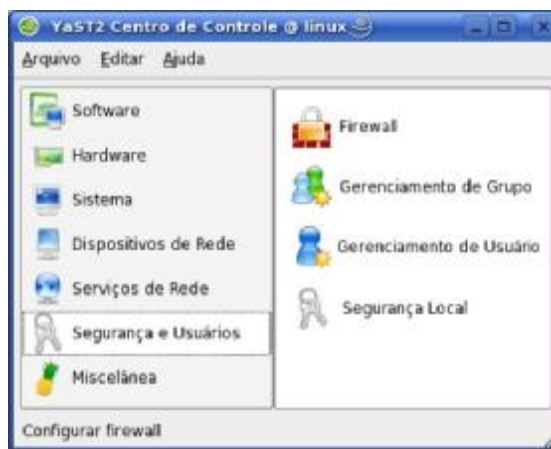


Figura 19 – Configurando *Firewall*
(Fonte: Autoria própria)



Figura 20 – Desabilitando *Firewall*
(Fonte: Autoria própria)

Depois de configurada a rede TCP/IP e desabilitada a Firewall, o sistema operacional Linux está pronto para que seja instalada e configurada a plataforma Asterisk.

6.4 – OBTENDO E COMPILANDO O ASTERISK

Inicialmente, obtemos o *software* Asterisk v1.2.11 e seus pacotes gratuitamente via download do site oficial <http://www.asterisk.org/>.

Salvamos no diretório `/usr/src` os seguintes arquivos, conforme a figura 21:

- Asterisk Version v1.2.11 – Arquivos compactados que contém os arquivos fonte para compilação *software* Asterisk.
- Zaptel Version v1.2.9.1 – Arquivo de *Drivers* dos *hardwares* compatíveis com o Asterisk.
- Libpri Version v1.2.3 – Arquivo que contém as bibliotecas utilizadas pelo Asterisk.
- Addons Version v1.2.4 – Arquivo que contém os códigos fonte do módulos adicionais utilizados pelo Asterisk
- Sounds Version 1.2.1 – Arquivos de sons utilizados pelo Asterisk



Figura 21 – Diretórios do Asterisk
(Fonte: Autoria própria)

A compilação, tanto do *software* quanto dos seus pacotes, foi realizada via console no diretório `/usr/src #`.

Inicialmente efetuamos a compilação do arquivo principal do *software* Asterisk, `asterisk-1.2.11`, que contém os arquivos fonte, através dos comandos:

- | | |
|--|---|
| <code>cd asterisk-1.2.11</code> | – Acessa o diretório principal do Asterisk |
| <code>make clean</code> | – Apaga os arquivos de configuração do Asterisk |
| <code>make mpg123</code> | – Efetua a compilação do player MP3 |

- | | |
|----------------------------|---|
| <i>make</i> | – Compila os arquivos do diretório |
| <i>make install</i> | – Instala os arquivos nos diretórios do sistema operacional |
| <i>make samples</i> | – Cria os arquivos de configuração padrão |

Em seguida, efetuamos a compilação do arquivo que contém os códigos fonte dos módulos adicionais utilizados pelo Asterisk, asterisk-addons-1.2.4, através dos comandos:

- | | |
|--|---|
| <i>cd asterisk-addons-1.2.4</i> | – Acessa o diretório asterisk-addons |
| <i>make</i> | – Compila os arquivos do diretório |
| <i>make install</i> | – Instala os arquivos nos diretórios do sistema operacional |

Compilamos o arquivo padrão de sons utilizado nas facilidades e funcionalidades do Asterisk, asterisk-sounds-1.2.1, através dos comandos:

- | | |
|--|---|
| <i>cd asterisk-sounds-1.2.1</i> | – Acessa o diretório asterisk-sounds |
| <i>make</i> | – Compila os arquivos do diretório |
| <i>make install</i> | – Instala os arquivos nos diretórios do sistema operacional |

Por fim, efetuamos a compilação do arquivo que fornece as bibliotecas para canais digitais, libpri-1.2.3, através dos comandos:

- | | |
|-------------------------------|---|
| <i>cd libpri-1.2.3</i> | – Acessa o diretório libpri |
| <i>make</i> | – Compila os arquivos do diretório |
| <i>make install</i> | – Instala os arquivos nos diretórios do sistema operacional |

Até esta configuração, para quem desejaria somente uma plataforma VoIP sem hardware de telefonia, já poderia estar configurando o Asterisk, conforme

descrevemos no tópico 6.5. Porém, no projeto proposto necessita-se de um Hardware para realizar a conexão com o PABX Equitel Saturno 5000.

6.4.1 Instalando E Compilando O Hardware TDM400P

Para comunicação do servidor Asterisk com a central PABX, efetuamos a instalação de uma interface de comunicação, o hardware da Digium TDM400P, no *slot* PCI do microcomputador servidor Asterisk. Em seguida, efetuamos a compilação do driver Zaptel do módulo PCI TDM400P, através dos comandos:

<i>cd zaptel-1.2.8</i>	- Acessa o diretório zaptel
<i>make clean</i>	- Apaga os arquivos de configuração do zaptel
<i>make linux26</i>	- Compila os arquivos para sistemas com kernel apartir da versão 2.6
<i>make install</i>	- Instala os arquivos do driver nos diretórios do sistema operacional

Após termos instalado o *driver* zaptel, carregamos o módulo PCI da Digium TDM400P com as duas interfaces instaladas (FXO/FXS), através do comando modprobe.

O comando Modprobe é usado para carregar os *drivers* da zaptel na memória, de forma que se possa ter acesso ao *hardware* do sistema. Primeiramente carregamos o driver zaptel na memória, através do seguinte comando:

<i>modprobe zaptel</i>	- Carrega o driver da placa na memória, verificando suas dependências
-------------------------------	---

Depois de carregado o *driver* com o comando anterior, carregamos os drivers específicos das portas de comunicação disponibilizadas na placa da Digium TDM400P, uma interface FXS e uma FXO, através do seguinte comando:

modprobe wcfxs

- Carrega na memória o driver das interfaces instaladas na placa.

6.5 CONFIGURANDO A PLATAFORMA ASTERISK

Depois de compilados os arquivos do Asterisk, cria-se um diretório em */etc/asterisk/*, contendo os arquivos de configuração com *extension.conf*.

Foi definido, no IPBX Asterisk, uma faixa de numeração para os usuários SIP e outra faixa para usuários que utilizam o protocolo IAX. Os arquivos necessários para configuração das funcionalidades básicas do IPBX são os seguintes:

- Sip.conf
- lax.conf
- Extension.conf
- Zapata.conf
- Meetme.conf
- Musiconhold.conf

Na sequência, são descritas as funções de cada um destes arquivos, bem como os parâmetros que foram configurados para este projeto.

6.5.1 SIP.conf

Configuramos o arquivo do protocolo SIP em */etc/asterisk/sip.conf*. Este arquivo contém os parâmetros relacionados à configuração dos telefones SIP. Os clientes SIP devem ser configurados antes que possam ser originadas ou recebidas chamadas.

O arquivo SIP é executado de cima para baixo. A primeira seção contém as opções globais [general]. Segue abaixo a descrição de alguns parâmetros que podem ser configurados na seção *general*:

- **port:** Define a porta UDP que o Asterisk deve esperar por conexões de entrada SIP. O padrão é 5060 para sinalização.

- **bindaddr:** Define o endereço IP onde o servidor Asterisk irá esperar pelas conexões SIP. O comportamento padrão é esperar a conexão neste endereço IP para autenticação no Asterisk.

- **context:** Configura o contexto padrão para todos os clientes SIP.

- **allow:** Permite que um determinado codec seja utilizado.

Nas seções seguintes definem-se os parâmetros dos clientes SIP, tais como o nome do usuário, senha, endereço IP *default* e tipo de conexão, conforme abaixo:

- **type:** Configura a classe de conexão, a qual depende do sentido de autenticação da mesma. O tipo *user* é utilizado para autenticar terminais (ramais) que só recebem chamadas. O parâmetro *friend* é utilizado para autenticar terminais que realizam e recebem chamadas. O tipo *peer* é usado para autenticar terminais que executam chamadas de saída do PABX IP.

- **secret:** Senha alfanumérica utilizada para autenticar os clientes para receber e/ou efetuar chamadas.

- **username:** Configura a identificação do usuário que o Asterisk utiliza para autenticação dos clientes.

- **Host:** Configura o endereço IP ou o nome do *Host* de destino. Se for especificado um determinado endereço IP (ou nome de *Host*), isso limitará as chamadas de saída para um destino fixo (cliente). Pode-se usar a opção de configuração *dynamic*. Neste caso, a autenticação poderá ser realizada por qualquer cliente, independentemente do seu endereço IP.

- **notransfer:** Esse parâmetro configura, ativa ou desativa a facilidade de transferência de ligações nos clientes (ramais). Configurando-se com *no*, o ramal estará habilitado para efetuar a transferência de ligações. Configurado como *yes*, o ramal não possui permissão para realizar a transferência de ligações.

Definimos que os ramais SIP teriam a numeração na faixa do 8000 ao 8099. Segue o exemplo para a configuração de um ramal:

[general]

port = 5060

bindaddr = 200.19.73.18

context = voip

allow = all

[8000]

type = friend

secret = 8000

username = 8000

Host = dynamic

nottransfer = no

6.5.2 lax.conf

O *Inter-Asterisk eXchange Protocol* fornece controle e transmissão de voz sobre redes IP. O IAX pode ser usado com qualquer tipo de mídia como voz e vídeo, mas foi principalmente desenvolvido com o objetivo de trafegar voz.

Como padrão, utilizado pelo Asterisk, a configuração do IAX é realizada no arquivo */etc/asterisk/iax.conf*.

O arquivo de configuração do IAX é uma coleção de seções e cada qual representa uma entidade dentro do escopo do IAX .

A primeira seção é tipicamente a seção geral [general]. Nesta área, um número de parâmetros que afetam todo o sistema pode ser configurado. Realizamos, no projeto em questão, a configuração dos seguintes parâmetros:

- **bindport:** Configura a porta UDP que o protocolo IAX utilizará nas conexões. A porta *default* é a 4569 para sinalização e tráfego de dados.

- **bindaddr:** Endereço IP onde o servidor Asterisk irá esperar pelas conexões IAX. O comportamento padrão é esperar a conexão neste endereço IP para autenticação no Asterisk.

Nas seções seguintes definem-se os parâmetros dos clientes IAX, tais como o nome do usuário, senha, endereço IP *default*, tipo de conexão e tipo de codec, conforme abaixo:

- **bandwidth:** Este parâmetro permite a configuração da banda passante de acordo com o codec utilizado. Configurando como *high*, habilita todos os codecs que exigem conexões acima de 10Mbps. Escolhendo *medium*, habilita todos codecs com

largura de banda menor que 32Kbps. Configurando com *low* habilitam-se os codecs que utilizam baixíssima banda passante, G723 e GSM.

- **allow/disallow:** Permitem respectivamente, habilitar ou desabilitar um codec específico.

- **jitterbuffer:** Este parâmetro permite habilitar ou desabilitar o *buffer* para eliminar os efeitos de variação no atraso dos pacotes. Deve estar sempre habilitado a menos que todas as conexões estejam em uma LAN.

- **notranfer:** Esse parâmetro configura, ativa ou desativa a facilidade de transferência de ligações nos clientes (ramais). Configurando-se com *no*, o ramal estará habilitado para efetuar a transferência de ligações. Configurado como *yes*, o ramal não possui permissão para realizar a transferência de ligações.

- **auth:** Seleciona os métodos de autenticação dos clientes no servidor Asterisk. Múltiplos métodos podem ser habilitados, dentre eles: md5, *plaintext* e RSA. O *plaintext* oferece pouca segurança. Esse método exige que uma senha válida seja fornecida para autorizar uma conexão. No entanto, o fato da senha ser armazenada e transferida como texto puro, torna esse método de autenticação bastante inseguro. O método MD5 aumenta a segurança na conexão de rede, entretanto ambas as pontas ainda requerem uma senha em texto puro no arquivo *iax.conf*. O RSA fornece a melhor segurança. Neste caso, a autenticação será feita com um par de chaves RSA pública / privada.

Seguindo o mesmo raciocínio do *sip.conf*, configura-se os usuários que utilizam o protocolo IAX. Definiu-se que os ramais IAX teriam a numeração na faixa do 7000 ao 7099. Segue o exemplo para a configuração de um ramal:

[general]

bindport = 4569

bindaddr = 200.19.73.18

[7000]

type = friend

auth = md5

context = voip

secret = 7000

Host = dynamic
disallow = all
allow = gsm
bandwidth = medium
jitterbuffer = yes
nottransfer = no

6.5.3 extension.conf

O arquivo *extension.conf* é o plano de discagem do Asterisk, que gerencia as chamadas. Consiste em uma lista de instruções ou passos que o Asterisk utiliza para gerenciar as chamadas. Essas instruções são interpretadas a partir dos dígitos recebidos de um canal ou aplicação.

A configuração básica utilizada no arquivo *extension.conf* é a seguinte:

exten=>8000,1,dial(SIP/8000)

A instrução *exten=>* descreve qual o próximo passo para o encaminhamento de uma ligação. O *8000* é o conjunto de dígitos que foi recebido pelo Asterisk (número discado). O número *1* (pode-se utilizar 1, 2, 3, etc) indica a prioridade (maior número, menor prioridade) de execução da linha de comando. O parâmetro *dial*, efetua o encaminhamento da chamada para o canal especificado (SIP, IAX2 ou ZAP). Neste exemplo, qualquer ramal que discar *8000*, a ligação será encaminhada para o telefone IP registrado como *8000*.

Configurou-se o plano de discagem para os canais SIP e IAX, conforme os comandos a seguir:

exten=>8000,1,dial(SIP/8000)
exten=>8010,1,dial(SIP/8010)
exten=>8020,1,dial(SIP/8020)
exten=>8030,1,dial(SIP/8030)

exten=>7000,1,dial(iax2/7000)

```
exten=>7010,1,dial(iax2/7010)  
exten=>7024,1,dial(iax2/7024);  
exten=>7070,1,dial(iax2/7070);
```

Em seguida foi configurado o plano de discagem para o canal ZAP, interface FXO, ligações saintes do servidor Asterisk em destino ao PABX Equitel Saturno 5000. Seguem abaixo os comandos configurados:

```
exten=>7112,1,dial(ZAP/4/112)  
exten=>7113,1,dial(ZAP/4/113)  
exten=>7114,1,dial(ZAP/4/114)  
exten=>7115,1,dial(ZAP/4/115)  
exten=>7117,1,dial(ZAP/4/117)
```

O parâmetro *dial*, efetua o encaminhamento da chamada para o canal ZAP, placa TDM400P. O número “4” indica a posição do *slot* em que está instalada a interface, neste caso FXO, na placa. Os dígitos seguintes indicam o número do ramal do PABX (Equitel Saturno 5000) para o qual a ligação será encaminhada.

Para utilizarmos a facilidade de conferência no servidor Asterisk, foi necessário realizarmos a configuração do canal *meetme*, sala de conferência, no plano de discagem, conforme a seguir:

```
exten=>5555,1,Meetme(9999)
```

Neste caso, quando o usuário disca 5555 a ligação é encaminhada para uma sala de conferência. A função *Meetme* efetua o encaminhamento da chamada para a sala de conferência 9999 configurada no arquivo *meetme.conf*.

6.5.4 Zapata.conf

O arquivo de configuração *zapata.conf* contém parâmetros relacionados à sinalização dos canais TDM utilizados pelas interfaces de *hardware* zaptel. As

sinalizações dos canais são definidas neste arquivo para que possa ser utilizado pelo servidor Asterisk.

No Projeto VoIP foi feita a configuração da sinalização da placa TDM400P com uma interface FXO e uma FXS. Segue abaixo a descrição dos parâmetros configurados:

- **signalling:** Configura o tipo de sinalização para o canal utilizado na placa. Estes parâmetros devem ser coerentes com os definidos no arquivo */etc/zaptel.conf* (driver da placa). Se as instruções não forem coerentes com o arquivo *zaptel.conf*, ou se o dispositivo estiver instalado ou configurado de forma não apropriada, as interfaces não funcionarão.

- **echocancel:** É um parâmetro que possui a função de habilitar ou desabilitar o cancelamento de eco do canal e, para isso, utiliza um algoritmo específico. É recomendável que permaneça ligado (yes).

- **group:** Permite agrupar canais com o mesmo tipo de sinalização.

- **channel:** Define os canais, nos quais estão instaladas as interfaces na placa. No projeto, as interfaces FXS e FXO foram instaladas, respectivamente, nos canais 1 e 4 da placa TDM400P.

- **echotraining:** quando este parâmetro está habilitado o Asterisk envia um tom para a linha no início da conversação, para medir o eco e reduzir o tempo de aprendizado do cancelador de eco.

É importante lembrar, de que os canais FXS são sinalizados com FXO e os canais FXO são sinalizados com FXS. O Asterisk efetua a comunicação com os dispositivos internos como se estivesse do lado oposto. Segue abaixo as configurações utilizadas no projeto:

signalling=fxo_ks

echocancel=yes

echotraining=yes

group=1

channel=1

signalling=fxs_ks
echocancel=yes
echotraining=yes
group=2
channel=4

O complemento KS nas sinalizações FXO e FXS indica a sinalização *kewlstart*, que é um tipo de sinalização *loop start* (comumente usada em linhas analógicas). A sinalização *kewlstart* possui a capacidade de detectar a desconexão da chamada, quando esta é feita a partir do terminal distante.

6.5.5 Zaptel.conf

No arquivo */etc/zaptel.conf* efetua-se a configuração dos canais das interfaces das placas da Digium. Dependendo dos módulos instalados na placa TDM400P as interfaces desempenham a função de conexão a um PABX ou à rede pública. No projeto, temos o modulo FXO instalado no canal 4 e o módulo FXS instalado no canal 1. Utilizou-se os seguintes comandos para a configuração da placa:

fxsks=4
fxoks=1

Os demais parâmetros deste arquivo configuraram-se como *default*.

6.5.6 MeetMe.conf

A conferência no Asterisk é feita usando o script MeetMe(). O MeetMe() é um recurso de videoconferência muito simples de usar, com a vantagem de funcionar com qualquer tipo de canal. Alguns telefones padrão SIP, por exemplo, têm seus próprios mecanismos de conferência que também podem ser usados, mas o método padrão para conferência é o MeetMe(). Basicamente, a sala de conferência é definida através da instrução “*conf=>*”. Exemplificando, o Asterisk

recebe a solicitação de um ramal, que digitou 5555, para participar da conferência, conforme configurado em *extension.conf*. A ligação é encaminhada para a sala configurada como 9999 e é solicitada a senha para acesso (2222) à conferência. Segue abaixo as configurações do arquivo:

```
[rooms]  
conf=>9999;  
conf=>9999,2222;
```

6.5.7 Musiconhold.conf

Existem diversas formas de usar a música em espera. As opções podem ser mpg123 com *madplay*, ou baixar o mpg123 que é a opção mais comum.

No projeto optamos por utilizar o *player* mpg123, baixado do site <http://www.mpg123.de/cgi-bin/siteexplorer.cgi./mpg123/>

No arquivo *zapata.conf*, foi necessário adicionar um comando para habilitar a música em espera, conforme abaixo:

```
[channels]  
musiconhold=default
```

No arquivo *musiconhold.conf*, é especificado o arquivo (mohmp3) da música que será utilizada, conforme abaixo:

```
default=>mp3:/var/lib/asterisk/mohmp3
```

6.6 SOFTPHONES

Para efetuar a conexão dos usuários (ramais), escolhemos e testamos softphones gratuitos. Existem vários softphones facilmente encontrados para downloads na Internet, porém será mostrado um utilizado para protocolo SIP e outro para IAX, que foram testados em no projeto.

6.6.1 SOFTPHONE SIP

Foi utilizado o softphone x-lite para conexões com o protocolo SIP. Optou-se por utilizar este aplicativo, dentre diversos testados, devido sua estabilidade e interface gráfica intuitiva e de fácil instalação.

O X-Lite é um *software freeware*, que baixamos do site <http://www.xten.com>.

Para autenticação no servidor Asterisk é necessário configurar o nome de usuário (*User Name*), senha (*password*) e o endereço do servidor(*Domain*), conforme figura 22.

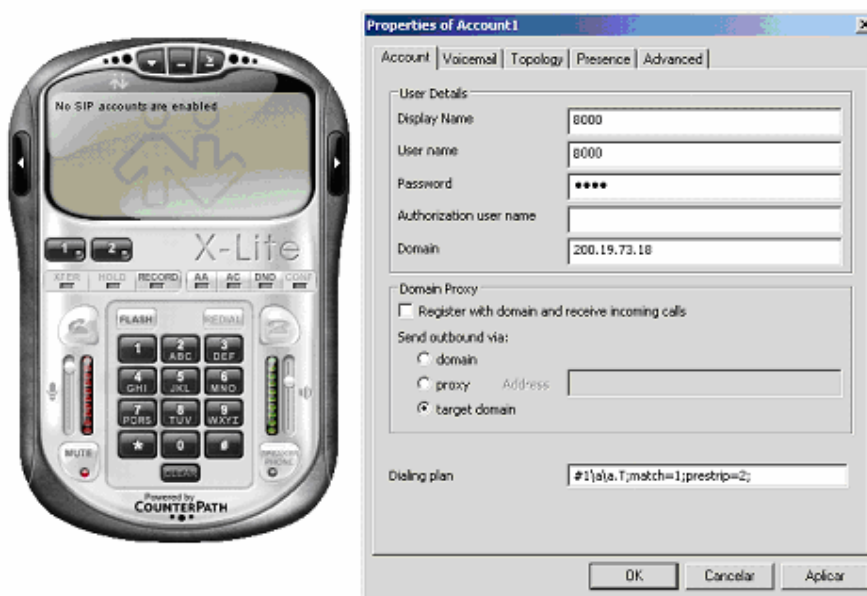


Figura 22 – Configuração X-Lite
(Fonte: Autoria própria)

6.6.2 SOFTPHONE IAX

Foi utilizado o softphone IaxComm para conexões com o protocolo IAX. Dentre os diversos aplicativos testados no desenvolvimento do projeto, optou-se por utilizar o IaxComm, devido suas características: simplicidade, eficiência e disponibilidade para o sistema operacional Linux.

O IaxComm é um *software freeware*, efetuando o *download* do site <http://iaxclient.sourceforge.net/iaxcomm/>.

A configuração deste aplicativo é relativamente simples. Para autenticação no servidor Asterisk é necessário preencher o campo *account name* com o número do ramal que foi criado no servidor. No segundo campo, *HOST*, é necessário digitar o endereço do servidor 200.19.73.18. No campo *password* é necessário digitar senha para a autenticação no servidor.

Na figura 23, temos a tela de configuração do IaxComm:



Figura 23 – Configuração IaxComm
(Fonte: Autoria própria)

Para efetuar uma ligação, é necessário digitar o número do ramal no campo *extension* e clicar em *dial*. Para se atender a uma ligação, na coluna *state*, aparece a palavra *ring*. Clicando nessa palavra aparecerá *active*, indicando que a chamada está ativa.

7 – INICIANDO A PLATAFORMA ASTERISK

Tendo-se concluído a compilação e configuração da plataforma Asterisk, sua inicialização é feita a partir do diretório do Asterisk. Para acessar o diretório é necessário abrir o console e digitar `cd /usr/sbin`, via usuário *root*, conforme a figura 24:

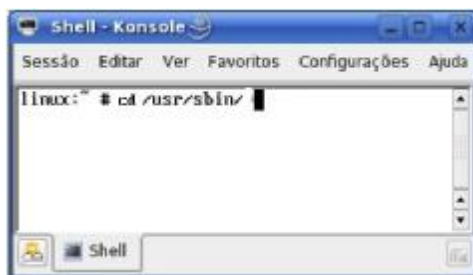


Figura 24 – Acessando o Diretório do Asterisk
(Fonte: Autoria própria)

Como foi usado um hardware de telefonia, placa TDM400P, inicia-se o serviço da placa (interfaces FXO e FXS) antes da plataforma Asterisk, através do comando `# ztcfg -vvv`, conforme a figura 25.

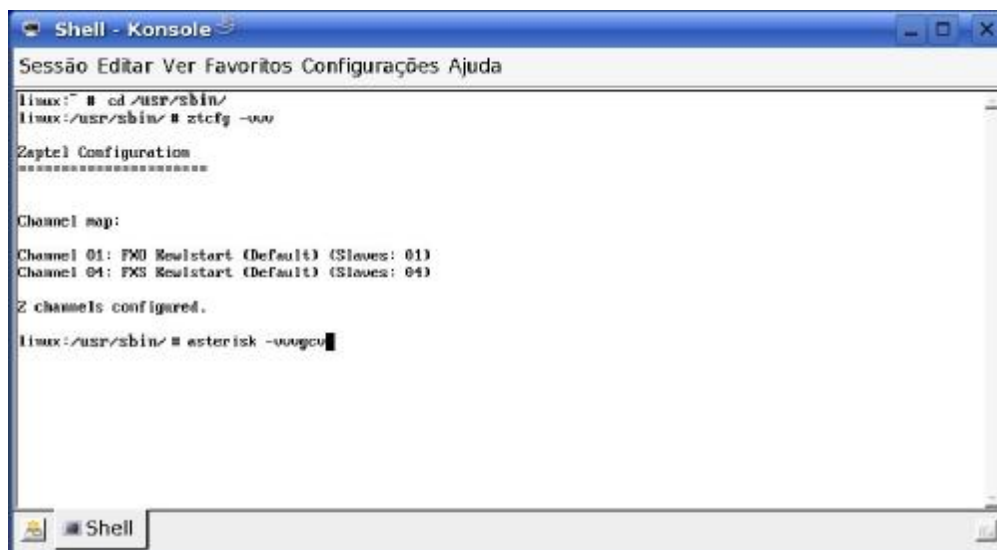


Figura 25 – Iniciando Hardware TDM400P e Plataforma Asterisk
(Fonte: Autoria própria)

Com a placa TDM400P já iniciada, inicia-se a plataforma Asterisk, propriamente dita, utilizando o comando:

#asterisk -vvvvcv

Têm-se assim, o *software* em processo de inicialização, conforme a figura 26.

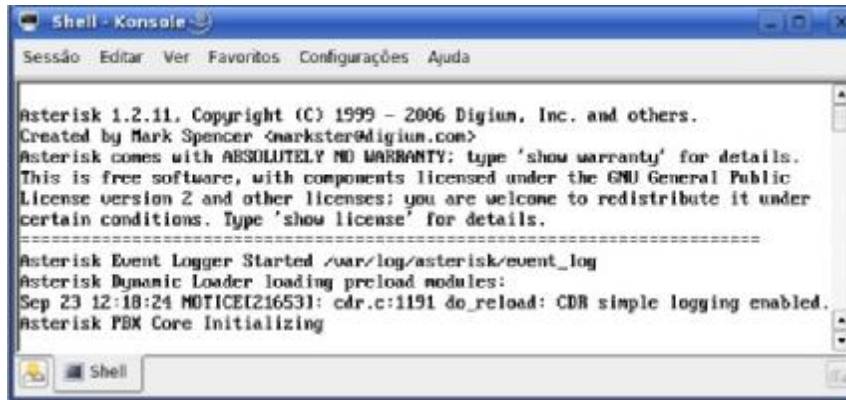


Figura 26 – Inicialização do Asterisk
(Fonte: Autoria própria)

Para se encerrar a plataforma Asterisk, utiliza-se o comando:

*CLI> stop now

7.1 VISUALIZANDO USUÁRIOS

Com a plataforma Asterisk em operação, pode-se monitorar todos os ramais configurados, através do comando:

*CLI> lax2 show peers

Com este comando, têm-se telas da seguinte forma, conforme a figura 27.

The screenshot shows a terminal window titled "Asterisk Console on 'linux' (pid 7457) - Shell - Konsole". The window contains a list of registered applications and their descriptions, followed by a command prompt where the command "iax2 show peers" has been executed. The output of the command is a table showing the status of four peers: 7070/7070, 7024, 7010/7010, and 7000/7000. The 7000/7000 peer is the only one with a status of "Unmonitored" and a port of 4569.

```
== Registered application 'MP3Player'
[app_lookupcidname.so] => (Look up CallerID Name from local database)
== Registered application 'LookupCIDName'
[format_uaw.so] => (Microsoft WAV format (8000hz Signed Linear))
== Registered file format uaw, extension(s) uaw
[app_snddtmf.so] => (Send DTMF digits Application)
== Registered application 'SendDTMF'
[format_vox.so] => (Dialogic VOX (ADPCM) File Format)
== Registered file format vox, extension(s) vox
[app_directed_pickup.so] => (Directed Call Pickup Application)
== Registered application 'Pickup'
[app_waitforring.so] => (Waits until first ring after time)
== Registered application 'WaitForRing'
[app_readfile.so] => (Stores output of file into a variable)
== Registered application 'ReadFile'
[app_setcidnum.so] => (Set CallerID Number)
== Registered application 'SetCIDNum'
[app_talkdetect.so] => (Playback with Talk Detection)
== Registered application 'BackgroundDetect'
[app_math.so] => (Basic Math Functions)
== Registered application 'Math'
[app_db.so] => (Database Access Functions)
== Registered application 'DBget'
== Registered application 'DBput'
== Registered application 'DBdel'
== Registered application 'DBdeltree'
[app_waitforsilence.so] => (Wait For Silence)
== Registered application 'WaitForSilence'

Asterisk Ready.
*CLI> iax2 show peers
Name/Username      Host                Mask                Port                Status
7070/7070           (Unspecified)      (D) 255.255.255.255 0                    Unmonitored
7024                (Unspecified)      (D) 255.255.255.255 0                    Unmonitored
7010/7010           (Unspecified)      (D) 255.255.255.255 0                    Unmonitored
7000/7000           200.17.97.40       (D) 255.255.255.255 4569                Unmonitored
4 iax2 peers [0 online, 0 offline, 4 unmonitored]
*CLI>
```

Figura 27 – Visualização dos Clientes Ativos
(Fonte: Autoria própria)

Como pode ser visualizado na figura 27, o ramal 7000 foi autenticado no servidor. Os demais ramais, 7010, 7024 e 7070, estão aguardando autenticação.

7.2 ENCAMINHAMENTO DAS LIGAÇÕES NO IPBX ASTERISK

No desenvolvimento deste trabalho, uma das primeiras questões que precisou ser definida foi o tipo de protocolo que seria utilizado para o tráfego VoIP. Inicialmente, optou-se por utilizar o protocolo SIP, devido ao grande número de aplicativos VoIP compatíveis com este protocolo, também, por ser um protocolo bastante indicado na literatura técnica (Meggelen, Smith e Madsen, 2005) para tráfego VoIP.

Porém, desistiu-se de utilizar esse protocolo, devido à política rígida de segurança na rede da UTFPR, que não permitiu efetuar a liberação no *firewall* da rede, das portas utilizadas pelo protocolo SIP, para acesso externo. A porta 5060 é

utilizada para sinalização, e as portas de 10000 a 20000, para o tráfego de pacotes, além da porta tcp 2000 utilizada para iniciar a autenticação VoIP (independente do protocolo utilizado).

Tendo em vista a impossibilidade da utilização do protocolo SIP, iniciaram-se estudos e testes com o protocolo IAX, proprietário do Asterisk.

O protocolo IAX utiliza uma única porta UDP, 4569, para realizar sinalização e tráfego de pacotes VoIP. A UTFPR efetuou a liberação da porta 4569 no *firewall* da rede, o que possibilitou a continuidade deste projeto.

O protocolo IAX explora uma característica dos pacotes UDP (protocolo de transporte), de poderem transmitir em tempo real, importantíssimo para o tráfego VoIP. Outra característica é que o UDP possui baixo *overhead*, ou seja, os cabeçalhos dos pacotes têm um tamanho reduzido. O UDP não solicita retransmissão dos pacotes que por ventura tenham sido perdidos, o que para a voz é muito importante.

Outro parâmetro qualitativo, essencial para o desenvolvimento do projeto, foi a escolha do tipo de codec a ser utilizado. Realizou-se algumas conexões com os Codecs: G726, GSM, iLBC e Speex. Dentre eles, o codec GSM apresentou a melhor qualidade de voz e compatibilidade com o *softphone* laxComm. Esse codec utiliza 13Kbps de largura de banda, sem a necessidade de licenças.

No projeto desenvolvido existem diversas formas de conexões e ligações, conforme será descrito na sequência.

7.2.1 Efetuando Ligações VoIP – VoIP

Após várias tentativas de comunicação entre ramais VoIP, conseguiu-se realizar uma primeira ligação, utilizando dois ramais IAX (Softphone). Ao longo do trabalho, foram criados ramais adicionais e conseguiu-se gerar várias ligações simultâneas, sendo todas gerenciadas pelo IPBX.

Com uma qualidade sonora audível e de fácil manuseio, foram estabelecidas conversações entre clientes Linux – Linux, entre ambientes Windows – Windows e até mesmo entre os dois sistemas operacionais, Linux – Windows, conforme a figura 28. Nesta figura, exemplifica-se uma conexão entre dois ramais (7000 e 7070)

autenticados no servidor Asterisk, através da Internet. Neste exemplo, o ramal 7000 efetua uma ligação para o 7070.

A chamada é encaminhada com base no número de destino, o servidor Asterisk destina a chamada para o endereço IP associado ao ramal 7070. Um sinal de *ringing* é então enviado ao *softphone* do ramal 7070. Após o atendimento, inicia-se a fase de conversação. A desconexão da chamada pode ser iniciada tanto pelo ramal 7000 (originador da chamada) como pelo 7070 (receptor da chamada).

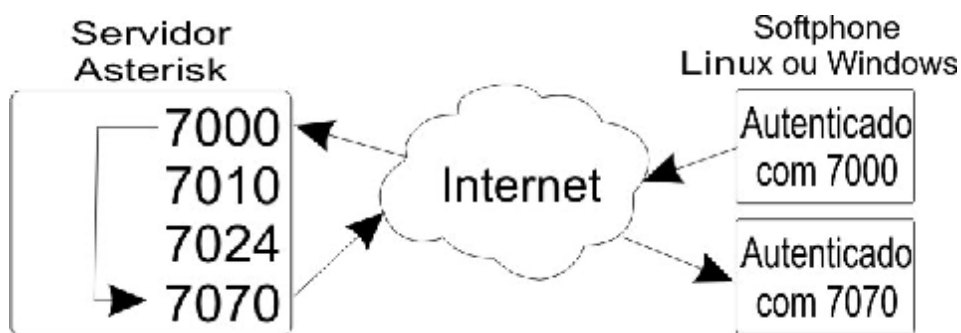


Figura 28 - Ligação entre Ramais VoIP
(Fonte: Autoria própria)

Como pode ser visualizado na figura 29, o ramal 7010 (softphone IAX) realiza uma chamada para o ramal 7000 (softphone IAX), a chamada é atendida e depois desligada.


```
Asterisk Console on 'linux' (pid 7908) - Shell - Konsole
Sessão Editar Ver Favoritos Configurações Ajuda

== Registered application 'WaitForRing'
[app_readfile.sol => (Stores output of file into a variable)
== Registered application 'ReadFile'
[app_setcidnum.sol => (Set CallerID Number)
== Registered application 'SetCIDNum'
[app_talkdetect.sol => (Playback with Talk Detection)
== Registered application 'BackgroundDetect'
[app_math.sol => (Basic Math Functions)
== Registered application 'Math'
[app_db.sol => (Database Access Functions)
== Registered application 'DBget'
== Registered application 'DBput'
== Registered application 'DBdel'
== Registered application 'DBdeltree'
[app_waitforsilence.sol => (Wait For Silence)
== Registered application 'WaitForSilence'
Asterisk Ready.
*CLI> -- Accepting AUTHENTICATED call from 200.19.73.18:
> requested format = gsm,
> requested prefix = (),
> actual format = gsm,
> host prefix = (gsm),
> priority = nine
-- Executing Dial("IAx2/7010-2", "iax2/7000") in new stack
-- Called 7000
-- Call accepted by 200.17.97.40 (format gsm)
-- Format for call is gsm
-- IAX2/7000-4 is ringing
-- IAX2/7000-4 answered IAX2/7010-2
-- Attempting native bridge of IAX2/7010-2 and IAX2/7000-4
-- Channel 'IAX2/7000-4' ready to transfer
-- Channel 'IAX2/7010-2' ready to transfer
-- Releasing IAX2/7010-2 and IAX2/7000-4
-- Hungup 'IAX2/7000-4'
-- Spawn extension (default, 7000, 1) exited non-zero on 'IAX2/7010-2'
-- Hungup 'IAX2/7010-2'
```

Figura 29 – Visualização de Chamada *Softphone / Softphone*
(Fonte: Autoria própria)

7.2.2 Efetuando Ligações PABX – PABX

As ligações entre os ramais do PABX funcionam de forma direta bastando discar o número do ramal desejado. No desenvolvimento do projeto não foram necessárias alterações no plano de discagem do PABX. A figura 30, exemplifica uma ligação do ramal 112 para o 117.

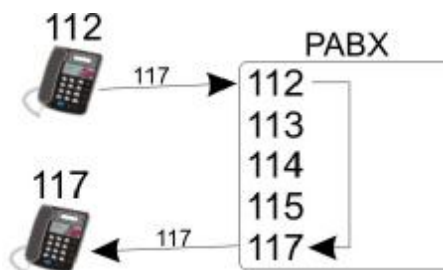


Figura 30 - Ligação entre Ramais da Central Saturno 5000
(Fonte: Autoria própria)

7.2.3 Efetuando Ligações entre o IPBX Asterisk e a Central PABX

Para conexão do IPBX (PABX VoIP Asterisk) com a central PABX Saturno 5000 do laboratório da UTFPR, efetuou-se a configuração e conexão física das rotas de saída e de entrada no IPBX e PABX, conforme a figura 31.

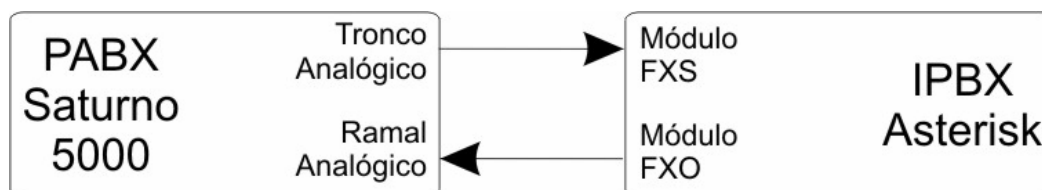


Figura 31 – Interligação entre o IPBX Asterisk e Central PABX
(Fonte: Autoria própria)

7.2.3.1 Ligações Saintes do IPBX Asterisk

As ligações saintes do IPBX Asterisk com destino aos ramais do PABX são efetuadas com a configuração de um canal dedicado, entre a interface FXO e uma posição de ramal analógico.

Discando-se de um ramal softphone o número 7117, por exemplo, a plataforma Asterisk recebe a informação, gerencia e comuta para o módulo FXO (ver tópico 6.5.3 para entender o roteamento da ligação). Este, por sua vez, transmite a sinalização da chamada para o ramal 117 do PABX, conforme a figura 32.

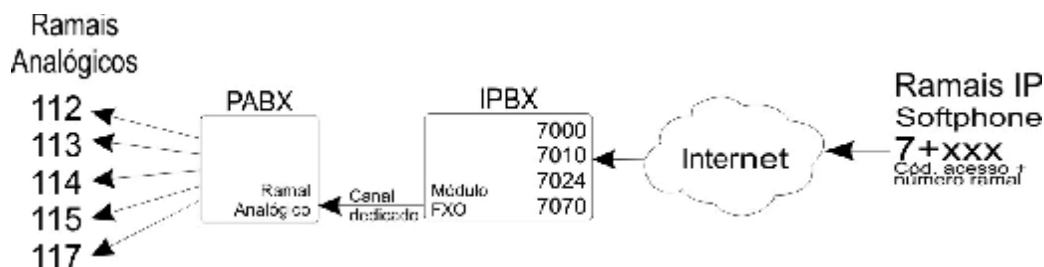
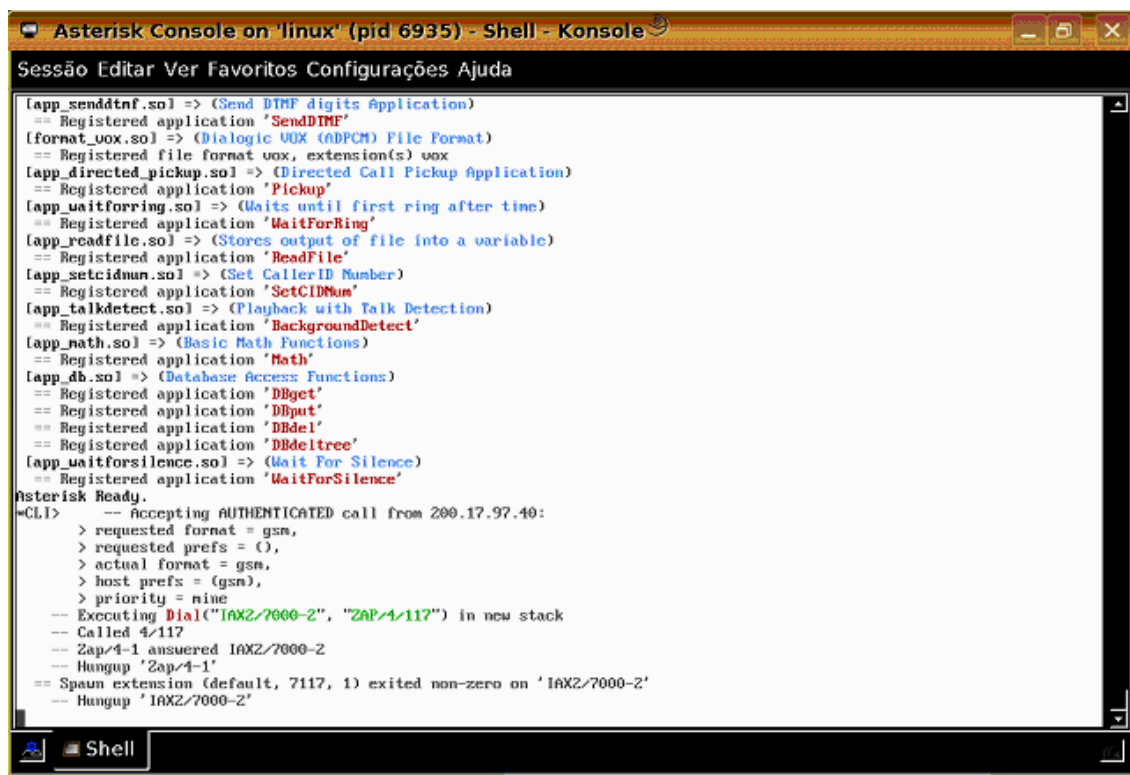


Figura 32 - Ligações *Softphone* / PABX
(Fonte: Autoria própria)

Como pode ser visualizado na figura 33, o ramal 7000 (softphone IAX) realiza uma chamada para o ramal 7117 (ramal analógico 117 da central Equitel Saturno 5000), a chamada é atendida e logo após desligada.



```
Asterisk Console on 'linux' (pid 6935) - Shell - Konsole
Sessão Editar Ver Favoritos Configurações Ajuda

[app_senddtmf.sol => (Send DTMF digits Application)
== Registered application 'SendDTMF'
[format_vox.sol => (Dialogic VOX (ADPCM) File Format)
== Registered file format vox, extension(s) vox
[app_directed_pickup.sol => (Directed Call Pickup Application)
== Registered application 'Pickup'
[app_waitforring.sol => (Waits until first ring after time)
== Registered application 'WaitForRing'
[app_readfile.sol => (Stores output of file into a variable)
== Registered application 'ReadFile'
[app_setcidnum.sol => (Set CallerID Number)
== Registered application 'SetCIDNum'
[app_talkdetect.sol => (Playback with Talk Detection)
== Registered application 'BackgroundDetect'
[app_nath.sol => (Basic Math Functions)
== Registered application 'Math'
[app_db.sol => (Database Access Functions)
== Registered application 'DBget'
== Registered application 'DBput'
== Registered application 'DBdel'
== Registered application 'DBdeltree'
[app_waitforsilence.sol => (Wait For Silence)
== Registered application 'WaitForSilence'
Asterisk Ready.
=CLI> -- Accepting AUTHENTICATED call from 200.17.97.40:
> requested format = gsm,
> requested prefs = (),
> actual format = gsm,
> host prefs = (gsm),
> priority = nine
-- Executing Dial("IAX2/7000-2", "ZAP/4/117") in new stack
-- Called 4/117
-- Zap/4-1 answered IAX2/7000-2
-- Hungup 'Zap/4-1'
-- Spawn extension (default, 7117, 1) exited non-zero on 'IAX2/7000-2'
-- Hungup 'IAX2/7000-2'
```

Figura 33 – Visualização de Chamada *softphone* / PABX.
(Fonte: Autoria própria)

7.2.3.2 Ligações Entrantes no IPBX Asterisk

As ligações entrantes no IPBX Asterisk oriundas dos ramais do PABX são realizadas com a utilização de um canal dedicado configurado, entre a interface FXS e uma posição de tronco analógico.

A central PABX Equitel Saturno 5000 foi configurada através do acesso à porta de comunicação serial de um microcomputador do laboratório, via o *software* específico para tal procedimento (Hyperterminal do Windows).

Programou-se um código de acesso (digito 7) na central PABX destinado à conexão de um canal dedicado entre o PABX (tronco analógico) e a plataforma IPBX Asterisk (Interface FXS).

A configuração do tronco analógico na central Equitel Saturno 5000 foi realizada utilizando os seguintes comandos: *featacc*, *trkgrp*, *trkassn*, *service* e *custdata*.

Com o comando *trkgrp* foram programados os parâmetros relativos ao grupo de acesso, *trunk group number* = 2, com as configurações de sinalização de *loop* e *DTMF* para tronco analógico a dois fios.

No comando *featacc* configurou-se o dígito, *access code range* = 7, para acesso ao grupo de tronco, definido no comando *trkgrp*.

O comando *trkassn* foi utilizado para efetuar configuração do circuito de tronco analógico (3020) no grupo 2, definido no comando *trkgrp*. Para realizar esta configuração, foi necessário utilizar o comando *service* para tirar o circuito 3020 de serviço, possibilitando assim a alteração do circuito para o grupo 2. Efetuada a configuração no circuito de tronco analógico, é necessário recolocá-lo em serviço com o comando *service*.

Efetuu-se a configuração da central PABX de tal modo que, digitando o código de acesso 7 dos ramais analógicos da central PABX (112, 113, 114, 115 e 117) estabelecemos a conexão ao canal dedicado, entre o tronco analógico da central PABX e a interface FXS do servidor IPBX. Após o acesso ao canal, discando os quatro números do ramal desejado, estabelecendo a conexão da origem, ramal analógico da central PABX, ao destino, ramal softphone do IPBX Asterisk.

O *custdata* salva todas alterações realizadas na base de dados da central telefônica.

Exemplo de discagem para o ramal 7000: primeiramente digita-se o código de acesso 7 (conexão do canal dedicado), e em seguida os dígitos 7000 para encaminhamento da ligação ao usuário autenticado, conforme a figura 34.

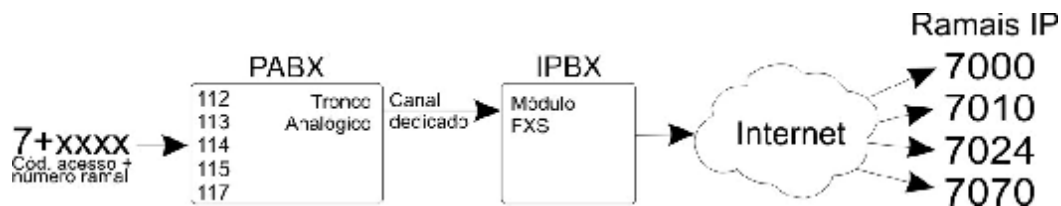
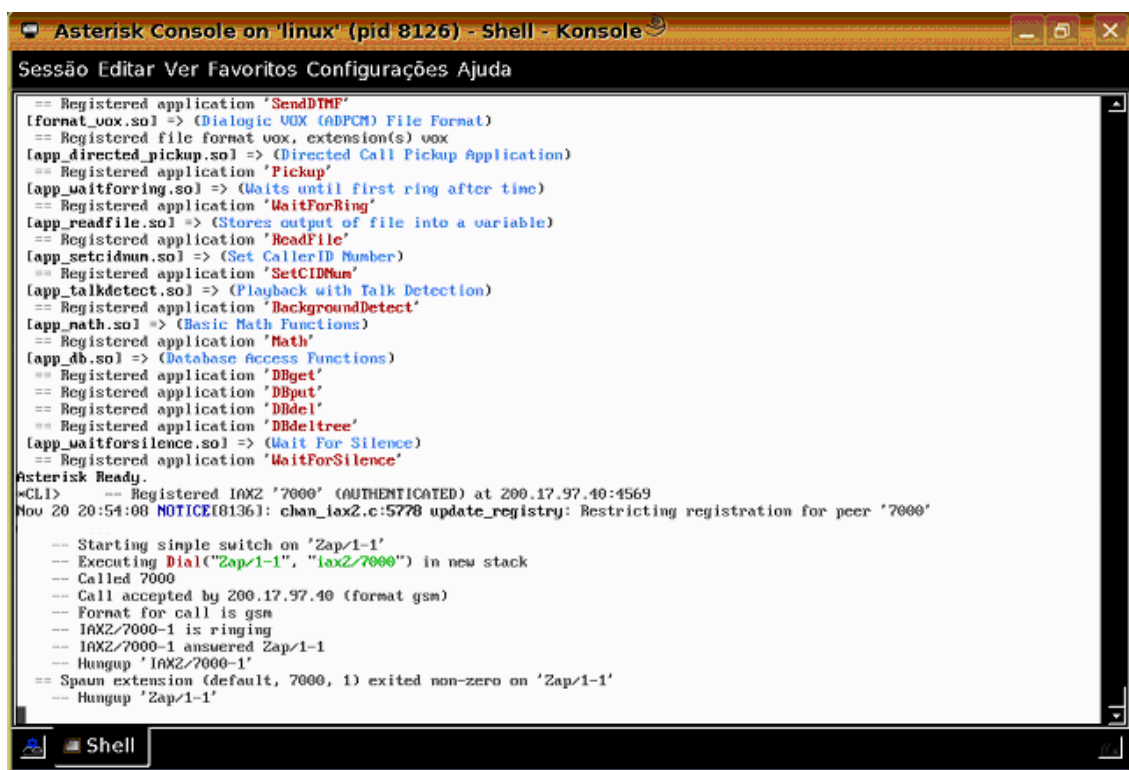


Figura 34 - Ligações PABX / Softphone
(Fonte: Autoria própria)

Como pode ser visualizado na figura 35, um ramal analógico da central Equitel Saturno 5000 (discagem: 7 + 7000), realiza uma chamada para o ramal 7000 (softphone IAX), a chamada é atendida e logo após desligada.



```
Asterisk Console on 'linux' (pid 8126) - Shell - Konsole
Sessão Editar Ver Favoritos Configurações Ajuda

== Registered application 'SendDTMF'
[format_vox.sol => (Dialogic VOX (ADPCM) File Format)
== Registered file format vox, extension(s) vox
[app_directed_pickup.sol => (Directed Call Pickup Application)
== Registered application 'Pickup'
[app_waitforring.sol => (Waits until first ring after time)
== Registered application 'WaitForRing'
[app_readfile.sol => (Stores output of file into a variable)
== Registered application 'ReadFile'
[app_setcidnum.sol => (Set CallerID Number)
== Registered application 'SetCIDNum'
[app_talkdetect.sol => (Playback with Talk Detection)
== Registered application 'BackgroundDetect'
[app_math.sol => (Basic Math Functions)
== Registered application 'Math'
[app_db.sol => (Database Access Functions)
== Registered application 'DBget'
== Registered application 'DBput'
== Registered application 'DBdel'
== Registered application 'DBdeltree'
[app_waitforsilence.sol => (Wait For Silence)
== Registered application 'WaitForSilence'

Asterisk Ready.
*CLI> -- Registered IAX2 '7000' (AUTHENTICATED) at 200.17.97.40:4569
Nov 20 20:54:00 NOTICE[8136]: chan_iax2.c:5778 update_registry: Restricting registration for peer '7000'

-- Starting simple switch on 'Zap/1-1'
-- Executing Dial("Zap/1-1", "iax2/7000") in new stack
-- Called 7000
-- Call accepted by 200.17.97.40 (format gsm)
-- Format for call is gsm
-- IAX2/7000-1 is ringing
-- IAX2/7000-1 answered Zap/1-1
-- Hungup 'IAX2/7000-1'
-- Spawn extension (default, 7000, 1) exited non-zero on 'Zap/1-1'
-- Hungup 'Zap/1-1'
```

Figura 35 – Visualização de Chamada PABX / Softphone
(Fonte: Autoria própria)

No apêndice B são apresentadas as configurações detalhadas do tronco analógico na central telefônica Equitel Saturno 5000 para comunicação com IPBX Asterisk, através da interface FXS.

8 – DIFICULDADES ENCONTRADAS

Houve a dificuldade na escolha do tema do projeto. Hoje, o assunto VoIP ainda causa medo, desconfiança e curiosidade a muita gente que não convive diretamente com esta tecnologia. Quanto ao Asterisk, nem idéia tinha do que se tratava.

Outra dificuldade foi encontrar material bibliográfico com conteúdo detalhado sobre o funcionamento de todo esse sistema, principalmente a respeito do Asterisk.

Acredita-se que a maior dificuldade foi não entender, de primeiro momento, o motivo de não conseguir estabelecer uma comunicação VoIP com o protocolo SIP (primeiro utilizado). Descobriu-se que o protocolo SIP trabalha com a porta UDP 5060 para o acesso e para a troca de sinalização do IPBX, também utiliza uma faixa de portas UDP (10000 a 20000) para o tráfego de mídia. No caso, quando a UTFPR disponibilizou um ponto de rede para conexão do servidor, por motivos de segurança, o *firewall* bloqueou essas portas. Tinha-se só a porta TCP 80 liberada para os testes.

Após os estudos realizados sobre o tema e principalmente sobre a dificuldade encontrada, descobriu-se que o protocolo IAX (proprietário do Asterisk) utiliza a porta UDP 4569, tanto para sinalização como tráfego de mídia.

Com a intervenção dos professores Luis Vieira e Vagner Leitão, conseguiu-se o desbloqueio da porta UDP 4569 junto ao departamento de Informática (AINFO) da UTFPR, o que proporcionou tanto a comunicação interna quanto o acesso externo pelos usuários, provido pelo endereço IP fixo e válido.

9 - CONCLUSÃO

A tecnologia VoIP já é uma realidade e está revolucionando o mundo das comunicações. Seus recursos avançados, facilidade de uso e menores custos são grandes atrativos, tanto para consumidores quanto para empresas de telefonia.

Por se tratar de um assunto ainda relativamente novo, ainda não é incorporado ao currículo de muitos cursos de formação técnica. Entretanto, devido à demanda cada vez maior do mercado, por profissionais capacitados a entender sistemas VoIP, nos próximos anos certamente haverá mudanças neste quadro.

Na fase da elaboração desta proposta de trabalho, não foi encontrado nada de semelhante ao Asterisk, dentre as plataformas VoIP disponíveis no mercado. O Asterisk tem com principal vantagem o fato de ser um *software* livre, e de poder ser executado em plataforma Linux.

Neste trabalho, foram úteis os conceitos de telefonia convencional e de rede de computadores vistos em algumas disciplinas do curso de tecnologia em eletrônica – modalidade comunicações, da UTFPR. Novos temas, tais como protocolos utilizados em VoIP e PABX IP, embora não tenham sido abordados nas disciplinas do curso, serviram como desafio de pesquisa e aprendizado para a equipe. Uma das contribuições deste trabalho foi, justamente, ampliar as possibilidades do ensino de plataformas VoIP nos cursos de tecnologia e/ou engenharia da UTFPR. Estas possibilidades não são apenas teóricas, mas também práticas, com a implementação de um servidor IPBX (Asterisk) funcionando de forma integrada à central PABX convencional (Equitel Saturno 5000), instalada no laboratório de comunicações (Q-201) do departamento de eletrônica da UTFPR – campus Curitiba.

Com o IPBX Asterisk implementado e a interface com o PABX convencional instalada e configurada, é possível utilizar este sistema para outros trabalhos de diplomação, tais como:

- Desenvolvimento de uma plataforma didática de *call center*.
- Implementação e teste de conexão digital (link E1) entre o IPBX Asterisk e um PABX digital convencional. Para tanto, seria necessária a aquisição de uma interface E1 para barramento PCI, a ser instalada no IPBX.

- Implementação de um projeto piloto de comunicação VoIP entre a sede da UTFPR – campus Curitiba, localizado no centro, e as novas instalações do campus, localizado no bairro Ecoville.

REFERÊNCIAS BIBLIOGRÁFICAS

TANENBAUM, A.S. **Redes de Computadores**. Tradução por Insight Serviços de Informática. 3. ed. Rio de Janeiro: Campus, 1997. Tradução de: Computer Networks.

FAGUNDES, E. **A convergência das redes de voz**.

Disponível em: <<http://www.efagundes.com/Artigos/A%20Convergencia%20das%20redes%20de%20voz.htm>>. Acesso em: 26 maio 2006.

PEREIRA, L. G. A. **Tutorial IPv6**. Apresenta a nova versão do protocolo da camada de rede da arquitetura internet, ou TCP/IP.

Disponível em: <<http://penta2.ufrgs.br/redes296/ipv6/>>. Acesso em: 2 junho 2006.

PUC-RJ. Apostila de “**Internet e arquitetura TCP/IP**”. 2. ed. Rio de Janeiro: [s.n.], [199-]. 2v. Disponível em: <<http://www.pegar.com.br/arquivos/1215.zip>>. Acesso em: 2 junho 2006.

XAVIER, S. **Voz sobre IP na PBH**. PUC-MG, 2000. Monografia (Pós Graduação em Informática Pública).

Disponível em: <<http://www.pbh.gov.br/informatica/programaformacao/especializacao/sidney.pdf>>. Acesso em: 21 julho 2006.

GONÇALVES, F. E. A. **Asterisk PBX**.

Disponível em: <www.asteriskbrasil.org/downloads>. Acesso em: 22 agosto 2006.

PINHEIRO, B. O. **Voz sobre IP Utilizando Asterisk**. LAVRAS-MG, 2005. Monografia (Pós Graduação em Administração de Redes Linux).

Disponível em: <<http://www.ginux.ufla.br/documentacao/monografias/mono-BrunoPinheiro.pdf>>. Acesso em: 7 junho 2006, 23h52

Apêndice A

Instalação do Sistema Operacional Linux

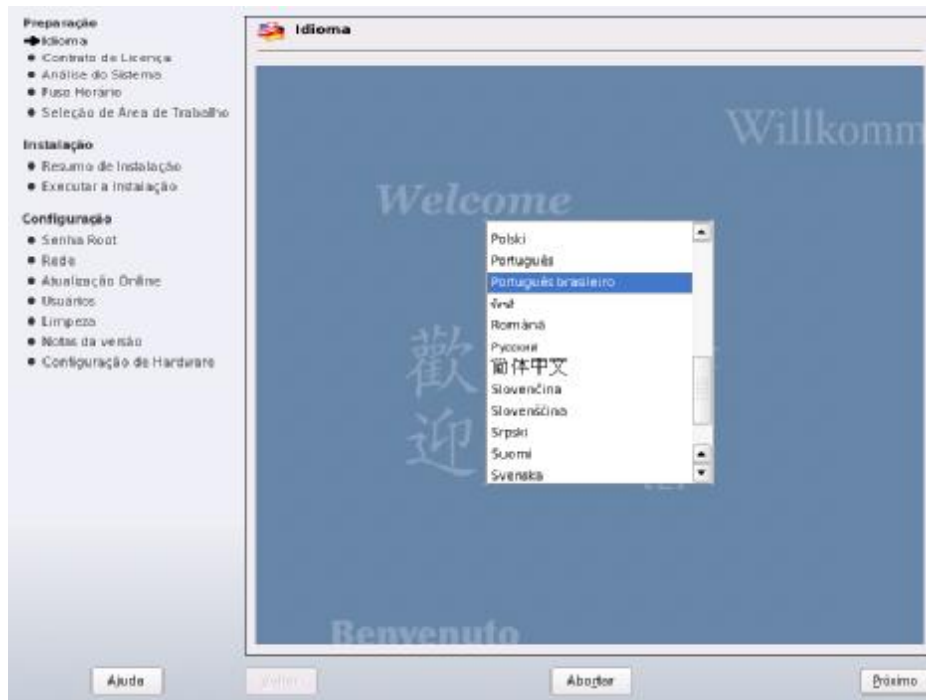
Neste trabalho foi utilizado o Suse Linux. Este é uma distribuição desenvolvida e mantida pela Novell.

O início da instalação e o decorrer da mesma são simples. Ligamos o microcomputador já com o CD1 no leitor e selecionamos no *setup* do microcomputador o tipo de Boot para *Fist Boot CD-ROM*.

Passo 1: boot usando o disco 1 do Suse 10.



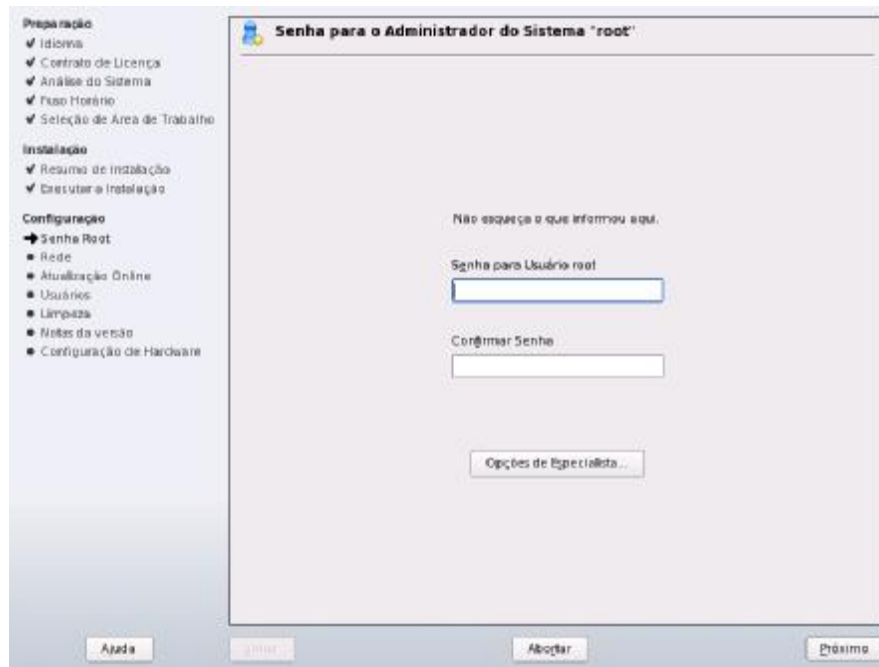
Passo 2: Escolha do idioma.



Passo 3: Escolha o item Software



Passo 4: Digitação da senha do super usuário Root



Passo 5: Após a instalação utilizando o Yast instale os seguintes pacotes:

- Kernel Sources
- gcc - GNU C Compiler and Support Files
- cvs – Concurrent Versions System
- ncurses – New curses libraries
- ncurses-devel – Bibliotecas para desenvolvimento com ncurses
- bison – The GNU parser generator
- Termcap – Termcap library
- openssl – Secure Sockets and TLS Layer Security
- openssl-devel – Bibliotecas do openssl.
- zlib-devel

Apêndice B

Configuração do PABX Siemens Saturno 5000

>DIS FEATACC

ACCESS CODE OR FEATURE(A,F) ?=A

ACCESS CODE RANGE=0

DISPLAY FEATACC 92-09-01 20:10:39

ACC CODE	FEATURE	DATA
----------	---------	------

0 TRUNK GROUP ACCESS	TRKGRP GROUP =1
---	--------------------------	-----------------

DISPLAY COMPLETE

>DIS FEATACC

ACCESS CODE OR FEATURE(A,F) ?=A

ACCESS CODE RANGE=1

DISPLAY FEATACC 92-09-01 20:12:56

ACC CODE	FEATURE	DATA
----------	---------	------

7 DT_VACANT_CODE	
---	----------------------	--

DISPLAY COMPLETE

>ADD FEATACC

NEW ACCESS CODE/RANGE(MAX.250)=7

FEATURE ASSIGNED TO ACCESS CODE=TRKGRP

TRUNK GROUP NUMBER(0-95)=2

ADD FEATACC 92-09-01 20:26:10

DIS F

FEATURE/ACCESS CODE ADDED

>DIS FEATACC

ACCESS CODE OR FEATURE(A,F) ?=A

ACCESS CODE RANGE=7

DISPLAY FEATACC 92-09-01 20:26:36

ACC CODE	FEATURE	DATA
----------	---------	------

7 TRUNK GROUP ACCESS TRKGRP GROUP =2
DISPLAY COMPLETE

>DIS TRKASSN

PORT EQUIPMENT NUMBER (WXYZ)=3020

DISPLAY TRKASSN 92-09-02 12:18:20

PORT EQUIPMENT NUMBER = 3020

TRUNK GROUP NUMBER = 2

TRUNK TYPE = THKZSR

TRUNK MEMBER NUMBER = 0

TRUNK TYPE = 2WAY

DIGITE DIRECT TRUNK = NO

NEXT TRUNK IN LINK = 3020 ;

DIT EXTENSION NUMBER =

NITE ANSWER TYPE = NONE

ANA EXTENSION NUMBER = ----

APPEARANCE MAP NUMBER = ----

END OF DISPLAY

>

DISPLAY TRUNK GROUP(S) (0-95)=2

DISPLAY ALL TRUNKS(Y,N) ?=Y

DISPLAY TRKGRP 92-09-02 12:21:42

TRUNK GROUP NUMBER = 2

TRUNK GROUP TYPE = 2WAY

ALTERNATE TRUNK GROUPS = -- -- --

ALTERNATE LINK GROUP(S) = --

ATTENDANT DISPLAY CHARACTER= NONE

BUSY LAMP THRESHOLD = 0

TRKGRP NAME DISP FLAGS:

NONE

TRUNK GROUP PRIORITY LEVEL = 0

DISP USE OF RESTRICTION LISTS:

NONE

DISP STATION COS RESTRICTION:

NONE

TRKGRP DASS2 RELATED? : N

DISPLAY MISCELLANEOUS FLAGS:

NONE

TRKGRP ASSIGNED TO ATND GRP: 0

ACCESS CODES IN TG-DATA-TBL: 7

OUTGOING ADDRESS SIGNALLING= DTMF

OUTGOING CALL USAGE = CO

OUTGOING CONTROL SIGNALLING= DIALTN

TOLL CO DIGIT COUNT = 0

TOLL CO DIGITS = --

OUTGOING FLOAT LOOP NUMBER = --

DIGIT ECHO COUNT = 0

INCOMING ADDRESS SIGNALLING= AUTO

INCOMING CALL USAGE = CO

INCOMING CONTROL SIGNALLING= IMMED

MFC GROUP NUMBER = 0

PCM PROTOCOL NUMBER = NONE

ACD GROUP NUMBER = --

STATION COS FOR TRK GROUP = 0

TRUNK DATA COS = --

INCOMING FLOAT LOOP NUMBER = --

INCOMING DID DIGIT EXPECTED= 0

INC DIGIT DELETE COUNT = 0

INC DIGIT PREFIX COUNT = 0

INC DIGITS TO BE PREFIXED = ----
INC DIGIT CONV TABLE NUMBER= --
DID NITE ANSWER TYPE = NONE
NITE ANSWER EXTENSION NUM =

CONN DELAY TIMER TO TRUNK = 1
AWAIT DIAL TONE TIMER = 100
EM SEIZURE TIMER = 8
TRUNK RELEASE TIMER = 30
TRUNK RESEIZE TIMER = 30
LONG OUTDL INTER DGT TMR = 100
SHORT OUTDL INTER DGT TMR = 40
OUTPUT INTER DIGIT TIMER = 8
TRK REANSWER TIMER = 900

COS RSTR : BLOCK/TONE : BUSY
STN BUSY : BLOCK/TONE : BUSY
STN DND : BLOCK/TONE : BUSY
STN OOS : BLOCK/TONE : BUSY
DATA PRV : BLOCK/TONE : BUSY
STN UNAS : BLOCK/TONE : BUSY

DISPLAY ALL TRUNKS:
3020

END OF DISPLAY

>DIS SERVICE
PORT EQUIPMENT NUMBER (WXYZ)=302
MODE (<CR> CURRENT SERVICE MODE)=
PORT EQUIPMENT NUMBER (WXYZ)=3020
MODE (<CR> CURRENT SERVICE MODE)=
DISPLAY PHANTOM DIGITE LST?(N,Y)=

EXTENSION NUMBER? =

DISPLAY SERVICE 92-09-01 20:21:39

PEN 3020 IN SERVICE

END OF SERVICE DISPLAY

>SAVE CUSTDATA

SAVE TO (ACTIVE,STANDBY,BOTH)=ACTIVE

SAVE CUSTDATA 92-09-01 20:31:16

.....

CUSTOMER MEMORY DATA SAVED

END OF SAVE DISK BACKUP

OUTROS TRABALHOS EM:
www.projetoderedes.com.br