



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

THIAGO PLANET DA SILVA

SOLUÇÕES VOIP BASEADAS EM SOFTWARE LIVRE

ASSIS
2009



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

SOLUÇÕES VOIP BASEADAS EM SOFTWARE LIVRE

THIAGO PLANET DA SILVA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito de Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Analisador (1): _____

Analisador (2): _____

ASSIS
2009



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

THIGO PLANET DA SILVA

SOLUÇÕES VOIP BASEADAS EM SOFTWARE LIVRE

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Área de Concentração: INFORMÁTICA

Assis
2009

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus por acreditar que nossa existência pressupõe uma outra infinitamente superior, a minha família, amigos e professores que acreditaram em mim.

AGRADECIMENTOS

Primeiramente a Deus, pela coragem, força e saúde que tem me dado durante todos os dias de minha vida.

Aos meus pais, Dirceu e Marinalva, que sempre me apoiaram em toda a minha caminhada, apoiando nos momentos mais difíceis que passei, também pela educação que me permitiram ter, pelas pessoas de caráter e honestidade que são e que assim me ensinaram a ser.

A minha irmã, Elaine, por sempre me incentivar, desde o início até agora no final desta jornada da graduação.

Aos amigos, pela amizade e incentivos constantes.

A professora Marisa, pela orientação sempre oportuna, esclarecedora, inteligente e pelo seu incentivo, fazendo com que este trabalho pudesse ser realizado, minha eterna gratidão.

A comunidade de Software Livre, pela disponibilidade e colaboração.

E a todos os demais que contribuíram, direta ou indiretamente, para a realização desse trabalho.

RESUMO

Este trabalho se propõe a trazer informações que definam o que é Voz sobre IP, conhecida também pelos termos VoIP, telefonia IP, telefonia Internet, voz sobre banda larga. Também para explicar seu funcionamento e principais características como: arquitetura, topologias, protocolos e dificuldades técnicas enfrentadas.

O trabalho tem como objetivo o desenvolvimento de um protótipo para demonstração da tecnologia VOIP. O software foi desenvolvido com o uso do protocolo SIP da IETF que fará toda a transmissão de pacotes em uma rede TCP/IP usando protocolo UDP.

O sistema PLANETSIP foi desenvolvido na plataforma de programação JAVA, a linguagem foi escolhida devido às facilidades que ela oferece para o desenvolvimento desta aplicação. Contudo o usuário poderá realizar chamadas em uma rede local ou de mesmo domínio, porque o sistema é muito simples de ser utilizado.

Palavras-chaves: Voz, VoIP, Telefonia IP, SIP,IETF, JAVA, UDP

ABSTRACT

This paper aims to bring information to define what is Voice over IP, also known by the terms VoIP, IP telephony, Internet telephony, voice over broadband. Also to explain their operation and main characteristics such as: architecture, topologies, protocols and technical difficulties encountered.

The work aims to develop a prototype for demonstration of the VOIP technology. The software was developed using the SIP protocol that the IETF will make the transmission of packets over a TCP / IP using UDP protocol.

PLANETSIP

The system was developed on the Java programming language was chosen due to the facilities it offers to the development of this application. However, the User can make calls on a local network or the same domain, because the system is very simple to use.

Keywords: Voice, VoIP, IP Telephony, SIP, IETF, JAVA, UDP

LISTA DE ILUSTRAÇÕES

FIGURA 1 - PILHA DE PROTOCOLO H.323.....	13
FIGURA 2 - TELEFONIA IP ENTRE DOIS COMPUTADORES	18
FIGURA 3 - ARQUITETURA COMPLETA DA REDE VOIP. FONTE: [2].	20
FIGURA 4 - TOPOLOGIA PONTO A PONTO EM UMA REDE VOIP.	22
FIGURA 5 - REDE COM PROVEDOR VOIP.	23
FIGURA 6 - ESTRUTURA DE PROTOCOLOS VOIP.....	26
FIGURA 7 - ENTIDADE SIP. FONTE: [7].	29
FIGURA 8 - ORGANIZAÇÃO DOS PROTOCOLOS – SDP. FONTE: [9].	30
FIGURA 9 - CABEÇALHO RTP. FONTE: [10].	31
FIGURA 10 - SINALIZAÇÃO SIP PC PARA PC.	39
FIGURA 11 - SINALIZAÇÃO SIP PC PARA SERVIDOR. FONTE: [8].	40
FIGURA 12 - TERMINAL IP PARA TELEFONE.	45
FIGURA 13 - COMUNICAÇÃO DOS EQUIPAMENTOS COM SERVIDOR SIP.	46
FIGURA 14 - COMUNICAÇÃO ENTRA EQUIPAMENTOS COM BANCO DE DADOS.....	47
FIGURA 15 - DIAGRAMA CASO DE USO.	51
FIGURA 16 - DIAGRAMA DE CLASSES.....	52
FIGURA 17 - DIAGRAMA DE SEQÜÊNCIA INICIANDO CHAMADA	53
FIGURA 18 - DIAGRAMA DE SEQÜÊNCIA RECEBENDO CHAMADA.	53
FIGURA 19 - DIAGRAMA DE SEQÜÊNCIA ENVIAR PEDIDO BYE.....	54
FIGURA 20 - DIAGRAMA DE SEQÜÊNCIA RECEBENDO PEDIDO BYE	55
FIGURA 21 - INTERFACE SISTEMA PLANETSIP	60

LISTA DE TABELAS

TABELA 1 - CAMPOS DO CABEÇALHO SIP	34
TABELA 2 - ESPECIFICAÇÕES DO SOFTWARE.....	49
TABELA 3 - MÉTODOS SIPLISTENER.....	59
TABELA 4 - FUNCIONALIDADES DA INTERFACE.	60

SUMÁRIO

INTRODUÇÃO.....	12
1.1- OBJETIVO	15
1.2- JUSIFICATIVA.....	15
1.3- MOTIVAÇÕES	16
1.4- ESTRUTURA DO TRABALHO.....	16
FUNDAMENTAÇÃO TEÓRICA BÁSICA.....	18
2.1- AMBIENTE PARA TELEFONIA IP.....	18
2.2- VOZ	19
2.3- ARQUITETURA VOIP.....	19
2.3.1- TOPOLOGIAS	21
2.3.1.1-Ponto a Ponto.....	21
2.3.1.2- Com provedor Voip	23
2.3.2- Vantagens e desvantagens com uso de Voip	24
2.3.2.1- Vantagens de usar Voip	24
2.3.2.2- Desvantagens de usar Voip	25
2.3.3- Tipos de protocolos	25
2.3.3.1- H.323	26
2.3.3.2- PROTOCOLO SIP	27
2.3.3.3- PROTOCOLO SDP (<i>SESSION DESCRIPTION PROTOCOL</i>)	30
2.3.3.4- RTP (REAL-TIME PROTOCOL)	31
2.3.3.5- MGCP (Media Gateway Control Protocol)	32
2.3.3.6- Aplicação do protocolo SIP	33
2.3.3.6.1- Mensagens de Requisição	33
2.3.3.6.2- MENSAGENS DE RESPOSTA	34
2.3.3.6.3- CAMPOS DE CABEÇALHO.....	34
2.3.3.6.4- FLUXOS DE CHAMADAS	38
2.3.3.6.4.1- SIP PC para PC	38
2.3.3.6.4.2- SIP PC / servidor / PC	39
2.4- Banco de Dados	40
2.5- Linguagem de Programação	41
2.5.1- Linguagem Java	41
2.5.2- Máquina Virtual	41
2.5.3- Edições de JAVA.....	42
2.5.4- IDEs Java	42
DESENVOLVIMENTO DO SISTEMA DE VOZ	44
3.1-Descrição do problema.....	44
3.2- Modelagem do problema	44
DESENVOLVIMENTO DA APLICAÇÃO	48
4.1- REQUISITOS PRINCIPAIS DO PROBLEMA.....	48

4.1.1- ESPECIFICAÇÃO.....	48
4.1.2- USE CASES	48
4.1.3- DESCRIÇÃO DOS USE CASES	49
4.2- Modelo de Caso de Uso	50
4.2.1- Diagrama de classes.....	51
4.2.2- Diagrama de Seqüência.....	52
4.2.3- Assunto iniciando chamada SIP	52
4.2.4- Recebendo chamada	53
4.2.5- Enviar um pedido de encerramento de chamada (BYE).....	54
4.2.6- Recebendo um pedido de encerramento da chamada(BYE).....	54
IMPLEMENTAÇÃO DA APLICAÇÃO	56
5.1- Transmissão e recepção de dados.....	56
5.2- Interface do Sistema PLANETSIP	60
CONCLUSÃO.....	62
REFÊRENCIAS BIBLIOGRÁFICAS	64

CAPÍTULO 1

INTRODUÇÃO

O surgimento da tecnologia VoIP (Voz sobre IP ou *Voice over IP*) vem mudando, em ritmo acelerado, a forma como as pessoas se comunicam, sendo uma das grandes revoluções tecnológicas ocorridas no século 21. O VoIP é a tecnologia que utiliza as redes de dados baseadas no protocolo IP para a transmissão de voz, em tempo real e na forma de pacotes de dados. Mas, para que essa tecnologia pudesse se tornar funcional, foi criado o conceito de Telefonia IP, que são os serviços diversos de telefonia oferecidos por provedores e/ou operadoras VoIP afim de diversificar as aplicações e utilização de soluções de voz em redes de dados IP. Esta tecnologia vem ganhando cada vez mais a atenção de empresas e usuários da telefonia convencional, ampliando a diversidade de serviços de voz a serem oferecidos, aproximando pessoas geograficamente distantes, aumentando as funcionalidades dos softwares e reduzindo os custos de comunicação quando comparada às convencionais ligações telefônicas interurbanas e internacionais [1].

O software está presente em praticamente todas as atividades e relações humanas e além de ser um conjunto de instruções executadas automaticamente pelo computador, é também uma representação de conhecimento. Combinando este conhecimento, com o poder dos computadores e a inovação da tecnologia VoIP temos a oportunidade de criar uma solução de comunicação de voz por meio de redes de dados que provê toda a estrutura necessária para a redução de custos com telefonia e com licenças de software, permitindo investimentos em áreas vitais (empresariais ou governamentais), o acesso a informação sem restrições e promovendo o desenvolvimento tecnológico.

VoIP (Voz sobre IP ou *Voice over IP*) é o método que utiliza as redes de dados baseadas no protocolo IP (TCP/UDP/IP) para a transmissão de voz, em tempo real e na forma de pacotes de dados. Para que essa tecnologia pudesse funcionar, foi criado o conceito de Telefonia IP, que são os serviços diversos de

telefonia oferecidos por provedores e/ou operadoras VoIP afim de diversificar as aplicações de soluções de voz em redes de dados IP. Uma arquitetura que é bem difundida no mercado para transmitir voz sobre *Local Area Network* (LAN) é o padrão H.323, proposto pela *Internation Telecom Union* (ITU-T). A figura 1 mostra a arquitetura H.323.

Aplicações de áudio	Aplicações de Video	Controle e Gerenciamento			
G.711 G.723 G.728 G.729	H.261 H.263	RTCP	H.225.0	H.225.0 Canal de Sinalização de Chamada	H.245 Canal de Controle
RTP			Canal RAS	X.224 Classe 0	
UDP			TCP		
Nível de Rede (IP)					
Nível de Enlace					
Nível Físico					

Figura 1 - Pilha de protocolo H.323

A H.323 fornece três protocolos de controle: sinalização de chamada H.225.0, RAS H.225.0 e o controle de mídia H.245. O H.225.0 é usado em conjunto como o H.323 e fornece a sinalização para controle de chamada. É uma pilha de protocolos que está focado na conexão e controle da chamada, que são separados da transmissão de conteúdo (voz) entre os computadores.

Para a transmissão do fluxo de voz, será utilizado o protocolo do modelo de referência *Transmission Control Protocol/Internet Protocol* (TCP/IP) chamado *Real Time Protocol* (RTP). O RTP utiliza o serviço de transporte *User Datagram Protocol* (UDP) para transmitir os pacotes. O RTP é necessário porque na Telefonia IP uma taxa de transmissão constante é fundamental, enquanto que a perda de pacotes pode ser desprezada.

Na arquitetura VoIP, a rede é projetada para o roteamento e transporte de pacotes de dados, e suporta um número maior de possibilidades de serviços. Os terminais são inteligentes, seu endereçamento independe de sua

localização geográfica, e funções específicas vinculadas a uma chamada VoIP podem ser aplicadas a equipamentos diferentes, localizados em qualquer área da rede.

A arquitetura de rede TCP/IP é o nome que se dá a toda a seqüência de protocolos usados pela Internet. Esta seqüência de protocolos foi desenvolvida pela *Defense Advanced Research Project Agency* (DARPA) no Departamento de Defesa dos Estados Unidos (DoD). Este conjunto de protocolos foi desenvolvido para permitir que os computadores compartilhem recursos (arquivos, documentos, etc.) em uma rede. Toda a família de protocolos inclui um conjunto de padrões que especificam os detalhes de como comunicar computadores, assim como também convenções para interconectar redes e ratear o tráfego. O TCP é o protocolo da camada de transporte, que oferece um serviço confiável e orientado à conexão, já o protocolo IP foi implementado como um protocolo de comunicação para controle de tráfego baseado na regra do melhor esforço (*Best-effort Service*), que não provê nenhum mecanismo de qualidade de serviços e conseqüentemente, nenhuma garantia de alocação de recursos da rede. Com o crescimento da Internet, a tendência atual é a integração de voz (telefonia) e dados numa única infra-estrutura de redes de pacotes IP.

Neste projeto serão utilizadas ferramentas computacionais *open source* (software livre) com o objetivo de diminuir os custos operacionais do sistema. O Software Livre ou Free Software, conforme a definição de software livre criada pela Free Software Foundation, é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição, a cada dia vem crescendo este tipo de filosofia, beneficiando a todos em comunidades de software livre. A forma usual de um software ser distribuído livremente é sendo acompanhado por uma licença de software livre (como a GPL ou a BSD), e com a disponibilização do seu código-fonte. É bom saber que software livre é diferente de software com domínio publico, com o primeiro, usando-se (como as licenças GPL e BSD), os direitos autorais são todos do programador e/ou organização, o segundo abre mão de todos os direitos autorais, renunciando a propriedade do programa, beneficiando a todos, tornando-se um bem comum. O Free Software teve seu início em 1983, dando abertura ao projeto GNU e,

em seguida, à Free Software Foundation. Software Livre se refere à existência simultânea de quatro tipos de liberdade, que são:

- **liberdade de executar o programa**, para qualquer propósito (liberdade nº 1)
- **liberdade de estudar como o programa funciona**, e adaptá-lo para as suas necessidades (liberdade nº 2). Acesso ao código-fonte é um pré-requisito para esta liberdade.
- **liberdade de redistribuir cópias**, de modo que você possa ajudar ao seu próximo (liberdade nº 3).
- **liberdade de aperfeiçoar o programa**, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 4). Acesso ao código-fonte é um pré-requisito para esta liberdade.

O software é considerado livre se tiver as quatro liberdades, contudo será livre para redistribuir cópias, podendo ser modificadas ou não [3].

1.1- OBJETIVO

O objetivo do trabalho é desenvolver um protótipo de um software de Sistema de Voz sobre IP (VoIP) para a Internet através da especificação de um sistema telefônico baseado na arquitetura de uma Intranet. O trabalho visa especificar uma solução alternativa para a utilização da infra-estrutura de rede na comunicação entre funcionários de uma empresa que não dependa da infra-estrutura de telefonia tradicional, possibilitando que empresas obtenham uma sensível redução de custos operacionais.

O desenvolvimento será efetuado com uso de software livre, e os dados dos perfis dos usuários, serão armazenados em um banco open-source para que todo o custo do sistema seja o mais acessível possível, com isso empresas de grande ou médio porte possa ter acesso a este sistema.

1.2- JUSTIFICATIVA

O desenvolvimento deste trabalho se justifica pelo fato da tecnologia VoIP

ser definida como qualquer aplicação telefônica usada em uma rede de dados que utiliza o Protocolo de Internet (IP). Esta tecnologia utiliza um novo modo para originar e receber chamadas telefônicas que usa a conexão de internet banda larga em vez da linha telefônica convencional. Ela permite montar sua rede de comunicação, interligando filiais, fornecedores, distribuidores, representantes, etc. Além de permitir levar sua linha telefônica com seu número local para qualquer parte do mundo. Basta ter acesso à internet. Com isso, você pode fazer e receber chamadas como se estivesse na sua empresa. E um dos atrativos desta tecnologia sem dúvida é a possibilidade de se comunicar com baixíssimo custo.

1.3- MOTIVAÇÕES

O interesse nesta tecnologia surgiu com a necessidade de reduzir custos no setor de comunicação da empresa na qual trabalho. E pelas análises realizadas esta tecnologia está sendo utilizada em muitas empresas, mas ainda pouco eficiente com relação a software de gerenciamento e monitoramento. A falta de solução disponível para empresas corporativas é que motivou a desenvolver uma pesquisa nesta área usando software livre.

1.4- ESTRUTURA DO TRABALHO

O trabalho está subdividido em capítulos que serão explicitados a seguir.

O primeiro capítulo apresenta a contextualização e justificativa para o desenvolvimento da proposta do trabalho.

O segundo capítulo apresenta toda a teoria para o desenvolvimento da tecnologia VOIP, detalhando alguns conceitos importantes sobre: arquitetura, protocolos, etc.

O terceiro capítulo mostra os passos do problema abordado no trabalho, que será dividido em módulos para facilitar o entendimento.

O quarto capítulo trata sobre o desenvolvimento do trabalho, mostrando os diagramas de classe, casos de uso e diagramas de seqüência.

O quinto capítulo aborda a implementação do protótipo, com trechos de código para exemplificar o funcionamento do Protocolo SIP (Session Initiation Protocol).

O sexto capítulo apresenta as considerações finais, abrangendo as conclusões do desenvolvimento deste trabalho, as dificuldades encontradas e as sugestões para próximos trabalhos.

O sétimo capítulo mostra as referências bibliográficas.

CAPÍTULO 2

Fundamentação Teórica Básica

Neste capítulo, será feita a fundamentação teórica básica utilizada para o desenvolvimento do projeto. Esta fundamentação envolve os principais ambientes para telefonia IP e os protocolos mais adequados para a solução de problemas, o banco de dados utilizado para o armazenamento de informações e linguagem de programação que será utilizada para a implementação do sistema.

2.1- AMBIENTE PARA TELEFONIA IP

O termo telefonia IP não é limitado somente às chamadas entre dois computadores. Na realidade, a telefonia IP se refere à comunicação de voz sobre pacotes transportados na rede que utilizam o protocolo IP para terminais onde o fluxo é decodificado em voz novamente. Esses terminais podem tanto ser computadores quanto telefones convencionais modulados.

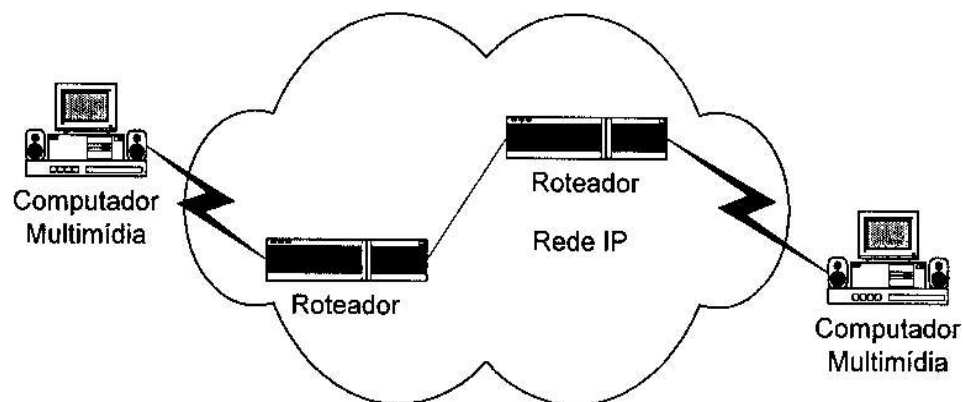


Figura 2 - Telefonia IP entre dois computadores [13]

A figura dois mostra a arquitetura de telefonia IP entre dois computadores, sendo o modelo mais simples da telefonia IP. Esta arquitetura é composta por dois usuários se comunicando sem que haja utilização de rede telefônica convencional. A codificação da voz é feita pelos computadores envolvidos e a transmissão é feita através da rede IP.

2.2- VOZ

Pode-se definir a voz humana como uma forma de onda mecânica com frequências principais na faixa que vai de 300 Hz a 3,4 kHz. No uso da voz para telefonia há uma grande preocupação com a reprodução com relação a distância do terminal receptor, pois existe uma perda na qualidade.

Já para a telefonia digital, como o próprio nome já diz a voz é codificada em formato digital, que é multiplexado no tempo de forma a compartilhar meios de transmissão. Com o sinal digital de áudio a telefonia tem um grande ganho em termos de performance, tendo vantagens como: baixa taxa de ruído, uma certa estabilidade e reprodutividade.

São duas as formas usadas para a codificação da voz: o baseado na forma de onda e o baseado nos padrões de voz. A codificação baseada na forma de onda já é utilizada hoje na telefonia convencional para digitalizar a voz, de forma a permitir a multiplexação dos circuitos. Já os baseados nos padrões de voz, são conhecidos como *voice codecs*, ou simplesmente *vocoders*, que utilizam os dispositivos que são responsáveis pela codificação da voz, esses elementos são responsáveis pela codificação da voz em um fluxo de bits, utilizando técnicas de compressão de voz e supressão de silêncio.

2.3- ARQUITETURA VOIP

Na telefonia tradicional (comutada), a arquitetura é toda baseada nas grandes centrais telefônicas. Elas são interligadas de forma hierárquica e concentra toda sua inteligência da rede, os terminais dependem do local onde

se encontram, e não possuem inteligência. Já na arquitetura VoIP, ela é toda projetada para o roteamento de pacotes de dados, suportando maior possibilidades de serviços, e seus terminais são inteligentes, não dependendo de localização, suas funções podem ser aplicadas em qualquer equipamento, que pode estar em qualquer local da rede. A figura 3 apresenta toda a arquitetura VoIP.

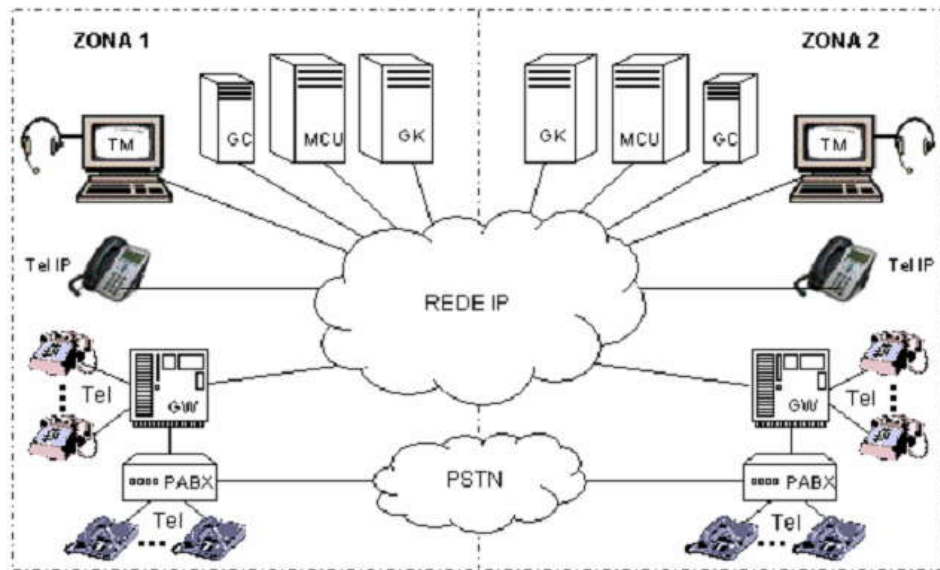


Figura 3 - Arquitetura completa da rede VOIP. Fonte: [2].

A seguir serão feitas as descrições de cada termo citado da arquitetura VOIP:

- **Rede IP:** Os dados em uma rede IP são enviados em pacotes, usando o protocolo TCP/IP.
- **PABX** (Private Automatic Branch Exchange): É o centro de distribuição de telefonia pertencente a uma empresa(privada ou publica) que não inclua como sua atividade o fornecimento de serviços telefônicos ao público em geral, permitindo fazer ligações internas, ou mesmo interurbanas e internacionais.
- **PSTN** (Public Switched Telephone Network): É a rede comutada, que faz a ligação da telefonia entre as residências e as empresas, nacional ou internacional.

- **Terminal Telefônico Convencional** (Telefone): É o aparelho telefônico usado em residências ou empresas.
- **Terminal Telefônico IP** (Telefone IP): É o telefone que disponibiliza todas as funcionalidades da tecnologia Voip.
- **Terminal Multimídia** (TM): São os terminais que foram desenvolvidos para atender as necessidades de comunicação Voip, e tem softphones instalados.
- **Gateway** (GW): É o aparelho que faz a interligação com IP e PSTN, sua principal função é converter mídia em tempo (voz analógica x voz digital comprimida) e a conversão de sinalização para chamadas telefônicas.[2]
- **Gateway Controller** (GC): É o equipamento que faz o controle das chamadas pelo GW, fazendo acompanhamento, desde início ao término da ligação entre 2 terminais.
- **Multipoint Control Unit** (MCU): É responsável pela conferência de 3 ou mais terminais, contendo Controlador Multiponto (MC), Processador Multiponto (MP).
- **Gatekeeper** (GK): Faz o gerenciamento de um conjunto de telefonia IP, IP, TM, GW, GC e MCU.
- **Zona**: É um conjunto de terminais, GW'S e MCU's gerenciados por um único GK, em uma zona deve ter ao menos um terminal, e deve ter somente um GK.

2.3.1- TOPOLOGIAS

Para VOIP existe 2 tipos de rede que são: topologia ponto-a-ponto e com provedor Voip. Em seguida iremos ver suas funcionalidades.

2.3.1.1-Ponto a Ponto

A figura 4 mostra a topologia ponto a ponto.

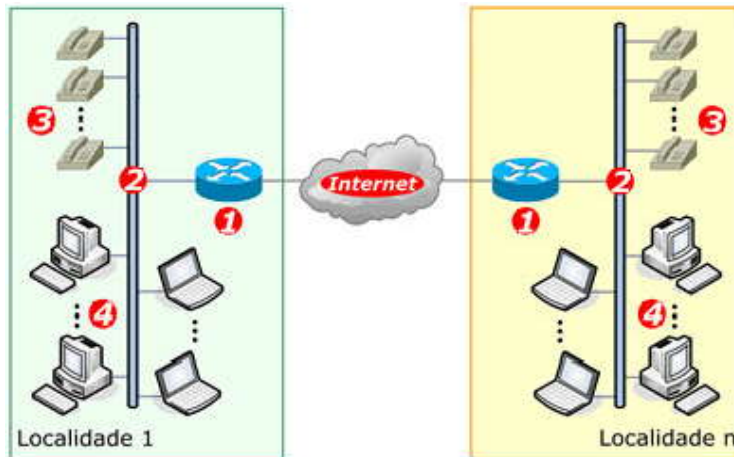


Figura 4 - Topologia ponto a ponto em uma rede Voip.

Detalhes da topologia:

1. **Modems / Roteadores:** aparelhos que fazem a interface da rede local com a Internet.
2. **Rede Local:** abrange toda a rede local.
3. **Telefone IP ou telefone analógico:** Possui adaptador para VoIP (ATA).
4. **Computadores desktop ou Notebooks:** que podem ser utilizados para telefonia IP desde que possuam softwares *softphones*.

A topologia ponto a ponto não tem nenhum tipo de equipamento que faz o gerenciamento de suas conexões, desta forma a transmissão é feita com os dados à rede IP ou Internet, devido a não ter nenhum aparelho que faz o gerenciamento, não é possível fazer chamadas a telefonia convencional (PSTN), limitando assim suas funcionalidades somente aos terminais, a chamada é feita da seguinte forma:

- O terminal 3 ou 4 disca para o número de um dos terminais da localidade 2.
- O protocolo de sinalização tenta emitir um sinal de toque ao terminal remoto da localidade 2 através da rede 2 e do equipamento 1 e, em seguida, a Internet.
- Quando a outra extremidade aceitar a chamada é estabelecido um fluxo de dados sobre RTP para a transferência da voz em pacotes de dados.
- A conexão é mantida até que uma das extremidades cancele a conexão.

2.3.1.2- Com provedor Voip

A figura 5 mostra a topologia com provedor VOIP.

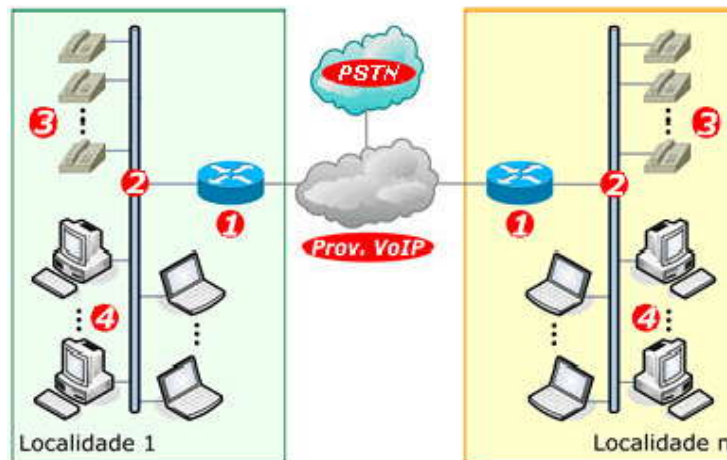


Figura 5 - Rede com provedor VoIP.

Detalhes da Topologia:

1. **Modems / Roteadores:** Aparelhos de interface da rede local com a Internet ou a rede do provedor VoIP.
2. **Rede Local:** Abrange toda a rede local.
3. **Telefone IP ou telefone analógico:** Possui adaptador para VoIP (ATA).
4. **Computadores desktop ou Notebooks:** que podem ser utilizados para telefonia IP desde que possuam softwares *softphones*.

Esta topologia possui um controlador, que tem como função gerenciar, todo o fluxo Voip, com isso a ligação é feita entre o terminal para o servidor, e assim o servidor quem faz a conexão entre o chamador e o chamado. Com esta tecnologia é viável fazer ligações para telefones convencionais (PSTN), e fazer todos os serviços que a rede Voip disponibiliza como e-mails de voz, secretária eletrônica, dentre outros.

A chamada é realizada da seguinte forma:

- O terminal 3 ou 4 disca para o número de um dos terminais da localidade 2;
- O protocolo de sinalização tenta emitir um sinal de toque ao provedor VoIP utilizado através da rede 2 e do equipamento 1;
- O provedor VoIP busca em sua base de dados e verifica se o número de destino é um número VoIP ou um número da telefonia fixa convencional;
- Caso seja um número VoIP o mesmo verificará se o número está registrado no servidor e está conectado, dessa forma redirecionando a conexão até o seu destino e estabelecendo um fluxo de dados sobre RDP para a transferência da voz em pacotes de dados. Caso contrário, o servidor não solicita a chamada ou retorna uma mensagem ao usuário;
- Caso seja um número da telefonia fixa convencional, o servidor faz a discagem para o número externo através de troncos de voz e ao efetuar a discagem converte tudo que chega digital para analógico e vice-versa;
- A conexão é mantida até que uma das extremidades cancele a conexão.

2.3.2- Vantagens e desvantagens com uso de Voip

Verificando o sistema de telefonia tradicional e o VoIP nota-se que se trata de uma verdadeira revolução. Tem muitas vantagens com o VoIP devido a sua natureza, que é atrelada a internet, e claro, algumas desvantagens. Seguem alguns exemplos.

2.3.2.1- Vantagens de usar Voip

O custo é o principal atrativo para a adoção da telefonia VoIP. Muitas empresas já estão enxergando isso, e aos poucos as pessoas estão descobrindo que se pode economizar muito nas ligações interurbanas e internacionais. Será uma verdadeira revolução quando todos adquirirem e confiarem neste serviço.

O telefone VoIP oferece Mobilidade, isto é sem dúvida um ótimo recurso, o mais interessante depois do custo. A pessoa poderá partir para qualquer lugar do mundo com seu aparelho, e onde existir conexão com a internet, poderá utilizar normalmente. As chamadas feitas e recebidas serão

automaticamente roteadas para ele. Pelo fato de funcionar na Web poderá se agregar com outros serviços, como conversação com vídeo, mensageiros instantâneos, compartilhamento de arquivos e outros.

2.3.2.2- Desvantagens de usar Voip

Uma grande desvantagem de usar Voip nas empresas ou órgãos públicos, é preciso um bom investimento financeiro inicial para a implementação da infra-estrutura necessária. Outra grande desvantagem é na falta de energia elétrica, em alguns lugares pode parar o funcionamento do serviço, diferente da telefonia convencional PSTN que mesmo na falta de energia elétrica, continua funcionando.

Um outro problema é no caso da rede não ter sido roteada corretamente, onde pode existir interferência na qualidade da voz nas ligações. Quando não há um mecanismo eficiente de QoS que garanta uma largura de banda dedicada para essas chamadas, os usuários irão escutar um som distorcido na comunicação. Há técnicas para soluções VoIP que fazem detecção de silêncio e cancelamento de eco com intuito de melhorar a qualidade da ligação Voip.

2.3.3- Tipos de protocolos

O mecanismo de suportar uma conexão VoIP geralmente envolve uma série de transações de sinalização entre os pontos-finais (e *gateways* entre eles), como por exemplo: estabelecer conexão, determinar ponto de destino, roteamento entre os pontos. Há também questões de sinalização de telefonia como campainha, identificador de chamada e desconexão. Existem diversos protocolos para manipular isso, como mostra a Figura 6. Para isso, detalharemos a seguir as características dos protocolos utilizados pela tecnologia VoIP.

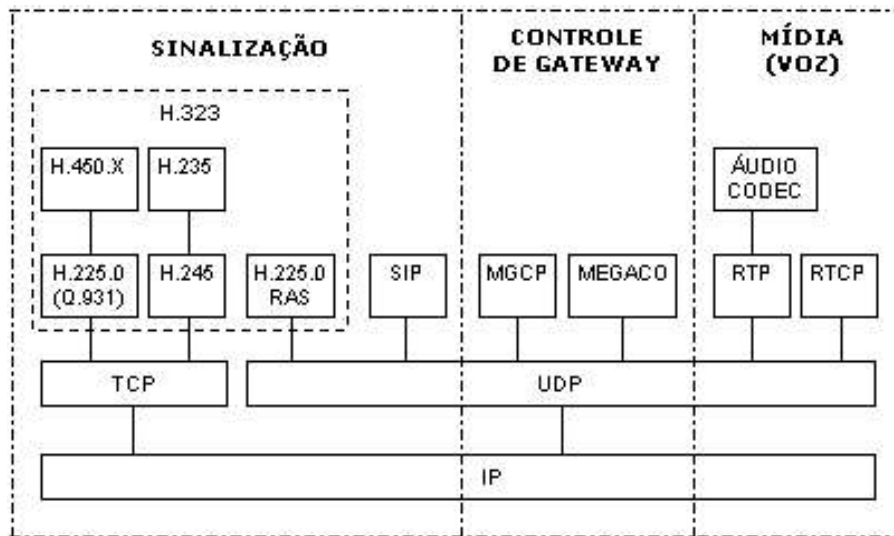


Figura 6 - Estrutura de Protocolos Voip.

Os protocolos principais para gateway de VoIP, o H.323, MGCP (Media Gateway Control Protocol), SDP (Session Description Protocol), SIP (Session Initiation Protocol).

2.3.3.1- H.323

O padrão H.323 é parte da família de recomendações ITU-T (International Telecommunication Union Telecommunication Standardization sector) H.32x, que pertence a série H da ITU-T, e que trata de "Sistemas Audiovisuais e Multimídia". A recomendação H.323 tem o objetivo de especificar sistemas de comunicação multimídia em redes baseadas em pacotes e que não provêm uma Qualidade de Serviço (QoS) garantida. Além disso, estabelece padrões para codificação e decodificação de fluxos de dados de áudio e vídeo. Redes baseadas em pacotes incluem as redes IP (Internet Protocol) como a Internet, redes IPX (Internet Packet Exchange), as redes metropolitanas, as redes de longa distância (WAN) e ainda conexões discadas usando PPP. O padrão H.323 é completamente independente dos aspectos relacionados à rede. Dessa forma, podem ser utilizadas quaisquer tecnologias de enlace, podendo-se escolher livremente entre as que dominam o mercado atual como Ethernet, Fast Ethernet, FDDI, ou Token Ring. Também não há restrições quanto à topologia da rede, que pode consistir tanto de uma única ligação

ponto a ponto, ou de um único segmento de rede, ou ainda serem complexas, incorporando vários segmentos de redes interconectados. O padrão H.323 especifica o uso de áudio, vídeo e dados em comunicações multimídia, sendo que apenas o suporte à mídia de áudio é obrigatório. Mesmo sendo somente o áudio obrigatório, cada mídia (áudio, vídeo e/ou dados), quando utilizada, deve seguir as especificações do padrão.

2.3.3.2- PROTOCOLO SIP

Tecnologia adotada no desenvolvimento do trabalho é protocolo SIP (Session Initiation Protocol) é um padrão da Internet Engineering Task Force (IETF). Como citado anteriormente ele tem semelhanças com o protocolo http, esta semelhança está no cabeçalho do protocolo, ou seja, quando falamos de requisição de resposta para qualquer início de sinalização entre usuários. O protocolo foi criado para constituir sinalização com mais de um usuário ao mesmo tempo, em conferência utilizando a rede, de um modo totalmente livre do conteúdo de mídia da chamada. Assim como o HTTP, o SIP leva os controles da aplicação para o terminal, eliminando a necessidade de uma central de resposta. Uma transmissão de chamadas permite que os usuários especifiquem onde eles se encontram para que possam receber a chamada. As pessoas que estão na chamada podem gerenciar a mesma, com isso o usuário faz a conferência ou desfaz a conferência.

Serão apresentados os componentes que possui SIP.

a) TERMINAL SIP

Este componente é o software de estação final. Ele trabalha como um usuário solicitando a inicialização de sessão e também funciona como um servidor pois ele emite resposta a um pedido de chamada. Assim sendo pode-se dizer que a arquitetura é formada por cliente/servidor. Este componente é muito inteligente, ele faz o gerenciamento e o armazenamento das chamadas. Consegue também estabelecer chamadas de um endereço similar a um endereço de web ou número de telefone.

b) SERVIDOR PROXY SIP

Este componente é conhecido como next-hop porque recebe um requerimento e envia para outro servidor ou para os usuários. Possui informações com o intuito de bilhetagem das chamadas, e corresponde a uma entidade intermediária contendo tanto um UAC (*User Agent Client*) quanto UAS (*User Agent Server*). Sua função é de encaminhamento das solicitações recebidas para os próximos servidores SIP ou terminais, que fazem parte do caminho até o destino (serviço de roteamento).

A entidade Servidora Proxy SIP também pode operar com comunicação stateful ou stateless. O stateful pode dividir as chamadas pela ordem que chegarem, assim faz com que muitas extensões estejam tocando de uma vez e o primeiro que atender pega a chamada. Isso significa que pode utilizar o telefone por meio de um desktop SIP, celular SIP e suas aplicações de videoconferência de qualquer lugar. Esta entidade usa métodos para resolver a solicitação ao endereço de host, inserindo busca de DNS, busca em banco de dados.

c) SERVIDOR DE REGISTRO SIP

Este servidor recebe a solicitação sobre a localização presente de cada usuário. Aceita solicitação de registro permitindo que os usuários registrem suas presenças. Pode estar localizado no mesmo meio físico de um servidor proxy ou servidor de redirecionamento.

d) SERVIDOR DE REDIRECIONAMENTO

Esta entidade também recebe solicitações, determina outro servidor e retorna aos usuários que solicitarão o endereço do usuário requisitado. Usualmente é utilizado, somente o servidor de redirecionamento ou o servidor de Proxy.

e) SERVIDOR DE LOCALIZAÇÃO

Esta entidade oferece o servidor de localização de um determinado usuário na rede. Segue na figura 7 o funcionamento SIP.

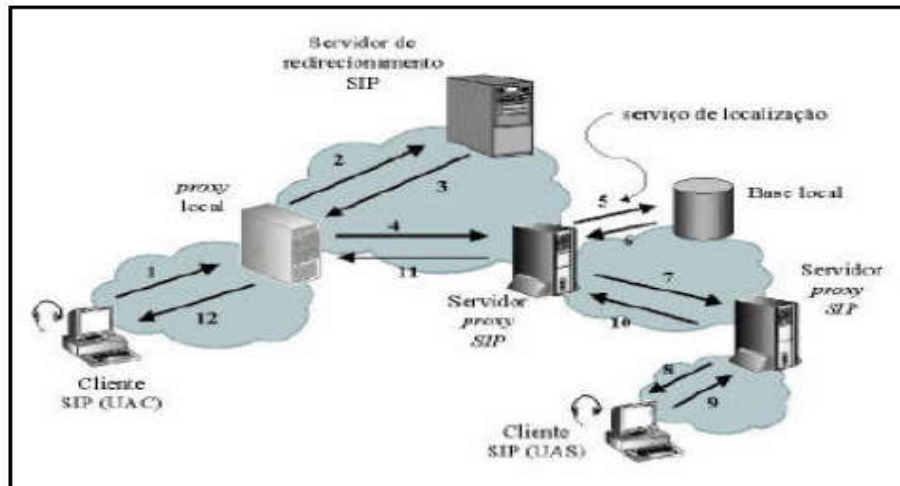


Figura 7 - Entidade SIP. Fonte: [7].

A figura 7 mostra o funcionamento dos componentes do SIP. O cliente SIP faz a solicitação para o servidor local, este servidor envia a solicitação ao servidor de redirecionamento que devolve a endereço ao servidor local, que solicita para o servidor proxy, que por sua vez envia a solicitação para o servidor de localização que faz a consulta na base de dados local. Após isso, é enviado ao outro servidor proxy que envia ao destino.

O SIP utiliza o SDP (Session Description Protocol). Esta ferramenta de conferência foi criada para descrever sessões de áudio, vídeo e multimídia ele também é um produto do grupo de trabalho MUSIC e é muito usado atualmente no contexto do MBONE, a rede de comunicação multicast que funciona na Internet.[7]. O principal objetivo do SDP é definir uma sintaxe padrão.

2.3.3.3- PROTOCOLO SDP (SESSION DESCRIPTION PROTOCOL)

O SDP foi projetado para descrever as sessões multimídias, anunciando sessões, convidando-as, ou agindo de outras maneiras para iniciar uma sessão multimídia. A figura 8 detalha como funciona o SDP.

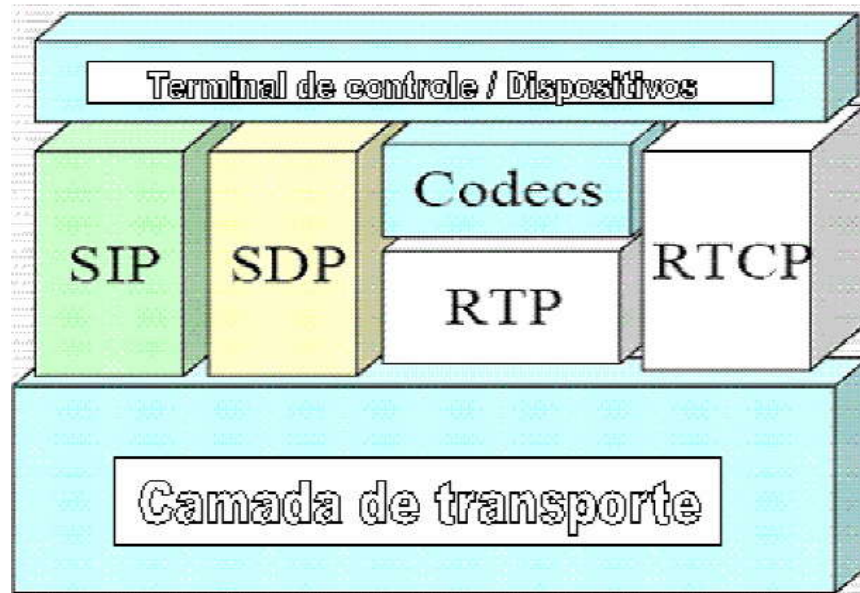


Figura 8 - Organização dos protocolos – SDP. Fonte: [9].

O SIP, previamente estudado, pode ser considerado o protocolo principal neste esquema. Ele define quem será o protocolo responsável por carregar fluxos de informações por meio de sessões para permitir que usuários participem de uma comunicação multimídia. No nosso caso, o escolhido para esta função, é o SDP. O SDP provê a informação necessária para habilitar a conversação em uma sessão multimídia. Este protocolo inclui descrição sobre a mídia a ser usada (*codecs*, taxas de amostragem), o destino da mídia (endereço IP e número da porta), o protocolo de transporte a ser utilizado, informações sobre a banda a ser utilizado, o nome da sessão e motivo da mesma, bem como o tempo de atividade da sessão.

2.3.3.4- RTP (REAL-TIME PROTOCOL)

O protocolo RTP é o responsável pelo transporte da mídia, a pedido de seu protocolo superior. Em nosso caso, como visto na figura 9, o SDP estaria solicitando ao RTP o transporte da mídia escolhida, de acordo com as características pretendidas pelo mesmo.

O RTP não tem capacidade de garantir a entrega da informação no tempo desejado, ou com QoS. Utilizando o protocolo UDP, que não é presa por mais velocidade e menores garantias, o RTP envia os dados encapsulados em pequenos pacotes que podem chegar a 20 ms, por exemplo. Os dados são enviados, e após recebidos pelo destinatários, enfileirados para leitura. Desta forma, um atraso considerável de um pacote, significa perda, pois o mesmo é desprezado a fim de não atrapalhar a leitura das outras mensagens.

Cabe ao RTP identificar que tipo de mídia está sendo enviada, a numeração dos pacotes, a marcação do tempo e monitorar a qualidade de sessão multimídia. Para este monitoramento, o RTP se utiliza do protocolo RTCP (*Real-Time Control Protocol*), que também trabalha sobre UDP.

A figura 9 apresenta os campos de cabeçalho do protocolo RTP.

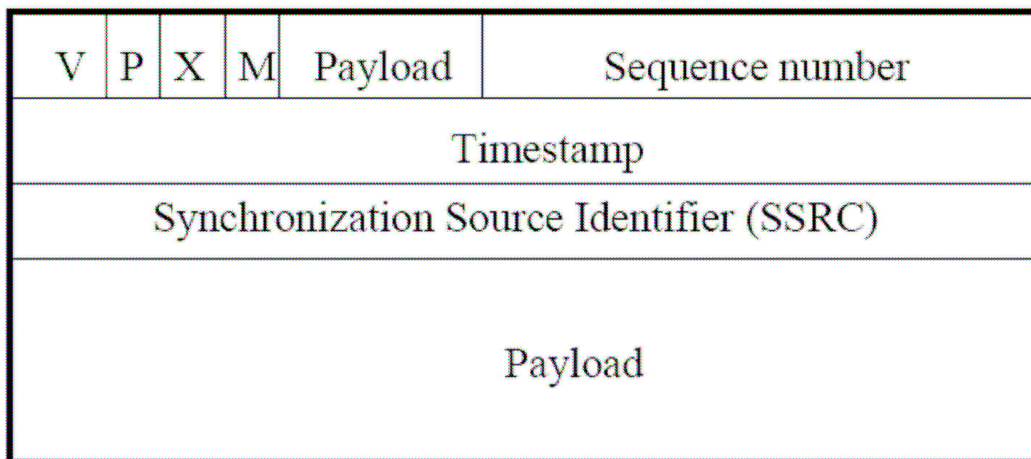


Figura 9 - Cabeçalho RTP. Fonte: [10].

O padrão da maioria dos protocolos, o cabeçalho RTP começa com a identificação de versão utilizada (V). Este campo ocupa dois bits. Logo após, segue um bit de *padding* (P), que, caso esteja “setado”, cria um pacote de tamanho fixo, ou habilita criptografia. Após, um bit adicional em (X), e um

marcador, também de um bit, que indica o início de um novo frame em vídeo (M), ou voz, caso esteja com supressão de silêncio.

Aparecem então um conjunto de 7 bits, que carregam informação referente ao tipo de *codec* que está sendo empregado. Um campo de 16 bits cria um número seqüencial incrementando a cada pacote enviado, ajudando ao receptor a detectar uma possível perda de mensagens. O *timestamp*, com 32 bits, envia informações sobre a amostragem do código, ajudando assim o receptor a reduzir alguns problemas, como o de atraso.

Por último, um binário de 32 bits, chamado SSRC (*Synchronization Source Identifier*), único, randômico individual para cada participante de um enlace, é enviado, terminando o cabeçalho RTP. Ele permite que o receptor saiba quem está enviando o pacote.

2.3.3.5- MGCP (Media Gateway Control Protocol)

MGCP é um acrônimo para a expressão inglesa Media Gateway Control Protocol, um protocolo proposto pelo grupo de trabalho IETF (Internet Engineer Task Force) para integração da arquitetura SS#7 em redes VOIP. Embora o SS#7 se encontre presente na telefonia tradicional, o MGCP especifica com redes IP, Frame Relay e ATM. O sistema é composto por um Call Agent, pelo menos um media gateway (MG), responsável pela conversão dos sinais entre circuitos e pacotes, e pelo menos um Signaling Gateway (SG), quando conectado a um PSTN.

Durante a evolução do MGCP, o trabalho cooperativo de grupos do ITU-T e do IETF resultou na recomendação H.248, definida também com o protocolo Megaco (IETF), através do RFC 3015. O H.248, também conhecido como protocolo Megaco, é um padrão desenvolvido cooperativamente entre o ITU e a IETF para permitir que um Media Gateway Controller (MGC) desempenhem seu papel em um media gateway (MG). Competindo com outros protocolos como o MGCP e MDPCP, é considerado um protocolo complementar ao H.323 e ao SIP, no qual o MGC controla os MGs via H.248, mas comunicará com outro via H.323 ou SIP.

2.3.3.6- Aplicação do protocolo SIP

Para um melhor entendimento do protocolo SIP será feita uma aplicação que servirá de base para a implementação do modelo proposto neste projeto de pesquisa.

2.3.3.6.1- Mensagens de Requisição

Segue as mensagens de requisição mais comuns:

- **INVITE:** Indica a parte chamada que está sendo convidada a participar de uma sessão multimídia. Normalmente contém informações sobre o usuário, bem como QoS (Quality of Service) ou dados relacionados a segurança. No corpo da mensagem, uma descrição de sessão pode estar presente. Para tal, pode ser usado, o protocolo de descrição de sessão SDP (Session Description Protocol);
- **ACK:** Esta mensagem é recebida como resposta final para INVITE. Nesta mensagem, o protocolo SDP pode ser negociado. Caso não esteja indicado, pode-se assumir que a descrição enviada pelo primeiro INVITE será a padrão nesta conversaçãõ;
- **OPTIONS:** Questiona quais métodos e extensões são suportadas por ambos os pontos. O servidor pode indicar em sua resposta quais métodos e extensões são suportadas por ele, e pelo usuário (fornecidos na pergunta);
- **BYE:** Força a desconexão da ligação, e libera os recursos associados da mesma.
- **CANCEL:** Cancela uma requisição que ainda esteja pendente, ou seja, ainda não foi atendida com uma resposta final.
- **REGISTER:** Método utilizado pelo cliente para registrar o apelido do seu endereço em um servidor SIP.

2.3.3.6.2- MENSAGENS DE RESPOSTA

As mensagens de resposta são divididas em 6 grupos:

- *1xx Responses*: Respostas de informações.
Exemplo: 180 Tocando
- *2xx Responses*: Respostas de transações sucedidas.
Exemplo: 200 OK
- *3xx Responses*: Respostas de redirecionamento.
Exemplo: 302 Usuário deslocado temporariamente.
- *4xx Responses*: Respostas de requisições falhas.
Exemplo: 404 Não encontrado.
- *5xx Responses*: Respostas de falha no servidor.
Exemplo: 503 Serviço Indisponível.
- *6xx Responses*: Respostas de falha geral do sistema.
Exemplo: 600 Todos os serviços estão ocupados

2.3.3.6.3- CAMPOS DE CABEÇALHO

Seguem as duas categorias utilizadas que são: requisição e resposta, somente requisição, somente resposta, e cabeçalho de corpo de mensagem.

No protocolo SIP, as especificações de campos de cabeçalho estão descritas na RFC 3261, e seguem, na maioria dos casos, as mesmas regras que os cabeçalhos HTTP. Alguns cabeçalhos SIP mais comuns podem conter uma forma compacta, onde o nome do campo é designado por uma letra minúscula. Estes campos são mostrados no tabela 1.

Tabela 1 - Campos do cabeçalho SIP

Campo de cabeçalho	Forma compacta
<i>Accept-Contact</i>	a
<i>Allow-Event</i>	u

<i>Call-ID</i>	I
<i>Contact</i>	m
<i>Content-Encoding</i>	e
<i>Content-Length</i>	L
<i>Content-Type</i>	c
<i>Event</i>	o
<i>From</i>	f
<i>Refer-To</i>	r
<i>Referred-By</i>	b
<i>Reject-Contact</i>	J
<i>Subject</i>	s
<i>To</i>	T
<i>Via</i>	v

Segue os campos de cabeçalho SIP mais comumente utilizados. Para facilitar o entendimento, vai ser demonstrado um cabeçalho com a requisição

INVITE sip:thiagoplanet@192.0.0.3:5060;user=phone SIP/2.0

Via: SIP/2.0/UDP 192.0.0.2:5060

From: UserAgent<sip:teste@192.0.0.2:5060;user=phone>

To: 1<sip:thiagoplanet@192.0.0.3:5060;user=phone>

Call-ID: 96561418925909@192.0.0.2

Cseq: 1 INVITE

Subject: Comunicação multimídia em tempo real

Contact: <sip:teste@192.0.0.2:5060;user=phone>

Content-Type: application/sdp

Content-Length: 168

v=0

o=- 238540244 238540244 IN IP4 10.0.0.2

s=VOVIDA Session

c=IN IP4 10.0.0.2

t=3174844751 0

m=audio 23456 RTP/AVP 0

a=rtpmap:0 PCMU/8000

a=ptime:20

Neste exemplo, nota-se que o cabeçalho se constitui de dois blocos. O primeiro, até o campo *Content-Length*, contém o cabeçalho SIP. Abaixo disto, tem o corpo da mensagem com os parâmetros do protocolo SDP, que está especificado como padrão para a sinalização do pacote.

Na primeira linha, o comando *INVITE* vem precedido da identificação do protocolo (SIP), usuário de destino separado do domínio por uma arroba, porta utilizada na comunicação, e um ponto e vírgula. Este tem o objetivo de separar os dados do destinatário dos primeiros parâmetros do protocolo. O primeiro (*user*), informa que a origem da chamada vem de um telefone (*phone*), e não de um dispositivo que esteja emulando o aparelho. Em seguida, é informado que o protocolo SIP versão 2.0 é suportado.

- **VIA (Cabeçalho de Requisição e Resposta)** – Guarda dados do caminho que o pacote está percorrendo. Este histórico é importante para que a mensagem chegue ao seu destino, e não fica “rodando” eternamente entre servidores. A cada passagem por um Servidor Proxy, um novo campo VIA é acrescentado com informações deste servidor. Um parâmetro Branch pode ser adicionado junto ao VIA, informando a origem da mensagem em relação ao sistema. Assim, evita-se que o pacote caia em looping na rede.
- **FROM (Cabeçalho de Requisição e Resposta)** – Campo de extrema importância nesta categoria de mensagens, pois identifica quem está enviando o pacote. Neste exemplo, a SIP URL <sip:1@192.0.0.2:6060;user=phone> é relacionada ao nome User Agent.
- **TO (Cabeçalho de Requisição e Resposta)** – Trabalha da mesma forma que o FROM, porém identifica o destinatário, com o nome seguido de sua SIP URL.
- **Call-ID (Cabeçalho de Requisição e Resposta)** – Provê uma identificação única para convites ou registros de um mesmo usuário. Segue de @ e o endereço IP de origem.
- **Cseq (Cabeçalho de Requisição e Resposta)** – Número sequencial enviado junto a requisição. Permite ao destinatário saber a ordem correta

das chamadas, a fim de evitar confusões na interpretação das mesmas. No exemplo acima, verificamos que é a primeira mensagem sendo enviado do tipo INVITE.

- **Subject (Cabeçalho de Requisição e Resposta)** – Campo normalmente não utilizado. Serve para descreve a natureza da ligação.
- **Contact (Cabeçalho de Requisição e Resposta)** – Mostra uma SIP URL da origem a fim de fornecer ao destinatário um endereço de resposta.
- **Content-Type (Cabeçalho de Corpo de Mensagem)** – Informa o formato da mensagem de mídia. Neste caso, para tal será utilizado o protocolo SDP.
- **Content-Lenght (Cabeçalho de Corpo de Mensagem)** – Mostra o tamanho da mensagem de mídia enviada. Neste caso, 168 octetos enviados por SDP. Além destes cabeçalhos do exemplo, podemos citar alguns outros que são usados com considerável frequência por outras mensagens SIP:
- **Proxy-Authorization (Cabeçalho de Somente Requisição)** – Após receber uma resposta do servidor pedindo autorização (407 Proxy Authentication Required), utiliza este campo para enviar a chave que permitirá o fluxo dos pacotes na rede. Importante para o processo de registro.
- **Expires (Cabeçalho de Corpo de Mensagem)** – Também utilizado no processo de registro, fornece a data e hora que a autenticação perderá sua validade.
- **Priority (Cabeçalho de Somente Requisição)** – Define prioridades entre as chamadas. Cabeçalho opcional.
- **Proxy-Require (Cabeçalho de Somente Requisição)** – Indica características suportadas pelo Proxy, como, por exemplo, QoS (Quality of Service - Qualidade de Serviço).
- **Record-Route (Cabeçalho de Requisição e Resposta)** – Cabeçalho inserido por Servidores Proxy a fim de ser marcado como um caminho para nova sinalização.
- **Retry-After (Cabeçalho de Requisição e Resposta)** – Indica um intervalo para nova tentativa. Normalmente é utilizado após uma mensagem de resposta informando canal ocupado (486 Busy).

- **Timestamp (Cabeçalho de Requisição e Resposta)** – Provê informações sobre quando a mensagem foi enviada pelo cliente.
- **Warning (Cabeçalho de Somente Resposta)** – Indica se o telefone está apresentando algum problema de status.
- **Proxy-Authenticate (Cabeçalho de Somente Resposta)** – Especifica ao cliente qual o tipo de segurança requerida. Normalmente origina uma resposta REGISTER do usuário com os padrões solicitados.

2.3.3.6.4- FLUXOS DE CHAMADAS

Serão apresentadas os fluxos de chamadas entre dispositivos.

2.3.3.6.4.1- SIP PC para PC

A ligação tem início quando o usuário A, envia o INVITE para o usuário B. Neste momento a mensagem é roteada e devolvida ao usuário A, informando que a requisição está sendo transmitida. Quando a mensagem chega ao destino, o usuário B recebe uma mensagem de RINGING que informa ao usuário A que está aguardando uma resposta do usuário B, neste momento é enviada uma mensagem ACK ao usuário A.

Quando um dos participantes resolve finalizar a conexão, a mensagem BYE é enviada. Assim que o outro usuário recebe o BYE ele também envia o BYE para finalizar a conexão.

A figura 10 apresenta a sinalização PC para PC no SIP .

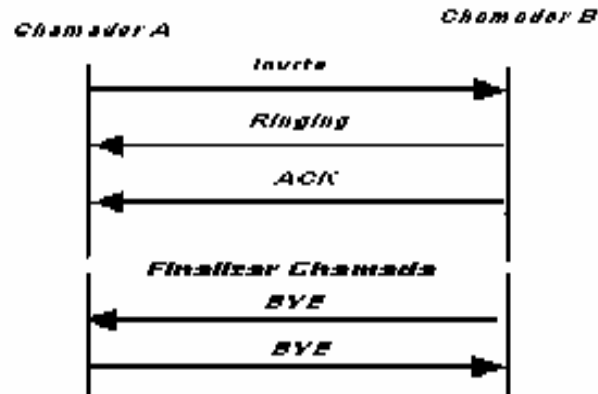


Figura 10 - Sinalização SIP PC para PC.

2.3.3.6.4.2- SIP PC / servidor / PC

A ligação tem início quando o usuário A e o usuário B se registram no servidor. Após o registro o usuário A envia o INVITE para o usuário B. Neste momento a mensagem é roteada e devolvida ao usuário A, informando que a requisição está sendo transmitida. Quando a mensagem chega ao destino, o usuário B recebe uma mensagem de RINGING que informa ao usuário A que está aguardando uma resposta do usuário B, neste momento é enviado uma mensagem ACK ao usuário A. Quando um dos participantes resolve finalizar a conexão, o BYE é enviado ao servidor, que desregistra o usuário e passa para o outro usuário que envia o BYE para o servidor que também desregistra o mesmo. A figura 11 mostra esta ligação.

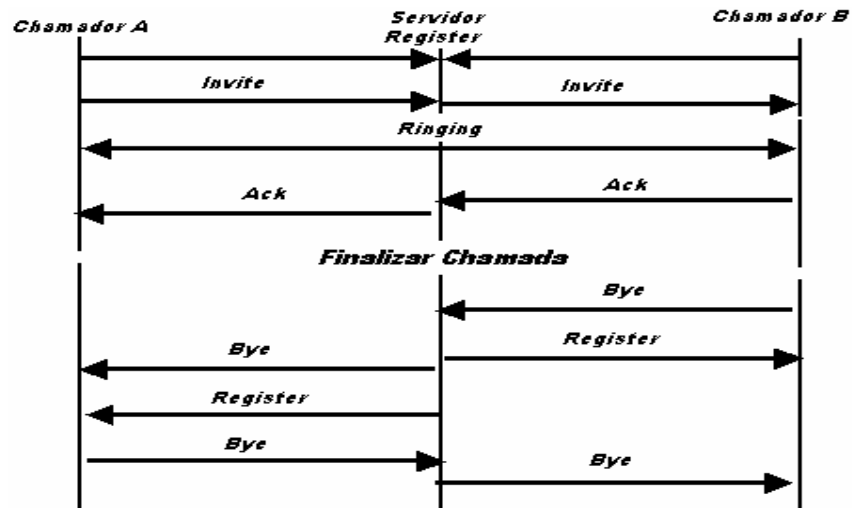


Figura 11 - Sinalização SIP PC para Servidor. Fonte: [8].

2.4- Banco de Dados

Um banco de dados é uma coleção de dados estruturados. Ele pode ser qualquer coisa desde uma simples lista de compras a uma galeria de imagens ou a grande quantidade de informação da sua rede corporativa. Para adicionar, acessar, e processar dados armazenados em um banco de dados de um computador é necessário um sistema de gerenciamento de bancos de dados como o Servidor MySQL, Oracle, PostgreSQL entre outros.

O sistema de gerenciamento de banco de dados MySQL será o utilizado no desenvolvimento do projeto de pesquisa e para isso será feita uma descrição um pouco mais detalhada. O banco de dados relacional armazena dados em tabelas separadas em vez de colocar todos os dados um só local. Isso proporciona velocidade e flexibilidade. A parte SQL do MySQL atenda pela "Structured Query Language - Linguagem Estrutural de Consultas". SQL é linguagem padrão mais comum usada para acessar banco de dados e é definida pelo Padrão ANSI/ISO SQL. O MySQL é um software Open Source, isto significa que é possível para qualquer um usar e modificar o programa, além dá vantagem de ser software livre ele é o servidor de banco de dados MySQL é muito rápido, confiável, e fácil de utilizar. Apesar de estar em constante desenvolvimento, o Servidor MySQL oferece hoje um rico e proveitoso conjunto de funções como por exemplo, a conectividade,

velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

2.5- Linguagem de Programação

A linguagem de programação que será utilizada para a implementação do Sistema de Voz sobre IP (VoIP) é o Java. A escolha por esta linguagem é devido ao fato da diminuição de custo do sistema utilizando software livre.

2.5.1- Linguagem Java

A linguagem de programação Java é toda orientada a objetos, ela é constituída essencialmente da linguagem de programação e um programa para execução chamado Máquina Virtual (ou Virtual Machine). A tecnologia Java foi desenvolvida por uma equipe de pessoas da Sun Microsystems. Inicialmente, foi elaborada para ser a linguagem-base de projetos de software para produtos eletrônicos, e teve seu grande *boom* em 1995, devido ao sucesso mundial da World Wide Web.

Atualmente é uma das linguagens de programação mais poderosas, devido sua grande flexibilidade de poder funcionar em qualquer plataforma, seja Windows, Linux, Mac etc.

2.5.2- Máquina Virtual

Java é multi-plataforma, isto significa que quando um programa Java é executado, um código intermediário é gerado, conhecido como *bytecode*. Este *bytecode* é interpretado pelas máquinas virtuais Java (JVMs) para a maioria dos sistemas operacionais. A máquina virtual é a responsável por criar um ambiente multiplataforma, se construírem um sistema operacional novo, basta desenvolver uma máquina virtual Java que traduza os *bytecodes* para código nativo e pronto. Todas as aplicações funcionarão. A máquina virtual Java ainda é responsável por carregar de forma segura todas as classes do código, averiguar se os *bytecodes* aderem à especificação JVM e se eles não violam a integridade e a segurança do sistema.

2.5.3- Edições de JAVA

A linguagem Java é dividida em três edições que são:

- **Java 2 Standard Edition (J2SE):** é a tecnologia Java para computadores pessoais, notebooks e PC's, é com elas que a maioria dos softwares são construídos e executados. O J2SE também é dividido em duas partes, Java Development Kit (JDK) e Java Runtime Edition JRE;
- **Java 2 Mobile Edition (J2ME):** é a tecnologia Java para dispositivos móveis com poucos recursos de memória ou processamento. Possui APIs bem simples e leves para economizar armazenamento, memória e processamento. Ela é muito utilizada para sistemas em celulares, palm tops, pocket pcs, smartphones, javacards e demais dispositivos. O J2ME se divide em dois grupos de bibliotecas, Connected Limited Device Configuration (CLDC) e Connected Device Configuration (CDC);
- **Java 2 Enterprise Edition (J2EE):** é a tecnologia Java para aplicações corporativas que podem estar na internet ou não. Tem um grande número de APIs onde a segurança é a principal preocupação. É ideal para a construção de servidores de aplicação, integração de sistemas ou distribuição de serviços para terceiros.

2.5.4- IDEs Java

Existem várias IDEs Java: o Eclipse e o NetBeans. A IDE NetBeans é um conjunto de bibliotecas, módulos e APIs que forma um ambiente integrado de desenvolvimento visual que possibilita ao desenvolvedor compilar, debugar, efetuar *deploying* de suas aplicações. É um produto *open source*, todo em Java e possui vários módulos de expansão que podem ser baixados do próprio site. <http://www.netbeans.org>. Podemos destacar alguns recursos desta IDE.

- Debugador e compilador de programas;
- Auto-completar avançado, depurador de erros, depurador de Servlets;

- Syntax highlighting à XML,HTML,CSS,JSP,IDL;
- Suporta linguagens Java, C, C++;
- Suporte à XML e HTML, JSP, JSTL, Servlets, etc.

O Eclipse é um dos IDE's mais conceituados para a programação Java, possuindo código aberto para o desenvolvimento de aplicações de computador. O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade. O Eclipse é a IDE Java mais utilizada no mundo, tendo como características marcantes o uso da SWT e não do Swing como biblioteca gráfica, a forte orientação ao desenvolvimento baseado em plug-ins e o amplo suporte ao desenvolvedor com centenas de *plugins* que procuram atender as diferentes necessidades de diferentes programadores. Algumas das principais qualidades do Eclipse são:

- **Auto complete:** O Eclipse mostra assinaturas de métodos, trechos de ajuda entre outros;
- **Várias JVM's configuradas simultaneamente:** Isso proporciona grande vantagem, pois a criação do projeto fica mais eficaz;
- **Perfis de servidores:** Podem instalar vários servidores diferentes, montando ambientes de *deploy* idênticos aos utilizados pelos seus clientes. Desde o Derby (banco de dados agora da Sun) até o JBoss (servidor da Red Hat), passando pelo Tomcat, Geronimo entre outros;
- **Perspectivas:** Pode-se visualizar o mesmo projeto de várias formas diferentes;
- **Workspace:** Possibilidade de customização do *workspace* ou até mesmo de trabalhar em vários *workspaces* diferentes, cada um com o seu conjunto de projetos.

Além dessas características, tem se ainda a geração de construtores e declaração automática, *auto-import*, geração de *getters & setters*, recomendações para resolução de problemas, integração com Subversion e CVS, gerenciamento automático de pacotes, etc. O eclipse pode ser baixado em www.eclipse.org.

CAPÍTULO 3

DESENVOLVIMENTO DO SISTEMA DE VOZ

Neste capítulo, serão descritos todos os passos para o desenvolvimento do sistema de controle de comunicação utilizando a tecnologia Voip.

3.1-Descrição do problema

O problema que será abordado neste trabalho de pesquisa é sobre a tecnologia Voip, consiste no desenvolvimento de um software de Sistema de Voz sobre IP (VoIP) para a Internet por meio da especificação de um sistema telefônico baseado na arquitetura de uma Intranet. O software será responsável pelo controle da comunicação em um ambiente em que um terminal IP possa se comunicar com um telefone. A implementação será feita em linguagem Java utilizando o banco de dados MySql.

3.2- Modelagem do problema

A modelagem do problema de comunicação de voz de terminal IP para telefone é mostrada na figura 12, e mostra como é a interação entre os serviços conversacionais de Voip entre um PC e um telefone, juntamente com os dispositivos necessários.

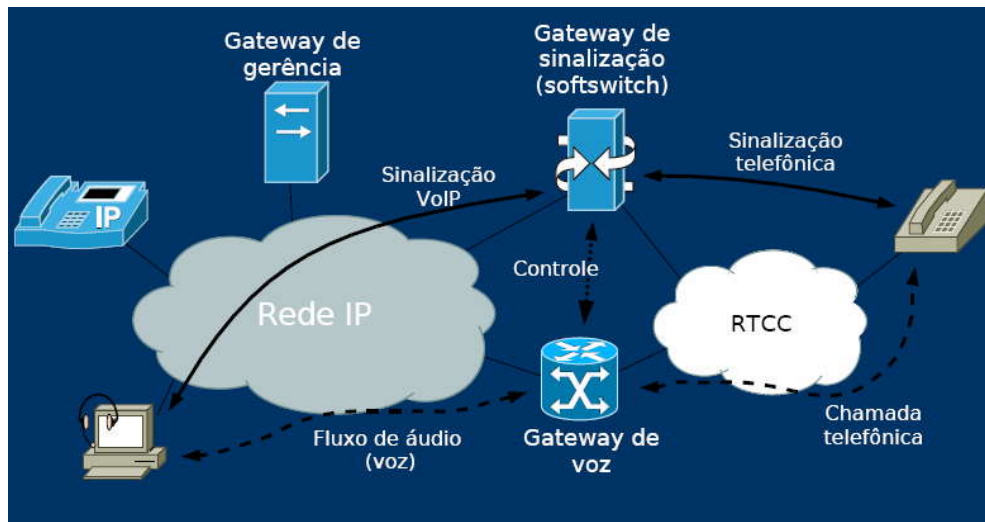


Figura 12 - Terminal IP para Telefone.

O problema será dividido em módulos para facilitar o desenvolvimento de cada parte que constitui o sistema.

- **Módulo 1: Desenvolvimento da interface**

A interface para apresentação das mensagens de protocolo e fluxos foi inteiramente implementada em Java. De forma que o usuário tenha facilidades para o uso do sistema. Nesta interface o cliente, faz busca de outros usuários com grande precisão, porque o painel de busca é bem simples.

- **Módulo 2: Desenvolvimento da aplicação**

Será desenvolvida na aplicação, a comunicação pode meio de uma rede IP, em que o computador (cliente) envia mensagens para outro computador (cliente), onde eles se interagem.

- **Módulo 3: Comunicação entre dois computadores**

Neste módulo será desenvolvida no sistema, a comunicação entre os terminais, com auxílio de um servidor DNS, no qual fará todo o controle da aplicação, também a comunicação auditiva, aplicando o uso de protocolo SIP para que exista a interação entre dois ou mais usuários em terminais IP distintos, em uma rede local, ou de mesmo domínio.

- **Módulo 4: Comunicação entre terminal IP com telefone**

Neste módulo será desenvolvida dentro do sistema uma funcionalidade que fará a comunicação entre o terminal IP e o telefone IP. O usuário poderá usar interface toda gráfica.

Além do software dá suporte protocolo SIP que provê serviços como de mensagens, videoconferência, transferência e redirecionamento de chamadas, entre outros, se o telefone utilizado tiver esses recursos. A figura 13 demonstra este cenário.

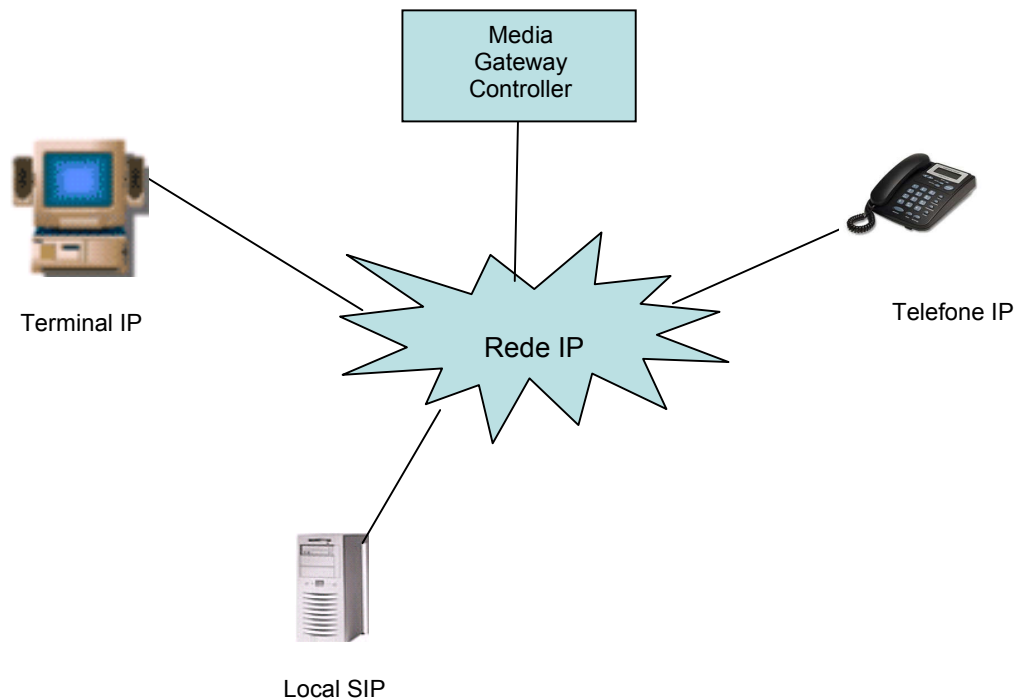


Figura 13 - Comunicação dos equipamentos com servidor SIP.

- **Módulo 5: Comunicação usuário com o sistema**

Neste módulo, será desenvolvida no software, a comunicação entre o sistema para com o usuário. Sendo assim, ele poderá personalizar o sistema com seu perfil, isto é, terá um usuário e uma senha; podendo assim acessar seu perfil em qualquer lugar da rede, sem perder seus contatos, fotos, mensagens. E todos os dados da pessoa serão gravados no banco de dados Mysql, para acesso em qualquer lugar da rede. A figura 14 mostra este cenário.

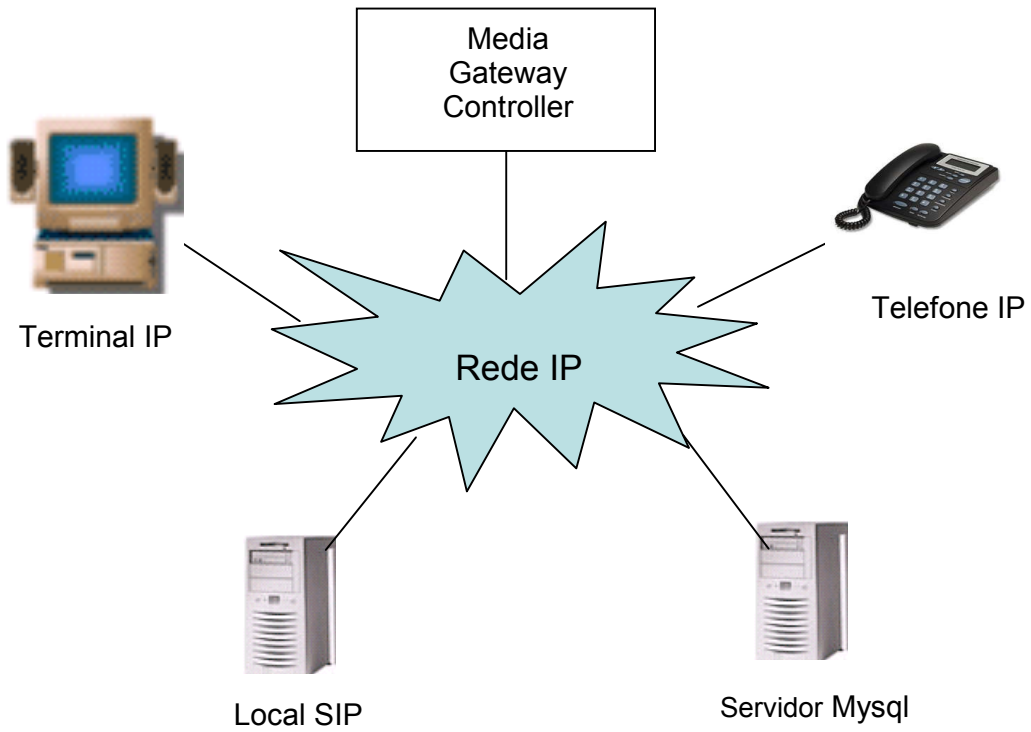


Figura 14 - Comunicação entre equipamentos com banco de dados.

CAPÍTULO 4

DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo será apresentado os passos para implementação do sistema, proposto no trabalho.

4.1- REQUISITOS PRINCIPAIS DO PROBLEMA

O protótipo desenvolvido neste trabalho utiliza o protocolo SIP para enviar e receber mensagens em uma rede TCP/IP, usando protocolo UDP.

Neste sistema são dois pontos chaves para implementação, o Cliente/Servidor, um único terminal fará este papel de envio de recebimento de mensagens ou mesmo áudio - conferência. Segue a descrição de cada um independente do outro.

- **Cliente:** o cliente envia as mensagens para o servidor, e em seu cabeçalho segue todas as necessárias das informações.
- **Servidor:** o servidor recebe os dados enviados do cliente, e transmite para o seu destinatário.

Deste modo é possível, por meio do protótipo, fazer uma comunicação de informações na rede. Para fazer a especificação deste sistema foi utilizada uma metodologia orientada a objetos, a *Unified Modeling Language* (UML), usando como ferramenta o *JUDE*.

4.1.1- ESPECIFICAÇÃO

4.1.2- USE CASES

Tabela 2 - Especificações do software

N°	Nome	Requisição	Resposta
01	FazerChamada	O PLANETSIP faz uma chamada para outro PLANETSIP	“Chamada Efetuada”
02	EncerrarChamada	O PLANETSIP é desconectado	“Chamada Encerrada”
03	IniciarSIPPHONE	Carrega as configurações padrão do PLANETSIP	Status = “Pronto”
04	EscolherDispositivo	Usuário escolhe o dispositivo de entrada	“dispositivo OK”

4.1.3- DESCRIÇÃO DOS USE CASES

01-FazerChamada

Descreve como o usuário faz uma chamada

1. O usuário cliente entra com os dados da chamada e envia o pedido INVITE pressionando a tecla “chamar”
2. O usuário servidor aceita a chamada e envia a resposta OK.
3. O usuário cliente envia a resposta ACK e inicia a transmissão

02-EncerrarChamada

Demonstra como o usuário termina uma chamada.

1. O usuário (cliente) envia o pedido BYE pressionando a tecla “encerrar”
2. O usuário (servidor) responde com um OK.
3. A ligação é encerrada.

03- Iniciar PlanetSIP

Inicia as configurações do PLANETSIP.

1. Usuário executa o PLANETSIP

2. O PLANETSIP carrega configurações e envia resposta “status = pronto”.

Problemas com PLANETSIP

1. Usuário executa o PLANETSIP
2. O PLANETSIP não carrega configurações e retornando “status não definido”

04 – Escolher dispositivo

Mostra de qual forma o usuário escolhe o dispositivo de entrada de áudio, vídeo ou texto.

1. O PLANETSIP detecta a lista de dispositivos de saída e envia a lista para o usuário
2. O usuário escolhe o dispositivo.
3. O PLANETSIP aceita e retorna com a mensagem “dispositivo OK”.

Problemas com PLANETSIP

1. O PLANETSIP detecta a lista de dispositivos de saída e envia a lista para o usuário
2. O usuário escolhe o dispositivo
3. O PLANETSIP rejeita e retorna com a mensagem “dispositivo não acessível”.

4.2- Modelo de Caso de Uso

O protótipo possui quatro casos de usos.

- 1- IniciaPlanetSIP:** O usuário executa o software, e o sistema é carregado na tela.
- 2- EscolherDispositivo:** O usuário define qual dispositivo irá usar, sendo texto ou voz, e o software retorna para ele quais estão disponíveis no momento.

- 3- **FazerChamada:** O usuário define com quem quer se comunicar iniciando uma chamada, o sistema retorna chamada efetuada, e estabelece a transmissão.
- 4- **DesligarChamada:** O usuário desliga a chamada, o sistema para a transmissão.

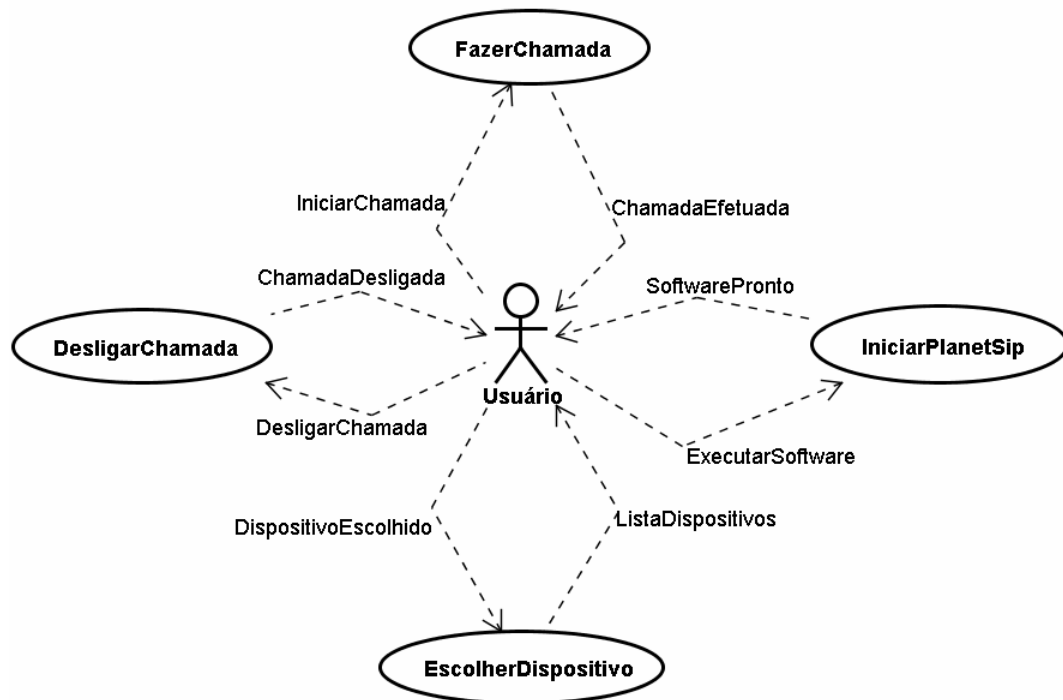


Figura 15 - Diagrama Caso de Uso.

4.2.1- Diagrama de classes

As classes utilizadas no sistema são:

- PlanetSip:** principal classe do sistema, esta classe que fará a interação com o usuário, obtendo seus dados;
- Controle:** está classe fará todo controle das chamadas, enviando dados ao seu destinatário;

c) **Cliente:** está classe fará toda a parte de envio da mensagem ao seu destinatário;

d) **Servidor:** está classe terá os dados de todos os clientes que estão on-line e suas localizações.

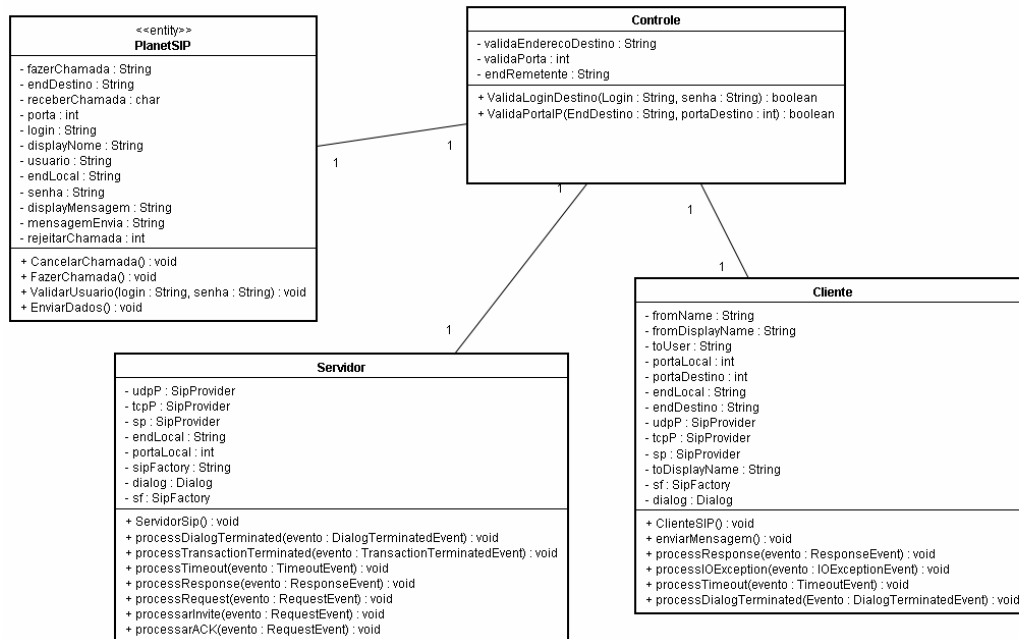


Figura 16 - Diagrama de classes.

4.2.2- Diagrama de Seqüência

Os diagramas de seqüência mostram a continuação das ações correntes no sistema. Eles demonstram como é feita a troca de mensagens entre as classes.

4.2.3- Assunto iniciando chamada SIP

Este diagrama de seqüência é executado pelo cliente (Chamador) sempre que o sistema é iniciado, logo que a interface é carregada, o usuário entra com seu login e senha, a interface valida os campos. Caso esteja correta, é enviado ao controlador os dados do cliente, a porta e o IP corrente no momento, logo é enviado a mensagem ao servidor, que processa a mensagem.

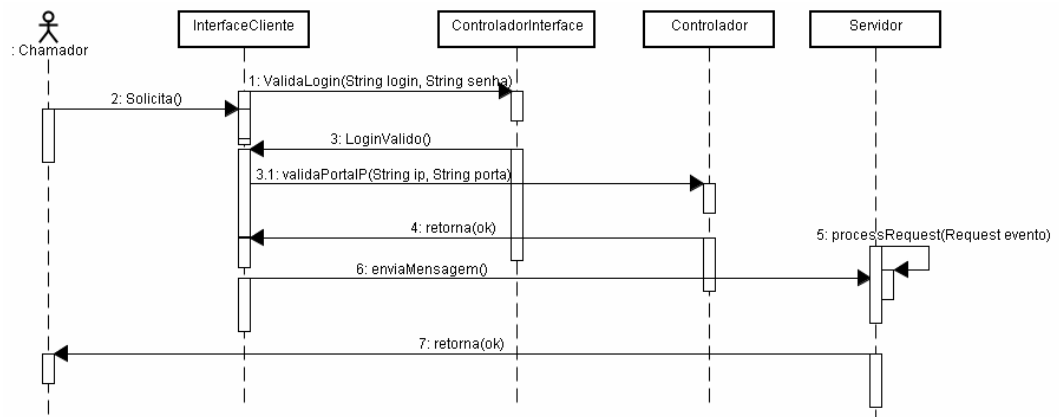


Figura 17 - Diagrama de seqüência iniciando chamada

4.2.4-Recebendo chamada

Este diagrama de seqüência demonstra como o servidor recebe uma chamada, repassando ao seu destinatário (Chamado). A mensagem chega, ele envia ao controlador que processa (Invite) o pedido de mensagem, enviando para o seu destino, o cliente responde (ACK) ao servido que recebeu a mensagem.

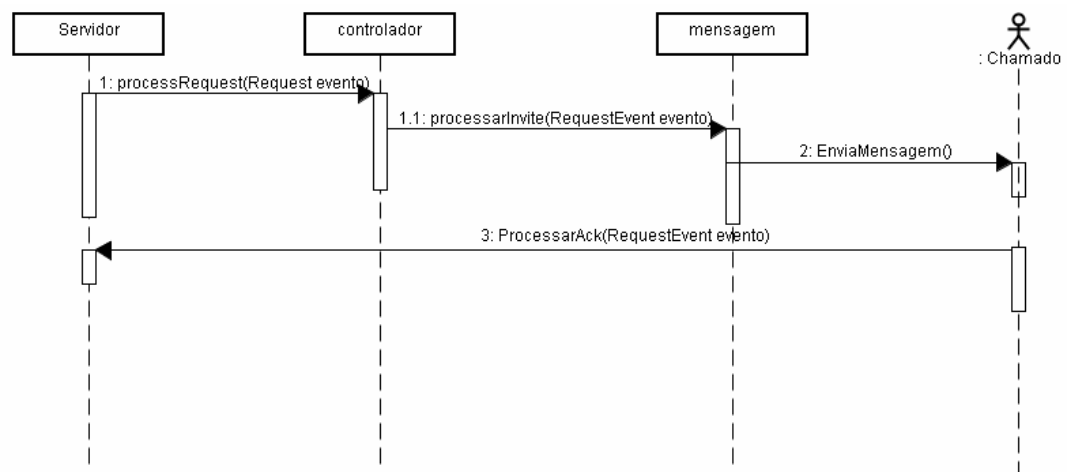


Figura 18 - Diagrama de seqüência recebendo chamada.

4.2.5- Enviar um pedido de encerramento de chamada (BYE)

Este diagrama de seqüência mostra o usuário enviando o pedido de BYE para o servidor. Com isso o servidor recebendo a mensagem encerra a conversa.

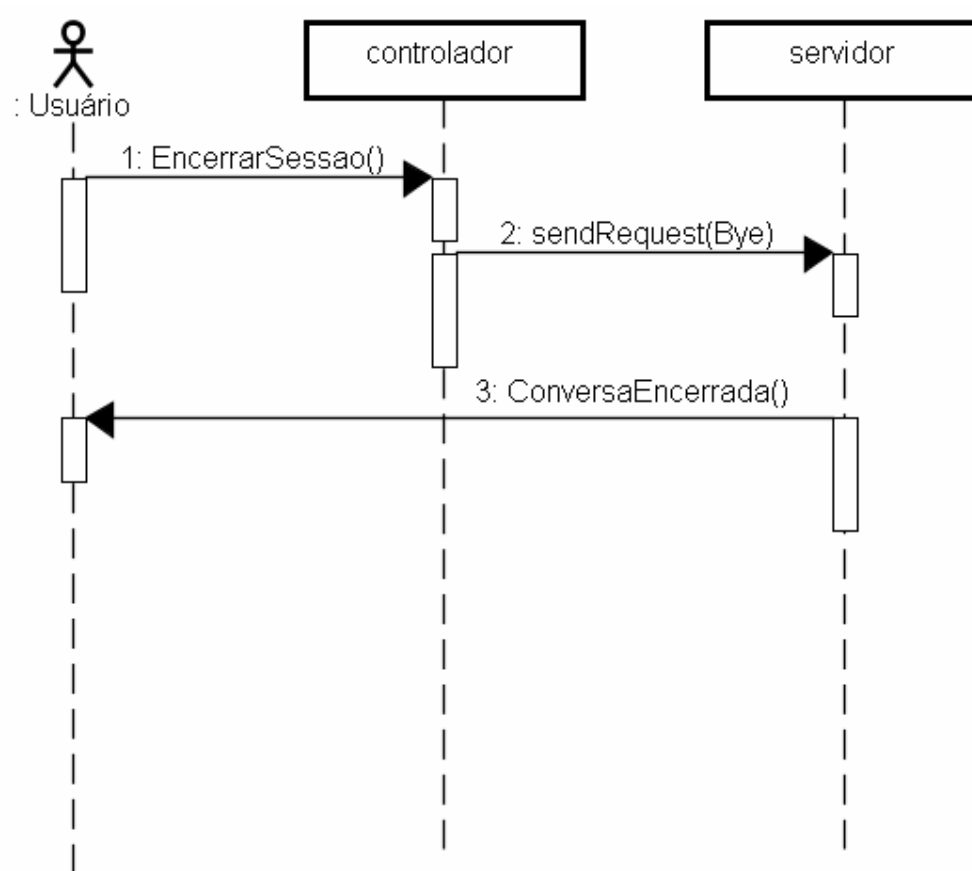


Figura 19 - Diagrama de seqüência enviar pedido BYE

4.2.6- Recebendo um pedido de encerramento da chamada(BYE)

Neste diagrama o servidor recebe a mensagem do cliente (remetente) e envia para o controlador que processa a mensagem e retorna um OK para o servidor.

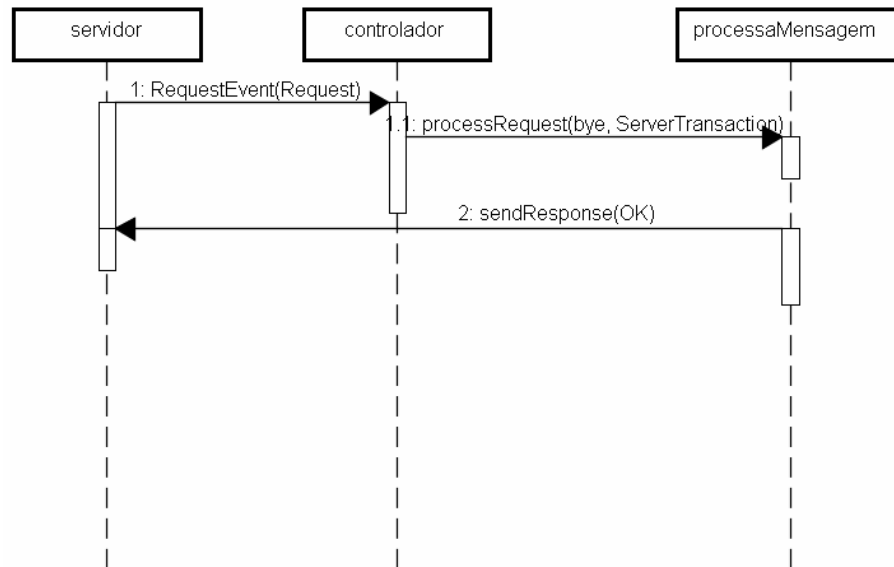


Figura 20 - Diagrama de seqüência recebendo pedido BYE

Este capítulo mostrou todo o funcionamento do software.

CAPÍTULO 5

IMPLEMENTAÇÃO DA APLICAÇÃO

O projeto da implementação do PLANETSIP foi baseado na linguagem de programação Java, pela sua portabilidade, segurança, ferramentas para a programação gratuitas e o também pela sua diversidade de bibliotecas disponíveis. Para o desenvolvimento do projeto foram utilizadas as bibliotecas JAIN SIP desenvolvida pela Sun [11] e pelo grupo NIST [12]. Esta biblioteca possui classes para controle de sessão e mídia do SIP.

5.1 - Transmissão e recepção de dados

A implementação da comunicação de informações em Java pode ser feita por meio da biblioteca RTP da API do JMF e a biblioteca JAIN SIP que contém as classes SIP e SDP. A biblioteca RTP tem algumas funções adicionais com relação a qualidade da transmissão de dados em tempo real, previstas no RTP deixa subentendido a programação em baixo nível na rede. As classes SIP e SDP auxiliam na montagem das estruturas de mensagens do protocolo SIP e de sessão com o SDP. A implementação da preparação de envio Jain_SIP é mostrada no código fonte.

```
1 - sf = SipFactory.getInstance();
2 - sf.setPathName ("gov.nist");
3 - prop.setProperty ("javax.sip.STACK_NAME", "gov.nist");
4 - prop.setProperty ("javax.sip.IP_ADDRESS", endLocal);
5 - SipStack pilhaSip = sf.createSipStack (prop);
6 - ListeningPoint esperarUdp = pilhaSip.createListeningPoint (endLocal,
portaLocal, "udp");
7 - sp = pilhaSip.createSipProvider (esperarUdp);
8 - headerFactory = sf.createHeaderFactory();
9 - addressFactory = sf.createAddressFactory();
10 - messageFactory = sf.createMessageFactory();
```


Na linha 1 é iniciada com a criação do SipFactory, um objeto desse tipo é responsável pela criação de outros elementos ligados direta ou indiretamente à comunicação. Na linha 2, é estabelecido o diretório base para API, de acordo com a implementação desenvolvida pelo NIST (National Institute of Standards and Technology), que é a mais usada. Nas linhas 3 e 4 é criado um objeto Properties, este objeto configura uma pilha SIP, essa pilha será a base para toda a comunicação, na linha 5 é criado um objeto SipStack. Após isso, deve se configurar os pontos de recepção das mensagens de mensagens SIP, esses pontos da aplicação, irá “escutar” pelo recebimento de mensagens, na linha 6 é criado esse ponto, de acordo com o endereço, a porta e o protocolo de transporte especificado, este ponto de recepção criado deve ser associado a um objeto SipProvider, que é criado na linha 7. Na linha 8 é criado o objeto HeaderFactory, que será utilizado na construção de cabeçalhos das mensagens SIP, já o objeto da linha 9, addressFactory, estará responsável pela criação de endereços SIP, por fim na linha 10 a mensagem como um todo será montada com auxílio de um objeto do tipo MessageFactory.

Após as configurações iniciais, será criada uma mensagem SIP de tipo MESSAGE. A implementação do MESSAGE é mostrado no trecho do código.

```
1 - SipURI fromAddress = addressFactory.createSipURI(remetente,
remetenteSIP);
2 - Address fromNameAddress = addressFactory.createAddress(fromAddress);
3 - fromNameAddress.setDisplayName(remetente);
4 - FromHeader fromHeader =
headerFactory.createFromHeader(fromNameAddress, null);
5 - SipURI enderecoDestino = addressFactory.createSipURI("destino",
endDestino + ":" + portaDestino);
6 - ArrayList viaHeaders = new ArrayList();
7 - ViaHeader viaHeader =
headerFactory.createViaHeader(endDestino,portaDestino,"udp",null);
8 - ContentTypeHeader contentTypeHeader =
headerFactory.createContentTypeHeader("text", "plain");
9 - Request request =
messageFactory.createRequest(enderecoDestino,Request.MESSAGE,
```

```
sp.getNewCallId(), headerFactory.createCSeqHeader((long)1,
Request.MESSAGE), fromHeader, toHeader, viaHeaders,
headerFactory.createMaxForwardsHeader(70));
```

Na linha 1, um objeto SipURI é criado, para representar um endereço SIP de origem, utilizando para tanto um objeto AddressFactory como base, este endereço é utilizado para criação de um endereço tipo Address na linha 2, que por sua vez é aproveitado para criação de um objeto FromHeader nas linhas 3 e 4. Na linha 5 define o destino da mensagem SIP, na criação desse endereço são especificados o endereço IP de origem e a porta. Nas linhas 6 e 7 é criada a parte Via da mensagem SIP, que, inclusive, pode indicar mais de um caminho, logo em seguida na linha 8 é especificado o tipo da informação carregada pela mensagem SIP, uma mensagem SIP pode carregar qualquer informação, ou até mesmo nenhuma, no exemplo a demonstrado o objetivo é que uma mensagem de texto seja enviado para UAS(*User Agent Server*), o tipo text/plain é especificado. Na linha 9, a mensagem SIP é criada através do método createMessage () do objeto MessageFactory, esse método recebe oito parâmetros, sendo eles: endereço de destino em formato Sip URI, o tipo da mensagem SIP sendo criada, um valor para CallID(identificação da mensagem), o número de seqüência da mensagem, a indicação da origem(informação), a indicação do destino da mensagem(informação de cabeçalho) e a parte Via da mensagem.

Após a mensagem criada, é necessário adicionar um contato, e isto é mostrado no trecho do código.

```
1 - SipURI contato = addressFactory.createSipURI(remetente, endLocal);
2 - Address contactAddress = addressFactory.createAddress(contato);
3 - ContactHeader contactHeader =
headerFactory.createContactHeader(contactAddress);
4 - request.addHeader(contactHeader);
5 - request.setContent(mensagem, contentTypeHeader);
6 - sp.sendRequest(request);
```

Nas linhas de 1 a 3 é adicionado o campo de contato, logo em seguida é adicionado à mensagem na linha 4, para completar o conteúdo da mensagem, bem como seu tipo, é adicionado na linha 5. A partir do SipProvider criado nas linhas anteriores, a mensagem é enviada para seu destino com a instrução na linha 96. Na criação deste exemplo, é necessário que a classe implemente a interface SipListener, assim todo terminal SIP deve utilizar essa interface, que obriga a classe implementar seis métodos. Como segue na tabela 3 abaixo.

Tabela 3 - Métodos SipListener

Método	Descrição
public void processRequest(RequestEvent)	Chamado com recebimento de uma mensagem de requisição
public void processResponse(ResponseEvent)	Chamado com recebimento de uma mensagem de resposta SIP.
Public void processTimeout(TimeoutEvent)	Chamado com notificações de timeout
Public void processIOException(IOExceptionEvent)	Chamado quando ocorrem notificações de IO
Public void processTransactionTerminated (TransactionTerminatedEvent)	Chamado com o encerramento de um Transaction(relacionado com a comunicação SIP)
Public void processDialogTerminated (DialogTerminatedEvent)	Chamado com o encerramento de um Dialog(relacionamento com a comunicação SIP)

Na recepção da mensagem, o UAS (*User Agent Server*) também deve implementar a interface SipListener, o UAC e UAS são bem parecidos, com a diferença no UAS é necessário implementar o método conhecido como processRequest(RequestEvent), já que a solicitação SIP será recebida. Será mostrado um trecho do código fonte para tratamento e recebimento da mensagem SIP.

- 1 - Request requisicao = evento.getRequest();
- 2 - new String (requisicao.getRawContent());
- 3 - requisicao.getMethod();
- 4 - requisicao.toString();

Na linha 1, o objeto relacionado à requisição é capturado, a partir daí é possível verificar o corpo da mensagem SIP na linha 2, o tipo de mensagem na linha 3, e a mensagem propriamente dita na linha 4.

5.2- Interface do Sistema PLANETSIP

A figura 21 mostra a interface do programa PLANETSIP.

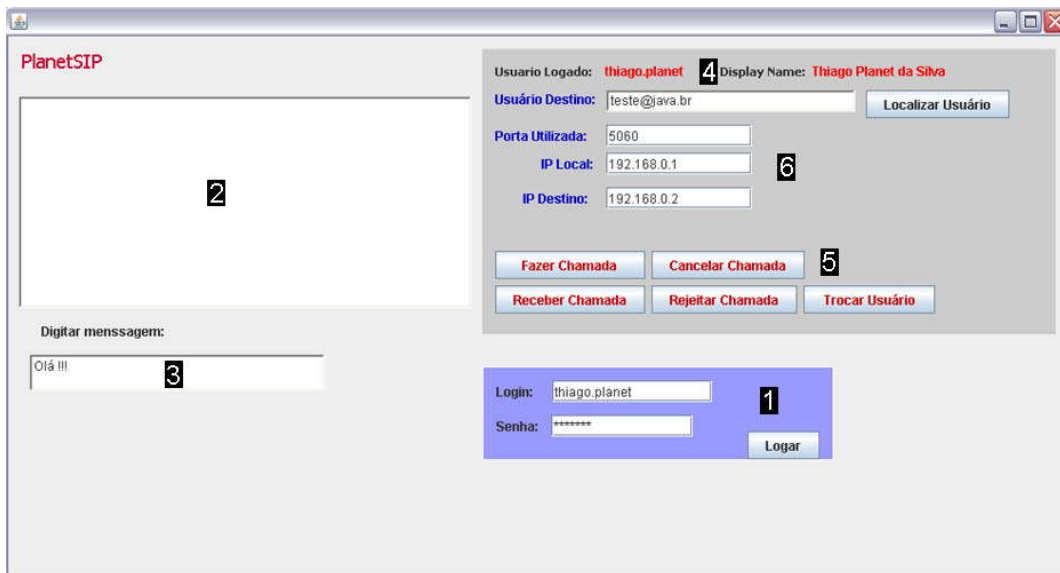


Figura 21 - Interface sistema PlanetSIP

A tela principal do programa desenvolvida ao longo do trabalho. Foram numerados todos os campos da tela para facilitar o entendimento da interface, a tabela 4 demonstra as funcionalidades de cada um desses números.

Tabela 4 - Funcionalidades da Interface.

Numero	Nome	Descrição
1	Login	Aqui onde usuário faz autenticação de seu usuário.
2	Mensagens	Onde os usuários se interagem por meio de texto neste local.
3	Escrita	Aqui o usuário digita mensagem para seu destinatário.
4	Identificação	Onde aparece o usuário logado, e também o seu nome completo.
5	Botão	Botão de interação entre os usuários.

6	Configurações	Configurações de IP local, IP destino, usuário destino e a porta.
---	---------------	---

Este programa foi desenvolvido para comunicação usando tecnologia Voip em uma rede local, ou de mesmo domínio.

CAPÍTULO 6

CONCLUSÃO

Neste trabalho foi possível aperfeiçoar os conhecimentos de rede e telefonia utilizando redes baseadas em IP. Este projeto mostrou também todas as dificuldades que um desenvolvedor pode ter ao realizar um projeto, como a falta de recursos para implementação, e também o desconhecimento em certas áreas, o planejamento de tempo necessário para uma implementação, e como lado positivo a auto-aprendizagem neste processo.

A transmissão de dados comprova a tendência para a integração, e seguindo esta tendência a telefonia IP começa a migrar de um paradigma de comutação por circuitos para o paradigma comutação por pacotes (VOIP). As características do H.323 e do SIP são distintas como pode observar neste trabalho, mas não é possível afirmar que um seja melhor que o outro. A tendência futura para a transmissão de dados e mídia, incluindo a VoIP, é que os dois protocolos coexistirão, porém em aplicações diferentes. A utilização da VoIP é realizada em software e na Internet, a sua implementação em SIP é o caminho mais simples, já que o SIP é baseado nos protocolos da Internet, como o HTTP. O protocolo SIP é um protocolo baseada na Internet, e simples como o http, que é baseado em requisições e respostas. A sua lógica não depende de hardware, somente de alguma aplicação para implementar a sua estrutura, sendo portátil e de fácil implementação.

Foi escolhida a biblioteca JAIN SIP por que é uma nova tecnologia, que está sendo adotada pelos programadores Voip. A maior parte do tempo foi gasto para entender como funcionavam os objetos, e como eles se comunicavam. A SUN está produzindo uma biblioteca que diminui a complexidade na implementação de aplicativos utilizando SIP, chamada JAIN LITE. Esta biblioteca diminui a dificuldade na implementação de um aplicativo VoIP pois é uma especificação de alto nível, abstraindo do desenvolvedor os detalhes de seu funcionamento.

Sugestão para projetos futuros é a implementação completa deste projeto, com todas as características descritas no decorrer do projeto, e ainda pode se acrescentar a multiconferência para a conversa entre três ou mais participantes através de *multicasting* e outras funções de PABX, como a chamada em espera e transferência de chamada.

CAPÍTULO 7

Referências bibliográficas

- [1] Lima, V.: **Telefonia e Cabeamento de Dados**, Editora Érica, 2004.
- [2] TELECO. **Informação em Telecomunicações**. Disponível em:
<<http://www.teleco.com.br/voip.asp>> Acesso em Junho 2009.
- [3] SOFTWARELIVE.ORG.BR. **Portal – Software Livre**. Disponível em:
<<http://www.softwarelivre.gov.br>> Acesso em Junho de 2009.
- [4] VOIPCENTER – **Artigos. VoIP**. Disponível em:
<<http://www.voipcenter.com.br/>> Acesso em Junho de 2009.
- [5] WIKIPÉDIA. **Voz sobre IP**. Disponível em:
<<http://pt.wikipedia.org/wiki/VoIP>> Acesso em Junho de 2009.
- [6] Colcher, S...[et al]. **VOIP: Voz sobre IP**, Editora Elsevier, 2005-2^a Reimpressão.
- [7] **Funcionamento Protocolo SIP** Disponível em < [http://www.ietf.org./](http://www.ietf.org/)> Acesso em outubro de 2009.
- [8] GUIDE, DAVID; HERSENT, OLIVIER; PETIT, JEAN-PIERRE - "**Telefonia Ip**". Editora Makron. São Paulo. 1^a Edição. 2005.
- [9] Fonte: SISALEM, Dorghan; KUTHAN, Jiri. **Understanding SIP. Iptel**, 2005.
- [10] Fonte: SISALEM, Dorghan; KUTHAN, Jiri. **Undertanding SIP**. 187 p.
- [11] **SUN.Java Communit Process JSR-000032 JAIN SIP Specification** Disponível em <http://www.jcp.org/aboutJava/communityprocess/final/jsr032/>> Acesso em outubro de 2009
- [12]. **National Institute of Standard and Technology.NIST SIP Parser and Stack (v1.2)** disponível em <<http://www-x.antd.nist.gov/proj/iptel/jain-sip-1.2/>> Acesso em outubro de 2009.
- [13] OLIVEIRA, Sérgio. **Telefonia IP para ambientes móveis usáveis: Simpósio Brasileiro de Redes de Computadores**, 19. 2001, Florianópolis. **Anais...** Florianópolis: UFSC, 2001. p. 542-558.