

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**PROTÓTIPO DE UM SISTEMA DE TELEFONIA IP PARA  
LAN S BASEADO NO PADRÃO SIP**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**DÉBORA LUCIANI NAZARIO**

BLUMENAU, JUNHO/2003

2003/1-14

# **PROTÓTIPO DE UM SISTEMA DE TELEFONIA IP PARA LANS BASEADO NO PADRÃO SIP**

**DÉBORA LUCIANI NAZARIO**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Francisco Adell Péricas — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

**BANCA EXAMINADORA**

---

Prof. Francisco Adell Péricas

---

Prof. Sérgio Stringari

---

Prof. Miguel A. Wisintainer

## **AGRADECIMENTOS**

Agradeço em primeiro lugar a Deus por tudo que vivi e aprendi em todos esses anos na universidade.

Aos meus amados pais Dilvo Neto Nazario e Georgina de Souza Nazario pelo amor, dedicação, educação e confiança, sentimentos que fizeram com que eu lutasse com determinação por isso que conquisto agora.

À minha querida irmã Lilian Bianca Nazario e a Leonardo Müller Dellagiustina pelo apoio e amor com que sempre pude contar.

Ao meu orientador Francisco Adell Péricas que no sentido pleno da palavra orientou, incentivou sempre disposto a ajudar com extremo desprendimento, qualidades que devem sempre acompanhar um orientador.

A todos os professores, colegas e amigos que de alguma forma participaram de mais esta etapa que cumpro da minha vida.

A todos o meu muito OBRIGADO.

## RESUMO

A velocidade com que a Telefonia IP se transformou em uma indústria é impressionante, principalmente no que diz respeito à convergência de diferentes tipos de redes de transmissão de dados e à popularidade da internet como meio de troca de informações a baixo custo. Este trabalho especifica e implementa um protótipo de um sistema de telefonia IP para LANs baseado no padrão SIP do IETF, desenvolvido para comunicação *full-duplex* de voz em ambiente Windows.

## **ABSTRACT**

The velocity that the IP Telephony was transformed in an industry is impressive, mainly concerning the convergency of different kinds of data transmission networks and the popularity of internet as a low cost way for interchanging information. This work specifies and implements a prototype of an IP Telephone system for LANs based on the SIP standard from IETF, developed for voice full-duplex communication in an Windows plataform.

## LISTA DE FIGURAS

Figura 1 - Ambiente para Telefonia IP .....	13
Figura 2 - (a) Onda de áudio; (b) Amostragem; (c) Quantização.....	15
Figura 3 - <i>Codecs</i> de áudio do ITU-T .....	16
Figura 4 - Arquitetura do Protocolo TCP/IP .....	18
Figura 5 - Formato do pacote UDP .....	19
Figura 6 - Pacote RTP .....	20
Figura 7 - H.323&SIP.....	22
Figura 8 - Mensagem de pedido SIP .....	25
Figura 9 - Resposta SIP .....	26
Figura 10 - Sinalização Completa da Chamada.....	27
Figura 11: Tipos de Endereços SIP .....	27
Figura 12 - Mensagem SIP .....	30
Figura 13 - Diagrama de Caso de Uso.....	31
Figura 14 - Diagrama de Classes.....	32
Figura 15 - Classes (Criação da Sessão ).....	35
Figura 16 - Classes responsáveis pelo trabalho de voz .....	36
Figura 17 - Tela Inicial .....	37
Figura 18 - Menu Opções .....	37
Figura 19 - Cadastro de Usuários .....	38
Figura 20 - Seleção do Usuário .....	38
Figura 21 - Mensagem de Contato para Ligação.....	40
Figura 22 - Sobre .....	43

## LISTA DE QUADROS

Quadro 1 - Descrição dos Casos de Uso.....	31
Quadro 2 - Definição JThread .....	36
Quadro 3 - Mensagem Pedido Enviada .....	39
Quadro 4 - Fonte do Botão Ligar .....	39
Quadro 5 - Fonte de Construção da Mensagem SIP.....	39
Quadro 6 - Mensagem Resposta de Confirmação .....	40
Quadro 7 - Criação da Sessão.....	41
Quadro 8 - Mensagem de Pedido de Confirmação.....	41
Quadro 9 - Fonte do Botão Desligar.....	42
Quadro 10 - Mensagem Pedido "BYE" .....	42
Quadro 11 - Finalização da Sessão.....	42

# LISTA DE ABREVIações

*PSTN – Public Switched Telephonic Network*

*SIP – Session Initiation Protocol*

*IETF – Internet Engineering Task Force*

*ITU-T – International Telecom Union*

*NGN – Next Generation Network*

*CNAME – Canonical Name*

*RFC – Request For Comment*

*SDES – Source Description*

*RR – Receiver Reports*

*SR – Sender Reports*

*UAC – User Agent Client*

*UAS – User Agent Server*

*URL – Uniform Resource Locator*

*SDP – Session Description Protocol*

*DARPA – Defense Advanced Resourcer Projects Agency*

*SMTP – Simple Mail Transfer Protocol*

*SNMP – Simple Network Management Protocol*

*VoIP – Voice over IP*

*IP – Internet Protocol*

*LAN – Local Area Network*

*UDP – User Datagram Protocol*

*RTP – Real Time Protocol*

*RTCP – Real Time Control Protocol*

*FTP – File Transfer Protocol*

*HTTP – Hypertext Transfer Protocol*

# SUMÁRIO

1	INTRODUÇÃO .....	8
1.1	OBJETIVOS DO TRABALHO .....	9
1.2	ESTRUTURA DO TRABALHO .....	10
2	TELEFONIA IP .....	11
2.1	QUALIDADE DE SERVIÇO .....	13
2.2	ÁUDIO .....	14
2.3	PROTOCOLOS DE TELEFONIA IP .....	16
2.3.1	TCP/IP .....	16
2.3.2	UDP .....	18
2.3.3	RTP .....	19
2.3.4	RTCP .....	21
2.3.5	SIP & H.323 .....	21
3	SIP .....	23
3.1	ENDEREÇOS SIP .....	27
3.2	SDP .....	28
4	DESENVOLVIMENTO DO TRABALHO .....	29
4.1	REQUISITOS DO PROTÓTIPO .....	29
4.2	ESPECIFICAÇÃO .....	29
4.2.1	DIAGRAMA DE CASO DE USO .....	30
4.2.2	DIAGRAMA DE CLASSES .....	32
4.3	IMPLEMENTAÇÃO .....	34
4.3.1	TÉCNICAS E FERRAMENTAS UTILIZADAS .....	34
4.3.1.1	Borland C++ .....	34

4.3.1.2 JVoIPLib.....	34
4.3.2 O PROTÓTIPO.....	37
5 CONCLUSÕES.....	44
5.1 EXTENSÕES.....	45
REFERÊNCIAS BIBLIOGRÁFICAS.....	46

# 1 INTRODUÇÃO

O futuro das telecomunicações está na convergência das redes de dados e das redes de voz. Quando se fala em convergência na área de telecomunicações, refere-se à redução para uma única conexão de rede, fornecendo todos os tipos de serviços, com conseqüente economia de dinheiro.

As comunicações de voz baseadas na tecnologia *Internet Protocol* (IP) vêm se tornando uma alternativa cada vez mais viável à substituição dos modelos de telefonia convencional, em função do desenvolvimento de novas tecnologias que permitem dar suporte à transmissão de áudio em tempo real. Entende-se por transmissão de áudio em tempo real em redes de dados o termo *Telefonia IP*.

Segundo Mesquita (2002), estima-se que o serviço de voz sobre IP (VoIP), começará a ter maior representatividade a partir do segundo semestre de 2003. Esse avanço se dará com a utilização da tecnologia IP, suportando serviços de voz com mais qualidade de serviço, garantindo mobilidade, confiabilidade e flexibilidade tecnológica, fatores já existentes no serviço da *Public Switched Telephonic Network* (PSTN).

O princípio da comunicação da tecnologia IP é basicamente o mesmo da telefonia pública. Para realizar uma chamada são necessários protocolos de controle e sinalização para executar algumas tarefas como localização do usuário, notificação de chamada, início de transmissão de voz, finalização de transmissão de voz e desconexão.

Dentre os vários protocolos para transmissão de voz em uma *Local Area Network* (LAN) existentes no mercado, tem-se uma nova tecnologia do *Internet Engineering Task Force* (IETF), equivalente ao padrão H.323 do *International Telecom Union* (ITU-T), chamada *Session Initiation Protocol* (SIP) que difere por sua simplicidade, velocidade, uso de *multicast*, uso de URLs, priorização de chamadas e codificação de texto. O padrão SIP é uma pilha de protocolos onde a conexão pode ser estabelecida utilizando-se sinalização *User Datagram Protocol* (UDP) ou *Transmission Control Protocol* (TCP).

O SIP é um protocolo de controle/sinalização usado para iniciar, modificar ou terminar sessões ou chamadas multimídia entre usuários.

A IETF propôs o padrão SIP utilizado por este trabalho, e que tem as seguintes funcionalidades:

- a) a localização de usuários;
- b) estabelecimento de chamadas;
- c) suporte a *unicast* ou *multicast*.

Para a transmissão do fluxo de voz, utiliza-se o protocolo da camada de aplicação do modelo de referência *Transmission Control Protocol/Internet Protocol* (TCP/IP) denominado *Real Time Protocol* (RTP). O RTP utiliza o serviço de transporte *User Datagram Protocol* (UDP) para transmitir os pacotes.

O RTP é o protocolo padrão para dados de áudio e vídeo que necessitam ser transmitidos em tempo real, tais como áudio na telefonia IP. O protocolo RTP utiliza conjuntamente um protocolo de controle chamado *Real Time Control Protocol* (RTCP), que provê suporte a aplicações com características de transmissão de dados em tempo real, como por exemplo, controle sobre as informações individuais da conexão de cada usuário participante da sessão.

Este trabalho tem como objetivo aproveitar as facilidades da tecnologia IP propondo o uso desta tecnologia em substituição à infra-estrutura do sistema telefônico convencional, utilizando o uso de tecnologias abertas que estão se tornando padrão no mercado, evitando a dependência por tecnologias proprietárias.

Quanto ao protótipo, será implementado um software para a comunicação entre computadores de uma mesma LAN, utilizando a telefonia sobre IP, através da utilização do protocolo SIP e RTP, ambos do IETF, que farão o transporte da voz entre os recursos de rede utilizados, com o propósito de reduzir sensivelmente os custos inerentes à telefonia convencional utilizada atualmente.

## **1.1 OBJETIVOS DO TRABALHO**

Este trabalho tem como objetivo, desenvolver um protótipo de software de telefonia via Internet (Telefonia IP), conectando-se equipamentos de uma mesma LAN (Intranet), utilizando o padrão SIP.

Pelo fato do padrão SIP ser uma nova tecnologia e em crescente evidência no mercado, a sua utilização para a transmissão de voz sobre IP abre uma nova perspectiva dentro das redes dedicadas e corporativas que terão a possibilidade de suprir a infraestrutura da comunicação tradicional a custos mais baixos.

## **1.2 ESTRUTURA DO TRABALHO**

O capítulo inicial apresenta na introdução uma visão geral sobre o assunto que será tratado neste trabalho, sua importância para os novos conceitos dentro da telefonia e objetivos.

O segundo capítulo apresenta didaticamente a fundamentação sobre a Telefonia IP e tudo o que ela envolve. Assuntos que tratam das visões gerais de voz sobre IP, protocolos que serão utilizados, áudio e qualidade de serviço na telefonia IP.

O terceiro capítulo apresenta o SIP, principal embasamento deste trabalho.

O quarto capítulo apresenta o software desenvolvido, descrevendo o protótipo, especificação, características e seu funcionamento através das principais telas.

O quinto e último capítulo apresenta as conclusões obtidas com o desenvolvimento deste trabalho e sugestões para a sua continuidade.

## 2 TELEFONIA IP

O sistema telefônico mais comum nos dias de hoje ainda é o analógico assim como, para que haja a comunicação telefônica, é necessário o estabelecimento de um circuito entre assinantes (como são chamados os usuários do sistema telefônico). A comunicação telefônica atual, também conhecida como convencional, é representada pela rede de telefonia pública comutada ou *Public Switched Telephonic Network* (PSTN) onde o circuito ainda é digitalizado, multiplexado e necessário.

A convergência da área de telecomunicações (redes de dados e voz), dar-se-á pelo empacotamento da voz que será transmitida juntamente com os dados utilizando o protocolo IP e, sendo assim, a necessidade da presença de um circuito deixará de existir destacando-se o objetivo da Telefonia IP. Segundo Fagundes (2002), a convergência das redes de transmissão, é um dos temas mais discutidos atualmente, e a esta nova concepção é dado o nome de *Next Generation Network* (NGN).

Segundo Oliveira (2001), o termo Telefonia IP, Telefonia Internet ou ainda Voz sobre IP, tem se aplicado à utilização de redes baseadas no protocolo IP na camada de rede para transporte de voz, em especial através da internet.

Comparando-se a tecnologia de Telefonia IP com a tecnologia da telefonia analógica atual, esta última apesar de antiga, possui ainda vantagens que viabilizam a sua utilização, como por exemplo: seu custo se torna baixo quando poucos usuários estão falando ao mesmo tempo e quando não estão muito distantes entre si. Mas para isso, a tecnologia requer um par de cabos por conversação ativa, o que se torna rapidamente impraticável e caro (HERSENT, 2001). Segundo Oliveira (2001), as vantagens da utilização da tecnologia de Telefonia IP são:

- a) compartilhamento da rede para o tráfego de voz com o tráfego de dados (e-mail, FTP, etc);
- b) unificação de redes de transporte, sinalização e gerência sobre a mesma rede, com economia de infraestrutura e manutenção;
- c) meio de transmissão de baixo custo, comparado ao sistema telefônico;
- d) possibilidade de compactação e supressão de silêncio, reduzindo a largura de banda utilizada;
- e) utilização de rede já instalada;

- f) possibilidade de oferecer outros serviços como correio de voz, *call centers* via internet, segunda linha virtual, com reconhecimento e síntese de voz, através de interface telefônica.

Dentre as vantagens citadas acima, destaca-se não apenas a possibilidade na redução de custos, mas a facilidade que haverá na criação de serviços pela unificação das redes (voz e dados).

A redução de custos na Telefonia IP, deve-se principalmente pela tecnologia em si, ou seja, a comutação da rede IP é feita através de software, por dispositivos como roteadores e *switches*, e não através de hardware como na rede pública atual. A rede IP não garante qualidade de serviços, como por exemplo: atrasos na voz, perda de pacotes e ruídos na voz não previstos, serviços estes garantidos na telefonia convencional que a deixam com um custo mais elevado. Outro fator que faz com que os custos da Telefonia IP sejam mais reduzidos em comparação ao da telefonia convencional, é o compartilhamento de canal. Em uma rede IP, o mesmo meio físico é compartilhado para várias ligações, com inclusão de algumas características como supressão do silêncio e compressão que podem reduzir a largura de banda utilizada na conversação. Na telefonia convencional dois canais são reservados para cada conversação, independente da existência ou não de tráfego de voz nos circuitos, ou seja, mesmo nos momentos de silêncio (OLIVEIRA, 2001).

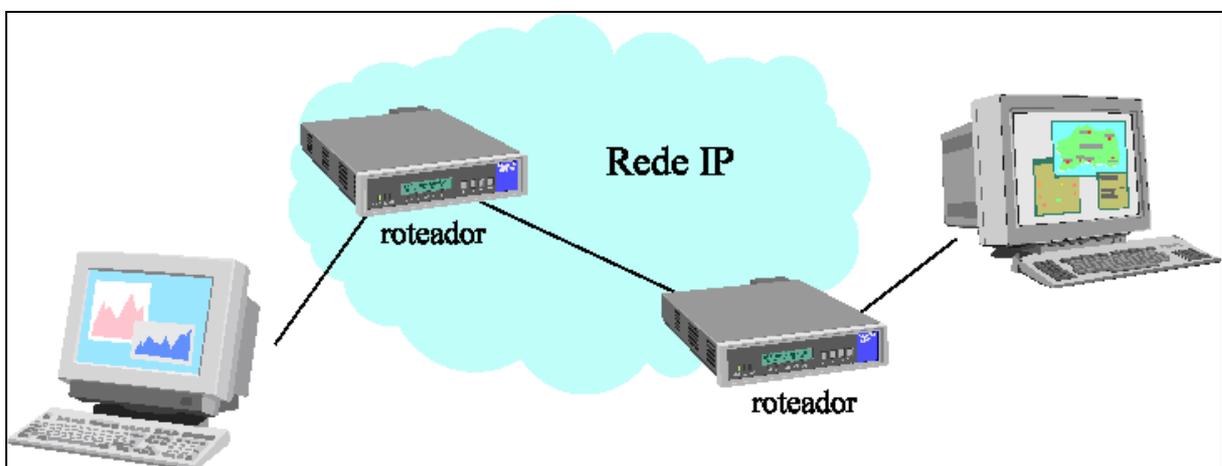
A vantagem na unificação das redes de transporte existentes atualmente na telefonia convencional, trará facilidade na manutenção e economia de infraestrutura. As operadoras de telecomunicações utilizam três sub-redes distintas para uma rede de voz: a rede de transporte de voz, de sinalização e de gerência, onde faz-se necessário a manutenção de cada rede distintamente. A unificação das redes de voz e dados, fará com que a manutenção, gerência e suporte e desenvolvimento sejam aplicados sobre uma única rede e não em duas como atualmente.

Sendo a Telefonia IP baseada em software, torna-se mais fácil adicionar e integrar os serviços a serem oferecidos, utilizando tecnologias agregadas como reconhecimento e síntese de voz. Por exemplo segundo Souza (2001), é possível ouvir todos os e-mails recebidos por um telefone convencional, utilizando a síntese de voz e transmissão sobre IP para que estes e-mails cheguem por telefone. Desta forma, serviços como fax, HTTP, correio de voz e outros podem ser acessados através de telefones comuns.

Inicialmente a tecnologia de Telefonia IP não fica restrita somente à ligação entre dois computadores, as chamadas podem ser feitas através de ligações entre computador e telefone (especial), telefone e telefone e finalmente, entre computador e computador. Na realidade, a Telefonia IP refere-se ao fluxo de voz sobre pacotes transportado sobre redes que utilizam o IP para terminais onde o fluxo é decodificado em voz novamente. Estes terminais podem ser computadores, ou da mesma forma telefones convencionais analógicos (OLIVEIRA, 2001).

A Figura 1 mostra um exemplo de ambiente de Telefonia IP entre dois computadores, no qual está baseado este trabalho.

**Figura 1 - Ambiente para Telefonia IP**



Neste exemplo, dois usuários estão se comunicando através da rede IP, ou seja, não há a utilização de rede telefônica convencional. De acordo com Oliveira (2001), vários softwares estão disponíveis para esta utilização, podendo utilizar protocolo padrão ou proprietário. Quando da utilização de protocolo padrão, há a possibilidade de interar softwares de diferentes fabricantes. A codificação de voz é feita pelas placas multimídia dos computadores.

## 2.1 QUALIDADE DE SERVIÇO

Desde a grande importância que a internet adquiriu mundialmente, toda ordem de aplicações tem sido desenvolvidas com infra-estrutura e capacidade para comunicação.

Como citado anteriormente, a rede IP não garante qualidade de serviço, o que indiretamente influencia no que diz respeito ao custo de tal tecnologia. De acordo com Oliveira (2001), se a rede estiver sobrecarregada, podem ser inseridos atrasos que podem

atrapalhar a qualidade final da voz. Caso a Telefonia IP esteja sendo usada em uma rede privada dedicada, a utilização de mecanismos de garantia de qualidade de serviço poderá fazer com que a qualidade final da voz fique dentro do nível aceitável pelos usuários.

Qualidade de Serviço (QoS) é a capacidade da rede de prover serviços não convencionais para um subconjunto de tráfegos agregados. O protocolo da camada de transporte é o responsável pelo controle de QoS.

Aplicações que apresentam dados multimídia em tempo real como a Telefonia IP, são ditas aplicações intolerantes de tempo real. Isso porque pessoas não toleram atrasos na voz de mais de 200 ms, ou seja, o retardo fim-a-fim do tráfego de voz deve estar dentro de limites que dependem da aplicação, podendo variar de:

- a) 0 a 150 ms – aceitável para a maioria das aplicações;
- b) 150 a 400 ms – aceitável cuidando com o impacto do atraso na aplicação;
- c) acima de 400 ms – inaceitável para a maioria das aplicações.

As aplicações de Telefonia IP necessitam de uma largura de banda fixa mínima, dependente do *vocoder* utilizado no processo de codificação de voz. Estas taxas variam entre 5,6kbps e 64kbps.

## 2.2 ÁUDIO

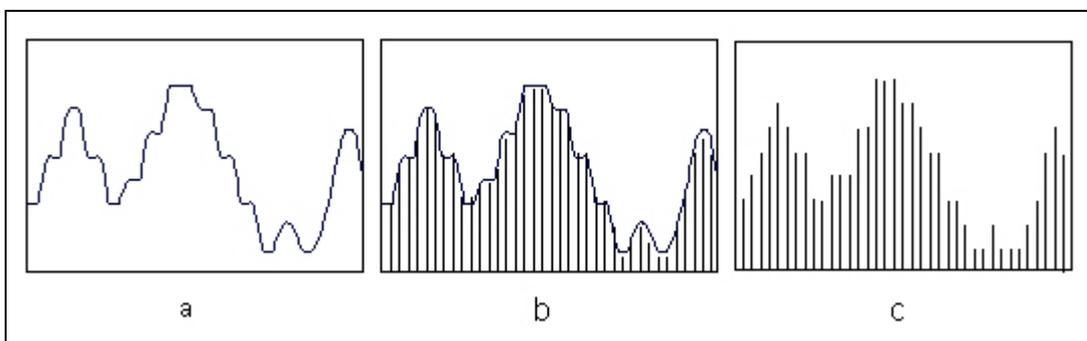
O ouvido humano é sensível a variações de som que duram milésimos de segundo e numa transmissão multimídia isso pode significar perda da qualidade da voz na aplicação.

Conforme Hersent (2001), para comparar a telefonia convencional com a Telefonia IP, primeiro uma pessoa faz uma chamada telefônica comum para alguém que conhece, e diz ‘1’, a pessoa do outro lado da linha diz ‘2’ e assim sucessivamente até 25. Marca-se então o tempo gasto para contarem até 25. Em seguida, faz-se uma chamada de telefonia IP e repete-se o mesmo jogo. Em todas as tentativas, demora muito mais tempo utilizando-se a Telefonia IP porque ocorrem atrasos e ecos que na telefonia convencional ficam imperceptíveis ao ouvido humano. Ao transportar voz sobre IP, torna-se muito mais difícil controlar o eco e o atraso.

Segundo Tarouco (2002), o áudio é obtido a partir de fontes analógicas, necessitando-se portanto, de uma conversão analógico-digital (A/D). O processo de conversão A/D pode ser dividido em três fases:

- a) amostragem: trata-se do processo de medir valores instantâneos de um sinal analógico em intervalos regulares. O intervalo entre as amostras é determinado por um pulso de *clock* e a frequência deste *clock* é chamada Taxa de Amostragem. De acordo com Hersent (2001), a taxa de amostragem está diretamente relacionada com a largura da banda de frequência do sinal que precisa ser transmitida ou armazenada. A conversão de um sinal analógico em um sinal digital envolve a captura de uma série de amostras da fonte analógica. A agregação das amostras forma o equivalente digital de uma onda sonora analógica. Quanto maior a taxa de amostragem, maior a qualidade do som digital, pois foram utilizados mais pontos de referência para replicar o sinal analógico. De acordo com a teoria de Nyquist de processamento de sinais, para representar fielmente um sinal, a taxa de amostragem precisa ser de no mínimo, o dobro da frequência do sinal;
- b) quantização: o sinal medido na amostragem é convertido para valores discretos na quantização, ou seja, é considerada a amplitude deste sinal apenas em níveis discretos. A Figura 2 mostra como uma onda de áudio (a) é amostrada (b) e quantizada (c) em valores discretos;

**Figura 2 - (a) Onda de áudio; (b) Amostragem; (c) Quantização**



- c) compressão: para reduzir a banda do canal necessária para a transmissão de voz digitalizada são utilizadas técnicas de compressão de voz que devem acontecer em tempo real para possibilitar a comunicação e interação. A compressão de sinais é baseada em técnicas de processamento que retiram informações redundantes, imprevisíveis e inúteis. Os dispositivos responsáveis pela codificação da voz são conhecidos como *voice codecs*, ou simplesmente *vocoders*. A Figura 3 mostra os principais *codecs* e suas aplicações segundo recomendações do *International Telecom Union (ITU-T)* (OLIVEIRA, 2001).

**Figura 3 - Codecs de áudio do ITU-T**

Recomendação (Ano de Aprovação)	Algoritmo	Taxa de Bits (kbit/s)	Largura de Banda (Hz)	Delay entre Pontos (ms)	Aplicação
G.711(1972)	PCM	56, 64	300 - 3,4k	< 1	Telefonia GSTN, videoconferência H.320/H.323
G.722(1988)	Sub-ADPCM	48, 56, 64	50 - 7k	< 2	Telefonia e Videoconferência ISDN
G.723.1(1995)	ACELP - 5,3MP-MLQ - 6,3	5,3, 6,3	300 - 3,4k	67-97	Videotelefonia GSTN, Telefonia H.323, VoIP (básico)
G.728(1992)	LD-CELP	16	300 - 3,4k	< 2	GSTN, Videoconferência H.320
G.729(1995)	CS-ACELP	8	300 - 3,4k	25-35	Telefonia GSTN, Modem GSTN, Videotelefonia H.324 GSTN
G.729 - A(1996)	CS-ACELP	8	300 - 3,4k	25-35	Modem GSTN, Videotelefonia H.324 GSTN

Segundo Souza (2001), a técnica de codificação de amostragem da voz humana utilizada atualmente pela telefonia convencional e que utiliza a forma de onda para digitalizar a voz é chamada de *Pulse Code Modulation* (PCM)

## 2.3 PROTOCOLOS DE TELEFONIA IP

Alguns dos protocolos tradicionalmente utilizados para uso em Telefonia IP serão citados no texto que segue.

### 2.3.1 TCP/IP

A pilha de protocolos TCP/IP apresentada na Figura 4 surgiu com a criação em 1966 da ARPAnet, uma interligação de computadores iniciada pelo governo americano através da agência *Defense Advanced Research Projects Agency* (DARPA), com o objetivo de criar um sistema de comunicação e controle distribuído para fins militares.

O TCP/IP é um acrônimo para o termo *Transmission Control Protocol/Internet Protocol*, recebendo este nome devido a seus dois protocolos mais importantes: o IP e o TCP. O protocolo IP, base da estrutura de comunicação da internet, é um protocolo baseado em comutação de pacotes utilizando datagramas.

Segundo Souza (2001), 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, permitindo assim que os computadores compartilhem seus recursos na rede.

Os protocolos TCP/IP podem ser utilizados sobre qualquer estrutura de rede, seja ela simples como uma ligação ponto-a-ponto ou uma rede de pacotes complexa. Como exemplo pode-se citar sua utilização em estruturas de rede em Telefonia IP.

O TCP é o protocolo da camada de transporte, que oferece um serviço confiável e orientado à conexão, garante a entrega de pacotes livre de erros sem perdas ou duplicação, implementa a multiplexação, gerencia as conexões, dentre várias outras capacidades. Além do TCP, a camada de transporte utiliza o protocolo UDP que opera no mesmo nível que o TCP. Por ser sem conexão e não confiável, não possui recursos de confiabilidade, sendo adequado para aplicações orientadas para transações. Ambos realizam funções de transporte de dados fim-a-fim, ou seja, considerando apenas a origem e o destino da comunicação.

Segundo Souza (2001), desde sua origem, o protocolo IP foi desenvolvido e implementado como um protocolo de comunicação sem conexão para controle de tráfego baseado na regra do melhor esforço (*best-effort service*), sem nenhum mecanismo de qualidade de serviço.

Atualmente o TCP/IP está presente na maioria dos sistemas operacionais e é usado em uma variedade de aplicações. Dentre os serviços ditos mais complexos oferecidos pelo TCP/IP, tem-se: controle de erros e fluxo, serviço com e sem conexão e envio de fluxos de dados. Tais serviços são utilizados por protocolos que fazem parte da arquitetura TCP/IP, que constituem a camada de aplicação, tais como (Figura 4):

- a) o *Filer Transfer Protocol* (FTP) utilizado para a transferência de arquivos entre sistemas computacionais usando dois tipo de conexões TCP em paralelo para a transferência;
- b) o *Simple Mail Transfer Protocol* (SMTP) responsável por encaminhar mensagens de correio eletrônico às máquinas servidoras de correio, permitindo que usuários enviem correspondência de uma LAN bem como da internet;
- c) o *Simple Network Management Protocol* (SNMP), projetado para operar com o UDP, suporta troca de mensagens de gerência de rede entre *hosts*;
- d) o *Hypertext Transfer Protocol* (HTTP) utilizado para a transferência de páginas de navegação da internet entre sistemas computacionais.

**Figura 4 - Arquitetura do Protocolo TCP/IP**

Telnet FTP SMTP HTTP NNTP	DNS SNMP RTP	Camada de Aplicação
Fluxo Confiável (TCP)	Datagrama de Usuário (UDP)	Camada de Transporte
Internet Protocol (IP)		Camada de Rede
Interface Física		Camada de Enlace e Físico

Segundo Souza (2001), com o TCP/IP um processo leva dados para um computador que por sua vez os encaminha para outro processo. Um computador transmissor fornece à rede o endereço do computador receptor para garantir encaminhamento correto pela rede.

### 2.3.2 UDP

O protocolo UDP opera no mesmo nível da camada de transporte do protocolo TCP, fornecendo uma forma simples de acesso ao sistema de comunicação permitindo que as aplicações criem e utilizem conexões de forma independente. As aplicações que utilizam o UDP como protocolo, têm a possibilidade de enviar pacotes IP brutos encapsulados sem que seja necessário estabelecer e gerir uma conexão entre as entidades UDP de transmissão e recepção.

Como se trata de um protocolo não confiável e sem conexão, a principal função que o UDP desempenha na camada de transporte é a multiplexação de acesso ao sistema de comunicação. Esta função permite que vários processos ou programas em um mesmo computador possam acessar recursos de comunicação e tráfego de dados utilizando *buffers* individuais.

O protocolo UDP apresentado na Figura 5 utiliza portas e *sockets* que orientam adequadamente o tráfego de informações a cada aplicação de nível superior. Essas portas são:

- a) porta de destino: parte do datagrama (extremidade) que indica o aplicativo ao qual deve-se enviar a informação que chega;
- b) porta de origem: localiza-se na outra extremidade do datagrama e indica o aplicativo que enviou a mensagem. Usado para a resposta e caso não se aplicar pode ser preenchido com zeros.

**Figura 5 - Formato do pacote UDP**

	16 bits		32 bits
Porta de origem		Porta de destino	
Comprimento do datagrama		Soma de verificação	
Dados			

### 2.3.3 RTP

Segundo Loureiro (1999), o uso singular do protocolo TCP/IP não atende o exigente tráfego do fluxo de voz, pois não é possível especificar e reservar a quantidade de largura de banda necessária. O uso do UDP (não orientado à conexão), adequa-se ao tráfego multimídia, onde o reenvio de dados não se faz necessário, pois pequenas perdas são suportadas pelo sistema fim-a-fim. Por outro lado, o UDP não permite configurar parâmetros de requisitos de largura de banda, podendo o serviço ser prejudicado por um congestionamento. Para integrar estes protocolos com tráfegos multimídia, alguns protocolos de camadas superiores foram propostos, dentre eles o *Real Time Transmission Protocol* (RTP) e o *Real Time Control Protocol* (RTCP).

O RTP é um protocolo padrão para transporte de dados com características de tempo real, como o áudio usado na Telefonia IP. Este protocolo atua sobre a pilha UDP/IP, não garantindo a entrega dos pacotes (não orientado à conexão), nem os requisitos de largura de banda (LOUREIRO, 1999).

O RTP possibilita a especificação de requisitos de tempo, tanto na transmissão quanto na recepção dos pacotes. Durante a transmissão dos pacotes de dados, estes podem ser perdidos, ter atrasos variados ou serem entregues fora de ordem. Dentre os serviços fornecidos pelo protocolo para permitir que o receptor detecte estes problemas, tem-se: informação sobre o tipo de dado transportado (identificação do conteúdo), *timestamps* (reconstrução temporal dos pacotes recebidos) e números de seqüência dos pacotes recebidos. Portanto, o RTP foi projetado para permitir que os receptores compensem o *jitter* (variação do tempo de atraso definido no *timestamp*) e a perda de seqüência dos pacotes introduzidos pelas redes IP.

O número de seqüência e o *timestamp* são transportados por cada pacote de RTP sendo utilizados para gerenciar o buffer de recepção dos pacotes em uma aplicação de áudio como no caso da Telefonia IP, evitando que os atrasos sejam perceptíveis pelo usuário.

Quando uma transmissão é iniciada, uma sessão RTP é criada a partir da comunicação entre os usuários participantes da sessão. Cada participante usa dois endereços de transporte para a transmissão dos pacotes, no caso do IP por exemplo, duas portas UDP na máquina local: uma porta é utilizada para o fluxo RTP e uma para as mensagens RTCP (HERSENT, 2001).

Quando o fluxo RTP é o resultado de uma combinação de vários fluxos contribuintes (utilizado principalmente em *multicast*) feita por um *mixer* RTP (dispositivos que misturam mídias de vários usuários – fluxo contribuinte - resulta em um fluxo resultante), cada fluxo é adicionado ao cabeçalho RTP e incrementado no contador de CSRC (como é chamada a lista que contém os fluxos contribuintes) representado pelo campo CC (Contador de CSRC) do pacote de RTP apresentado na Figura 6.

Todos os campos que precedem a lista CSRC estão presentes em um pacote RTP. O campo SSRC identificada por 32 bits no cabeçalho, refere-se à fonte de sincronização de um fluxo RTP. O campo *v* indica a versão do protocolo RTP; *P* representa se o *payload* (área onde fica contido o áudio) foi preenchido; o *flag X* sinaliza a presença de um cabeçalho estendido entre o cabeçalho fixo e os dados; *M* indica a presença da supressão do silêncio nos pacotes para codificações de áudio com esta capacidade; *PT* informa o tipo dos dados, ou seja, o tipo de conteúdo que o campo dados contém; o número de seqüência é incrementado a cada pacote RTP; o *timestamp* contém a freqüência do *clock* utilizada por um determinado tipo de dado.

**Figura 6 - Pacote RTP**

v	P	X	CC	M	Tipo de payload	Número de Seqüência
Timestamp						
Identificador de fonte de sincronização (SSRC)						
Identificador de fonte contribuinte (CSCR)						
Depende do Perfil					Tamanho	
Dados						

### 2.3.4 RTCP

Segundo Loureiro (1999), como o RTP não tem um mecanismo de controle sobre a conexão (transmissão e recepção), utiliza-se de outro protocolo: o *Real Time Control Protocol* (RTCP). O RTCP é geralmente usado com o RTP para permitir o transporte de algum retorno sobre a qualidade de transmissão e informações sobre os participantes da sessão.

Ambos os protocolos (RTP e RTCP) permitem aos receptores compensarem o *jitter* da rede, por meio do controle do *buffer* e seqüenciamento apropriado para que medidas corretivas possam ser tomadas. Assim como o RTP, o RTCP é usado sobre o UDP, neste caso o RTP é tradicionalmente associado a uma porta de número par e o RTCP, à próxima porta UDP de número ímpar (HERSENT, 2001).

Os participantes da sessão recebem, de tempos em tempos, pacotes RTCP de controle relativos a uma sessão RTP em particular. Esses pacotes de controle podem ser de diferentes tipos, um para cada tipo de informação. São eles:

- a) *Sender Reports* (SR): contém informações de transmissão e recepção para transmissores ativos. Descrevem a quantidade de dados enviados até o momento;
- b) *Receiver Reports* (RR): contém informações de recepção para ouvintes que não sejam transmissores ativos. Cada um desses relatórios traz informações sobre *timestamp* e o atraso desde o último relatório do transmissor recebido;
- c) *Source Description* (SDS): são pacotes usados para controle da sessão que descrevem os parâmetros da fonte e contém o *Canonical Name* (CNAME), um identificador global usado para associar diferentes fluxos de mídia gerados pelo mesmo usuário. Seu formato é similar ao de um endereço de e-mail;
- d) BYE: enviado por um participante quando ele abandona a sessão;
- e) APP: adicionam funções específicas de uma aplicação ao pacote RTP.

### 2.3.5 SIP & H.323

Atualmente o protocolo mais utilizado para a transmissão de áudio em Telefonia IP é o H.323 do *Internation Telecom Union* (ITU-T), um organismo que define padrões para redes de computadores e telecomunicações.

Segundo Hersent (2001), o que impressiona quando um documento preliminar SIP é lido, é a sua simplicidade. As diferenças entre o H.323 e o SIP são logo demonstradas em uma

transação. O SIP faz em uma transação o que a primeira versão do H.323 fazia em quatro ou cinco trocas de mensagens e pode usar o UDP ao contrário do H.323 que nas primeiras duas versões tinham que usar o TCP. O resultado final foi identificado no tempo de configuração que no SIP é menor.

Além da velocidade inferior na efetivação de uma transação no SIP, este ainda tem a capacidade de:

- a) usar sinalização UDP;
- b) funciona em *backbones* com capacidade de *multicast*, não apenas para fluxos de mídia como também para mensagens de sinalização;
- c) usar URLs como identificadores, facilitando a integração de aplicações de áudio;
- d) priorização de chamadas, negligenciado no H.323;
- e) codificação de texto.

O H.323 também tem capacidades distintas como o SIP, por exemplo: criação de canais lógicos, controle de conferências e codificação binária. A Figura 7 mostra os protocolos utilizados pelo H.323 e o SIP.

**Figura 7 - H.323&SIP**

H.323	SIP
H.323 Versão 2	SIP
H.225.0 Versão 2	SDP
H.245 Versão 3	
X.691 (ASN.1 PER)	
H.450.1	
H.450.2	
H.450.3	

A Figura 7 mostra simplicidade na utilização de protocolos por parte do SIP, utilizado neste trabalho.

### 3 SIP

Segundo o *Request For Comment* 3261 (RFC 3261), muitas aplicações da internet necessitam da criação e gerência de uma sessão (uma troca de dados entre uma associação de participantes). A implementação dessas aplicações é complicada pelos hábitos dos participantes: usuários podem mover-se entre *endpoints*, eles podem ser acessíveis por múltiplos nomes, e eles podem comunicar-se utilizando diferentes mídias, às vezes simultaneamente.

O *Session Initiation Protocol* (SIP) é um protocolo de controle da camada de aplicação usado para criar, modificar e terminar sessões com um ou mais usuários (participantes). Estas sessões incluem conferências multimídia para internet, chamadas de telefone para internet e distribuição multimídia. Participantes em uma sessão podem se comunicar via *multicast* ou *unicast*, ou uma combinação de ambos. O SIP foi definido pelo *Internet Engineering Task Force* (IETF) e está definido no RFC 3261 (2003) do IETF.

Segundo Oliveira (2001), cada requisição SIP consiste de um conjunto de campos de cabeçalho que descrevem a chamada seguidos por uma mensagem que descreve uma sessão individual que está sendo realizada pela chamada. É um protocolo cliente-sevidor parecido tanto em sintaxe como em semântica ao protocolo HTTP.

De um modo geral, as requisições são geradas por uma entidade cliente e enviadas para uma entidade receptora ou servidor. O servidor processa a mensagem e então envia uma resposta para o cliente.

As entidades que participam da comunicação e transmissão de dados multimídia e que utilizam o protocolo SIP são denominadas de entidades SIP. São elas: cliente SIP e servidor SIP. A comunicação é estabelecida entre a entidade cliente SIP e a entidade servidor SIP utilizando-se de requisições denominadas de transações. O SIP chama cada transação de pedido e cada pedido provoca uma ou mais respostas. O cliente SIP é o responsável por iniciar a comunicação, feita através do envio de um pedido SIP para o servidor SIP que é o responsável pelo envio das respostas. Pedidos e respostas SIP são tipos de mensagens (HERSENT, 2001).

Segundo Oliveira (2001), todo software SIP tem um sistema final que interage com um usuário conhecido como *User Agent*. Um *user agent* tem dois componentes: um *User Agent Client* (UAC) e um *User Agent Server* (UAS).

Inicialmente em uma comunicação SIP é necessário criar uma conexão de sinalização entre os pontos de origem e de destino da chamada. Os pontos finais SIP podem usar sinalização TCP ou UDP. O presente trabalho utiliza o protocolo UDP, e neste caso o endereço e a porta a serem usados para as repostas aos pedidos SIP, estarão contidos no parâmetro de cabeçalho denominado “Via” do pedido SIP, representado pela Figura 8.

A linha de início do pedido SIP contém o tipo de pedido enviado pelo cliente SIP, o endereço SIP do usuário destino e a versão SIP utilizada. Segundo Hersent (2001), o cabeçalho geral, comum tanto nos pedidos como nas respostas, contém os seguintes campos:

- a) Via: já citado anteriormente traz a versão SIP, o protocolo da camada de rede, o endereço IP do usuário que faz a chamada e a porta utilizada. É uma das características mais poderosas do SIP, pois mostra que o SIP foi construído levando em consideração possíveis problemas de rede;
- b) Call-ID: a primeira parte deste campo deve ser um padrão único dentro de cada computador e a última parte o nome de domínio ou endereço IP. Um novo “Call-ID” deve ser gerado para cada chamada;
- c) From: este campo contém um nome que pode ser mostrado opcionalmente e o endereço do originador da chamada. Deve estar presente tanto nos pedidos como nas respostas SIP. Nestas, o campo “From” simplesmente é copiado a partir do pedido e, portanto, não designa ser o originador da chamada;
- d) To: este campo indica o destino da chamada, sendo obrigatório em todos os pedidos e respostas SIP, onde são simplesmente uma cópia do campo presente nos pedidos;
- e) Cseq: este campo é composto por um número escolhido aleatoriamente sem sinal, que é incrementado a cada novo pedido (com exceção dos pedidos do tipo ACK e CANCEL, onde o número é o mesmo da resposta recebida – para o ACK – e o mesmo do pedido cancelado – para o CANCEL) e pelo nome do método, ou seja, que identifica o tipo de pedido que está sendo enviado. No caso de mensagens do tipo resposta, o servidor deve copiar o valor “Cseq” do pedido para as respostas correspondentes. Em ambos os tipos de mensagens (pedidos e respostas), é obrigatória a presença do campo “Cseq”.

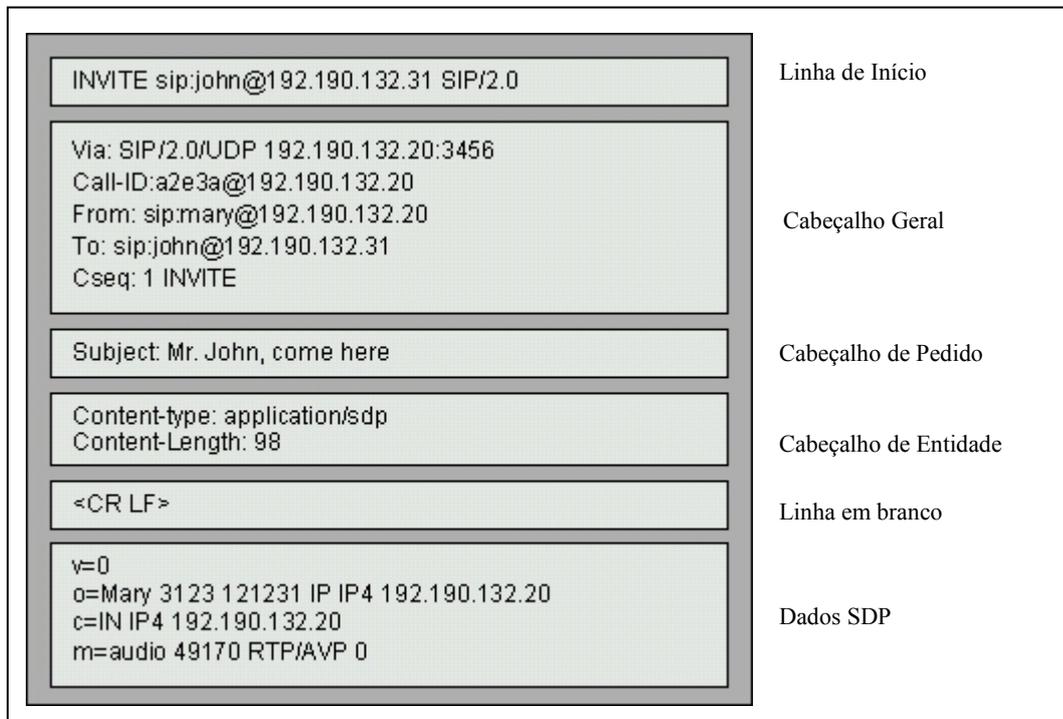
Além dos campos do cabeçalho geral (utilizado por pedidos e respostas), uma mensagem de pedido pode transportar campos com informações específicas dos pedidos, no

cabeçalho de pedido. Para indicar, por exemplo, os tipos de mídia aceitáveis na resposta, utiliza-se do campo “Accept” e para transportar informações sobre a natureza da chamada pode-se usar o campo “Subject”.

Os campos que terminam com CR e LF, são utilizados para determinar o uso de uma linha em branco entre os campos. O cabeçalho da entidade é formado por campos de cabeçalho que se aplicam diretamente ao corpo da mensagem, são eles:

- a) Content-Type: descreve o tipo de conteúdo da mensagem;
- b) Content-Length: contém o número de octetos do corpo da mensagem.

**Figura 8 - Mensagem de pedido SIP**



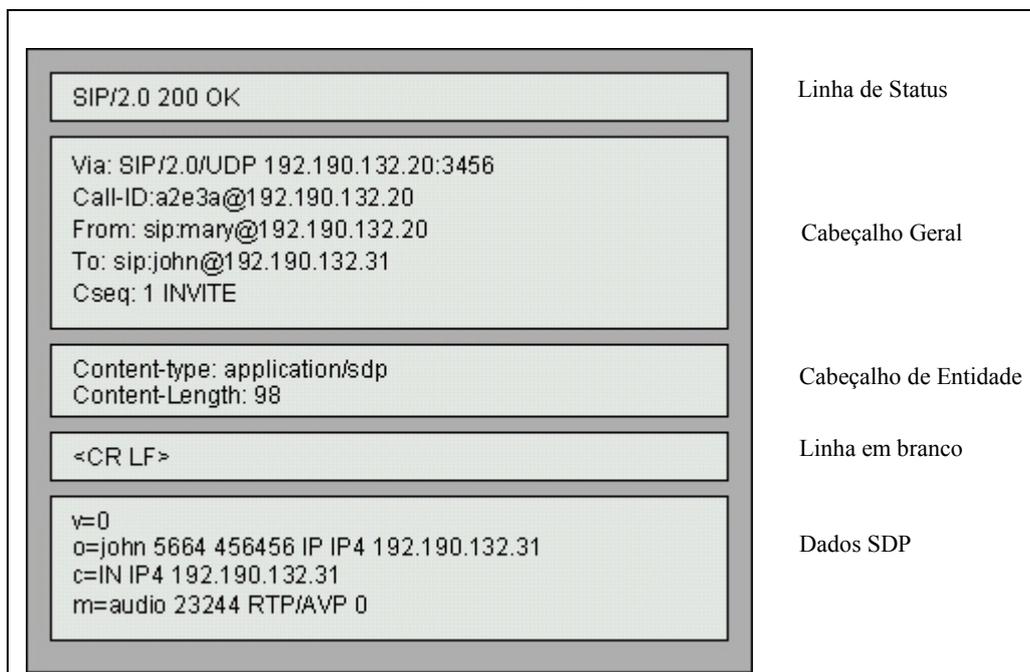
Todos os pedidos SIP são enviados do terminal cliente para o terminal servidor e podem representar diferentes funções, são elas:

- a) ACK: este pedido é enviado pelo cliente para confirmar que ele recebeu uma resposta final do servidor;
- b) BYE: este pedido é enviado pelo agente de origem ou pelo agente de destino para interromper uma chamada;
- c) CANCEL: este pedido pode ser enviado para interromper um pedido que foi enviado anteriormente enquanto o servidor ainda não tiver enviado uma resposta final;

- d) INVITE: este pedido é usado para iniciar uma chamada;
- e) OPTIONS: este pedido é enviado ao servidor pelo cliente para saber as capacidades que o servidor suporta;
- f) REGISTER: este pedido é enviado pelo cliente ao servidor para informar sua localização.

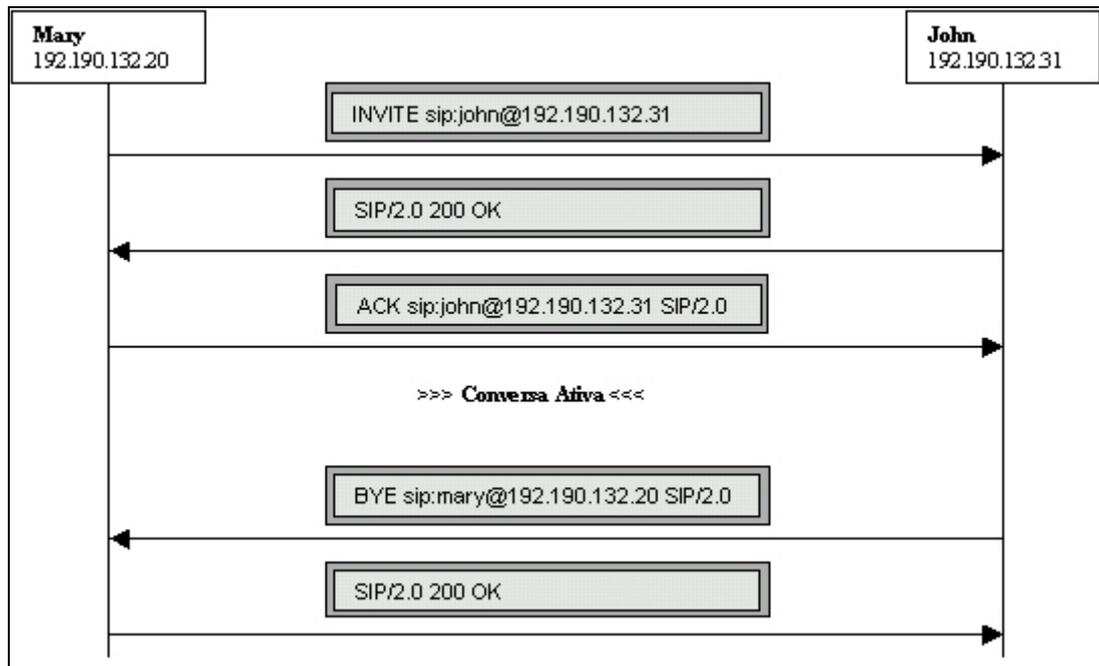
A comunicação é iniciada com a criação de uma sessão feita através da mensagem de pedido do tipo INVITE (Figura 8) e tão logo chegue ao destino, este tem condições de saber através das informações que o pedido INVITE traz, o tipo de mídia que o originador da chamada pode receber e o endereço de transporte onde o originador espera receber os dados de mídia. Para indicar que o pedido foi aceito, o destino da chamada deve enviar ao originador da chamada uma resposta 200 OK (Figura 9), que contém as capacidades de mídia do destino da chamada e a porta onde deseja receber os dados de mídia. A primeira linha de uma resposta SIP sempre contém um código de status e uma frase de justificativa. A maior parte da seção de cabeçalho é copiada a partir da mensagem do pedido original.

**Figura 9 - Resposta SIP**



Segundo Hersent (2001), um servidor SIP responde a um pedido SIP com uma ou mais respostas SIP. A maioria das respostas (2xx, 3xx, 4xx, 5xx, 6xx) são respostas que finalizam a transação SIP, enquanto que respostas 1xx são provisórias e não finalizam a sessão. A Figura 10 demonstra a sinalização da chamada completa.

Figura 10 - Sinalização Completa da Chamada



### 3.1 ENDEREÇOS SIP

Em muitos casos, o endereço SIP de um usuário será o mesmo que seu endereço de *e-mail*, mas, como padrão, os endereços SIP são URLs (*Uniform Resource Locators*) não se referindo a endereço de transporte a ser chamado.

No presente trabalho o tipo de endereço SIP a ser usado será o endereço IP (neste caso fixo) dentre os vários URLs SIP que podem ser utilizados (conforme Figura 11).

Figura 11: Tipos de Endereços SIP

john@netcentrex.net: 1234	URL SIP comum
userdomain.com	Sem a parte do usuário a porta padrão é 5060
support@company.fr:2345;transporte=UDP	Deseja contactar usando UDP
192.190.234.3:8001	Contacte o servidor neste endereço IP
support@netcentrex.net;maddr=239.255.255.1;tli=32	Uso do nome do host, mecanismo de endereço transporte e multicast para 239.255.255.1 e TTL de 32
+33-231759329@cybercall.com;user=phone	Número de telefone global
ACD@netcentrex.net?priority=high&customercode=1234	Usando cabeçalhos de extensão proprietários para controlar a prioridade de um sistema ACD..
Newcomer@reg.usergroup.com; METHOD=REGISTER	Inicia registro junto ao registrar do usergroup: reg.usergroup.com

Segundo Hersent (2001), o documento preliminar SIP especifica que o formato geral das URLs SIP é *user@host*. A parte *host* também pode ser um nome de domínio.

Opcionalmente, uma URL SIP pode conter um número de porta e caso isso não ocorra a porta padrão a ser utilizada será a 5060.

## 3.2 SDP

Segundo Hersent (2001), o SIP usa um protocolo de descrição de sessão chamado de *Session Description Protocol* (SDP) especificado no RFC 2327.

O SDP leva informações sobre tipo do fluxo de mídia, endereços, portas, tipo de conteúdo, em formato textual assim como o SIP. Quando uma chamada é estabelecida, a mensagem de pedido INVITE contém os parâmetros SDP trazendo as capacidades do originador da chamada. No caso da resposta SIP, os parâmetros SDP são modificados trazendo as capacidades do destino da chamada que responde ao pedido.

Os campos utilizados por este trabalho e que foram descritos na área de dados SDP dos pedidos (mostrado na Figura 8) e respostas SIP são:

- a) v: traz a versão do protocolo;
- b) o: nome do usuário/proprietário/criador e identificador da sessão;
- c) c: informação sobre a conexão (c=<tipo da rede> <tipo de endereço> <endereço de conexão>). Ex.: IN IP4 192.190.132.20;
- d) m: nome da mídia e endereço de transporte (m=<mídia> <porta> <transporte> <lista de formatos>). Ex.: m=áudio 49170 RTP/AVT 0.

O SDP é um protocolo em formato legível por pessoas. O uso do formato textual facilita a programação e a depuração, ao custo de uma maior largura de banda (HERSENT, 2001).

## **4 DESENVOLVIMENTO DO TRABALHO**

Em síntese, para a realização deste trabalho foram executados procedimentos de análise, especificação e implementação visando o cumprimento dos objetivos utilizando-se de metodologias de desenvolvimento de software, ferramenta de especificação e ambiente de programação já consagrados tecnologicamente de acordo com a categoria em que o trabalho se enquadra.

### **4.1 REQUISITOS DO PROTÓTIPO**

Para que seja viável a comunicação entre dois usuários que utilizam a Telefonia IP, o presente trabalho tem como funcionalidades principais: o estabelecer ou criar uma sessão VoIP através da negociação de mensagens SIP, capturar e comprimir a voz, enviar a voz comprimida em pacotes utilizando-se do RTP através do protocolo IP e, reproduzi-la na placa de som do usuário de destino.

A principal característica do trabalho é também o embasamento deste, ou seja, a utilização da negociação de mensagens baseadas no padrão SIP do IETF para a criação/finalização da sessão, feita de forma muito mais simples em comparação a protocolos tradicionalmente utilizados.

Resumidamente, este protótipo de sistema de Telefonia IP deve estabelecer e prover a comunicação entre usuários (chamados de assinantes na telefonia convencional) através do uso do protocolo IP (utilizado para a internet) dentro de uma LAN, sem fazer uso de números telefônicos convencionais e sim utilizando-se de endereços IP (que neste caso será fixo para cada usuário).

### **4.2 ESPECIFICAÇÃO**

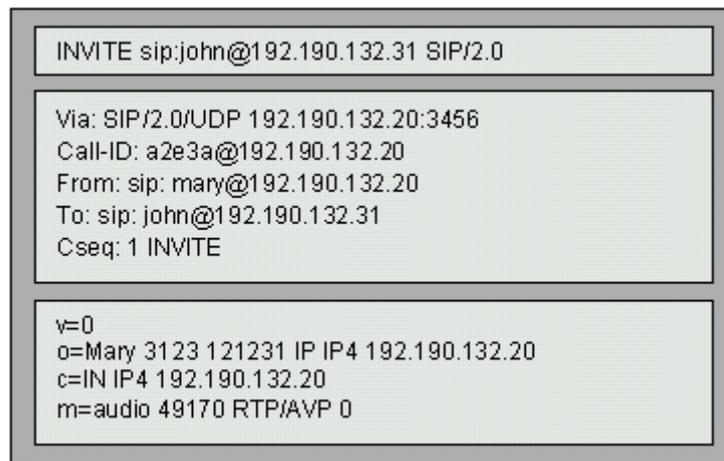
Após a análise - que é parte do processo de desenvolvimento de software e tem a finalidade de formular um modelo de domínio do problema - realizada através da orientação a objetos utilizando-se a linguagem UML, o presente trabalho foi especificado através da ferramenta Rational Rose e demonstrado através de diagrama de caso de uso e diagrama de classes.

Ambos os diagramas são utilizados para representar as classes constituintes da implementação e o entendimento das funções que elas executam.

O protótipo de Telefonia IP foi proposto com o objetivo de diminuir os custos com relação à telefonia, e neste caso desenvolvido para o uso em LANs. O entendimento quanto ao seu funcionamento é relativamente simples e pode ser apresentado na Figura 13 que representa o diagrama de caso de uso para este protótipo.

Neste protótipo, o modelo da mensagem construída é representado na Figura 12.

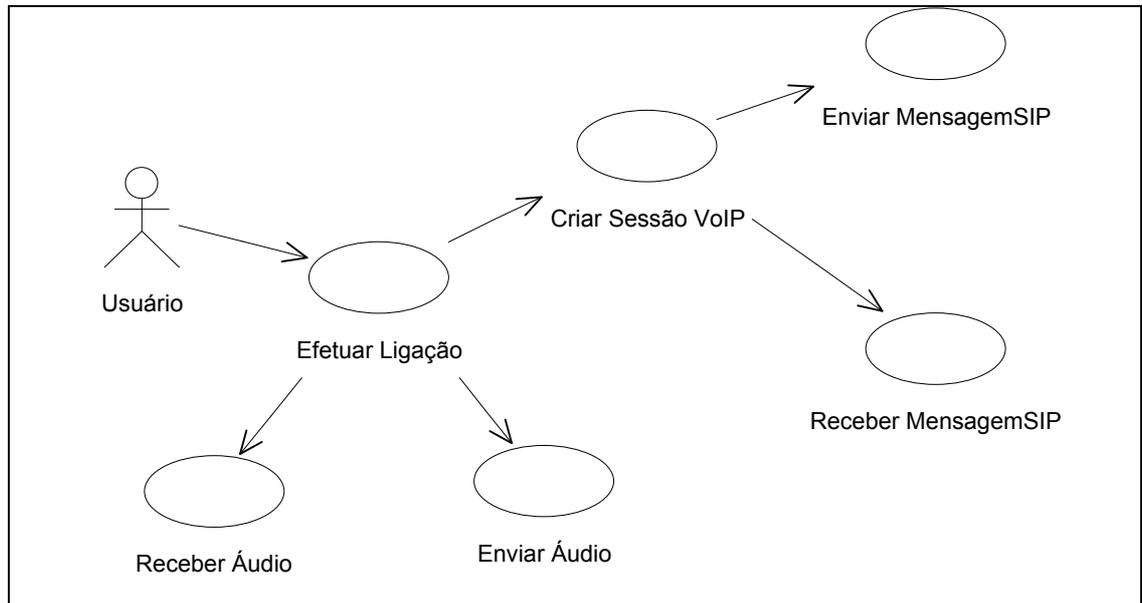
**Figura 12 - Mensagem SIP**



#### 4.2.1 DIAGRAMA DE CASO DE USO

Segundo Furlan (1998), o diagrama de caso de uso é usado para demonstrar o comportamento de uma classe sem revelar sua estrutura interna. A Figura 13 demonstra o diagrama de caso de uso do protótipo.

Figura 13 - Diagrama de Caso de Uso



O Quadro 1 descreve os casos de uso do protótipo, indicando o nome do caso de uso, o respectivo ator e a descrição de cada um deles.

Quadro 1 - Descrição dos Casos de Uso

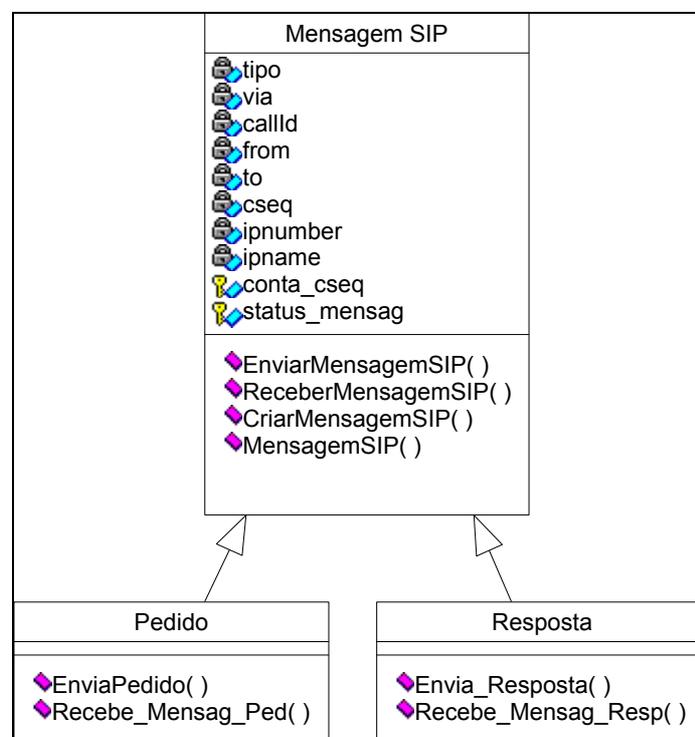
Caso de Uso	Ator	Descrição
Efetuar ligação	Usuário	O usuário envia uma mensagem de pedido para um usuário destino (previamente escolhido por ele) indicando que deseja estabelecer a conversação.
Criar Sessão VoIP	Efetuar Ligação	A criação da sessão é feita durante a negociação das mensagens SIP exemplificado na Figura 10. Após o recebimento da mensagem de resposta “200 OK” pelo originador da chamada e do envio da resposta “200 OK” pelo receptor da chamada a sessão está propriamente estabelecida.
Receber Áudio	Efetuar Ligação	Após a criação da Sessão VoIP o usuário está apto a receber o áudio capturado, comprimido, enviado via RTP e reproduzido via biblioteca utilizada.
Enviar Áudio	Efetuar Ligação	Após a criação da Sessão VoIP o usuário está apto a enviar o áudio capturado, comprimido e enviado via RTP.
Enviar MensagemSIP	Criar Sessão VoIP	O envio da mensagem dá-se de acordo com a mensagem recebida, ou seja, o envio poderá ser de uma mensagem de pedido ou uma mensagem de resposta.

Receber MensagemSIP	Criar Sessão VoIP	Caso a mensagem recebida seja de pedido, imediatamente é criada a mensagem de resposta a esse pedido. Caso a mensagem recebida seja de resposta, imediatamente é criada a resposta de pedido.
---------------------	-------------------	---

## 4.2.2 DIAGRAMA DE CLASSES

Segundo Furlan (1998), o diagrama de classes é usado para mostrar a estrutura lógica mostrando elementos tais como: classes, tipos, atributos e métodos. A Figura 14 demonstra o diagrama de classes do protótipo.

Figura 14 - Diagrama de Classes



A execução do processo “Efetuar Ligação” é iniciada com a chamada ao método `EnviaPedido()` da classe `Pedido`. Este método inicia a criação de mensagens pedido do tipo `INVITE`, `ACK`, `CANCEL` e `BYE`, sendo finalizada com a chamada ao método `CriarMensagemSIP()` da classe `MensagemSIP`. O método `CriarMensagemSIP()` verifica se o tipo de mensagem é pedido ou resposta e faz a construção da mensagem propriamente dita. Ao final da construção é chamado o método `EnviarMensagemSIP()` que encontrará o IP e a porta de destino enviando desta maneira a mensagem construída.

A execução do processo “Criar Sessão VoIP” é iniciada quando o usuário de destino recebe a mensagem através do método `Envia_Resposta()` da classe `Resposta`. Este método extrai cada campo da mensagem recebida e através da chamada ao método `CriarMensagemSIP`, dará início a criação da mensagem de resposta. Caso a mensagem a ser enviada seja “200 OK”, ao enviá-la - utilizando-se novamente o método `EnviarMensagemSIP()` - é criada uma sessão VoIP. Caso contrário a mensagem é simplesmente enviada.

O originador da chamada ao receber a resposta, verifica se ela foi aceita (“200 OK”) ou negada (“BusyHere”). Caso a resposta recebida seja um “200 OK”, o originador cria a sua sessão VoIP (neste caso se a resposta foi em decorrência de um pedido `INVITE`) ou destrói a sua sessão (se a resposta foi em decorrência de um pedido `BYE`) e em seguida faz uma chamada ao método `Recebe_Mensag_Ped()` da classe `Pedido`. Caso contrário ele simplesmente fará a chamada ao método `Recebe_Mensag_Ped()`. Este método extrai cada campo da mensagem recebida e chama o método `ReceberMensagemSIP` que irá verificar o usuário de destino. Em seguida faz uma chamada ao método `CriarMensagemSIP` que executará da mesma maneira descrita acima a construção e o efetivo envio da mensagem SIP, que neste caso será do tipo pedido `ACK` e será simplesmente recebida pelo receptor da chamada.

Após a negociação de mensagens SIP cada usuário estará apto a receber e enviar áudio (conversa), o que implicitamente será possível através do uso da biblioteca `JVOIPLIB`.

A qualquer momento um dos usuários poderá encerrar a comunicação e isso será feito através do envio de uma mensagem SIP pedido do tipo `BYE`. A construção da mensagem pedido `BYE` tem o mesmo tratamento que qualquer outra mensagem SIP. O receptor da mensagem ao recebê-la utilizará o método `Recebe_Mensag_Resp()` da classe `Resposta`. Este método dará início a construção da mensagem de resposta chamando em seguida o método `ReceberMensagemSIP()` que encontrará o usuário de destino, finalizando com a utilização do método `CriarMensagemSIP()` e o envio da mensagem através do método `EnviarMensagemSIP()`.

## 4.3 IMPLEMENTAÇÃO

Nos itens seguintes serão abordadas as técnicas e ferramentas utilizadas para a elaboração e desenvolvimento deste trabalho e detalhadamente o funcionamento do Protótipo de Sistema de Telefonia IP.

### 4.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para a implementação do protótipo foi utilizada a linguagem visual de programação Borland C++ Builder 5.0 baseando-se na biblioteca JVoIPLib escrita por Liesenborg (2002).

#### 4.3.1.1 BORLAND C++

Segundo descrito por Souza (2001), Borland C++ é uma linguagem visual descendente do C que possibilita o uso e criação de novas bibliotecas de funções. Por trabalhar sobre o paradigma da orientação a objetos como definições de classes e objetos, onde as variáveis e funções são agrupadas dentro das classes, é que o Borland C++ foi escolhido para ser utilizado na implementação deste protótipo.

Segundo Schildt (1998), Borland C++ é na verdade dois compiladores. Um compilador C++ (linguagem propriamente dita) e um compilador do C++, podendo produzir programas para ambientes gráficos.

#### 4.3.1.2 JVOIPLIB

A biblioteca JVOIPLIB foi escrita por Jori Liesenborg e significa *Jori's VoIP Library*. Tem basicamente a função de criar uma sessão VoIP, transmitir voz e destruir a sessão VoIP. Segundo Liesenborg (2002), o propósito desta biblioteca é a criação de aplicações de voz sobre IP, provendo de maneira simples a criação e gerenciamento das sessões VoIP. Dentre as diversas funções que esta biblioteca possui está a definição e especificação dos componentes necessários para a sessão VoIP e a interação entre eles.

A vantagem na utilização desta biblioteca fica evidente diante de algumas de suas características como a facilidade na criação e destruição de uma sessão VoIP e sessões

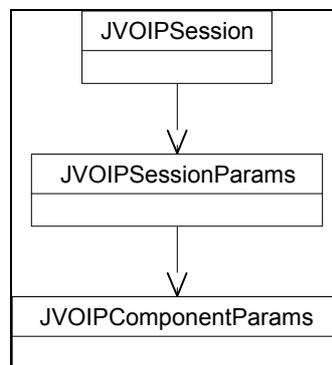
altamente configuráveis, ou seja, as configurações podem ser feitas durante a sessão estabelecida.

Segundo Liesenborg (2002), os componentes a serem utilizados em uma aplicação VoIP podem ter uma ou mais implementações incluídas nos seguintes módulos:

- a) leitura de som na entrada e saída da placa de som;
- b) compressão de áudio nos padrões DPCM, Mu-Law, GSM e LPC 5.4;
- c) transmissão de dados via RTP;
- d) um voice mixer (junta sinais enviados pela conexão VoIP criada);
- e) método de localização.

Uma sessão VoIP é representada pela classe `JVoIPSession` e a criação dá-se pela chamada da função `Create` utilizando parâmetros do tipo `JVoIPSessionParams`, que contém parâmetros para a sessão. A utilização do RTP dá-se através da classe `JVoIPParams`. O diagrama exemplificando estas três classes é representado pela Figura 15

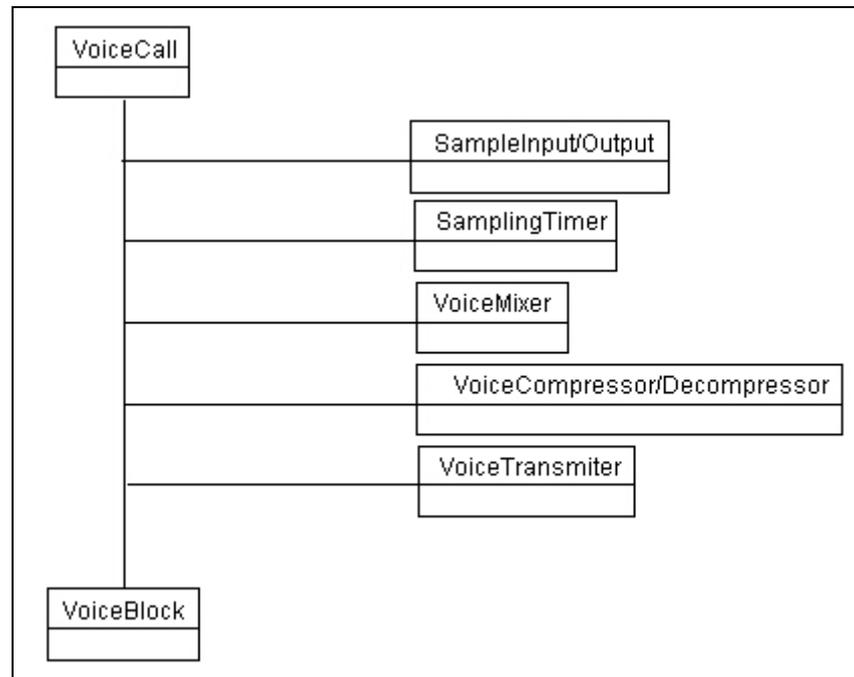
**Figura 15 - Classes (Criação da Sessão )**



A partir do momento em que a sessão se encontra ativa é necessário saber quem receberá os dados a serem enviados: isto é feito pelas funções `AddDestination` e conseqüentemente na finalização da sessão é utilizada a função `DeleteDestination`.

Após a criação da sessão é necessário que a voz seja transmitida e para isso uma das classes que torna o trabalho de voz sobre IP possível é a classe `VoiceCall` que, juntamente com outras subclasses (Figura 16), executa funções tais como: captar o sinal de voz na placa de som, comprimir e descomprimir a voz.

Figura 16 - Classes responsáveis pelo trabalho de voz



Para a destruição de uma sessão utiliza-se simplesmente a função membro da classe JVOIPSession chamada *Destroy*.

Juntamente com a biblioteca JVOIPLIB, utiliza-se implicitamente as bibliotecas JRTPLIB que serve para dar suporte ao RTP facilitando o envio e recebimento de pacotes, e JThread (definição Quadro 2) que serve para facilitar o uso de *threads* em diferentes plataformas.

Quadro 2 - Definição JThread

```

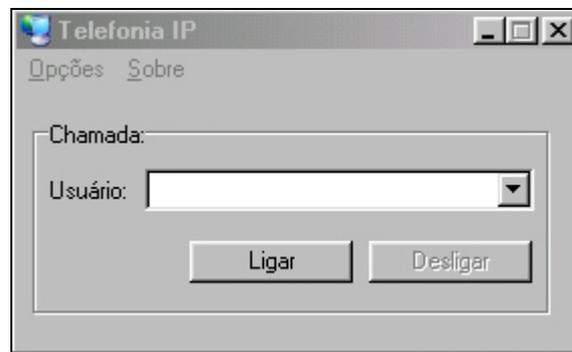
Class Jthread
{
    public
    JThread ();
    Virtual ~Jthread ();
    int Start ();
    int Kill ();
    virtual void *Thread () =0;
    bool IsRunnig ();
    void *GetReturnValue();
}
  
```

### 4.3.2 O PROTÓTIPO

Como citado anteriormente, o presente protótipo de Sistema de Telefonia IP tem a relevância de poder promover a comunicação dentro das redes dedicadas e corporativas suprimindo a infraestrutura tradicional de telefonia a custos menores.

Imagine-se trabalhando em uma grande empresa e realizando ligações utilizando a tecnologia do protocolo IP com facilidade. Ao executar o programa de Telefonia IP, é aberta a tela representada na Figura 17.

Figura 17 - Tela Inicial

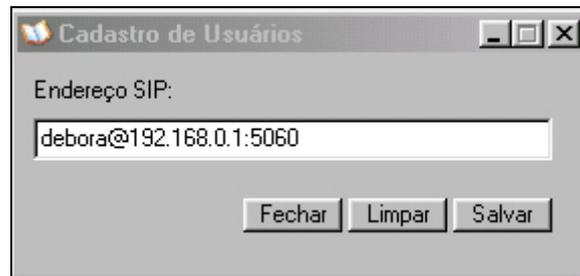


No menu opções representado na Figura 18, o usuário terá a oportunidade de acessar a tela de cadastro ou escolher a opção Sair para encerrar o programa.

Figura 18 - Menu Opções

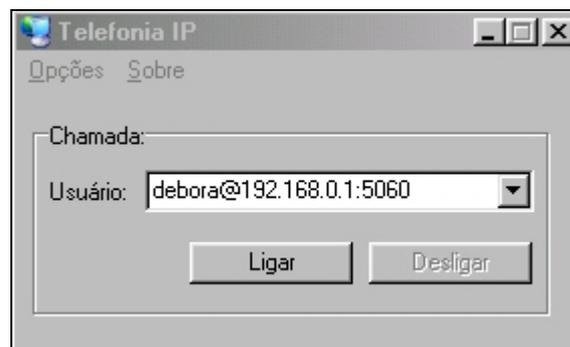


Para cadastrar um novo usuário é acessada a tela de cadastro (Figura 19) onde deverá ser colocado o endereço SIP dentro do formato `usuario@ip:porta`.

**Figura 19 - Cadastro de Usuários**

Após digitado o endereço, para salvá-lo utiliza-se o botão Salvar. O botão Limpar é utilizado antes de salvar o endereço e serve para limpar e poder corrigir o endereço digitado. O botão Fechar fecha a janela de Cadastro de Usuários retornando para a tela inicial onde será efetuada a ligação.

Na tela inicial escolhe-se um usuário para a realização da ligação, conforme Figura 20.

**Figura 20 - Seleção do Usuário**

Após selecionar o usuário com quem se deseja realizar a ligação, para estabelecer a comunicação é utilizado o botão Ligar. Ao clicar no botão Ligar é chamada uma função (Quadro 4) para a construção de uma mensagem de pedido (Quadro 3) ao usuário de destino. Neste ponto inicia-se o processo de criação de uma sessão SIP de acordo com a seqüência apresentada na Figura 10.

### Quadro 3 - Mensagem Pedido Enviada

```
INVITE sip:rubik@192.168.0.1:5060 SIP/2.0
via: SIP/2.0/UDP 192.168.0.2:5060
callid: debora@192.168.0.2
from: sip:debora@192.168.0.2
to: sip:rubik@192.168.0.1:5060
cseq 1 INVITE
v=0
o=debora 3123 121231 IP IP4192.168.0.2
c=IP IP4192.168.0.2
m=audio 5060 RTP/AVP 0
```

### Quadro 4 - Fonte do Botão Ligar

```
void __fastcall TFrmPrincipal::BtLigarClick(TObject *Sender)
{
    if (CBUsuario->Text == "")
    {
        ShowMessage("Escolha para quem ligar!!");
        CBUsuario->SetFocus();
    }
    else
    {
        Pedido mensagem;
        mensagem.Envia_pedido(tInvite, CBUsuario->Text, &sData);
    }
}
```

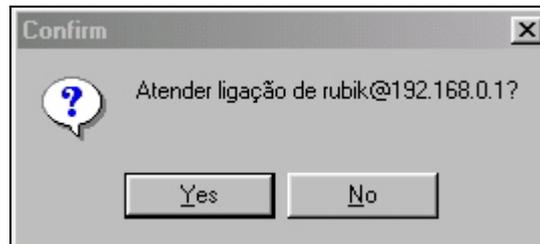
O código correspondente à construção da mensagem está representado pela Quadro 5, onde buff contém a primeira linha da Mensagem SIP apresentada na Quadro 3, ipNumber contém o número do IP da própria máquina, ipName contém o nome do computador, usuario contém o destinatário da mensagem, conta\_cseq contém o número cseq atual, id\_mensag indica o tipo da mensagem e nome indica a quem pertence a sessão.

### Quadro 5 - Fonte de Construção da Mensagem SIP

```
tipo = buff;
via = "via: SIP/2.0/UDP "+ipNumber+":5060\n";
callId = "callId: "+ipName+"@"+ipNumber+"\n";
from = "from: sip:"+ipName+"@"+ipNumber+"\n";
to = "to: sip:" + usuario + "\n";
cseq = "cseq " + IntToStr(conta_cseq) + ' ' + ind_mensag + "\n";
sdp.v = "v=0\n";
sdp.o = "o=" + nome + " 3123 121231 IP IP4" + ipNumber + "\n";
sdp.c = "c=IP IP4" + ipNumber + "\n";
sdp.m = "m=audio 5060 RTP/AVP 0 \n";
```

O usuário destinatário ao ser contactado, recebe a mensagem exemplificada na Figura 21 e pode aceitar ou recusar a ligação.

**Figura 21 - Mensagem de Contato para Ligação**



Caso aceite a ligação, implicitamente uma mensagem de confirmação “200 Ok” (Quadro 6) é enviada ao originador da chamada. Neste momento a criação da sessão é estabelecida para o originador da chamada, conforme Quadro 7 que representa o código implementado.

**Quadro 6 - Mensagem Resposta de Confirmação**

```
SIP/2.0 200 OK
via: SIP/2.0/UDP 192.168.0.1:5060
callid: rubik@192.168.0.1
from: sip:rubik@192.168.0.1
to: sip:debor@192.168.0.2:5060
cseq 1 INVITE
v=0
o=rubik 3123 121231 IP IP4 192.168.0.1
c=IP IP4 192.168.0.1
m=audio 5060 RTP/AVP 0
```

### Quadro 7 - Criação da Sessão

```
JVOIPSessionParams params;
JVOIPRTPTransmissionParams rtpparams;
unsigned long long_ip;

if (sData.session) //verifica se já existe sessão
{
    sData.session->ClearDestinations(); //limpa a lista de destinatários
    sData.session->ClearAcceptList(); // limpa a lista de destinatários aceitos
    sData.session->Destroy(); //destroi a sessão
    sData.stat->SimpleText = "";
    delete (sData.session);
}

rtpparams.SetAcceptOwnPackets(true); //habilita o RTP a aceitar o proprietário dos pacotes
params.SetTransmissionParams(&rtpparams); //parâmetros de transmissão RTP.
sData.session = new JVOIPSession; //instancia nova sessão
sData.session->Create(params); //chama construtor da classe JVOIPSession
long_ip = inet_addr(ip.c_str()); //converte IP
long_ip = ntohl(long_ip); //converte IP
sData.stat->SimpleText = ip;
sData.session->AddDestination(long_ip,5061); //adiciona na lista de destinos
sData.session->AddToAcceptList(long_ip,5061); //adiciona na lista de destinos aceitos
```

O originador da chamada por sua vez, envia também de forma implícita uma mensagem caracterizando que entendeu a confirmação (Quadro 8) e neste momento é criada sessão VoIP (Quadro 7) no receptor da chamada.

### Quadro 8 - Mensagem de Pedido de Confirmação

```
ACK sip:rubik@192.168.0.1:5060 SIP/2.0
via: SIP/2.0/UDP 192.168.0.2:5060
callid: debora@192.168.0.2
from: sip:debora@192.168.0.2
to: sip:rubik@192.168.0.1:5060
cseq: 1 ACK
v=0
o=debora3123 121231 IP IP4192.168.0.2
c=IP IP4192.168.0.2
m=audio 5060 RTP/AVP 0
```

Imediatamente após a negociação das mensagens, a conversação pode ser iniciada.

O botão Desligar presente na tela inicial é utilizado para encerrar a ligação. Quando clicado, é chamada uma função (Quadro 9) para a construção da mensagem pedido “BYE” (Quadro 10) que será enviado ao usuário de destino caracterizando que a sessão e a conversação serão encerradas. Este processo está representado seqüencialmente na Figura 10. O código correspondente à construção da mensagem está representado na Quadro 5.

**Quadro 9 - Fonte do Botão Desligar**

```

void __fastcall TFrmPrincipal::BtDeslClick(TObject *Sender)
{
    BtDesl->Enabled = false;
    BtLigar->Enabled = true;
    Pedido mensagem;
    mensagem.Envia_pedido(tBye, CBUuario->Text, &sData);
}

```

**Quadro 10 - Mensagem Pedido "BYE"**

```

BYE sip:rubik@192.168.0.1:5060 SIP/2.0
via: SIP/2.0/UDP 192.168.0.2:5060
callid: debora@192.168.0.2
from: sip:debora@192.168.0.2
to: sip:rubik@192.168.0.1:5060
cseq 73369512 BYE
v=0
o=rubik 3123 121231 IP IP4192.168.0.2
c=IP 192.168.0.2
m=audio 5060 RTP/AVP 0

```

Após enviar a mensagem de pedido “BYE”, o mesmo procedimento é realizado pelo protótipo, ou seja, o originador recebe uma mensagem resposta “200 OK” e envia uma mensagem de pedido “ACK” sempre implicitamente. No momento em que o originador recebe um “200 OK” e o receptor recebe um “ACK”, a sessão é desfeita conforme implementação representada na Quadro 11.

**Quadro 11 - Finalização da Sessão**

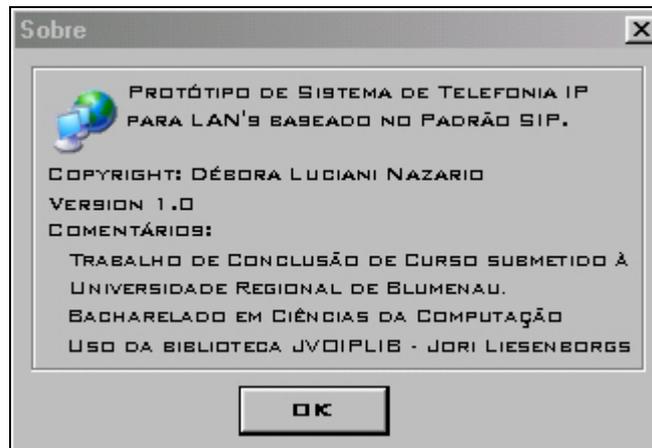
```

sData.session->ClearDestinations(); //limpa a lista de destinatários
sData.session->ClearAcceptList(); //limpa a lista de destinatários aceitos
sData.session->Destroy(); //destroi a sessão
delete (sData.session);

```

O menu Sobre citado anteriormente, ao ser clicado traz informações gerais sobre o protótipo e o autor e está representado na Figura 22.

Figura 22 - Sobre



## 5 CONCLUSÕES

Durante as etapas de pesquisa e desenvolvimento do trabalho, foi possível constatar que apesar da Telefonia IP ter se transformado rapidamente em uma indústria, a tecnologia envolvida é relativamente recente, encontrando obstáculos como a estrutura de rede a ser envolvida para a transmissão de áudio em tempo real.

O uso do padrão SIP, um protocolo extremamente simples se comparado com protocolos já utilizados para a tecnologia de voz sobre IP, foi dificultado pela limitação de material de pesquisa para a fundamentação teórica. O SIP é um protocolo especificado na RFC 3261 do IETF e que trata da negociação de mensagens com o objetivo de estabelecer a comunicação entre dois ou mais usuários, seja dentro de uma mesma LAN (objetivo deste trabalho) ou na rede mundial de computadores. A negociação das mensagens para estabelecimento e encerramento da comunicação é feita somente através de dois envios de mensagens pelo originador da chamada e do envio de uma mensagem pelo receptor da chamada. Durante a negociação das mensagens SIP a sessão é criada a partir do uso de bibliotecas e componentes que utilizam o protocolo RTP.

O protocolo RTP é o ideal para transmissão de dados em tempo real e seu uso foi possível graças ao uso das bibliotecas JVOIPLIB encontradas para C++, linguagem esta utilizada na implementação do protótipo através do ambiente de programação Borland C++ Builder. Assim como na implementação que utilizou o ambiente mais apropriado para o seu desenvolvimento, para a especificação foi utilizada uma das ferramentas mais utilizadas para a orientação a objetos e UML, o Rational Rose. Ambos supriram todas as necessidades de que este trabalho exigiu.

Durante a fundamentação deste trabalho algumas técnicas de compressão de áudio foram citadas, apesar disso a compressão utilizada foi a compressão *default* das bibliotecas utilizadas chamada DPCM.

A implementação do protótipo no ambiente de programação Borland C++ Builder utilizou um componente para que fosse possível o uso do protocolo UDP chamado NMUDP que necessita dentre outros atributos a porta local e o IP do receptor. Informações estas previamente cadastradas, por se tratar de uma LAN onde os IPs são eventualmente fixos.

A grande vantagem na utilização deste protótipo de transmissão de áudio em tempo real, é o estabelecimento da comunicação em uma LAN a custos menores em relação ao da telefonia convencional.

## 5.1 EXTENSÕES

Procurando dar continuidade ao desenvolvimento de um sistema completo de Telefonia IP com base no padrão SIP, sugere-se:

- a) a implementação desta tecnologia para uso em *multicast* para sistemas de áudio-conferência;
- b) o uso de criptografia no envio das mensagens SIP de forma segura.

## REFERÊNCIAS BIBLIOGRÁFICAS

FAGUNDES, Eduardo. **A Convergência de Dados e Voz na Próxima Geração de Redes (NGN)**, [S.l.], [2002]. Disponível em: <<http://www.efagundes.com/Artigos/A%20Convergencia%20de%20Dados%20e%20Voz%20na%20NGN.htm>> . Acesso em: 08 jun. 2003.

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

HERSENT, Oliver; GURLE, David; PETIT, Jean-Pierre. **Telefonia IP: comunicação, multimídia baseada em pacotes**. São Paulo: Makron Books, 2001. 451 p.

LIESENBORGS, Jori. **JVOIPLIB**, [S.l.], [2002]. Disponível em: <<http://lumumba.luc.ac.be/jori/jvoiplib/jvoiplib.html>>. Acesso em: 08 jun. 2003.

LOUREIRO, Hélio; SAVARIS, Nixon. **Protocolos Internet para Comunicação Multimídia**, Florianópolis, [1999]. Disponível em: <[http://www.lcmi.ufsc.br/redes/redes99/helio/protocolos\\_multimidia/](http://www.lcmi.ufsc.br/redes/redes99/helio/protocolos_multimidia/)> Acesso em: 08 jun. 2003.

MESQUITA, Renata. **InfoExame**, São Paulo, [2002]. Disponível em: <<http://info.abril.com.br/aberto/infonews/102002/14102002-19.shl>>. Acesso em: 19 mar. 2003.

OLIVEIRA, Sérgio. **Telefonia IP para ambientes móveis compactos**, Belo Horizonte, [2001]. Disponível em: <<http://www.lecom.dcc.ufmg.br/~sergiool/telefonia/telefoni.htm>>. Acesso em: 08 jun. 2003.

*INTERNET ENGINEERING TASK FORCE. RFC 3261:Session Internet Protocol*. New York, 2002. 269p.

*INTERNET ENGINEERING TASK FORCE. RFC 3264:Session Description Protocol*. New York, 2002. 16p.

SCHILDT, Herbert. **Borland C: completo e total**. São Paulo: Makron Books, 1998. 1114p.

SOUZA, José Marcio de. **Protótipo de um sistema de VoIP** (Voz sobre IP). 2001. 53 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TAROUCO, Liane. **Grupo de Trabalho RNP: aplicações educacionais em rede**, Porto Alegre, [2002]. Disponível em: <http://penta3.ufrgs.br/RNP/cap3/3.2%20Audio/>. Acesso em: 08 Jun. 2003.