Avaliação dos protocolos VoIP SIP e IAX utilizando simulação e parâmetros de qualidade de voz

Mateus Godoi Milanez

Avaliação dos protocolos VoIP SIP e IAX utilizando simulação e parâmetros de qualidade de voz

Mateus Godoi Milanez

Orientadora: ***Sarita Mazzini Bruschi***

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação — ICMC/USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Fevereiro/2009

Dedico este trabalho a minha família, minha namorada Talita e em especial ao meu amigo Renan Campos. Força meu irmão.

"Lutar sempre. Vencer talvez. Desistir jamais."

Agradecimentos

Agradeço primeiramente a Deus pelas minhas virtudes e defeitos e a todas as pessoas que contribuíram para a formação do meu caráter.

Agradeço ao suporte e oportunidade propostos pelo ICMC-USP, aos professores e funcionários, principalmente à minha orientadora Sarita Mazzini Bruschi, que me aceitou mesmo sabendo da minha não dedicação integral ao programa e às secretárias da seção de pós-graduação pelos avisos ininterruptos de datas definidas pelo programa.

Agradeço aos diretores e donos das empresas 3WT (*Wireless Web World Tech*) e SV Consultoria e Tecnologia, as quais trabalhei durante o desenvolvimento deste projeto, pela compreensão e profissionalismo nos períodos em que precisei de dedicação exclusiva ao mestrado.

Agradeço à minha família, aos meus amigos Renan Prates Lopes de Campos, Rafael Vital Aroca, Elton Bovolaro de Castro, Gecirlei Francisco da Silva (Aôôô Moreno !) e a todas as pessoas que me ajudaram, tecnicamente ou espiritualmente, a passar por mais esta etapa da minha vida.

Em muitos momentos durante o desenvolvimento desse trabalho chorei, sofri, pensei em desistir. Por isso, são salientados os agradecimentos ao meu irmão Douglas Henrique Milanez, ao meu amigo Diego Fiori de Carvalho e a minha namorada, amiga e companheirona Talita Cristina Corsi.

"Criei barriga, a minha mula empacou.
Mas vou até o fim !"
Até o fim - Chico Buarque de Holanda

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

HTML – *HyperText Markup Language*
HTTP – *Hypertext Transfer Protocol*
IAX – *Inter-Asterisk Exchange Protocol*
IETF – *Internet Engineering Task Force*
IP – *Internet Protocol*
ISDN – *Integrated Services Digital Network*
ITU – *International Telecommunication Union*
LAN – *Local Area Network*
MGCP – *Media Gateway Control Protocol*
MOS – *Mean Opinion Score*
NGN – *Next Generation Network*
NS – *Network Simulator*
RTP – *Real Time Transport Protocol*
RTSP – *Real Time Streaming Protocol*
RTCP – *Real-time Transport Control Protocol*
SDP – *Session Description Protocol*
SIP – *Session Initiation Protocol*
TCP – *Transmission Control Protocol*
UDP – *User Datagram Protocol*
URL – *Uniform Resource Locator*
UTF-8 – *8-bit Unicode Transformation Format*
URI – *Uniform Resource Indicators*
VOIP – *Voice Over IP*
WTG – *Web Traffic Generation*

Resumo

Recentemente, as tecnologias de telecomunicações estão convergindo para a concepção da *Next Generation Network*, onde propõe-se que todas as informações trocadas sejam classificadas por prioridade e segurança. Porém, como as redes atuais ainda não promovem tais práticas, protocolos VoIP, em conjunto a outras soluções, buscam a melhoria da qualidade das ligações. Como o protocolo VoIP IAX vem ganhando credibilidade na comunidade *open source* nos últimos anos, torna-se relevante compará-lo ao protocolo SIP, o qual é bastante investigado pela literatura.

Desta forma, o objetivo deste trabalho é o estudo e avaliação dos protocolos SIP e IAX, através de verificações de qualidade do áudio em ligações VoIP. Para a realização dos experimentos foi desenvolvida uma estrutura que representasse chamadas VoIP no simulador *Network Simulator* e, para tais ligações, empregou-se método de avaliação de qualidade PESQ.

Assim, foi possível a verificação das semelhanças compreendidas entre os protocolos SIP e IAX diante dos problemas de perda de pacotes, atraso, limitação da taxa de dados e *jitter*.

palavras-chave: VoIP, SIP, IAX, PESQ, qualidade, simulação.

Abstract

Telecommunications technologies are recently converging to the Next Generation Network conception, where it is proposed that all exchanged information should be classified by security and priority. As the currently available networks do not provide such practices, VoIP protocols, among other solutions, aim for the improvement of the calls quality. As the IAX VoIP protocol had been receiving credibility in the open source community in the last years, it is relevant to compare it to the SIP protocol, which is widely investigated in the literature.

In this way, the objective of this work is the study and evaluation of the SIP and IAX protocols through verifications of audio quality in VoIP calls. To implement the experiments, a structure that represents VoIP calls was developed in the "Network Simulator"software. For these calls, the PESQ method was used to evaluate the calls quality.

Using this approach, it was possible to verify similarities between the SIP and IAX protocols regarding the problems of packet loss, delay, limitation in the data rate and jitter

Keywords: VoIP, SIP, IAX, MOS, PESQ, voice quality .

Capítulo

1

Introdução

As primeiras formas de comunicações telefônicas surgiram no final do século XIX. No Brasil, o primeiro telefone foi trazido ao Rio de Janeiro por Dom Pedro II em 1877 [Casson, 2004]. Todavia, a expansão das capacidades de telecomunicação ocorreu no século XX, mais precisamente nas décadas de 30 e 40. O surgimento de mecanismos para troca de dados através de redes de computadores estabeleceu-se entre 1960 e 1970, tendo sua popularização nas décadas subseqüentes [Kessler, 1990]. Com o avanço das tecnologias, tornou-se possível a realização de conversas telefônicas por meio de redes de computadores. Esta técnica foi então nomeada de Voz sobre IP ou VoIP.

Nos últimos anos as telecomunicações vão em direção à concepção da *Next Generation Network* [ITU-T, 2004a], onde almeja-se a convergência de todas as redes de comunicação e propõe-se uma classificação das informações de acordo com funcionalidades e segurança necessárias. Como as redes atuais ainda não promovem interpretação e diferenciação de dados por meio de um padrão único, diversas práticas de melhoria da qualidade do áudio trafegado durante ligações VoIP foram desenvolvidas.

Participantes ativos deste processo, os protocolos VoIP, além de gerenciar as informações envolvidas no estabelecimento das chamadas, definem estruturas que otimizam o transporte dos dados de voz trocados nas ligações. Tais estruturas são utilizadas tam-

bém na resolução de problemas de qualidade gerados por limitação das redes, as quais destacam-se: perda de pacotes, atraso, *jitter*[1] dentre outros.

Dentre os protocolos VoIP encontrados, o SIP destaca-se pela ampla investigação nos últimos anos e o IAX por ganhar credibilidade na comunidade *open source* [Abbasi and Prasad, 2005]. A procura por análises detalhadas dos citados protocolos na literatura resultou apenas no trabalho de Abbasi e Prasad [Abbasi and Prasad, 2005]. Todavia, em tal pesquisa, o uso de diferentes esquipamentos nos experimentos causou uma grande dispersão nos resultados, impedindo uma análise isolada dos protocolos.

Devido às características incipientes do trabalho de Abbasi [Abbasi and Prasad, 2005], tornou-se necessária uma maior investigação sobre o funcionamento dos protocolos VoIP SIP e IAX, e a conseqüente possibilidade de uma análise detalhada sobre os mesmos. Neste contexto, a utilização de simulação para manipulação dos experimentos é bastante adequada pela facilidade de obtenção de resultados, alteração de cenários e possível inclusão de outros protocolos. Para a execução das análises foi empregado o simulador de redes *Network Simulator* [Ns, 2006] que, apesar não ter implementado em seu núcleo protocolos VoIP, possui uma série de facilidades que minimizam o processo de inclusão à sua estrutura e é muito utilizada no meio acadêmico.

O foco deste projeto de mestrado é o estudo e avaliação detalhada dos protocolos VoIP SIP e IAX através de verificações de qualidade em chamadas simuladas.

Para alcançar tal objetivo, foi necessário inicialmente o desenvolvimento de uma extensão ao simulador NS. Em seguida, realizou-se estudos sobre métodos quantitativos empregados em análises de qualidade em ligações VoIP, os quais foram indispensáveis para a escolha do método PESQ [ITU-T, 2001]. Depois, exigiu-se ainda a criação de uma estrutura capaz de representar os dados trocados nas simulações por sons que pudessem ser avaliados.

De forma a relatar os passos percorridos para o desenvolvimento do trabalho, esta monografia está dividida em seis capítulos: um de introdução, dois capítulos de revisão bibliográfica, dois capítulos que relatam o desenvolvimento do trabalho e descrição dos experimentos e um capítulo de conclusões.

O capítulo 2 descreve inicialmente as dessemelhanças da comutação por pacotes e comutação por circuitos. Em seguida apresenta descrição sobre as estruturas empregadas pelo protocolo VoIP H.323[ITU-T, 2006b], e ilustra detalhes sobre a organização e funcionalidades dos protocolos SIP e IAX. Por fim, é exibida uma comparação teórica entre SIP e IAX e comentados diversos trabalhos têm como objetivo a avaliação de desempenho VoIP.

[1] medida de variação do atraso entre os pacotes sucessivos de transmissão de dados

O capítulo 3 apresenta informações sobre métodos subjetivos e objetivos de análise de qualidade. Dentre tais métodos, o funcionamento do PESQ[ITU-T, 2001], método utilizado nas análises deste trabalho, é destacado nesse capítulo.

O capítulo 4 apresenta inicialmente uma breve descrição do simulador de redes NS. Em seguida, informações sobre implementações dos protocolos VoIP são destacadas em conjunto à estrutura criada para representação do tráfego de áudio das simulações. Para finalizar, são descritos os experimentos e cenários definidos para a realização das análises.

O capítulo 5 exibe os resultados obtidos pela análise de qualidade das ligações resultantes das simulações dos protocolos SIP e IAX, através de análises estatísticas de testes de hipótese.

Por fim, o capítulo 6 apresenta as conclusões alcançadas, discute as contribuições deste trabalho à comunidade de pesquisa na área e exibe algumas propostas de trabalhos futuros.

Capítulo

2

Voice over IP (VoIP)

2.1 Considerações Iniciais

A comunicação empregada no sistema de telefonia tradicional é baseada na comutação por circuitos. Neste tipo de comunicação, dados são trocados após o estabelecimento de um caminho fixo por um meio de transmissão, isto é, um circuito fim a fim é inicialmente estabelecido, delimitando uma rota entre estações que permanece operante até que uma das extremidades participantes decida finalizar a comunicação.

Por ter o caminho estabelecido exclusivo, a entrega dos dados de forma ordenada e com pouca variação de tempo é garantida, adaptando muito bem o tráfego de mídias de tempo real à comutação por circuitos. Entretanto, se os recursos disponíveis não forem utilizados de forma coordenada, a capacidade do meio físico é desperdiçada, ou seja, o estabelecimento de novas conexões podem não ocorrer pela dedicação de canais a outras conexões.

Diferentemente do modelo adotado pelo sistema telefônico, a transferência de informações entre computadores é feita através da divisão dos dados em pequenas partes denominadas pacotes. Neste tipo de comunicação, uma conexão lógica é estabelecida entre as estações, onde os fragmentos de dados são multiplexados através dos canais físicos disponíveis, não obrigando o uso de uma especifica conexão fim a fim.

As chamadas VoIP estão inseridas no ambiente de comunicação por pacotes, e justamente por não estabelecerem uma rota física dedicada para a transmissão dos dados dependentes do tempo, apresentam problemas relacionados a desorganização, atrasos e variações de tempo na chegada das informações. Contudo, apesar dos problemas citados, consegue-se obter boa qualidade nas chamadas VoIP. A medição desta qualidade é calculada a partir de limitações de estrutura, tais como: atraso, *jitter* e pacotes perdidos.

Em praticamente todas as etapas de uma comunicação VoIP, atrasos são gerados. Assim, para facilitar o entendimento de tais barreiras, a figura 2.1 ilustra os processos responsáveis pelos atrasos envolvidos em uma transmissão VoIP.

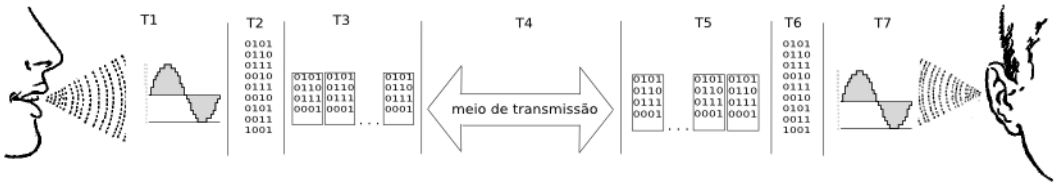

Figura 2.1: Etapas de atraso ocorridas em ligações VoIP.

O atraso se incia na etapa de digitalização pois, assim que o áudio é capturado por algum equipamento, ele passa pelo processo de transformação dos dados analógicos para digitais (T1). Em seguida, vem a fase de codificação (T2), onde, efetua-se o processo de compressão do áudio capturado. Porém, este procedimento não pode ter muita demora e deve manter um nível significativo de semelhanças em relação ao som original, quando descompactado. A terceira etapa causadora de atraso é a de empacotamento de dados (T3). Em tal procedimento, as informações codificadas são encapsuladas ao protocolo que realizará o transporte e transmissão da voz até o receptor. O transporte dos dados (T4) é a etapa onde os valores de atraso estão intimamente ligados aos recursos encontrados no caminho entre a gerador e o receptor final dos dados. Por fim, os passos de desempacotamento (T5), descompressão (T6) e transformação dos dados em áudio sonoro (T7) também contribuem para a soma do atraso total.

O *jitter* é a medida de variação dos atrasos entre sucessivos pacotes e este ocorre principalmente na etapa de transporte de dados. Quando uma seqüência de pacotes é transmitida, o atraso sobre cada pacote pode mudar dependendo do meio de comunicação, refletindo inclusive em possíveis desorganizações na chegada dos pacotes. Para amenizar estes problemas, alguns métodos foram desenvolvidos. Um dos procedimentos mais simples faz uso de memória temporária no receptor (*buffer*), onde as informações recebidas são armazenadas durante um período de tempo e depois consumidas de forma constante.

Já a perda de pacotes pode ocorrer por problemas físicos em equipamentos ou por congestionamento de dados. Esta última forma ocorre quando estações necessitam descartar

pacotes por motivos de sobrecarga de dados. Os efeitos causados pelas perdas podem ser minimizados através de algoritmos que tentam reconstituir as informações extraviadas na comunicação, usando dados predecessores e sucessores dos pacotes perdidos. Adicionalmente, o uso da técnica de retransmissão não é muito praticada em chamadas VoIP pelo tempo gasto no reenvio de pacotes perdidos.

O tempo de espera para realização de chamadas também é importante em uma comunicação VoIP. Seguindo o conceito da comutação por circuitos, as chamadas VoIP estabelecem canais lógicos de sinalização antes do início do tráfego de voz, classificando os protocolos em duas categorias: protocolos de sinalização, responsáveis pelo estabelecimento, criação e encerramento de uma chamada, e protocolos de mídia, que fazem o transporte de dados de voz quando a chamada está estabelecida.

Este capítulo apresenta na seção 2.2 características de dois protocolos de sinalização, mais precisamente H.323 e SIP. A seção 2.3, define informações sobre a estrutura do protocolo de transporte de mídia RTP. Na seqüência, a seção 2.4 exibe detalhes de funcionamento sobre o protocolo IAX, o qual provê tanto sinalização quanto transporte de áudio. Por fim, a seção 2.5 ilustra uma comparação teórica entre os protocolos SIP e IAX e a seção 2.6 discute trabalhos da área de avaliação de protocolos VoIP encontrados na literatura.

2.2 Protocolos de Sinalização

Os protocolos de sinalização são responsáveis pela criação, modificação e encerramento dos canais telefônicos virtuais existentes em chamadas VoIP. Fundamentalmente, representam o funcionamento lógico do gerenciamento dos canais físicos em ligações telefônicas que usam a comutação por pacotes. Dentre os protocolos de sinalização existentes, serão abordados nesta seção o padrão H.323 e o protocolo SIP.

2.2.1 H.323

O padrão H.323 foi criado originalmente pelo ITU (*International Telecommunication Union*) com o objetivo de prover mecanismos de transporte de aplicações multimídia em redes locais. Contudo, com o crescimento das redes e integração com telefonia pública tradicional, o padrão H.323 necessitou de rápida adaptação.

Visto como o resultado do agrupamento de uma série de protocolos que foram criados de acordo com o surgimento das necessidades, o H.323 especifica um sistema de telefonia IP funcional formado por camadas. Inclusive, como foi um dos primeiros padrões de comu-

nicação VoIP, o H.323 acabou abrangendo também protocolos de telefonia tradicional. A figura 2.2 exibe a disposição dos protocolos aplicados ao padrão H.323 [Tanenbaum, 1999].

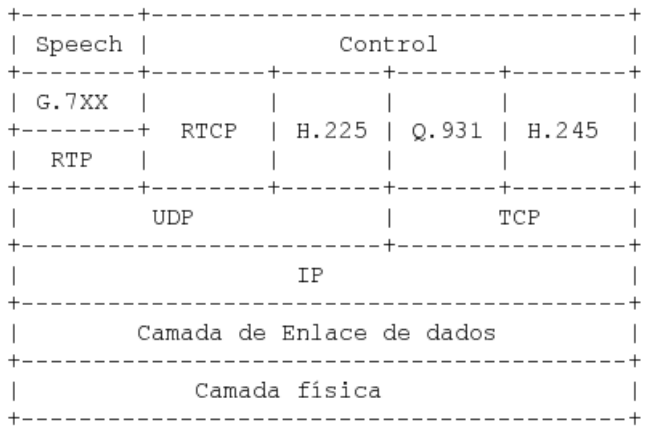

Figura 2.2: Modelo de camadas aplicado ao padrão H.323 [Tanenbaum, 1999].

A sinalização das chamadas no padrão H.323 se diferencia de acordo com a integração com telefonia tradicional. Quando a troca de informações entre os elementos participantes de uma ligação não utiliza telefones convencionais, o padrão H.323 faz uso do protocolo H.225 [ITU-T, 2006a]. Porém, se a telefonia tradicional estiver compreendida na comunicação, o protocolo Q.931 [ITU-T, 1998b] é empregado.

Outro protocolo de suma importância é o H.245[ITU-T, 2008], o qual é necessário para negociação do canal de voz utilizado na passagem de dados de tempo real. Tais informações são transferidas através do *Real Time Transport Protocol* (RTP), descrito na seção 2.3.

Para ilustrar o funcionamento dos tipos de sinalização, a figura 2.3 exibe um modelo de aplicação VoIP utilizando H.323 integrado com a rede de telefonia tradicional. A sinalização de uma chamada entre um terminal localizado na rede local e um terminal localizado na Internet é realizada através do protocolo H.225, enquanto que, uma chamada feita entre um terminal da rede local e um telefone localizado na rede de telefônica é feita com o protocolo Q.931.

O padrão H.323 tem sua devida importância por definir o modelo básico de chamada VoIP. Todavia, devido ao seu crescimento desordenado, tornou-se um emaranhado de protocolos, propiciando o desenvolvimento de soluções estruturalmente mais simples.

Diante deste contexto, surge o protocolo *Session Initiation Protocol* (SIP) [Rosenberg et al., 2002] apresentando vantagens quando a facilidades de implementação e configuração.

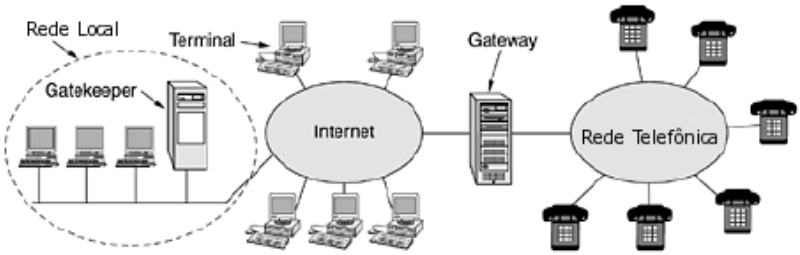

Figura 2.3: Modelo de aplicação VoIP utilizando H.323 integrado com rede de telefonia tradicional [Tanenbaum, 1999].

2.2.2 Session Initiation Protocol (SIP)

O protocolo SIP teve seu projeto iniciado em 1996 por Henning Schulzrinne (*Columbia University*) e Mark Handley (*University College London*), sendo especificado em conjunto ao IETF (*Internet Engineering Task Force*) que mantém a última versão (RFC3261 [Rosenberg et al., 2002]).

Localizado na camada de aplicação da pilha TCP/IP, o protocolo SIP foi modelado para ser independente da camada de transporte, verificando e retransmitindo mensagens tanto sobre TCP como UDP.

Utilizado basicamente para gerenciar sessões onde se trafega voz, vídeos e outros tipos de mídia, o SIP necessitou de outros protocolos para moldar toda sua arquitetura multimídia, destacando-se: *Real Time Transport Protocol* (RTP), realiza o transporte dos dados de tempo real; *Real Time Streaming Protocol* (RTSP) [Schulzrinne et al., 1998], aplicado em conjunto ao RTP para análise e controle da entrega de dados; *Media Gateway Control Protocol* (MGCP) [Cuervo et al., 2000], permite integração com telefonia pública e *Session Description Protocol* (SDP) [Handley and Jacobson, 1998], gerencia parâmetros para criação de sessões multimídia.

O protocolo SIP exibe uma estrutura de rede dividida em clientes e servidores onde, através da troca de mensagens definidas entre requisições e respostas, quatro tipos de entidades lógicas foram definidas: *User Agents*(UA), *Redirect Servers*, *Registrars* e *Proxy Servers*.

Em uma chamada estabelecida pelo protocolo SIP, os *User Agents* (UA) representam os elementos envolvidos na comunicação. Estes subdividem-se em *User Agent clients*, os responsáveis pela geração de requisições e *User Agente Servers*, encarregados de responder aos clientes. Os *Proxy Servers* são roteadores da camada de aplicação e fazem a comunicação entre os *User Agents* de forma indireta, transladando requisições e respostas. Os *Redirect Servers* atuam apenas redirecionando pedidos a específicos *Proxy Servers*, informando aos clientes endereço da nova rota. Os *Registrars* gerenciam dados

de endereços referentes aos UA, para que *Proxies* possam localizar informações durante o estabelecimento de chamadas.

Para ilustrar o funcionamento entre as entidades lógicas descritas, a figura 2.4 exibe a troca de mensagens durante a realização de uma chamada.

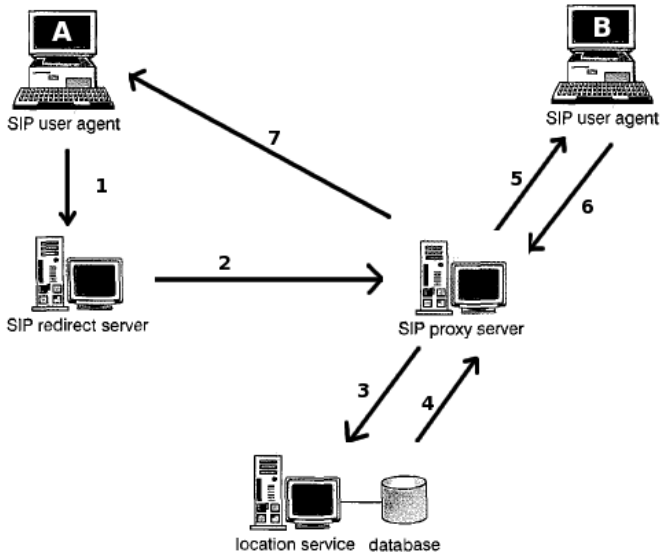

Figura 2.4: Funcionamento das entidades lógicas de rede no protocolo SIP durante a realização de uma chamada.

O processo se inicia quando o *user agent A* requisita a conexão ao *user agent B* através de um *redirect server* (1). O pedido é então redirecionado a um *sip proxy* (2), que verifica a existência do usuário *B* em seu domínio através de um *registrar* (3). O *registrar*, após encontrar informações do *user agent B*, envia ao *proxy* dados sobre a localização deste usuário (4). Por sua vez, O *user agent B* é avisado pelo *proxy* de que está sendo procurado pelo *user agent A* (5). Após o aviso, o *user agent B* retorna ao *proxy* uma mensagem de confirmação (6). Ao mesmo tempo, a cada mensagem que é enviada ao *user agent B* e confirmada, uma outra mensagem é enviada ao *user agent A*, alertando sobre a tentativa de contato ao usuário requisitado (7). Por fim, a comunicação entre os dois *user agents* é estabelecida.

As mensagens trocadas pelo protocolo SIP são baseadas em texto seguindo a codificação UTF-8(*8-bit Unicode Transformation Format*) e sintaxe similar a do *HyperText Transfer Protocol*(HTTP). Conforme exibe a figura 2.5, uma mensagem genérica trocada entre entidades é moldada por uma linha inicial, um ou mais campos de cabeçalho, uma linha vazia indicando o final do cabeçalho (CRLF) e corpo da mensagem.

A linha inicial de cada mensagem a classifica em método ou resposta. Os métodos são padrões de pedidos aplicados na determinação de funcionalidades. Seis métodos são definidos pelo protocolo SIP, porém outros são facilmente adicionados. As respostas são

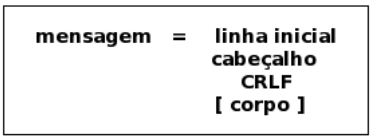

Figura 2.5: Estrutura das mensagens utilizadas pelo SIP [Rosenberg et al., 2002].

indicadores de resultados às requisições efetuadas e baseiam-se em códigos numéricos inteiros de três dígitos, sendo que o primeiro dígito do código define a classe de respostas e os últimos dois a categorização da regra. As tabelas 2.2 e 2.1 exibem, respectivamente, os métodos e as classes de códigos de resposta definidas pelo protocolo.

Tabela 2.1: Métodos de mensagens utilizadas pelo protocolo SIP

Método	**Descrição**
INVITE	convida um usuário para participar de uma chamada
ACK	utilizado para confirmar o convite de chamada
OPTIONS	solicita informação sobre capacidade do servidor
BYE	termina uma conexão entre usuários ou desliga uma chamada
CANCEL	termina uma requisição a busca por um usuário
REGISTER	registra um usuário

Tabela 2.2: Padrão numérico de respostas utilizadas pelo protocolo SIP

Código	**Descrição**	**Exemplo**
1XX	Faixa de códigos para informações	100-*Trying* 180-*Ringing*
2XX	Faixa de códigos de sucesso	200-*OK*
3XX	Faixa de códigos para redirecionamento	302-*Moved Temporarily*
4XX	Faixa de códigos para erro de clientes	404-*Not Found*
5XX	Faixa de códigos para erro de servidores	501-*Not Implemented*
6XX	Faixa de códigos para falhas globais	603-*Decline*

A primeira linha das mensagens emprega também o URI (*Uniform Resource Indicators*), que são identificadores de localização similares a endereços de e-mail. Contudo, os URIs também têm capacidade de noticiar endereços de rede, porta dentre outros.

Os cabeçalhos das mensagens são variáveis de acordo com cada aplicação do protocolo. Sintaticamente seguem o formato de mapa chave-valor, sendo que diversos valores do mesmo campo ficam separados por vírgulas. Por sua vez, o corpo das mensagens muda de acordo com a aplicação, ou seja, não possui um padrão sintático definido.

O protocolo SIP consegue promover suas funcionalidades básicas através de modelos de troca de mensagens com o uso dos métodos de requisição e resposta. Estas são descritas nas seções 2.2.2.2 a 2.2.2.5. Para um melhor entendimento das funcionalidades, a seção 2.2.2.1 apresenta o conceito dos diálogos existentes na comunicação SIP.

2.2.2.1 Diálogos

Os diálogos são responsáveis para haja reconhecimento entre *User Agents* (UAs), quando distintos canais de comunicação são estabelecidos. Localizados pela composição dos campos de identificador do canal, *URI* de origem e *URI* de destino, os diálogos representam o relacionamento ponto a ponto entre UAs. Iniciados a partir de uma resposta de sucesso na tentativa de estabelecimento de chamadas, os diálodos se mantêm ativos até que a sessão seja encerrada.

Durante o período de atividade, uma classificação de estados baseada na união do identificador do diálogo, número de seqüência local e remoto das mensagens é definida pelo próprio protocolo, fazendo com que os *UA*s consigam distinguir diferentes destinatários na comunicação sem a necessidade de implementações alternativas para tal procedimento.

2.2.2.2 Registro

O registro é necessário para que UAs divulguem suas informações de endereço aos *Registrars*. Para que uma sessão seja estabelecida, é essencial que o destinatário seja encontrado e receba o convite proposto pelo iniciador da chamada. Este processo de localização é frequentemente realizado por *Proxy Servers*, os quais recebem pedidos e determinam o roteamento lógico das mensagens através da interpretação dos dados recebidos pelos iniciadores, em conjunto ao conhecimento da posição exata de cada *User Agent* pertencente ao seu contexto. O armazenamento e gerenciamento destes endereços são realizados pelos *Registrars*, que mantém registros no formato de *URI*s.

As informações armazenadas devem ser frequentemente atualizadas evitando problemas de sincronização de endereços. O protocolo SIP, então, implementa a técnica de registro através do envio de requisições do tipo REGISTER em intervalos configuráveis por parte dos *User Agents*, desde o momento da ativação dos mesmos. Estas mensagens são interpretadas pelos *Registrar*s que além de armazenar as informações de localização respondem com pedidos de autenticação, iniciando um ciclo de funcionalidade de registro.

Os *Registrar*s, ao receber pedidos de registro, verificam primeiramente se o URI contido nas mensagens fazem parte de seu domínio. Caso haja falha nesta etapa, o servidor responderá ao requisitante uma mensagem do tipo 404 (*Not Found*). Na seqüência, os campos do cabeçalho das mensagens são analisados quanto aos elementos que representam valores e aos tipos de autenticação. Caso a autenticação falhe uma mensagem do tipo 401 (*Unauthorized*) contendo o tipo de autenticação utilizada pelo servidor é enviada como resposta.

Se a autenticação for válida, ou seja, todos os passos descritos pelo tipo de autorização utilizado forem realizados com sucesso, uma mensagem de resposta do tipo 200 (*OK*) é enviada ao *UA* requisitador, informando a finalização do processo de registro.

Informações referentes ao tempo de registro também são gerenciados pelo *Registrar*. O período de expiração de cada *UA* pode ser renegociado em cada procedimento de registro. Para que um *UA* se mantenha disponível durante um período contínuo, necessita-se que o mesmo execute o procedimento descrito antes do tempo de expiração. A remoção de registros de forma explícita pode ser realizada através da execução do procedimento de registro com o campo que define o tempo de expiração de registro preenchido com zero.

Os tipos de autenticação utilizadas pelo protocolo SIP são baseadas em métodos do protocolo HTTP. A verificação de autenticidade além de ser realizada no período de registro, pode ser executada durante a troca de mensagens entre *Proxy*s ou *UA*s.

O método de autenticação básica do HTTP, no qual usuário e senha são concatenados e codificados em *base 64* antes de ser enviados ao destinatário, foi removido do protocolo SIP por questões de segurança tanto no armazenamento das senhas de forma explícita, quanto na facilidade na decifração dos dados quando capturados no meio de transmissão.

A forma mais simples de autenticação do protocolo SIP foi transferida para o método *Digest* [Franks et al., 1999], a qual armazena e transmite dados referentes a usuários e senhas na forma criptografada de MD5, dificultando a obtenção das informações.

A autenticação mais empregada na comunicação entre *Proxy*s é a S/MIME *((Secure / Multipurpose Internet Mail Extensions))*. Esta promove uma maior confiabilidade aos parâmetros das sessões, pois há uma troca prévia de certificados assimétricos antes do estabelecimento do canal de comunicação.

O protocolo SIP ainda provê uma maneira mais segura para troca de mensagens, classificado como SIPS (*SIP Secure*)[Rosenberg et al., 2002], este usa TSL (*Transport Socket Layer*) na cifragem da sinalização. Seu precedecessor, SSL (*Secure Sockets Layer*), foi inicialmente desenvolvido para segurança em transações *web* (HTTPS). Já o TLS tem seu método disposto em três etapas: verificação da chave pública, checagem de integridade e troca de chaves assimétricas. É importante observar que os dados de mídia não estão inclusos nos canais encriptografados estabelecidos na sinalização.

Para esclarecimento dos passos necessários para a realização do registro, a figura 2.6 apresenta um cliente tentando se registrar.

Inicialmente o *User Agent* 'a' envia uma mensagem do tipo REGISTER ao *Registrar*, informando os dados referentes ao tempo de seu registro (1). Ao receber a mensagem, o *Registrar* verifica as informações e retorna uma resposta do tipo 401, contendo informações referentes ao método de autenticação utilizado, não autorizando o usuário (2). Ao receber a mensagem de não autorizado que contêm especificações referentes a autenticação, o

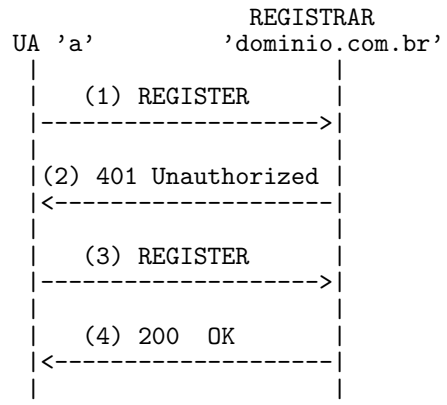

Figura 2.6: Passos para registro de um *user agent* SIP

usuário remonta uma nova mensagem do tipo REGISTER adicionando as informações necessárias para autenticação (3). Se todos os parâmetros estiverem corretos, o *Registrar* deve então responder com uma mensagem de OK, finalizando então o registro. Caso contrário, o usuário receberá outra mensagem de não autorizado.

2.2.2.3 Verificação de Capacidades

Antes do início da efetiva troca de dados de mídia em uma sessão SIP, os *UA*s coordenam configurações relacionadas a métodos de transmissão de dados de mídia tais como codificadores de voz e outras informações. Em uma chamada VoIP, por exemplo, informações sobre protocolo, codificadores de voz e endereços utilizados na transmissão do áudio devem ser combinadas pelos extremos.

Através do método OPTIONS, *UA*s podem requisitar informações a outros *UA*s e a *Proxy*s sem a necessidade do estabelecimento de diálogos. As mensagens do tipo OPTIONS não verificam dados relacionados a autenticação de usuários. Os receptores simplesmente retornam mensagens do tipo 200 (*OK*) com as informações desejadas adicionadas aos campos pré-definidos do cabeçalho e, se necessário, ao corpo das mensagens. Dados relacionados a mídia são armazenadas no corpo das mensagens através do protocolo *Session Description Protocolo* (SDP) [Handley and Jacobson, 1998], um padrão definido pela *IETF* na descrição de parâmetros de inicialização de mídias de tempo real.

Além de adquirir informações de elementos específicos, o método OPTIONS possibilita, através de uma única requisição, a obtenção de informações relacionadas a todos servidores *Proxy* intermediários entre o requisitador e seu destinatário. Este tipo de informação pode ser importante quando necessita-se que o áudio seja trespassado pelos roteadores da camada de aplicação.

2.2.2.4 Inicializando sessões

O estabelecimento de sessões é a principal funcionalidade do protocolo SIP. A criação de chamadas iniciam-se quando *UA*s enviam requisições a outros *UAs* através do método INVITE. Esses pedidos podem ser ou não roteados entre servidores *Proxy* até a chegada aos destinos requeridos.

Após o recebimento das requisições, os receptores poderão aceitar ou não os convites. Mensagens de resposta provisórias contendo informações referentes a estados dos convidados são enviadas periodicamente até a confirmação final da chamada. Padrões para estes tipos de mensagens são classificados pela faixa de respostas *1XX*, já a aceitação das ligações é representada pela classe *2XX*. Após o recebimento de um aceite, o criador da chamada deve confirmar a aprovação com *ACK*. Em seguida, são trocadas informações referentes a configuração da sessão multimídia que deverá se estabelecer.

Para exemplificar a troca de mensagens na criação de chamadas, a figura 2.7 apresenta o estabelecimento de uma conexão SIP entre os *UA*s *alice* e *bob* pertencentes aos domínios *atlanta.com* e *biloxi.com* respectivamente.

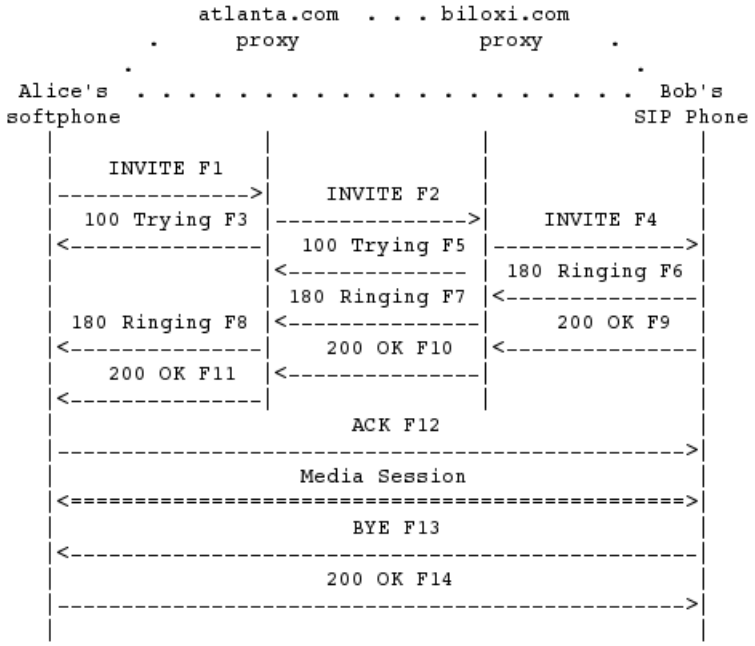

Figura 2.7: Passos para o estabelecimento de uma chamada SIP [Rosenberg et al., 2002].

O usuário *alice@atlanta.com* envia inicialmente uma mensagem ao seu *Proxy*, através do método INVITE, dizendo que necessita entrar em contato com o *UA bob@biloxi.com*. Como *bob* pertence ao domínio *biloxi.com*, o *Proxy atlanta.com* entra em contato com o outro *Proxy*, avisando a *alice* sobre sua tentativa de estabelecimento da chamada. O *Proxy biloxi.com*, ao receber a requisição, entra em contato com o *UA bob*, que responde

com dados relativos ao seu estado. Estas informações são então repassadas ao *Proxy atlanta.com*, que por sua vez avisa *alice*. Esse procedimento se repete até que o *UA bob* decida aceitar ou não o convite da chamada. Como neste exemplo a tentativa de estabelecimento é aprovada, o *UA bob* envia uma mensagem de *OK* ao seu *Proxy*, que repassa informação ao *Proxy atlanta.com* que envia a *UA alice*. Após o recebimento do aceite, *alice* envia uma mensagem ACK, confirmando a criação da chamada. Na seqüência, se inicia a troca de dados de voz, que permanece ativa até a finalização da sessão por um dos *UA*.

2.2.2.5 Modificando uma sessão existente

Uma sessão ativa pode ser modificada em tempo de execução. Essas mudanças referem-se à possíveis transferências de chamadas, adição e remoção de *UA*s e outros interesses. Para isso, o protocolo SIP precisa gerenciar alterações de endereços, modificações nos canais de dados de mídia e realizar outras complicações.

A modificação de uma sessão existente ocorre através do envio de uma nova mensagem de requisição (INVITE) preenchida com os mesmos dados de identificação do diálogo estabelecido na sessão. Qualquer *UA* envolvido no contexto pode pedir pelas modificações da sessão. Classificada como re-INVITE, esse tipo de requisição pode ser programada pelos *UA*s para reconfigurações automáticas de canais de mídia, quando identificadas falhas no envio ou recebimento dos dados. Entretanto, a geração de mensagens desse tipo é classificada como perigosa, por problemas de aumento de tráfego quando os *UA* não conseguem estabelecer novo curso para a transferência dos dados.

Para que as novas configurações requisitadas por re-INVITEs entrem em vigor, são necessárias respostas do tipo 2XX na confirmação da atualização. Caso contrário, o diálogo estabelecido será finalizado.

2.2.2.6 Finalizando uma sessão

O estado de uma sessão está intimamente ligado ao diálogo a ela associado. Desta maneira, o término de uma sessão implica na finalização do(s) diálogo(s) estabelecido(s) entre os *UA*s participantes. Todavia, diálogos podem ser terminados antes mesmo do início de sessões, pela negação de convites por *UA*s.

A finalização explícita de uma sessão deve ser realizada através do método BYE por qualquer *UA*, que após envio da requisição, espera respostas do tipo 481 (Chamada/-Transação não existe) ou 408 (requisição expirou). Caso não receba qualquer resposta, o requisitador finaliza todos os diálogos relacionados ao destinatário.

O método BYE não deve ser utilizado durante a tentativa de estabelecer uma sessão. Neste caso, os *UA*s devem recorrer à requisições do tipo CANCEL para finalizar o diálogo ativo.

A noção de tom de ocupado em ligações não esta bem definida na documentação SIP, sendo interpretado de diferentes maneiras por aplicações que utilizam o protocolo. Isto é, diversos procedimentos podem ser tomados para modos distintos de destruição de diálogos, classificados como tom de ocupado na telefonia tradicional.

2.3 Protocolos para Transporte de Mídia

Os protocolos de mídia fazem o transporte de dados de voz quando a chamada já está estabelecida. Em aplicações VoIP, estes dados são classificados como mídias contínuas, ou seja, são dados dependentes do tempo. Dessa forma, os protocolos responsáveis pelo transporte desses tipos de dados devem se preocupar com a reordenação de pacotes e com a perda de tempo na transmissão dos dados.

2.3.1 Real Time Transport Protocol (RTP)

O protocolo *Real Time Transport Protocol* (RTP), descrito na RFC1889 [Group et al., 1996] e atualmente utilizando a RFC3550 [Schulzrinne et al., 2003], surgiu pela necessidade de criação de um padrão único a diversas aplicações que realizavam troca de dados de tempo real.

O nome *Real Time Transport Protocol* indica um protocolo da camada de transporte, ou seja, que é armazenado diretamente em pacotes IP. Entretanto, na prática, o encapsulamento no RTP ocorre através protocolo UDP, com configurações de portas negociadas antes do início da transmissão dos dados. O RTP pertence a camada de aplicação no modelo TCP/IP, trabalhando em conjunto com aplicativos que utilizam transferência de dados, geralmente na forma de biblioteca. A figura 2.8 exibe a localização do protocolo RTP dentro do modelo TCP/IP.

O RTP não garante a entrega de dados, pois a técnica de retransmissão causa atrasos subseqüentes a pacotes reenviados. Todavia, os datagramas gerados pelo protocolo RTP são preenchidos com valores de tempo de criação de cada pacote e número de seqüência. Estes dados são empregados respectivamente para reordenação de pacotes e controle de pacotes perdidos.

O cabeçalho do protocolo RTP, exibido na figura 2.9, contém 32-*bits*, sendo os campos mais importantes o tempo relativo de criação do pacote, o número de seqüência e o código

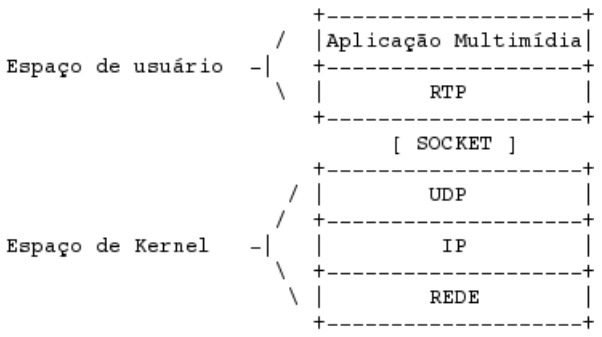

Figura 2.8: Posição do RTP no modelo TCP/IP [Tanenbaum, 1999].

identificador de sincronização. Esse último não descrito anteriormente, é utilizado para a sincronização de múltiplos canais de dados transmitidos em um único canal, como por exemplo um som *stereo* ou transmissão de áudio e vídeo.

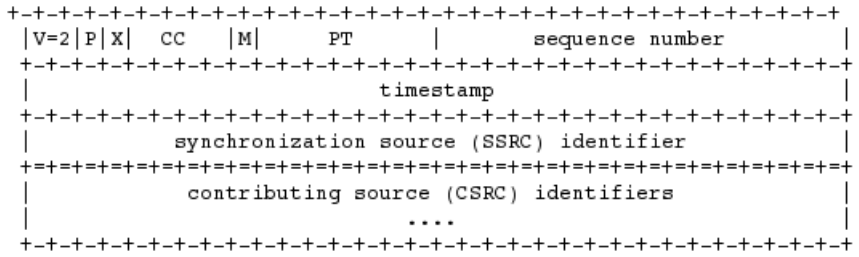

Figura 2.9: Cabeçalho do protocolo RTP [Group et al., 1996, Schulzrinne et al., 2003].

Vinculado ao protocolo RTP está o RTCP (*Real Time Transport Control Protocol*), que consegue obter diversas informações sobre a sessão RTP estabelecida. Através de troca de mensagens, o monitoramento do uso da rede, controle de fluxo e verificação da distribuição de dados em transmissões multi-destinatárias são realizados pelo RTCP. Aplicado também sobre o UDP e não tendo uma porta fixa determinada na definição do protocolo, o RTCP utiliza um número acima da porta estabelecida na sessão RTP. O RTCP não é fundamental para o funcionamento do RTP, mas é de suma importância para monitoramento a qualidade do meio de transmissão dos dados.

O protocolo RTP apresenta dificuldades ao trabalhar em conjunto a protocolos VoIP quando clientes tentam se comunicar através de alguns tipos de NAT (*Network Address Translation*). O NAT, forma com que endereços de rede não-roteáveis conseguem se comunicar com endereços de rede válidos, foi desenvolvido devido ao número limitado de endereços IP previsto no IPV4 [Tanenbaum, 1999]. Todavia, apenas as classes de endereços em redes locais definidas pelas regras da RFC1918[Rekhter et al., 1996] e IANA (Internet Assigned Numbers Authority) são permitidas. As citadas classes de endereços são exibidas na tabela 2.3.

Tabela 2.3: Classes de endereços não-roteáveis segundo a RFC1918.

Classe de Endereçamento	Faixa de Endereços
A	10.0.0.0 - 10.255.255.255
B	172.16.0.0 - 172.31.255.255
C	192.168.0.0 - 192.168.255.255

A figura 2.10 ilustra um exemplo de tradução de endereços. O indivíduo endereçado pelo IP local de uma rede classe C, se comunica com rede pública através de NAT. Verifica-se que o endereço de origem 192.168.0.4 torna-se acessível a endereços externos pelo IP 200.210.43.1.

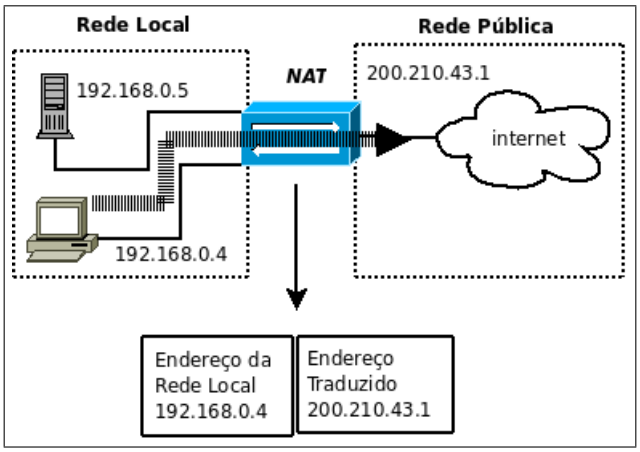

Figura 2.10: Exemplo de tradução feita pelo NAT na saida de dados de uma rede local.

O protocolo STUN (*Simple Traversal of User Datagram Protocol*) [Rosenberg et al., 2003] propõe, por meio de características de implementações, quatro classificações de NAT:

- *Full Cone*: todas as requisições de um mesmo endereço IP e porta internos são mapeados a um determinado endereço IP e porta externos. Qualquer comunicação interna/externa é realizada através do endereço externo mas com portas definidas pelo interno.

- *Restricted Cone*: todas as requisições de um mesmo endereço IP interno são mapeados pelo endereço IP externo. Entretanto, diferentemente do *Full Cone*, um elemento externo de determinado endereço IP apenas pode enviar um pacote para o indivíduo interno, quando este último iniciar a comunicação previamente.

- *Port Restricted Cone*: funciona da mesma maneira que o *Restricted Cone*, diferenciando-se pela restrição ao número de portas disponíveis.

- *Symmetric*: todas as requisições de um endereço IP interno são mapeadas na mesma porta do endereço externo. Além disso, apenas indivíduos externos que receberam

uma comunicação prévia podem responder aos elementos internos. Se um elemento interno tentar utilizar uma porta em uso, um diferente mapeamento de porta é realizado.

Essa classificação está saindo de uso, pois muitas implementações de NAT oscilam entre os vários tipos. Por exemplo, em um *symmetric* NAT, o padrão de portas internas/externas pode ser alterado na existência de comunicações simultâneas. Se dois elementos internos acessam ao mesmo endereço externo usando o mesmo número de porta, o segundo indivíduo terá sua porta externa atribuída de forma aleatória, tornando a classificação do NAT *restricted cone*.

O problema no uso do RTP em aplicações VoIP acontece nos tipos de NAT *symmetric* e *restricted*. Como o protocolo RTP escolhe dinamicamente uma porta para a transferência de dados, quando um elemento da rede interna inicia uma troca de mensagens RTP com um dispositivo localizado em um endereço externo, seus dados de endereço e porta são armazenados na tabela NAT. Entretanto, quando um indivíduo externo tenta iniciar a troca de mensagens RTP com um endereço da rede interna, se a porta utilizada na transmissão não estiver armazenada na tabela, não é possível relacionar o endereço interno com a porta de chegada de dados.

A figura 2.11 ilustra o problema do uso de RTP através de NAT em uma chamada SIP. Neste exemplo não se exibe todas as mensagens SIP trocadas para o estabelecimento da chamada. O indivíduo classificado como *USER2* pertence a uma rede local e estabelece sinalização com *USER1* localizado em um endereço externo. Para que o *USER2* seja visível ao *USER1*, uma tradução de endereços é necessária. A tabela NAT armazena essas informações a partir do momento que o *USER2* inicia sinalização com o *USER1*, mapeando os dados chegados pela porta 5089 no endereço 1.2.3.4 ao *host* 10.0.0.2 na porta 5060. O desencontro na troca de dados de mídia acontece na pactuação de portas. A porta escolhida pelo *USER2* para o RTP é mapeada pelo NAT em uma porta diferente da combinada anteriormente, mesmo assim os dados do *USER2* chegam para o *USER1*. Contudo quando o *USER1* envia os dados para o endereço 1.2.3.4 na porta 10001 o NAT não consegue identificar para qual *host* interno os dados devem ser traduzidos.

Para a solução desses problemas, algumas técnicas foram desenvolvidas, inclusive o surgimento do protocolo VoIP IAX.

2.4 Inter-Asterisk Exchange Protocol (IAX)

O protocolo *Inter Asterisk eXchange* (IAX)[Spencer and Miller, 2006] é um protocolo da camada de aplicação da pilha TCP/IP. Atualmente ocupando a terceira posição

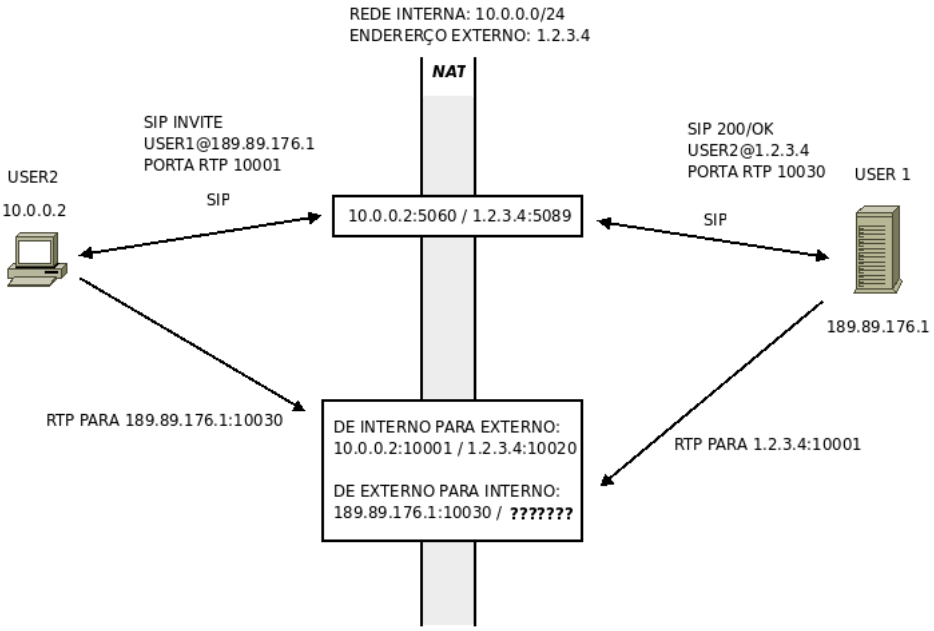

Figura 2.11: Problema de RTP em NAT.

na lista de protocolos VoIP mais utilizados, o protocolo IAX está sendo procurado e estudado na área acadêmica e por empresas de equipamentos VoIP nos últimos anos [Abbasi and Prasad, 2005].

Enquanto o H.323 e o SIP necessitam de estabelecimento de um novo canal lógico através do protocolo RTP para a transferência de dados de mídia, o IAX tem a capacidade de transferir os dados através do mesmo canal estabelecido na sinalização da chamada, evitando problemas de NAT discutidos na seção 2.3.

A criação do protocolo IAX teve inicio com Mark Spencer que precisava de um sistema de *PBX* [1], mas não tinha condições para adquirir o equipamento. Acostumado a participar de projetos *open source*, resolveu se aventurar no desenvolvimento de seu próprio sistema, recebendo e compartilhando idéias com outros entusiastas. Sua criação, batizada de *Asterisk* [ast, 2006], em pouco tempo continha recursos encontrados em aparelhos caros de *PBX*.

Percebendo a carência de hardware no mercado para aplicação efetiva do sistema *Asterisk*, Spencer criou a *Digium*, empresa responsável pela fabricação de placas e aparelhos. Hoje a *Digium* é uma provedora de soluções em telefonia e se dedica a vender hardware, custeando o desenvolvimento do projeto *open source Asterisk*.

Com o propósito de definir um modelo de comunicação entre vários servidores *Asterisk*, a criação do protocolo IAX se baseia em experiências relacionadas a outros protocolos VoIP, mostrando conceitos diferentes, tais como a multiplexação de dados de mídia, sinalização através do mesmo canal, provendo transparência em NAT e simplificando con-

[1] *Private Branch eXchange*

figurações de *firewall*. O protocolo IAX atualmente está na versão 2 e utiliza a porta 4569 sobre a camada de transporte UDP.

Dividindo-se basicamente em clientes e servidores, o protocolo IAX não contém em sua documentação um padrão lógico bem definido sobre servidores como o protocolo SIP, sub-entendendo-se que todos os padrões são agrupados em um único servidor.

A identificação de usuários no protocolo IAX, assim como o protocolo SIP, também é feita através de URIs. O padrão adotado se diferencia pelo prefixo "*iax*" utilizado na comunicação.

As mensagens trocadas pelo protocolo IAX são feitas de forma binária, o que o torna vantajoso quanto ao uso mais eficiente da largura de banda. Também evita-se o emprego de analisadores sintáticos utilizados nas mensagens textuais, que podem sofrer ataques de *buffer overflow* dependendo da implementação [Zhang, 2002]. Entretanto, destaca-se que mensagens binárias devem ter campos bem definidos para armazenamento dos dados.

De forma a evitar problemas relacionados a ordenação de pacotes, números de seqüência e tempos relativos das mensagens são utilizados pelo IAX. Em vista ao baixo consumo de banda ao multiplexar o transporte de dados de mídia e sinalização das chamadas, três formatos de mensagens classificados como *frames* são definidos pelo protocolo.

Os chamados *full frames* são responsáveis pela transmissão de dados de sinalização. Podem ser utilizados também para o transporte de mídia, apesar de causar maior consumo de banda em relação a outras alternativas por definir parâmetros não utilizados a essa aplicação. A estrutura empregada pelos *full-frames* é exposta na figura 2.12, e seus campos descritos na tabela 2.4.

Tabela 2.4: Campos utilizados pelos *full frames*.

`bit F`	indica se o *frame* é ou não um *full frame*
`bit R`	indica retransmissão
`source Call Number`	número do transmissor
`destination Call Number`	número do receptor
`timestamp`	tempo relativo de criação do pacote de acordo com a primeira transmissão (32 *bits*)
`OSeqno`	armazena o número de seqüência de mensagens enviadas
`ISeqno`	armazena o número de seqüência de mensagens recebidas
`Frametype`	identifica o tipo de mensagem
`bit C`	identifica se os dados estão codificados
`Data`	armazena estruturas definidas pelos tipos de mensagens

A classe de *frames* que possui um melhor desempenho no transporte de dados de mídia é a dos *mini frames*, pois tem um cabeçalho específico a este fim. Composta pelos campos

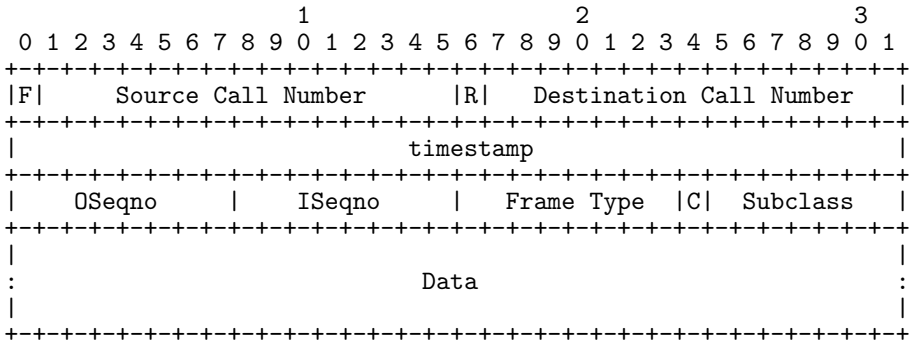

Figura 2.12: Estrutura dos *full frames* utilizados pelo IAX [Spencer and Miller, 2006].

exibidos na figura 2.13 e descritos na tabela 2.5, os *mini-frames* adotam o *timestamp* com apenas 16 bits de tamanho ao invés dos 32 bits definidos pelos *full-frames*, optimizando o uso de banda.

Tabela 2.5: Campos utilizados pelos *mini frames*

`bit F`	sempre preenchido com valor zero
`Source Call Number`	número do transmissor
`timestamp`	tempo relativo de criação do pacote de acordo com a primeira transmissão (16 bits)

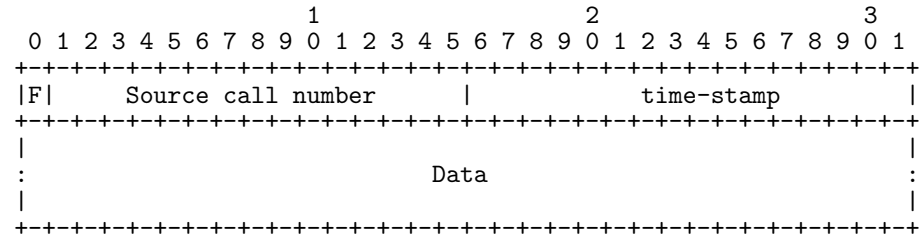

Figura 2.13: Estrutura dos *mini frames* utilizados pelo IAX [Spencer and Miller, 2006].

A terceira classe de *frames* é a dos *meta frames*, os quais têm dois propósitos. O primeiro ocorre em transferências de vídeo no qual é utilizado um único cabeçalho durante a chamada, minimizando assim o consumo da banda. Os *meta frames* possuem uma estrutura bastante parecida com os *mini frames*, diferindo-se apenas no emprego dos campos *Meta Indicator*, onde são armazenadas informações sobre dados, e *bit V* que indica se as informações são de vídeo ou não. A figura 2.14 apresenta a estrutura utilizada pelos *meta frames*.

Os dados transportados pelos *full frames* podem ser de mídia, como descrito anteriormente. Porém, fundamentalmente, estruturas que contém informações necessárias para realização das funcionalidades do protocolo também são empregadas, as quais são descritas na seção 2.4.1.

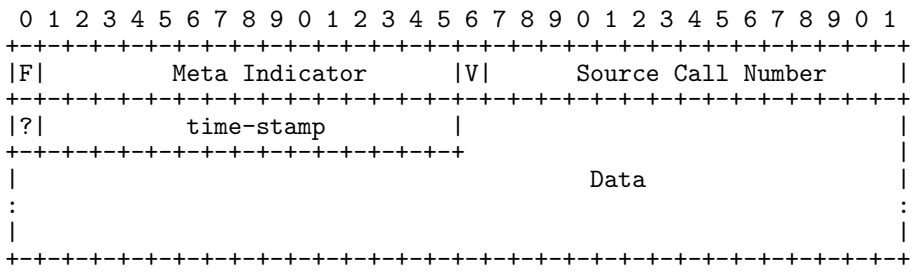

Figura 2.14: Estrutura dos *meta frames* utilizados pelo IAX [Spencer and Miller, 2006].

2.4.1 Descrição das funcionalidades do protocolo IAX

O protocolo protocolo IAX possui 11 áreas funcionais subdivididas entre obrigatórias e opcionais. As obrigatórias descrevem funções básicas necessárias para a implementações do protocolo, enquanto que as opcionais definem operações adicionadas durante a evolução do protocolo [Spencer and Miller, 2006].

Semelhante ao protocolo SIP, o IAX define uma série de mensagens utilizadas nas requisições e respostas. Entretanto, o IAX não possui um padrão de mensagens de resposta, estas são definidas por cada funcionalidade do protocolo.

2.4.1.1 Registro

Para que um agente IAX possa ser convidado à uma sessão, o realizador da chamada necessita saber a localização exata do receptor. Esse endereço pode ser configurado manualmente ou obtido pelo uso da funcionalidade de registro, que armazena a consulta de endereços de rede de clientes e servidores IAX.

Primeiramente clientes IAX devem se identificar a servidores para que suas informações sejam guardadas e consultadas por outros clientes. Durante esse processo, o servidor IAX requisitado pode exigir algum tipo de identificação ao cliente que tenta se registrar. Para tanto, três métodos de autenticação são definidos pelo protocolo IAX, sendo que dois deles utilizam criptografia.

No método sem criptografia, basicamente a senha requisitada é enviada como resposta sem preocupação com o meio de transmissão ou privacidade. No primeiro método que recorre a criptografia, antes de ser transmitida, a senha do cliente é agrupada a um número gerado pelo servidor e depois criptografada com o algoritmo MD5 (*Message-Digest Authentication*) [Rivest, 1992]. Neste caso, destaca-se o acesso a senha sem criptografia tanto pelo lado do cliente quanto pelo servidor já que a função matemática MD5 é de via única, ou seja, não é possível obter o valor original a partir do valor final.

O segundo método de autenticação com criptografia definido pelo protocolo IAX, classificado como mais seguro, apóia-se no uso do algoritmo RSA (*Rivest, Shamir and Adleman's*) [Kaliski and Staddon, 1998], que trabalha com senhas unidirecionais através de pares de chaves pública-privada. Neste método a chave pública pode ser conhecida por todos, mas a chave privada é mantida em sigilo. Todas as mensagens cifradas pela chave pública só podem ser decifradas pela respectiva chave privada, o que garante a privacidade dos dados no meio de transmissão.

Como citado anteriormente, para cada funcionalidade do protocolo IAX, um conjunto de mensagens são utilizadas. Para prover o registro, mensagens de requerimento e respostas utilizadas são exibidas na tabela 2.6.

Tabela 2.6: Mensagens utilizadas na funcionalidade de registro do protocolo IAX

Classificação	**Tipo**	**Descrição**
Requisição	REGREQ	pedido de registro
	REGREL	pedido de atualização de registro
Resposta	REGAUTH	resposta ao pedido de registro ou atualização de registro informando o método de autenticação correto
	REGACK	resposta do tipo *acknowledgment* a um pedido de registro
	REGREJ	resposta dada a um pedido de registro rejeitado

O entendimento das etapas seguidas por um cliente IAX durante uma tentativa genérica de registro, é ilustrado através da análise do diagrama de estados exibido pela figura 2.15.

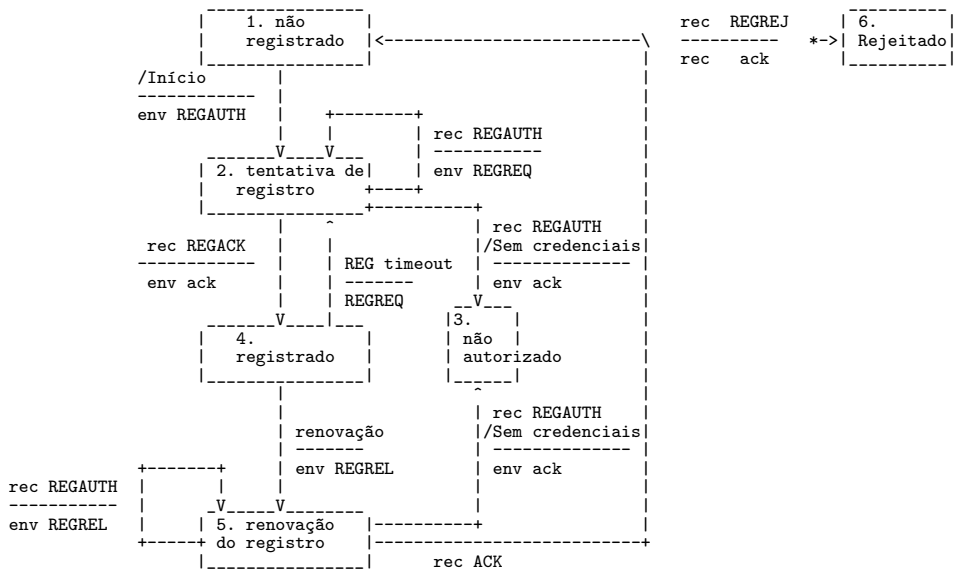

Figura 2.15: Diagrama de estados de registro no protocolo IAX [Spencer and Miller, 2006].

Inicialmente, localizado no estado ”não registrado”(1), o cliente envia uma mensagem do tipo REGREQ, com informações do número ou nome por ele utilizado e do tempo de expiração de seu registro. Após esta ação, o requerente passa para o estado ”tentativa de registro”(2). Se o servidor requerer autenticação, responderá com uma mensagem do tipo REGAUTH, contendo os tipos de autenticação por ele suportadas. O cliente então reenvia a mensagem REGREQ, contendo as informações necessárias de identificação. Se os dados estiverem corretos, o servidor enviará uma mensagem de REGACK, contendo o tempo de expiração até seu próximo registro, transportando o usuário ao estado ”registrado”(4). Caso contrário, o servidor responderá ao usuário, uma mensagem do tipo REGAUTH requisitando autenticação novamente. Após um número configurável de tentativas falhas, o usuário entrará no estado ”não autorizado”(3). A qualquer momento durante o registro, o servidor poderá enviar uma mensagem do tipo REGREJ, transferindo o usuário para o estado de ”rejeitado”(6).

Um cliente registrado pelo servidor deve atualizar suas informações antes da expiração do tempo combinado inicialmente. Caso isso aconteça, o cliente retrocede ao estado de ”não registrado”(1). Antes disso, o usuário translada ao estado de ”renovação de registro”(5), enviando uma mensagem do tipo REGREL. Se contestado pelo servidor com um REGAUTH, o cliente deverá responder com outro REGREL, contendo a autenticação necessária configurada anteriormente. Se a autenticação for satisfeita o servidor retornará uma mensagem REGACK reiniciando o ciclo de registro, caso contrário o usuário será classificado como ”não autorizado”(3).

2.4.1.2 Criação de chamadas

A funcionalidade de criação e estabelecimento de chamadas é essencial a qualquer protocolo VoIP. O gerenciamento das chamadas no protocolo IAX tem mensagens bem definidas de requisição e resposta, estas exibidas na tabela 2.7.

Tabela 2.7: Mensagens utilizadas na funcionalidade de criação de chamadas do protocolo IAX

Classificação	**Tipo**	**Descrição**
Requisição	`NEW`	criação de novas chamadas
	`HANGUP`	finalização de chamadas
	`AUTHREQ`	requisição de autenticação por parte do receptor a uma tentativa de criação de chamadas
Resposta	`ACCEPT`	resposta de aceite ao convite de chamadas
	`REGACK`	resposta de rejeição ao convite de chamadas
	`AUTHREP`	resposta a mensagem de requisição à autenticação

O processo de criação de chamadas utilizando o protocolo IAX pode ser analisado através do diagrama de estados exibido na figura 2.16.

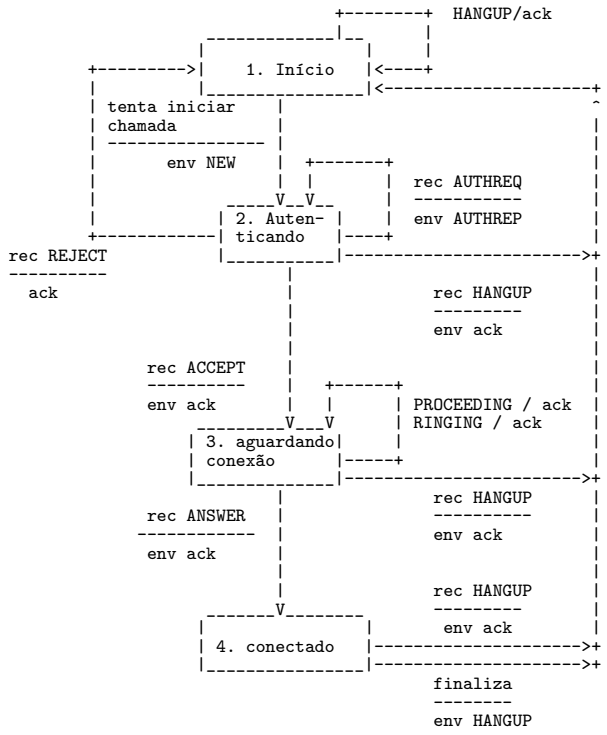

Figura 2.16: Diagrama de estados de chamadas no protocolo IAX [Spencer and Miller, 2006].

Um cliente IAX partindo do estado "início"(1), tenta iniciar uma ligação enviando uma mensagem NEW ao servidor ou convidando outro cliente para chamada. Nesse tipo de mensagem são armazenadas informações referentes aos tipo de codificadores de voz suportados pelo cliente criador e número de destino da chamada.

O servidor, após receber a mensagem de criação de chamada, pode solicitar algum tipo de autenticação através da mensagem AUTHREQ descrevendo os tipos de autenticação por ele suportadas. Caso isso aconteça, o usuário se deslocará ao estado de "autenticando"(2), respondendo à requisição de autenticação com uma mensagem do tipo AUTHREP, com as informações de segurança desejadas. As autenticações utilizadas na criação de chamadas são as mesmas descritas na funcionalidade de registro (sessão 2.4.1.1). Se todas as informações estiverem corretas, o criador da chamada receberá uma mensagem de ACCEPT, caso contrário a mensagem de HANGUP será enviada pelo servidor, finalizando a ligação. Se não houver autenticação, o cliente receptor ou servidor envia diretamente uma mensagem de ACCEPT. Independentemente do tipo de mensagem de autenticação recebida, um ACK de confirmação deve ser respondido pelo criador da ligação.

Quando autenticado, o criador da chamada passa então para o estado de ”aguardando conexão”(3). Na tentativa de localização do receptor, o servidor envia ao cliente que iniciou a sessão mensagens do tipo PROCEEDING. Se localizado, o receptor é avisado de que está recebendo uma chamada, sendo que mensagens de RINGING são enviadas pelo receptor informando estar ciente da ligação. Essas mensagens são interpretadas pelo iniciante da chamada como tons de chamando. Caso o receptor esteja ocupado, é retornado uma mensagem de HANGUP. Estas mensagens são sempre respondidas com uma confirmação ACK.

Se o receptor atender a ligação, uma mensagem de ANSWER deve ser enviada para finalizar a sinalização da chamada, iniciando a troca de dados de mídia. Dependendo da configuração do intermediário (servidor), a comunicação pode ocorrer de forma direta (sem interferência do servidor) ou indireta (passando pelo intermediário). A sessão 2.4.1.4 apresenta esta funcionalidade como uma opção para optimização do caminho dos dados.

Para finalizar uma chamada em operação, qualquer um dos clientes pode enviar uma mensagem do tipo HANGUP esperando um ACK por parte do receptor, pondo fim ao canal estabelecido. A interrupção de chamadas é classificada como outro fato dentro do protocolo IAX sendo descrita na sessão 2.4.1.5.

A criação de uma chamada pode ser observada de uma forma simplificada entre agentes IAX através da figura 2.17.

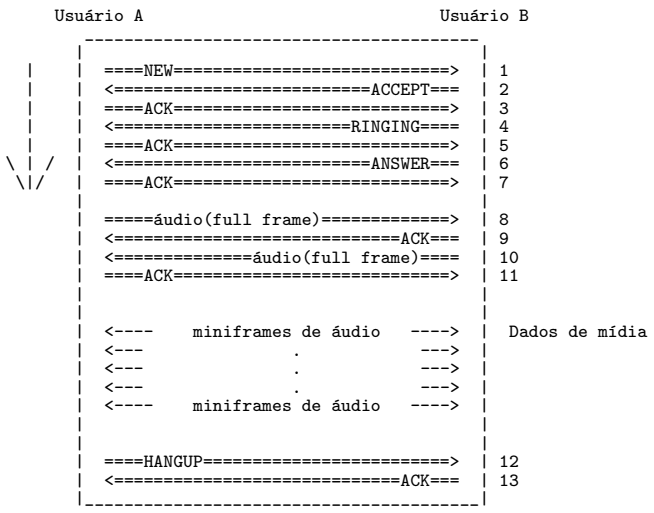

Figura 2.17: Exemplo do estabelecimento de uma chamada IAX [Spencer and Miller, 2006].

Inicialmente o agente A requisita uma nova conexão de chamada ao agente B(1). Na seqüência, o agente B aceita a conexão(2), o agente A então confirma o recebimento da mensagem(3). Em seguida, o agente B envia a informação ao usuário A de que está ciente

sobre a chamada(4), o agente A confirma, novamente, o recebimento da mensagem(5). A chamada é finalmente estabelecida quando o usuário B envia a mensagem confirmando o atendimento(6). Mais uma vez, o agente A confirma o recebimento da mensagem(7) e envia outra mensagem contendo informações sobre a codificação utilizada por ele na transmissão de dados de mídia(8). O agente B confirma a mensagem(9) compatibilizando o mesmo tipo de informação ao agente A(10) por uma nova mensagem a ser confirmada(11). Dados de voz são trocados até que um dos agentes finalize a chamada(12), que deve ser confirmada pelo correspondente(13).

Durante uma chamada, mudanças podem ser realizadas por qualquer elemento envolvido na ligação. Tais modificações serão descritas na sessão que segue.

2.4.1.3 Modificações chamadas estabelecidas

A funcionalidade de modificação de chamadas mantém o padrão funcional dos sistemas telefônicos tradicionais, como a transferência de chamadas e chamadas em espera. A operação consiste em substituir os sinais físicos representantes na telefonia tradicional por mensagens padronizadas no protocolo. A tabela 2.8 apresenta os tipos de mensagens utilizadas e suas devidas modificações no canal.

Tabela 2.8: Mensagens utilizadas na modificação de chamadas.

Tipo	Descrição
`FLASH`	mensagem que representa a tecla *flash* geralmente utilizada por adaptadores telefônicos para interrupção durante chamadas.
`HOLD / UNHOLD`	mensagens que representam a funcionalidade de chamada em espera. Enviadas ao receptor para que o mesmo ative ou desative a troca de dados de voz antes do estabelecimento efetivo das chamadas.
`QUELCH / UNQUELCH`	mesma funcionalidade do HOLD, entretanto, utilizado após o estabelecimento efetivo da chamada, ou seja, após o recebimento de uma mensagem de `ACCEPT`.
`TRANSFER`	mensagem que representa a transferência de chamadas. Enviada ao receptor informando que sua chamada será finalizada e na seqüência iniciada com outro agente IAX.

2.4.1.4 Optimização do caminho de dados

Considerada uma funcionalidade opcional, a optimização do caminho de dados é a exclusão do agente intermediário entre dois agentes IAX durante uma chamada. O servidor que sinalizou chamada entre clientes IAX pode se remover do caminho da comunicação entre os mesmos, caso não precise de algum monitoramento de progresso, conteúdo ou duração da chamada. Esta opção também é chamada de transferência supervisada, sendo uma maneira segura de não perder a ligação durante o processo de mudança. Para as partes envolvidas, o processo é basicamente uma operação de rede.

Tabela 2.9: Mensagens utilizadas na funcionalidade de transferência supervisionada do protocolo IAX

Classificação	Tipo	Descrição
Requisição	`TXREQ`	mensagem de início de processo de transferência
	`TXMEDIA`	mensagem que indica o caminho de mídia
Resposta	`TXCNT`	resposta dada a verificação de transferência
	`TXACC`	resposta de confirmação a mensagens `TXCNT`
	`TXREADY`	resposta para a confirmação da transferência
	`TXREL`	resposta para finalização com sucesso da transferência
	`TXREJ`	resposta para finalização com erro da transferência

A tabela 2.9 exibe as mensagens utilizadas para promover a optimização do caminho de dados, destacando-se que esse procedimento só pode ser empregado após o estabelecimento da chamada. A figura 2.18 ilustra o processo da transferência que se inicia quando o agente intermediário envia uma mensagem do tipo TXREQ a outros agentes, contendo informações de rede referentes ao canal estabelecido (1). Ao receber essa mensagem cada cliente responde entre si através do TXCNT, mensagens contendo informações referentes às contagens de pacotes, para que os mesmos possam entrar em sincronização (2, 7). Cada TXCNT é replicado com uma mensagem TXACC que, que é uma espécie de confirmação ao requerimento (3, 8). Ao sincronizar os dados de mídia, todos respondem com uma mensagem do tipo TXREADY, confirmando o novo caminho para os dados de mídia (5, 9). Por fim, um TXREL é enviado finalizando assim o processo (11). Se houver algum problema durante a transferência, uma mensagem do tipo TXREJ deve ser enviada, cancelamento da transferência e mantendo a sessão estabelecida.

Se o processo de transferência estiver completo, a sinalização continuará a passar pelo caminho inicial, ou seja, pelo agente intermediário. Isto ocorre para que o servidor possa obter informações referentes à chamada, porém os dados percorrem o novo caminho

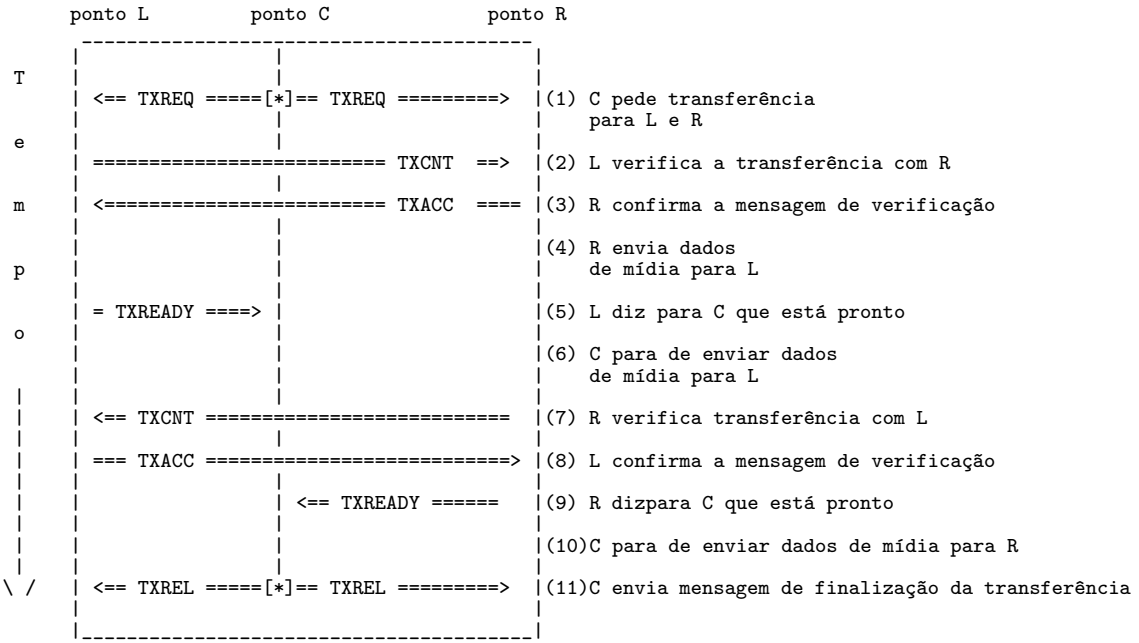

Figura 2.18: Processo de optimização do caminho das chamadas [Spencer and Miller, 2006].

estabelecido na transferência. A separação do caminho é feita através da mensagem TXMEDIA, que determina o caminho utilizado para a passagem dos dados de voz.

2.4.1.5 Finalização de chamadas

A interrupção de uma sessão é classificada separadamente da criação e estabelecimento de chamadas devido à variação existente neste processo. Além de aplicar alguns métodos para que ocorra a interrupção de uma sessão por vontade de um dos participantes, a finalização pode ocorrer por algum problema no meio de transmissão. A tabela 2.10 destaca os métodos e suas respectivas mensagens utilizadas em interrupções em sessões IAX:

Tabela 2.10: Métodos de interrupção de sessões IAX.

Tipo	**Descrição**
Desligar	método de finalização de uma sessão estabelecida. Utiliza-se a mensagem de HANGUP.
Rejeitar	método de finalização de uma sessão ainda não estabelecida. Utiliza-se a mensagem de REJECT (ver seção 2.4.1.2).
Estabelecimento de novo caminho de dados	método de finalização através de transferência supervisio nada de uma sessão estabelecida pelo procedimento descrito na sessão 2.4.1.4

2.4.1.6 Outras funcionalidades

Algumas funcionalidades suportadas pelo protocolo IAX, diversificam opções nas chamadas. Entretanto, estes procedimentos não são de suma importância para este projeto, ou seja, serão apresentados, mas não empregadas nos testes realizados.

- **Monitoramento de rede**: como o protocolo IAX é responsável por dados de mídia transferidos nas sessões, possui ferramentas para monitoramento das ligações estabelecidas. Os mecanismos para avaliação de chamadas tem mensagens de requisições e suas devidas respostas listadas na tabela 2.11. Informações relacionadas a atraso e *jitter* das ligações podem ser obtidas durante a realização das chamadas através da troca de mensagens do tipo PING/PONG. Para testes de conectividade são empregadas as mensagens POKE.

Tabela 2.11: Mensagens utilizadas para monitoramento de chamadas IAX

Classificação	Tipo	Descrição
Requisição	`PING`	teste de conectividade com outro usuário em uma chamada existente
	`POKE`	similar ao `PING`, testa a conectividade com outro usuário. Entretanto quando não durante uma chamada
	`LAGRQ`	requisita um pacote ao receptor para determinar atraso entre os dois pontos
Resposta	`PONG`	resposta dada às requisições `PING` ou `POKE`. O agente utiliza o *timestamp* do pacote recebido para calcular o *Round Trip Time* da conexão.
	`LAGRP`	envia uma resposta a mensagem `LAGRQ` contendo mesmo *timestamp* recebido no pacote

- **Planos de discagem** : esta funcionalidade é designada a dispositivos que não possuem planos de discagem. A inclusão de códigos de área e dígitos a um número na realização de uma chamada pode ser empregada tanto do lado do servidor quanto do cliente. A troca de informações é realizada por requisições com a mensagem do tipo DPREQ seguidos de respostas DPREP, que indicam os dígitos adicionados antes do início das chamadas. A tabela 2.12 indica os tipos de mensagens e descrições adotadas. Observa-se que este tipo de funcionalidade foi adicionada ao protocolo por motivos de roteamento utilizados pelos servidores *Asterisk*, não sendo interessante para esse projeto.

- ***Firmware Download***: essa funcionalidade providencia a obtenção de *firmwares* de dispositivos a partir de um servidor. Destaca-se a não segurança por parte do

Tabela 2.12: Mensagens utilizadas para troca de informações de planos de discagem no protocolo IAX

Classificação	Tipo	Descrição
Requisição	`DPREQ`	requisição para análise do número a ser chamado
Resposta	`DPREP`	resposta da análise com o plano de discagem atualizado de acordo com o número proposto.

protocolo IAX para essa opção. Este tipo de funcionalidade também foi adicionada ao protocolo por motivos técnicos aplicados a servidores *Asterisk*. As mensagens do processo são exibidas na tabela 2.13.

Tabela 2.13: Mensagens utilizadas para download de *firmware* no protocolo IAX

Classificação	Tipo	Descrição
Requisição	`FWDOWNL`	requisição de *download* de *firmware*
Resposta	`FWDATA`	resposta a mensagem `FWDOWNL` contendo dados referentes à requisição feita.

- **Configuração de dispositivos**: esta funcionalidade do protocolo IAX, também herdada dos servidores *Asterisk*, é utilizada por aparelhos que não mantém suas informações armazenadas. Basicamente a requisição é feita através da mensagem PROVISIONING e os dados são enviados após a confirmação da existência de informações corretas no requisito. A tabela 2.14 exibe as requisições utilizadas na operação.

Tabela 2.14: Mensagem de requisição de configuração de dispositivos no protocolo IAX

Classificação	Tipo	Descrição
Requisição	`PROVISIONING`	requisição de configuração de dispositivo

2.4.1.7 Mensagens globais e de mídia

Mensagens utilizadas por diversas funcionalidades são definidas como globais pelo protocolo IAX. Em conjunto a estas, estão as mensagens de mídia, que geralmente são armazenadas em *mini-frames* encapsulando dados de mídia e informações sobre codificação e formatação. As classes de mensagens, globais e de mídia, são apresentadas nas tabelas 2.15 e 2.16 respectivamente.

2.5 Comparação entre SIP e IAX

Tabela 2.15: Mensagens classificadas como globais pelo protocolo IAX.

Tipo	Descrição
`MWI`	mensagem usada para indicar a espera de uma ou mais mensagens
`ACK`	resposta padrão utilizada na confirmação de requisições
`INVAL`	resposta a mensagens recebidas inválidas
`VNAK`	resposta a mensagens recebidas fora de ordem. Utilizada particularmente quando um *mini-frame* é recebido antes de do primeiro *full-frame* de voz.
`UNSUPPORT`	resposta enviada a uma requisição indicando a falta de suporte por parte do receptor.

Tabela 2.16: Mensagens de mídia utilizadas em sessões IAX.

Tipo	Descrição
`DTMF`	mensagens com dados de cada DTMF (*Dual Tone Multiple Frequency*)
`Voice`	mensagens com dados de voz e informações sobre o CODEC utilizado.
`Video`	mensagens com dados de vídeo e informações sobre o formato utilizado.
`Text`	mensagens com dados de texto. Todas as mensagens de texto estão utilizando o formato UTF-8.
`HTML`	mensagens com dados em HTML (*HyperText Markup Language*).
`Comfort Noise`	parâmetros para configuração de níveis de ruídos gerados.

Spencer e Miller [Spencer and Miller, 2006] exibem uma lista de possíveis vantagens do protocolo IAX sobre o SIP. Pontos de comparação observados pelos autores, unidos a outros estudos, foram fatores essenciais na criação dos seguintes tópicos: tipos de mensagens utilizadas na sinalização das chamadas, eficiência de sinalização, transferência de dados de mídia, eficiência no uso de NAT(*Network Address Translation*), monitoramento de rede e documentação.

Os tipos de mensagens utilizadas no contexto deste projeto são focadas nas funcionalidades de registro e estabelecimento de chamadas, entretanto estas são conduzidas e formatadas de diferentes maneiras em cada protocolo. O protocolo SIP baseia-se em mensagens de texto com cabeçalhos padronizados, mas modificados de acordo com cada funcionalidade como descrito na seção 2.2.2 desta dissertação. O cabeçalho básico de uma mensagem SIP possui 4 *bytes*, porém no RFC3261 são listados mais de 60 tipos de cabeçalho que podem ser empregados na transmissão de informações. Por sua vez, o protocolo IAX emprega uma codificação binária em suas mensagens, fixando um tamanho

de 12 *bytes* de cabeçalho para todas mensagens na sinalização. O uso de mensagens binárias evitam tentativas de ataques do tipo *buffer overflow* aos analisadores sintáxicos de texto. Todavia, o uso de mensagens de texto contribui na facilidade de adição de novas funcionalidades, bastando ao desenvolvedor criar seu padrão e respectivo interpretador.

A sinalização das chamadas nos protocolos SIP e IAX são consideradas mais eficientes por serem simples, quando comparados ao H.323. O protocolo SIP realiza a troca de mensagens através dos protocolos de transporte UDP e TCP, enquanto que o IAX suporta apenas o protocolo UDP. Na sinalização, o protocolo IAX utiliza-se de métodos para confirmação da chegada das mensagens, baseando-se nas seqüências das mesmas, realizando este procedimento na camada de aplicação do protocolo TCP/IP. O protocolo SIP, apesar de poder ser aplicado ao protocolo TCP, também exige a confirmação da chegada dos pacotes de mensagens de sinalização. Como o padrão de troca de mensagens é bastante semelhante entre os protocolos, a eficiência da sinalização dos protocolos torna-se bastante dependente das condições de rede entre os agentes.

A transferência de dados de mídia no protocolo IAX, como descrito na seção 2.4, ocorre através do mesmo canal utilizado para estabelecimento das chamadas, enquanto o protocolo SIP estabelece outro canal para a comunicação da mídia, utilizando o protocolo RTP. Visando a redução no consumo da banda em ligações, estruturas de mensagens classificadas como *mini-frames* foram desenvolvidas pelos autores do protocolo IAX. Estes se caracterizam por cabeçalhos de 4 *bytes*, diferente do RTP que utiliza cabeçalho de 12 *bytes*.

Devido ao não estabelecimento de uma nova conexão para a transmissão de dados de mídia, o protocolo IAX classifica-se como mais eficiente quando empregado ao NAT (*Network Address Translation*). Entretanto o problema existente em chamadas intermediadas por NAT não é específico do protocolo SIP, mas de todos os protocolos que utilizam o RTP na transferência de dados em algum momento da sua comunicação. Isso ocorre porque o protocolo RTP não define uma porta padrão para a troca de informações, somente a cada início de sessão, resultando na distorção das portas combinadas pelo protocolo SIP na transmissão de dados de mídia por RTP ou até bloqueio por um possível NAT entre os comunicadores.

O monitoramento da rede tem como foco verificar o transporte de dados de mídia. Como o protocolo IAX além da sinalização provê o transporte de mídia, o monitoramento da rede é funcional, enquanto o protocolo SIP deixa essas funcionalidades ao protocolo responsável pela transmissão de dados. Neste caso, o protocolo RTP utiliza algumas técnicas para avaliar a troca de informações através do RTCP, como destacado na seção 2.3.

Por fim, a documentação salienta-se como o último tópico utilizado na comparação. O IAX não conta com uma documentação tão completa quanto a do SIP, o que reflete a dificuldade de propagação e expansão do protocolo. Adicionalmente, a inclusão de funcionalidades restritas a dispositivos é um fator negativo na padronização do protocolo IAX.

Na seqüência, são discutidos e comentados trabalhos encontrados na literatura.

2.6 Pesquisas na Área

Trabalhos específicos sobre os protocolos SIP e IAX são pouco encontrados em revistas científicas. Temas envolvendo contextos genéricos de VoIP, são deparados em maior quantidade, sendo que a maioria exibem experimentos relacionando aos meios de transmissão com qualidade de ligações.

Zhang [Zhang, 2002] mostrou entidades padronizadas pelo protocolo SIP, descrevendo seus respectivos comportamentos em conjunto a detalhes sobre mensagens trocadas pelo protocolo no estabelecimento de chamadas integradas a telefonia tradicional. As operações foram realizadas pelo protocolo MGCP (*Media Gateway Control Protocol*) que converte sinais entre circuitos e pacotes.

Zeadally e Siddiqui [Zeadally and Siddiqui, 2004] descreveram os passos para desenvolvimento e implementação de um *user agent* SIP que consiga se autenticar, criar e receber chamadas e, também, convidar outros usuários para conferências. Informações sobre atrasos ocorridos durante a execução das funcionalidades descritas foram utilizadas na comparação dos tempos obtidos nas simulações do presente trabalho.

Em [Camarillo et al., 2003], Camarino fez uma análise do comportamento dos protocolos da camada de transporte em sinalizações SIP utilizando o simulador NS-2. Os protocolos de transporte empregados nas simulações foram: UDP, TCP e SCTP (*Stream Control Transmission Protocol*). O SCTP é orientado a conexões foi desenvolvido pela IETF e mostra-se bastante eficaz na transferência de dados. Os resultados obtidos consideraram como vantajoso o uso de protocolos que garantam tanto a entrega de dados quanto na perda de pacotes. Neste caso, a própria aplicação gerencia o tempo de espera para o reenvio de pacotes quando aplica-se o protocolo UDP. Quanto ao controle de congestão, apesar de aumentar o tempo com gastos no envio de mensagens, existe a garantia de que a informação chegará ao seu destino.

Lulling e Vaughan [Lulling and Vaughan, 2006] tratam das mesmas aplicações discutidas no trabalho de Camarino. Modificações foram feitas para aplicação de três variações do protocolo TCP: TCP *Reno*, TCP *Vegas* e TCP *Sack*. O trabalho destaca-se por não

considerar a seqüência de mensagens trocadas pelo protocolo SIP, fazendo com que dados relacionados ao UDP sejam precipitados. Contudo, problemas observados no experimento de avaliação de perda de pacotes, cujos dados obtidos pelo atraso do protocolo UDP são constantes, não foram considerados nos resultados os tempos adicionais gerados pelos pacotes re-enviados.

O trabalho de Abbasi e Prasad [Abbasi and Prasad, 2005], foi a inspiração para o desenvolvimento deste projeto. Nele, um estudo comparativo entre os protocolos SIP e IAX foi efetuado. Exibindo algumas diferenças técnicas dos protocolos e realizando experimentos reais com um emulador de tráfego de rede, foi observado o comportamento dos protocolos através das análises de tempo de atraso, perda de pacotes e reordenação de pacotes. A avaliação da qualidade das ligações foi executada por métodos subjetivos, no qual destacou-se o protocolo IAX nos experimentos. Outro ponto que deve ser considerado é o enfoque dos experimentos apenas na qualidade de ligações, não sendo considerados os tempos para a realização de chamadas.

Em [Roychoudhuri et al., 2002], os autores descreveram as influências da Internet na qualidade de ligações VoIP. Por meio de um experimento realizado durante dose meses, trocas de arquivos de áudio e mensagens foram feitas a cada quatro horas para o calculo do RTT (*Round Trip Time*), que é a soma do tempo de ida e de volta de pacotes, entre diversos pontos dos Estados Unidos e outras localizações. Destacou-se através da análise dos resultados, que o número de estações intermediárias mensuradas pela distância da comunicação entre dois pontos, não é um indicador de qualidade no caminho estabelecido, em termos de atrasos e congestão, ou seja, a quantidade de estações não implica na qualidade do *link*. Outro ponto observado, foi que aumento do RTT é um indicador para perda de pacotes, entretanto essa relação não é simplesmente linear.

Em outro trabalho, [Furuya et al., 2003], os autores investigam quantitativamente através de experimentos, a relação entre a qualidade das ligações e problemas ocorridos na rede. Alterações no meio de comunicação, visando alterar parâmetros de *jitter*, taxas de pacotes perdidos e tráfego foram avaliados através do método objetivo PESQ (*Perceptual Evaluation of Speech Quality*). A análise dos experimentos realizados mostraram como principal fator de influência na qualidade das ligações a variação da limitação da taxa de transferência de dados no meio de comunicação. Adicionalmente, os outros parâmetros foram considerados de pouca influência nos resultados.

Ding e Goubran [Ding and Goubran, 2003] investigaram os efeitos dos pacotes perdidos em ligações VoIP utilizando o codificador de voz G.729 [ITU-T, 2007]. Em experimentos reais, analisados pelo método objetivo *E-Model* e experimentos simulados, analisados pelo método objetivo PESQ, os testes foram realizados com parâmetros de perda randômica de pacotes, diferenciação no tamanho dos pacotes e uso de técnicas de recuper-

ação de erros. Os resultados apresentaram que o tamanho dos pacotes de voz influenciam na variação do MOS e tem um impacto significativo no uso da técnica de recuperação que tenta gerar informações intermediárias por pacotes antecessores e sucessores aos pacotes perdidos.

Em [De Rango F., 2006] os autores exibem uma descrição geral sobre as técnicas para mensurar a qualidade das ligações VoIP, descrevendo as influências de cada um dos parâmetros que causam problemas na comunicação entre dois pontos e classificando hierarquicamente métodos utilizados para a verificação.

2.7 Considerações Finais

Neste capítulo, informações sobre o funcionamento dos protocolos de sinalização H.323 e SIP, do protocolo de transporte de mídia RTP e do protocolo IAX foram descritos. Em seguida, uma discussão sobre as diferença dos protocolos SIP e IAX foi apresentada. Por fim, foram exibidos trabalhos que tratam de análise de desempenho VoIP.

Para averiguar o desempenho dos protocolos SIP e IAX em relação à qualidade do áudio trafegado, foram necessários estudos sobre métodos empregados em análise de qualidade de voz. Estes são descritos no próximo capítulo.

Capítulo

3

Métodos para mensurar qualidade VoIP

3.1 Considerações Iniciais

Mensurar qualidade de áudio é considerado um procedimento subjetivo, pois o entendimento da mensagem de voz depende da capacidade auditiva de cada pessoa. Dessa forma, procedimentos devem ser seguidos para que valores relacionados a qualidade de chamadas VoIP sejam comparados. Para isso, as análises das propriedades de comunicações VoIP consideram problemas como: atraso, *jitter*, perda de pacotes e eco. Na busca da quantização da qualidade de voz em ligações, foram desenvolvidos métodos classificados globalmente como subjetivos e objetivos, sendo que estes são apresentados neste capítulo.

3.2 Métodos Subjetivos

A avaliação da qualidade das ligações através de métodos subjetivos fundamentalmente se resumem a gravações de amostras de som, que são ouvidas e avaliadas por pessoas, após passarem por diversas condições de comunicação.

De forma a mensurar a qualidade de uma comunicação, utiliza-se como medida padrão o *Mean Opinion Score* (MOS) desenvolvido pela ITU [ITU-T, 1996b]. O MOS associa

a percepção do som observado pelo ouvido humano com a escala numérica de 1 a 5, conforme a tabela 3.1.

Tabela 3.1: Tabela *Mean Opinion Score*

Qualidade	**MOS**
Excelente	5
Bom	4
Razoável	3
Pobre	2
Mau	1

Métodos de avaliação subjetiva apresentam desvantagens tais como quanto a experiência, humor e cultura dos avaliadores, o que pode fazer com que a opinião sobre o mesmo áudio seja divergente . Além de serem bastante custosos pela necessidade do uso de salas especiais e fatores de ambiente controlados, também exigem uma quantidade de pessoas para obtenção de resultados verídicos. Existem três tipos de classificação para os métodos subjetivos: *listening tests*, *conversational opinion tests* e *quantal-response detectability tests* [De Rango F., 2006].

3.2.1 Listening Tests

Os *listening tests* são feitos com transmissões unidirecionais, no qual conversas ou frases pré-gravadas são transmitidas sobre diferentes condições de comunicação. As amostras de referência, que são os dados originais antes da transmissão, podem ser ou não utilizadas na avaliação. Os *listening tests* mais conhecidos são: ACR (*Absolute Category Rating*), DCR (*Degradation Category Rating*) e CCR (*Comparison Category Rating*).

- *Absolute Category Rating* (ACR): método que visa obter o valor absoluto da qualidade dos dados. Os avaliadores escutam e determinam uma nota ao arquivo de som modificado sem o uso de amostras de referência para comparação.

- *Degradation Category Rating* (DCR): método utilizado para dados de alta qualidade. Os avaliadores ouvem os dados de referência seguido dos arquivos defeituosos e, então, outorgam suas notas por comparação.

- *Comparison Categoary Rating* (CCR): método similar ao DCR exceto pela seqüência de execução dos procedimentos. Enquanto no DCR a amostra de referência é apresentada inicialmente ao avaliador, no CCR a ordem das amostras aparecem

aleatoriamente, ou seja, a primeira amostra ouvida pode ser não ser amostra de referência.

3.2.2 Conversational opinion tests

Conversational opinion tests são ensaios de laboratório que objetivam reproduzir condições reais experimentadas. São extremamente importantes que as condições aplicadas a simulação sejam especificadas corretamente e reproduzidas fielmente às ligações reais.

3.2.3 Quantal-Response Detectability Tests

Quantal-Response Detectability Tests representam os métodos mais indicados para obtenção de informações sobre parâmetros que influenciam na qualidade das ligações e sons analógicos. Se baseia em uma escala de qualidade mais restrita que o MOS, cujas falhas encontradas nas amostras são classificadas como falha censurada, detectada e não detectada. Diferentes tipos de testes *quantal-response* utilizam essa escala, sendo necessária a conversão dos resultados ao padrão MOS.

3.3 Métodos Objetivos

De forma a solucionar os problemas decorrentes dos métodos subjetivos, tais como os altos custos para a realização de análises e a não possibilidade de avaliação durante a realização das chamadas, os métodos objetivos medem a qualidade das ligações através de cálculos de valores que representam diferentes fatores prejudiciais, combinados a parâmetros da comunicação. A meta dos métodos objetivos é alcançar resultados próximos aos valores de MOS obtidos em testes subjetivos.

Os métodos objetivos são categorizados em: métodos de comparação, que confrontam amostras transmitidas com suas referências; métodos absolutos: se fundamentam na estima da qualidade absoluta; e métodos de transmissão: obtêm valores do meio de comunicação, afim anteceder a analise da qualidade das ligações.

Uma outra classificação divide os métodos objetivos em intrusivos e não intrusivos. Os métodos não intrusivos fazem a avaliação durante a ocorrência das chamadas, tornam prescindível a utilização dos dados originais. Por outro lado, os métodos intrusivos fazem uso da comparação dos dados originais com as amostras obtidas nas ligações.

3.3.1 Métodos Objetivos Não-Intrusivos

Métodos não intrusivos ou também chamados passivos são desenvolvidos para mensurar o tráfego durante a ligação, baseando-se em previsões diretas através das informações de parâmetros dos meios de comunicação, tais como atraso, *jitter* e pacotes perdidos.

O modelo *In Service Non-intrusive Measurement Device* (INMD), um dos precursores dos métodos não-intrusivos, foi padronizado pela ITU P.562 [ITU-T, 2004b] para medir ruído, eco e níveis de perda em telefonia tradicional. Atualmente, o INMD suporta a comutação por pacotes, localizando e analisando a performance de danificação da voz pelas avaliações de distorções simples. Todavia, a falta de combinação dos parâmetros listados conduziu ao desenvolvimento do método *Call Clarity Index* (CCI) [ITU-T, 2004b].

O CCI foi desenvolvido para integrar o modelo INMD combinando diferentes parâmetros de ligações, através de um modelo matemático baseado na percepção humano para a obtenção da qualidade.

O *Non-intrusive Quality Assessment (NIQA)*, desenvolvido pela *Psytechincs*, se situa como uma extensão ao modelo CCI, gerenciando todos os tipos de distorções, tais como atrasos, falhas e baixas taxas de transmissão. O modelo NIQA possibilita a inclusão de uma quantidade de defeitos no áudio, permitindo uma classificação para cada codificador de voz utilizado.

Outro método de análise não-intrusiva é o *PsyVoIP*, similar ao modelo INMD. Este é mais aplicado a sistemas VoIP, avaliando a qualidade de várias chamadas simultaneamente.

O método *Perceptual Single ended Objective Measure (PSOM)* foi desenvolvido pela France Telecom R&D, e visa o oferecimento de previsão da qualidade de tráfego do meio de comunicação. O PSOM verifica distorções separadamente, analisando-os através de modelos estatísticos que resultam em probabilidades sobre cada parâmetro de diferença entre as amostras de referência e suas respectivas amostras degradadas durante a comunicação.

Por fim, o *European Telecommunications Standards Institute Computation Model* (E-Model) [ITU-T, 1998a], é o método não intrusivo mais popular. Difere-se pela presença de uma ferramenta de simulação de rede que se fundamenta na associação entre tipo de degradação e de fator de danificação. Os parâmetros observados na transmissões para gerar a análise de qualidade pertencem a uma escala de 0 a 100, classificado como fator R. Entretanto esses dados devem ser traduzidos ao padrão MOS.

3.3.2 Métodos Objetivos Intrusivos

Os métodos intrusivos são mais precisos na análise da qualidade das ligações, mas inadequados ao monitoramento do tráfego em tempo real. Tipicamente utilizam duas amostras para fazer a avaliação: a primeira contendo dados de referência e a segunda dados degradados na transmissão. Os métodos intrusivos podem ser classificados como:

- *Time domain groups*: métodos não apropriados a redes modernas e codificadores com baixa taxa de transferência de dados. A analise é feita pela verificação dos ruidos de sinais.

- *Sprectral domain measures*: métodos que calculam tamanhos variáveis dos seguimentos obtidos pelas ondas sonoras. São mais sensíveis à falta de sincronização entre as amostras.

- *Perceptual domain measures*: métodos baseados no modelo de percepção humana, considerados os melhores processos de testes de avaliação [De Rango F., 2006].

Os métodos *Perceptual domain measures* são mais precisos por usarem algoritmos que tentam representar o aparelho auditivo humano. Dentre os modelos mais utilizados destacam-se: *Perceptual Speech Quality Measure* (PSQM), *Measuring Normalizing Block* (MNB), *Perceptual Analysis Measurement System* (PAMS) e *Perceptual Evaluation of Speech Quality* (PESQ).

Padronizado pela ITU-T Rec. P.861 [ITU-T, 1996a], o PSQM é um algoritmo matemático que mede a qualidade do som, através da análise do ruído gerado pela intersecção entre as amostras de referência e defasada. A principal desvantagem do PSQM é não examinar parâmetros relacionadas a atrasos e pacotes perdidos.

O MNB surgiu como uma técnica alternativa ao PSQM, recomendado para avaliar os impactos aplicados a alguns parâmetros que afetam a qualidade das ligações. Os principais referem-se a problemas ocorridos nos canais de comunicação e, especialmente, em codificadores com baixas taxas de transmissão.

Desenvolvido pela British Telecommunications, o método PAMS, em contraste ao PSQM e ao MNB, inclui toda a sincronização e normalização das amostras defasadas e suas respectivas referências. A medida inclui fatores prejudiciais à comunicação, tais como pacotes perdidos, altas variações de atraso e distorções geradas por codificadores de voz utilizado nas transmissões. Entretanto, o PAMS não atinge as expectativas em casos de pouca variação de atraso, telefonia analógica, ruídos de fundo e sons musicais.

O *Perceptual Evaluation of Speech Quality* (PESQ) padrão P.862 desenvolvido pela ITU-T [ITU-T, 2001], surge como uma integração entre as melhores características dos modelos PAMS e PSQM, sendo aplicado na verificação de codificadores, optimização de equipamentos e monitoramento de redes.

Da mesma maneira que os outros métodos, o PESQ compara o sinal de som original com o degradado pela comunicação. A figura 3.1 ilustra os passos percorridos pelo método PESQ na análise da qualidade das amostras. O primeiro passo do algoritmo ocorre pela certificação dos atrasos obtidos com a comparação das amostras (1). Estes são calculados e separados em intervalos. As variações entre os pontos de início e fim dos intervalos obtidos devem ser levados em conta no conceito final da análise. Na seqüência, o intervalos obtidos são sincronizados e transformados em representações análogas a psicofísica de sinais de áudio no sistema auditivo humano, considerando a freqüência auditiva e sonoridade (2,3). A diferença entre as representações auditivas (4) unidas aos cálculos obtidos anteriormente resulta na informação de qualidade (5).

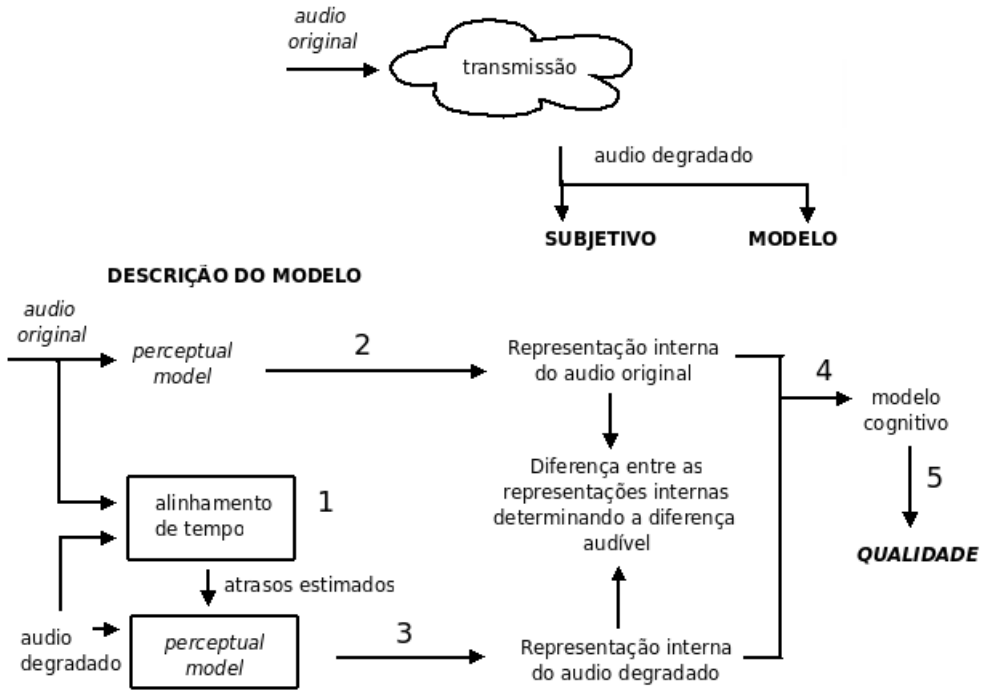

Figura 3.1: Passos percorridos pelo método PESQ para mensurar da qualidade do áudio [ITU-T, 2001].

3.4 Considerações Finais

Este capítulo apresentou inicialmente informações sobre o MOS, métrica quantitativa utilizada na análise de qualidade de áudio. Em seguida, foram ilustrados detalhes sobre as características empregadas em métodos subjetivos de avaliação. Por fim, uma classificação categórica dos métodos objetivos facilitou a verificação da evolução de algoritmos que buscam a representação do aparelho auditivo humano.

Após a descrição detalhada dos protocolos VoIP SIP e IAX e a apresentação de alguns métodos para análise de qualidade de voz, faz-se necessária a implantação dos métodos no ambiente de simulação proposto. Para isso, tenta-se estabelecer vantagens e desvantagens de cada um dos protocolos em cenários a serem estabelecidos no próximo capítulo.

Capítulo

4

Simulação dos Protocolos VoIP

4.1 Considerações Iniciais

As análises comparativas entre os protocolos SIP e IAX foram realizadas utilizando a técnica de avaliação de desempenho denominada simulação. Para isso, foi desenvolvida uma metologia capaz de efetuar e posteriormente examinar as chamadas geradas no ambiente fictício.

Este capítulo exibe inicialmente informações referentes à ferramenta *Network Simulator* (NS) [Ns, 2006]. Em seguida, descreve os passos percorridos para o suprimento da falta de protocolos VoIP na ferramenta de simulação e apresenta os métodos empregados para avaliação das ligações simuladas. Por fim, os cenários utilizados na avaliação dos protocolos são apresentados.

4.2 Network Simulator

O *Network Simulator*(NS) é um simulador de eventos discretos que possibilita simulações aplicadas a protocolos de rede, roteamento sobre redes locais, áreas distribuídas, sem fio e via satélite.

O NS iniciou-se como uma alteração do existente *REAL Network Simulator* [rea, 2006] no ano de 1989. Em 1995, financiado pelo DARPA (*Defense Advanced Research Projects Agency*), colocou-se em prática o projeto VINT [vin, 2006], que pretendia desenvolver um simulador de redes de computadores.

Atualmente na versão 2, o *Network Simulator* (NS-2), é modelado logicamente em duas camadas. A primeira, manipulada através de linguagem *OTcl*, é a interface para a criação topológica das simulações, englobando todas as possíveis opções de configuração. A segunda, implementada em linguagem C++, executa a simulação de eventos discretos de acordo com os atributos organizados pela primeira camada em um único processo.

Da mesma maneira que ocorre em sistemas reais, o NS define uma API (*Applications Programming Interface*) para acesso aos serviços de rede a partir das aplicações. Os *network sockets*, comunicadores de rede empregados em algoritmos tradicionais, são representados pelos agentes de rede da ferramenta, tornando o desenvolvimento semelhante ao de aplicações reais

A figura 4.1 ilustra o paralelo entre as aplicações reais e as desenvolvidas no universo do simulador. Como pode ser observado, os algoritmos de aplicação tradicionais, assim como os simulados, situam-se no mesmo nível e são vinculados a uma API para acesso aos serviços da rede. Todavia, *network sockets* e agentes de rede definem diferentes interfaces para uso dos recursos da camada inferior, fazendo com que as aplicações se diferenciem apenas pelas invocações às APIs.

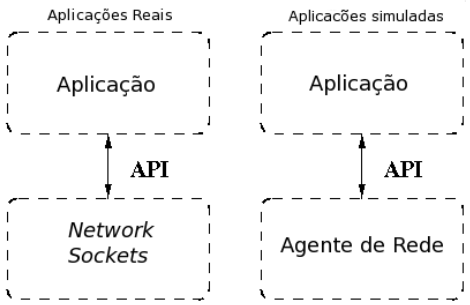

Figura 4.1: Comparação entre as aplicações reais e as desenvolvidas no contexto do NS.

Embora o *Network Simulator* mantenha em seu repertório um número elevado de protocolos de aplicação, os protocolos SIP e IAX não foram encontrados em suas listas, o que exigiu o desenvolvimento de ambos para a realização dos experimentos.

4.3 Implementação dos protocolos VoIP

Foram realizadas várias pesquisas na literatura científica objetivando a busca por implementações dos protocolos SIP e IAX no simulador NS. As buscas resultaram na

localização de dois algoritmos que compreendiam o protocolo SIP, mas nenhum módulo que simulasse o protocolo IAX foi encontrado. Dessa forma, foram realizados estudos de adequação das funcionalidades requeridas para a avaliação dos dois algoritmos SIP.

O primeiro, criado por Fasciana [Fasciana, 2003], apresentou documentação completa sobre os elementos SIP empregados nas chamadas simuladas. Porém, necessitou de adaptações para integração com versões mais atuais do simulador, além de não suprir todas funcionalidades de servidores *Proxy* do protocolo, estas descritas na seção 2.2.2 deste trabalho.

O segundo, apesar da falta de documentação e ainda necessitar de adaptações para obtenção dos eventos esperados, mostrou-se mais adequado por ter sido adotado nos trabalhos de Lulling e Vaughan [Lulling and Vaughan, 2006]. Assim, foi definido o uso do segundo algoritmo para as simulações do protocolo SIP.

A figura 4.2 ilustra a estrutura empregada pelo algoritmo SIP escolhido. Observa-se em tal implementação que a camada de aplicação é composta pelos agentes de sinalização e transporte de mídia. Os agentes de sinalização, através de cabeçalhos simplificados do protocolo, são responsáveis pela troca de mensagens para o controle das funcionalidades de registro, criação e finalização de chamadas, tanto em clientes quanto nos servidores SIP. Os agentes de transporte de mídia, os quais utilizam os cabeçalhos RTP, geram o tráfego de voz das chamadas, sendo criados e destruídos de acordo com o início e término das ligações. Destaca-se ainda que o acesso aos serviços da rede pelos elementos descritos são efetuados através dos agentes de rede UDP.

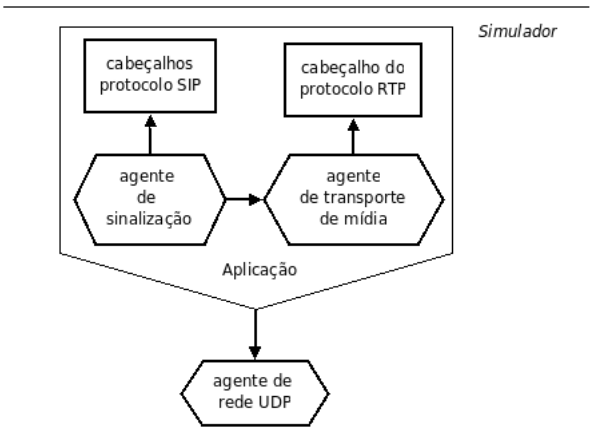

Figura 4.2: Estrutura do algoritmo empregado nas simulações SIP.

Como não foram encontrados algoritmos que capacitassem o NS para o gerenciamento de chamadas IAX, tornou-se necessária a criação de um módulo para a simulação e consequente análise de qualidade de protocolo.

Conforme ilustrado na figura 4.3, a criação do protocolo IAX seguiu os padrões lógicos adotados pela camada de aplicação do algoritmo SIP escolhido, ou seja, a definição sobre

os agentes de sinalização e transporte de mídia foi mantida. Porém foi inevitável a geração de agentes para criação, finalização e estabelecimento de chamadas e troca de dados de mídia. Estes novos elementos estão ilustrados de forma tracejada na figura. Afim de tornar as ligações IAX simuladas fiéis às reais, foram empregados cabeçalhos obtidos a partir da divulgação do código do aplicativo *Asterisk*, tanto nas mensagens de sinalização quanto na troca de dados de mídia. Para o acesso aos serviços da rede, também foram utilizados os agentes de rede UDP na implementação IAX.

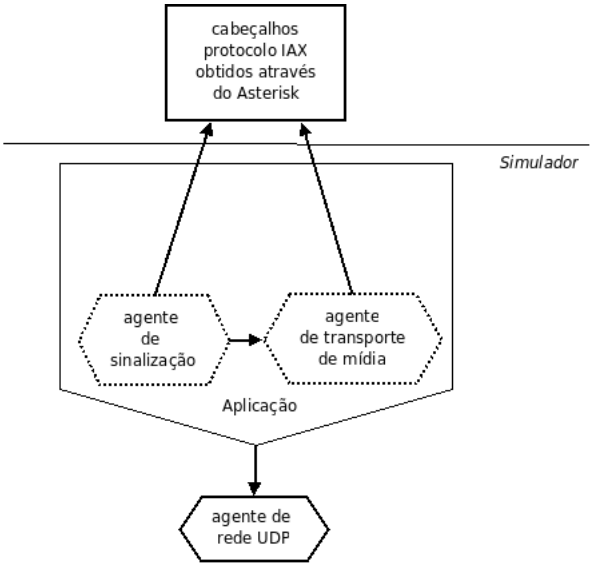

Figura 4.3: Estrutura do algoritmo do protocolo IAX adequado ao *Network Simulator*.

A versão obtida do algoritmo SIP e as primeiras implementações do protocolo IAX sofreram adaptações de modo a prover as funcionalidades desejadas. A figura 4.4 ilustra o modelo básico definido pelos protocolos SIP e IAX.

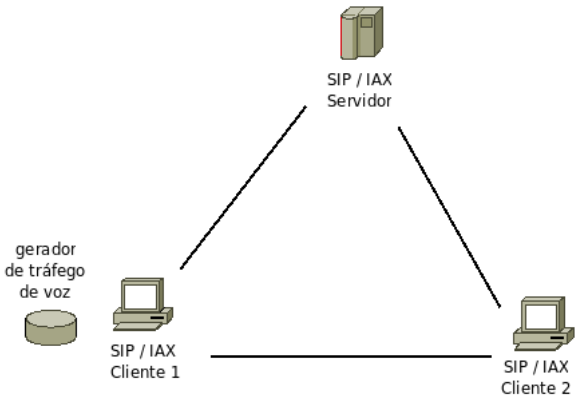

Figura 4.4: Modelo inicial de funcionamento dos algoritmos dos protocolos SIP e IAX no *Network Simulator*.

Nesta versão inicial, as ligações eram efetuadas sem necessidade de registro. O procedimento adotado começava com uma requisição por parte do *cliente 1* ao *cliente 2* através

do *servidor*, seguido do envio de uma quantidade de dados fixa a uma taxa de transferência definida por uma constante no gerador de tráfego de voz.

Posteriormente, elementos funcionais foram adicionados à representação inicial, estes são representados pelos desenhos tracejados na figura 4.5.

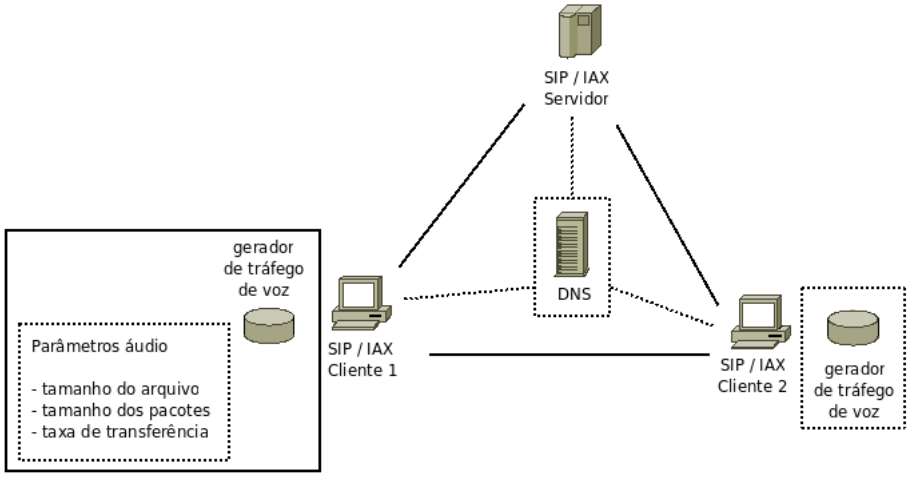

Figura 4.5: Modificações nos algoritmos dos protocolos SIP e IAX no *Network Simulator*.

Primeiramente, foi necessário o desenvolvimento de um servidor *Domain Name Service* (DNS) para a conversão entre endereços de rede definidos pelo simulador e nomes empregados por SIP *Proxy*s e servidores IAX. Em seguida, para que as simulações representassem arquivos de sons reais codificados durante as chamadas, foram adicionados parâmetros relacionados ao tamanho do arquivo de áudio, quantidade de *bytes* representados por cada pacote e freqüência de envio dos fragmentos.

A terceira mudança foi a adição de um mesmo gerador de tráfego de voz à clientes que recebem ligações, pois apenas os criadores de chamadas enviavam dados de mídia. Todavia, o procedimento executado pelo receptor não recorre ao envio de dados com taxas constantes, este basicamente responde a um pacote de mídia com outro.

A quarta alteração foi a realização do registro pelos clientes antes da criação das chamadas. Métodos de criptografia na autenticação de usuários não foram empregados, tornando a implementação inadequada para avaliações de segurança dos protocolos.

Em conjunto às implementações e modificações dos protocolos, uma estrutura capaz de unir os dados gerados pelas chamadas simuladas e arquivos de áudio reais foi construída para a medição da qualidade das ligações.

4.4 Qualidade das ligações no ambiente de simulação

A maneira encontrada para mensurar a qualidade das ligações no ambiente da ferramenta NS baseia-se na combinação dos dados de mídia obtidos como resultados das simulações com arquivos de áudio reais. O áudio resultante é utilizado para a avaliação de qualidade.

Para que as simulações representassem pacotes de sons foi necessário o uso de arquivos no formato G.711[ITU-T, 1988] pois, neste tipo de codificação, existe a correspondência direta entre fragmentos de áudio e dados trafegados na comunicação. Desta maneira, cada pacote obtido nas simulações representa um fragmento de som também codificado no formato G.711, o que aproxima os eventos sofridos pelos pacotes de áudios simulados aos aplicados em sistemas reais.

Como o som de referência será alterado pela simulação, a forma mais simplificada para obtenção de resultados de qualidade deve seguir o modelo dos métodos objetivos intrusivos, este explicado na seção 3.3.2 desta dissertação. Dentre os algoritmos definidos pelo modelo, o que apresentou resultados mais precisos quando comparados a métodos subjetivos de análise foi o PESQ [Rix et al., 2001], sendo este utilizado neste projeto.

Para o procedimento de geração de arquivos de áudio modificados com base nos resultados das simulações VoIP foi desenvolvido o aplicativo *nsTraceVoIP*.

4.4.1 Aplicativo nsTraceVoIP

O *nsTraceVoIP*, aplicativo desenvolvido durante a evolução deste projeto, essencialmente trabalha na geração de arquivos de áudio que representem pacotes recebido por clientes nas simulações VoIP executadas no *Network Simulator*. Para isso, um arquivo de som codificado em G.711 é fragmentado de acordo com a quantidade de *bytes* de dados armazenados em cada pacote das simulações.

A figura 4.6 ilustra a maneira com que o algoritmo emprega a associação entre as partes resultantes da divisão de um arquivo de som G.711 e os pacotes trocados na simulação. Como a codificação G.711 utiliza 8000 amostras por segundo, a uma taxa de transmissão de 64kbps[1], os fragmentos de som da figura representam um áudio de por 20ms e têm tamanho de 160*bytes* de dados. Adicionalmente, os pacotes obtidos nas simulações não são compostos apenas dados de mídia (160*bytes*). Cabeçalhos IP, UDP e RTP ou *mini-frames* empregados respectivamente para endereçamento e roteamento, transporte e transmissão de mídias contínuas, também são agrupados às mensagens.

Para avaliar a fidelidade dos resultados fornecidos pelo aplicativo, foram efetuados estudos sobre comportamento dos clientes VoIP no envio e recebimento de dados de som.

[1] *kilobytes* por segundo.

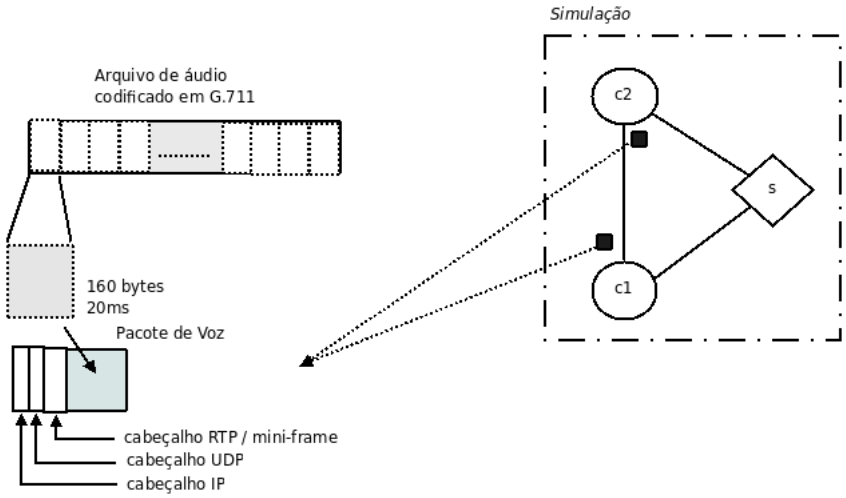

Figura 4.6: Funcionamento do aplicativo *nsTraceVoIP*.

Esta análise trouxe modelos de algoritmos que visam minimizar os efeitos causados pelo atraso e *jitter*, aumentando, assim a qualidade das ligações.

O *multi-buffer* [Baratvand et al., 2008], foi o padrão mais simples encontrado, sendo então incrementado à estrutura do *nsTraceVoIP* para manipular os pacotes recebidos pelos clientes SIP e IAX simulados.

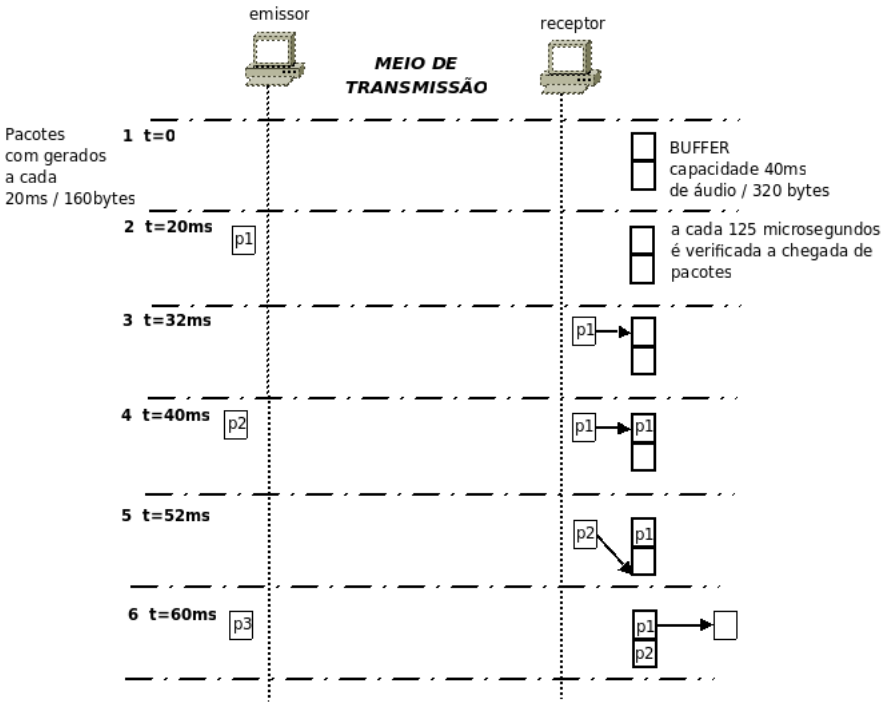

Figura 4.7: Exemplo de funcionamento do algoritmo *multi-buffer*.

A fim de exemplificar, a figura 4.7 exibe o funcionamento do algoritmo *multi-buffer* através de eventos ocorridos em uma ligação VoIP.

Neste exemplo, apenas os elementos extremos da comunicação estão representados. A codificação adotada é G.711 com intervalo de transmissão de 20ms, resultando em 160 *bytes* de dados. O *buffer* adicionado ao receptor tem tamanho de 320 *bytes*, seguindo indicadores positivos de [Baratvand et al., 2008] para o valor encontrado.

A finalização da sinalização para estabelecimento da chamada, seguido da troca de dados de voz se inicia na linha 1 (t=0s). Neste momento, conjectura-se que o emissor comece a gerar seu primeiro pacote de áudio (p1). Depois de 20ms, linha 2 (t=20ms), o pacote primogênito é gerado e enviado através do meio de transmissão, enquanto a criação de um segundo pacote é iniciada. No lado do receptor, a cada 125μs o ponteiro de recepção verifica a chegada de pacotes. Na linha 3 (t=32ms) o pacote p1 chega no receptor, este armazena a informação no *buffer*. Após mais 8ms, linha 4 (t=40ms), o segundo pacote (p2) é montado e transladado à rede. Quando o tempo se iguala a 53ms, o pacote p2 chega ao seu destino, também sendo adicionado ao *buffer*. A leitura da memória temporária se iniciará na linha 5 (t=60ms) devido ao tamanho do *buffer*, ou seja, 40ms depois do emissor enviar o primeiro pacote de áudio. Após o início do procedimento de leitura, a cada 20ms o receptor coleta 160 *bytes* do *buffer*, o que proporciona uma execução constante do áudio. Quando a absorção do pacote se inicia, o uso do *buffer* é verificado. Caso ele esteja vazio, 20ms de silêncio são empregados em equivalência ao segmento que deveria ser consumido. O algoritmo continua sua execução com os pacotes subseqüentes, que são adicionados ao *buffer* sempre que o tempo relativo referente ao fragmento analisado for menor que o tempo relativo de consumo. Caso contrário, como o tempo equivalente ao pacote já foi substituído pelo silêncio, o fragmento deve ser descartado.

A partir da estrutura criada para avaliação das chamadas simuladas no NS, foi possível a definição de cenários baseados em problemas relacionados às métricas empregadas na medição da qualidade das ligações.

4.5 Definição dos Cenários

Os cenários propostos buscam a comparação entre os protocolos SIP e IAX através da verificação da qualidade das ligações. Dentre os parâmetros utilizados na medição de qualidade, as métricas de atraso, *jitter* e pacotes perdidos serão avaliadas a partir das simulações.

Em cada cenário simulado, são incluídos ao menos dois clientes e um servidor VoIP. O procedimento adotado pelos elementos VoIP é sempre o mesmo. Inicialmente, os clientes realizam registro no servidor. Em seguida, o primeiro cliente referencia o segundo através da criação de uma chamada. Quando a ligação é estabelecida, o tráfego de voz é enviado

pelo criador da chamada e correspondido pelo receptor. A chamada é então finalizada quando a quantidade de pacotes de mídia trocados se iguala ao parâmetro correspondente ao tamanho do arquivo de áudio utilizado na geração do som defasado.

Em conjunto a estes elementos básicos, foram acrescentados outros elementos com o objetivo de gerar irregularidades nas chamadas através de manipulação de canais de conexão ou da adição de indivíduos que possam disputar os meios de transmissão aplicado às ligações simuladas. Para o segundo caso, o *Network Simulator* oferece um módulo gerador de tráfego denominado *Web Traffic Generation* (WTG).

O WTG, desenvolvido por pesquisadores do *Internet Traffic Research Group* da *Bell Labs*, foi criado a partir de estudos recentes de taxas de tráfego HTTP na *Internet* [Kevin Fall, 2007]. O componente, disponibilizado juntamente com o NS, não empreende interação entre aplicações clientes e servidores, apenas simula os dados gerados em nível da camada de rede. Assim, a completa funcionalidade do gerador de tráfego *Web* no NS necessita de ao menos um cliente e um servidor, nas quais as respectivas ações de requisição e resposta são efetuadas.

Por fim, na emulação dos pacotes de mídia trocados nas simulações, em todos os cenários, foi empregado um único arquivo de áudio codificado em G.711 com 48 segundos de duração, este obtido juntamente com a distribuição do algoritmo PESQ.

Para cada cenário, foram associadas categorias de possíveis defeitos, possibilitando a verificação dos resultados causados por cada problema dentro do contexto das análises.

4.5.1 Cenário 1

O cenário 1, ilustrado pela figura 4.8, faz uso apenas dos elementos que implementam os protocolos SIP e IAX. Na intenção de simular problemas de perda de pacotes, limitação da taxa de dados e atraso, três variações para este cenário foram definidas.

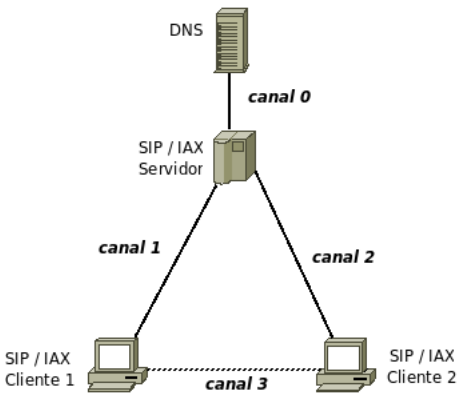

Figura 4.8: Cenário 1

4.5.1.1 Perda de pacotes

Seguindo os padrões adotados por [Abbasi and Prasad, 2005] que analisa a perda de pacotes de acordo com [int, 2008], observou-se que a taxa média de perda de pacotes no mundo varia entre 0% e 9%, sendo a média de perda de 6% na Europa, 5% na América do Norte e 0% na América do Sul. No intuito de avaliar as ligações simuladas, adotou-se médias de perdas de pacotes com uma variação de 0% a 10% e incremento de 0,25%, através de alterações no canal 3 definido pelo cenário 1. Para cada valor utilizado foram realizados trinta testes, para cada execução uma semente foi empregada na geração da aleatoriedade dos pacotes perdidos.

4.5.1.2 Limitação da taxa de dados

A taxa de transferência de dados utilizada pelo codificador G.711 é de 64kbps, como apresentado anteriormente. Quando encapsulados por protocolos RTP e *mini-frames*, essa taxa aumenta devido aos cabeçalhos acrescentados serem diferentes em cada um dos protocolos. Visando estas dessemelhanças, limitou-se a taxa de transferência de mídia trocada nas ligações simuladas. O canal 3 do cenário 1, teve redução na sua capacidade de envio e recebimento de pacotes variando de 10kbps a 100kbps com incremento de 500*bytes*. Neste caso, também foram efetuados trinta testes para cada valor adotado.

4.5.1.3 Atraso

A média de atraso de pacotes no mundo varia entre 50ms e 350ms [int, 2008], sendo que na América do Sul a média é 86ms, na América do Norte 88ms e na Europa 76ms. Para a comparação entre os protocolos, foram empregadas médias de atraso variando de 0ms a 1000ms com incremento de 1ms, no canal 3 do cenário 1. Nesta análise também foram executados 30 testes para cada valor de atraso.

4.5.2 Cenário 2

O cenário 2, exibido na figura 4.9, além dos componentes definidos pelos módulos dos protocolos SIP e IAX, contém dois elementos adicionais. Classificados como gerador de tráfego *web* servidor e cliente, os indivíduos acrescentados são responsáveis pela geração de um volume de dados no canal 5 por meio do módulo *Web Traffic Generator* fornecido pelo NS. Neste cenário, a métrica de *jitter* foi empregada nas análises de qualidade.

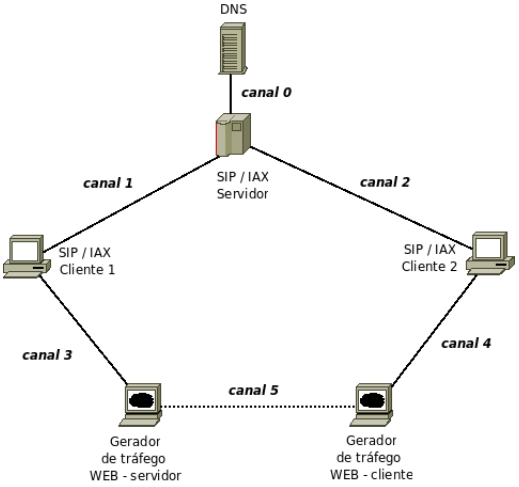

Figura 4.9: Cenário 2

4.5.2.1 Jitter

A concorrência direta entre o tráfego gerado pelos WTGs e as chamadas VoIP causam *jitter* sobre os pacotes trafegados no canal 5 do cenário 2. Desta forma, como o *Web Traffic Generator* gera tráfego por meio de parâmetros relacionados ao número de requisições criadas por segundo, o emprego de taxas fixas de transferências sobre os canais estabelecidos entre os clientes VoIP (canal 3, canal 4 e canal 5) em conjunto à adição de parâmetros relacionados ao número de requisições por segundo efetuadas pelo gerador de tráfego *web*, tornou possível a realização de um experimento que verificar o comportamento dos protocolos através do problema de *jitter*.

Para a realização de tais análises, foram utilizadas variações de 0 a 20 no número de requisições. Em cada variação foram realizados trinta testes, sendo que, para cada execução, é produzida uma semente de aleatoriedade sobre os intervalos das requisições HTTP.

4.6 Considerações Finais

Este capítulo apresentou informações relativas à estrutura do *Network Simulador*. Discutiu sobre as adaptações necessárias para implantação dos protocolos SIP e IAX no contexto do NS e as modificações realizadas na implementação para utilização de áudios reais nas simulações.

Em seguida, foram exibidas informações sobre o modelo adotado para representação dos pacotes de mídia trocados durante a simulação por fragmentos de arquivos codificados

em G.711 além da descrição do algoritmo de *jitter-buffer* associados aos receptores das ligações simuladas.

Para finalizar, o capítulo descreve os cenários e suas variações para avaliação da qualidade das ligações em cada protocolo por meio das métricas de atraso, perda de pacotes, *jitter* e taxa de transferência de dados. Os resultados obtidos pelas simulações são apresentados no próximo capítulo.

Capítulo

5

Resultados

5.1 Considerações Iniciais

Uma vez preparada a infra-estrutura e os cenários para execução das simulações que visam aferir métricas empregadas na avaliação de qualidade de chamadas VoIP, foi necessário o desenvolvimento de um ambiente capaz de efetuar dinamicamente as mudanças exigidas a cada tipo de problema aplicado.

Os procedimentos adotados para a realização dos experimentos são descritos na seção 5.2. Na seqüência, a seção 5.3 apresenta os métodos utilizados para a comparação pormenorizada dos protocolos em conjunto ao detalhamento dos resultados obtidos em cada problema proposto.

Destaca-se, ainda, que as simulações e análises foram realizadas em um computador de processador Intel(R) Core(TM)2 Quad de 2.40GHz e 2Gb de memória RAM, foi disponibilizado pelo Laboratório de Sistemas Distribuídos e Programação Concorrente(LASDPC) do Instituto de Ciências Matemáticas e de Computação da USP São Carlos.

5.2 Metodologia

A metodologia empregada na realização dos experimentos baseou-se na criação de um ambiente de testes flexível à manipulação de cenários e parâmetros. Tais procedimentos fazem uso fundamentalmente das tecnologias de base de dados e *scripts* TCL e *shell*. Os *scripts shell* são utilizados para gerenciamento do fluxo de operação, já os TCL responsabilizam-se por manipular dados dos cenários avaliados.

Participante efetivo do processo, a base de dados atua como centralizador de informações pois, além de armazenar e processar dados gerados nas simulações, ainda guarda os resultados obtidos pelo PESQ, facilitando a geração de relatórios de resultados.

A figura 5.1 ilustra os passos percorridos na execução de cada modificação nas variáveis do cenário para mensurar a qualidade das simulações.

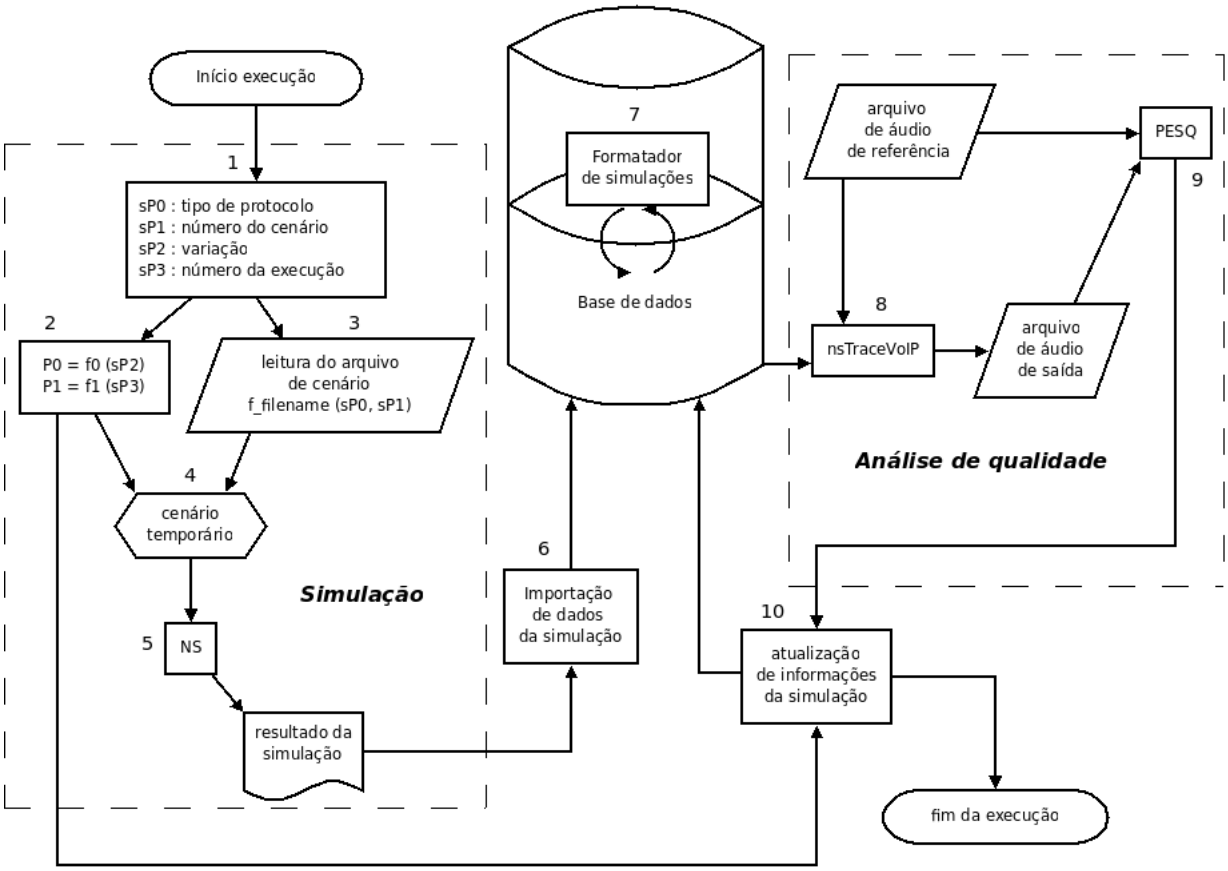

Figura 5.1: Etapas percorridas para medição da qualidade em cada execução de simulação.

Enumeradas de um a dez e divididas em duas áreas, simulação e análise de qualidade, as etapas seguidas no fluxograma são descritas na seqüência:

Etapa 1: Neste primeiro passo, são definidos por *script shell* os parâmetros correspondentes ao tipo de protocolo, numeração de cenário, variação do defeito e número da execução, classificados na figura pelas variáveis `sP0`, `sP1`, `sP2` e `sP3` respectivamente.

Etapa 2: As variáveis correspondentes à variação de cenário e ao número de execução, obtidos na etapa anterior, são utilizados como parâmetros de entrada para o *script* TCL.

Neste passo, os valores transpostos são manipulados através de funções específicas pré-programadas definidas para cada variação de cenário. Os números de execução são utilizados em cálculos matemáticos empregados na criação das variáveis dos cenários. As funções e seus valores de retorno são representados na figura por, respectivamente, `f0`,`P0` e `f1`,`P1`.

Etapa 3: Os parâmetros iniciais referentes ao tipo de protocolo e numeração do cenário são utilizados na localização do arquivo correspondente ao cenário da simulação. O procedimento que trata da leitura das informações e as variáveis de entrada são representadas na figura pela função `f_filename` e `sP0`, `sP1` respectivamente.

Etapa 4: Neste passo, as variáveis transformadas no passo 2 são aplicadas ao cenário obtido na etapa 3, resultando no cenário temporário que será executado na simulação.

Etapa 5: O cenário gerado no passo anterior é então aplicado ao *Network Simulator*, que por sua vez, armazena em arquivo texto com formatação própria, informações correspondentes a todos os pacotes trocados na simulação VoIP.

Etapa 6: Os dados resultantes da simulação são importados na base de dados neste momento. O mesmo padrão de formatação adotado no arquivo de texto é empregado no armazenamento.

Etapa 7: Este passo é realizado dentro da base de dados. A partir de dados brutos importados pela etapa 6, são calculados e armazenados os tempos de envio, recebimento e caminhos sobre todos os pacotes trocados na simulação.

Etapa 8: Com a utilização de um arquivo de áudio compatível com as variáveis aplicadas às simulações e às informações geradas no banco de dados pela etapa anterior, a execução do *nsTraceVoIP* é iniciada. Como descrito na seção 4.4.1, o aplicativo *nsTraceVoIP* gera um segundo arquivo de áudio a partir do arquivo de referência e dos dados obtidos pela comunicação VoIP simulada.

Etapa 9: O arquivo com defeitos gerado pela etapa anterior é comparado ao de referência através do aplicativo [ITU-T, 2001] fornecido pela própria ITU que aplica o algoritmo PESQ.

Etapa 10: O resultado obtido pelo PESQ é armazenado no banco de dados em conjunto com os parâmetros empregados na simulação para que futuras consultas possam ser realizadas.

A figura 5.1 apresenta uma a divisão de duas áreas de estudos específicos. A primeira, classificada com simulação, concentra informações relacionadas aos protocolos SIP e IAX e suas implementações no ambiente do *Network Simulator*. A segunda, nomeada de análise de qualidade, trabalha basicamente com manipulação de dados de mídia contínua em sistemas de tempo real. Como as etapas apresentadas estão bem segmentadas, o modelo empregado torna-se flexível e expansível a aplicações de futuros trabalhos.

A partir do estabelecimento dos processos efetuados para as realização dos experimentos, foi possível a obtenção dos valores correspondentes a qualidade das ligações simuladas sobre SIP e IAX e conseqüentemente a comparação entre os protocolos. As análises são apresentadas na próxima seção.

5.3 Análise dos Resultados

A avaliação dos resultados baseou-se nos valores de PESQ obtidos em chamadas VoIP SIP ou IAX ocorridas dentro do *Network Simulator*. A comparação efetiva dos protocolos foi realizada através de recursos estatísticos de teste de hipótese sobre populações independentes em pequenas amostras [W.C.Shefler, 1988] sobre cada grupo de resultados obtido a partir de modificações realizadas nas variações dos cenários.

Para cada parâmetro empregado, dois testes de hipótese (TH) foram efetuados. O primeiro teste, nomeado como TH1, propõe uma hipótese nula considerando que a qualidade do áudio transportado pelos protocolos SIP e IAX é igual, e uma hipótese alternativa coloca que protocolo SIP consegue obter melhor desempenho nas ligações.

- TH1

$$\begin{cases} H_0: & \text{Áudio transportado pelos protocolos SIP e IAX têm mesma qualidade} \\ H_1: & \text{Protocolo SIP transporta áudio com qualidade maior que IAX} \end{cases}$$

No segundo teste, TH2, a hipótese nula proposta é a mesma que no teste TH1, porém, a hipótese alternativa define que o protocolo IAX obtém melhor qualidade que o SIP sobre as chamadas simuladas.

- TH2

$$\begin{cases} H_0: & \text{Áudio transportado pelos protocolos SIP e IAX têm mesma qualidade} \\ H_1: & \text{Protocolo IAX transporta áudio com qualidade maior que SIP} \end{cases}$$

Desta maneira, a análise detalhada sobre determinado parâmetro é realizada através do cruzamento das informações obtidas nos dois testes. Se o resultado incluir-se na região da hipótese alternativa (H_1) em TH1, afirma-se que o protocolo SIP obtém melhor qualidade nas ligações que o IAX. Se o resultado classifica-se como hipótese alternativa (H_1) em TH2, o IAX é considerado melhor que SIP. Se o resultado incluir-se na hipótese nula em ambos os testes, conclui-se que o IAX é igual ao SIP naquela situação. Por fim, uma estatística é estabelecida situando em quais pontos o protocolo SIP ou IAX obtém melhor desempenho, e em quais condições eles são semelhantes.

Para a realização dos testes de hipótese, o nível de significância considerado foi de 0.1% e as informações detalhadas sobre cada uma das métricas aplicadas são apresentados na seqüência.

5.3.1 Perda de pacotes - Cenário 1

A perda de pacotes, métrica empregada na avaliação de qualidade de chamadas telefônicas, foi mensurada por meio de ligações simuladas no cenário 1, com taxas de perda variando de 0 a 10%, cujos incrementos foram 0.25%. O resultado das análises é ilustrado pelo gráfico de médias PESQ em função das taxas de perda de pacotes, representado na figura 5.2.

Informações mais precisas estão localizadas na tabela 5.1, a qual exibe em suas primeiras colunas valores de desvio padrão e média calculados a partir de cada taxa de perda utilizada. Nas três últimas colunas, encontram-se as verificações sobre as hipóteses alternativas em cada um dos testes propostos (TH1 e TH2) e a hipótese nula em ambos os testes. Para distinguir os resultados dos testes de hipótese, as colunas estão indicadas com 0 para as respostas negativas e com 1 para as positivas.

Desta maneira, ainda pelo gráfico, observa-se que os dados não apresentaram grandes diferenças nos intervalos de perda de pacotes proposto, a tabela confirma essa informação, descrevendo que quase 100% dos resultados obtidos pelos testes de hipótese considera que SIP é igual a IAX. Essa baixa variação indica que os protocolos comportam-se de maneira semelhante quanto ao transporte do mesmo arquivo de áudio.

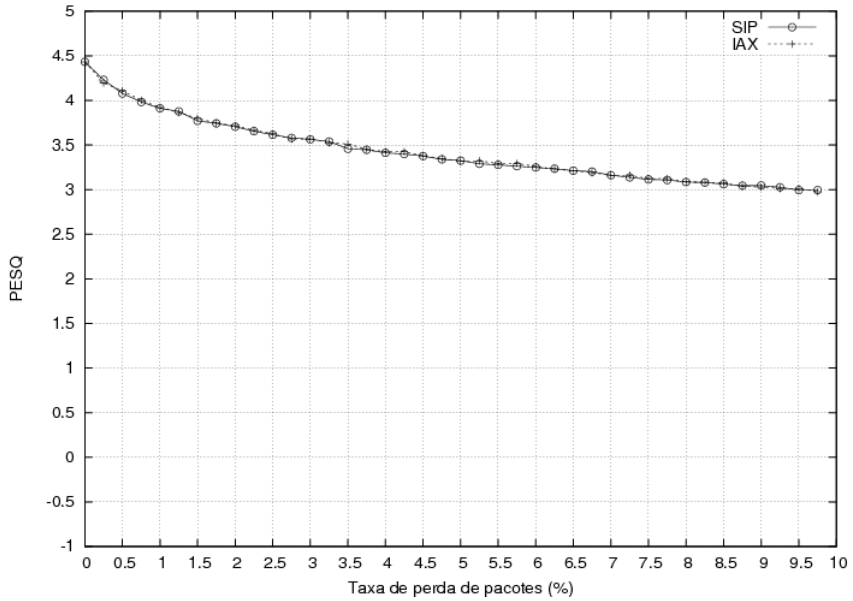

Figura 5.2: Valores de médias PESQ obtidas a partir de variações na taxa de perda de pacotes.

5.3.2 Limitação da taxa de dados - Cenário 1

Dados de comparação entre os protocolos SIP e IAX com variação de taxas de dados foram obtidos a partir de execuções do cenário 1 com variação de taxa de transferência de 10kbps a 100kbps com incremento de 500*bytes*. A figura 5.3 e as tabelas 5.2 e 5.3 apresentam respectivamente o gráfico das médias de PESQ com limitação da taxa de transferência no canal de comunicação VoIP e os valores de média, desvio padrão e testes de hipótese, obtidos em cada parâmetro de transferência empregado.

Os resultados adquiridos são classificados em cinco regiões no gráfico. Na primeira região, variação de 10kbps a 32kpbs, os pacotes enviados pelo criador da chamada estão sempre atrasados em relação à velocidade de leitura do receptor, o que causou resultados mínimos em todas as ocasiões. Na segunda região, taxa de 34kbps a 36kbps, são destacadas melhoras na qualidade do protocolo IAX, devido à chegada de alguns pacotes antes da leitura dos dados por parte do receptor. No caso do SIP, os pacotes não têm a mesmo desempenho pois o RTP tem cabeçalho 8*bytes* maior que os *mini-frames* IAX, o que causa um atraso maior no transporte. Na terceira região, localizada entre 37kbps e 79kbps, as médias PESQ obtidas são bastante próximas, sendo considerados iguais pelos testes de hipótese. No quarto fragmento, taxa de 80kbps a 84kbps, observa-se o aumento expressivo na qualidade das ligações pelo mesmo motivo identificado na segunda região. A quinta região já apresenta chamadas com o máximo de qualidade, tanto para o protocolo

SIP quanto para o IAX. Assim, nota-se que o protocolo IAX obtém melhores resultados em alguns casos, mas em geral eles podem ser considerados iguais.

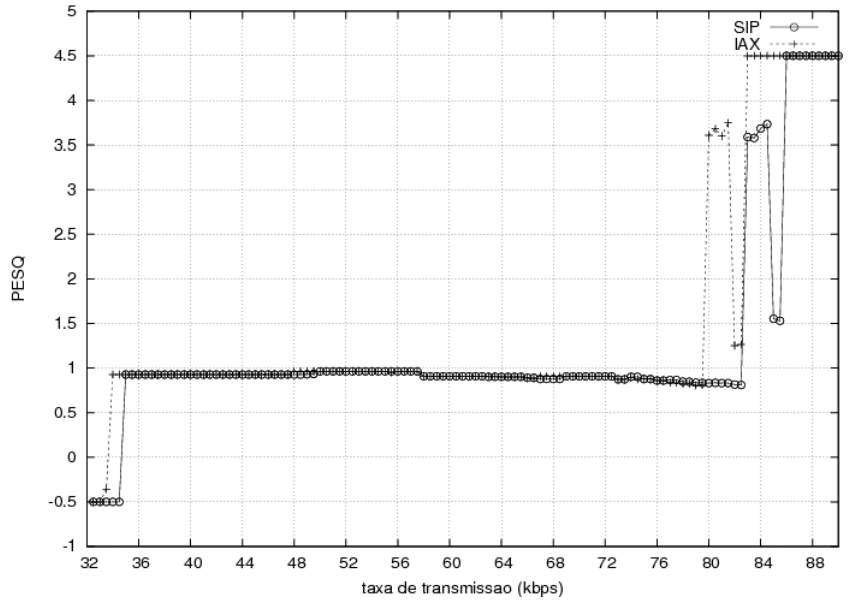

Figura 5.3: Valores de médias PESQ obtidas a partir da limitação na taxa de transferência de dados.

5.3.3 Atraso - Cenário 1

A métrica de atraso foi medida por meio de ligações SIP e IAX executadas pelo cenário 1 com taxas de 0 a 1000ms com intervalos de 1ms de atraso no canal de tráfego de voz.

As médias de PESQ obtidas nas análises mostraram diferenças significativas apenas na região onde os parâmetros de atraso se localizam entre 41ms e 45ms, como se observa na figura 5.4. Em todos os outros pontos, os resultados de PESQ foram considerados iguais, conforme os testes de hipótese apresentados pela tabela 5.4.

Desta maneira, três regiões no gráfico se destacam. Na primeira região, taxa de 0 a 40ms, todos os pacotes enviados pelo criador da chamada foram recebidos e consumidos pelo receptor, tanto no transporte SIP quanto no IAX. Na segunda região, taxa de 41ms a 45ms, o protocolo IAX obteve melhor desempenho, pois os cabeçalhos definidos pelos *mini-frames* são menores do que os do RTP. Assim, neste intervalo o receptor tentou consumir pacotes que ainda não haviam chegado quando transportado pelo protocolo SIP. Porém, no caso do IAX os dados já estavam disponíveis para leitura. A terceira região se destaca pelo atraso elevado de todos os pacotes, o que causaria silêncio no receptor em uma chamada real, independente do protocolo utilizado. De forma geral, os

protocolos responderam da mesma maneira ao ensaio realizado, destacando-se que o IAX uma tolerância maior em relação ao atraso no intervalo de 41ms a 45ms.

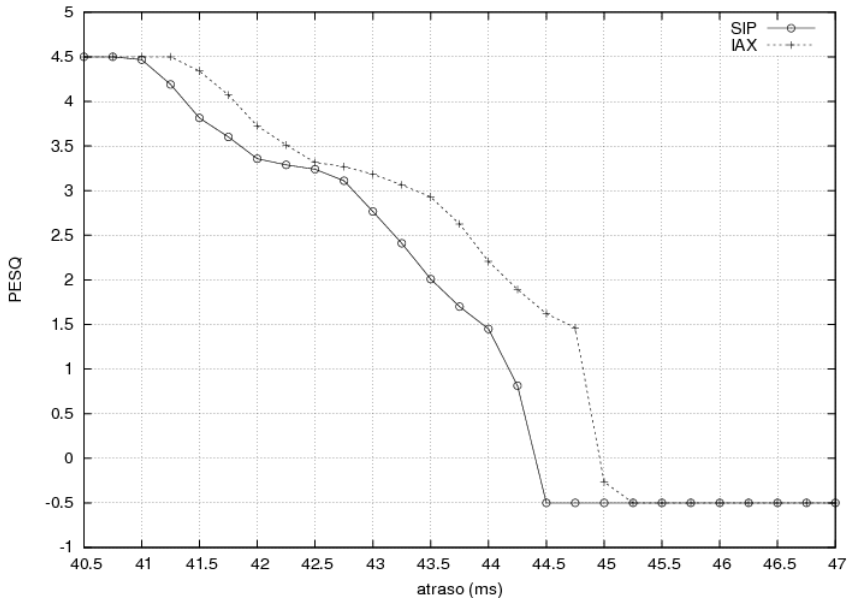

Figura 5.4: Valores de médias PESQ obtidas a partir de variações de atraso.

5.3.4 Jitter - Cenário 2

A métrica de *jitter* foi mensurada por meio de ligações simuladas sobre os protocolos através do ambiente proposto pelo cenário 2. Como descrito anteriormente, para avaliar esta medida, foi utilizado um módulo gerador de tráfego *Web*, fornecido pelo *Network Simulator*, para a geração de ruído de fundo no canal empregado pela troca de dados de mídia.

Os resultados obtidos são ilustrados pela tabela 5.5 e figura 5.5, os quais exibem médias do IAX com ligeiras vantagens na maioria dos pontos. Todavia em todos os testes de hipóteses verificados, os protocolos foram considerados iguais.

5.4 Considerações Finais

Este capítulo ilustrou inicialmente a metodologia empregada para a obtenção dos resultados, em conjunto da descrição de cada passo percorrido para a execução da simulação.

Em seguida, foram apresentados os métodos estatísticos utilizados na comparação dos protocolos, estes formalizados pelos testes de hipótese sobre populações independentes em pequenas amostras.

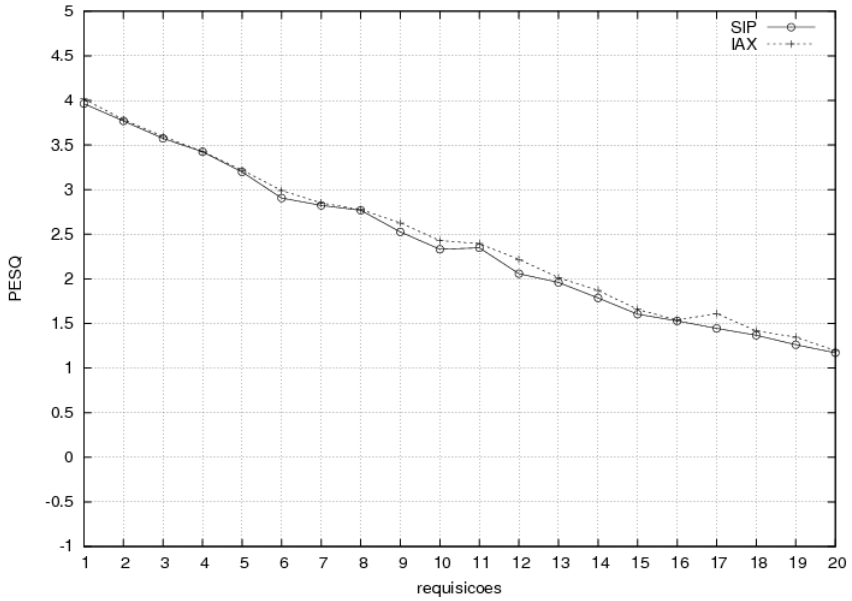

Figura 5.5: Valores de médias PESQ obtidas a partir de variações no número de requisições realizadas pelo *Web Traffic Generator*.

Por fim, análises completas e verificações de cada um dos problemas propostos inicialmente foram realizadas. As conclusões alcançadas por tais avaliações são apresentadas pelo próximo capítulo.

Tabela 5.1: Informações de média, desvio padrão e resultados dos testes de hipótese propostos obtidas a partir de variações na taxa de perda de pacotes.

Taxa de Perda	**SIP**		**IAX**		SIP > IAX	IAX > SIP	IAX = SIP
	Desvio Padrão	**Média**	**Desvio Padrão**	**Média**	TH1 H_1	TH2 H_1	TH1 TH2 H_0
0.00 %	0.000000000001	4.5000	0.000000000001	4.5000	0	0	1
0.25 %	0.000000204414	4.4330	0.000000204414	4.4330	0	0	1
0.50 %	0.075436629414	4.2302	0.080814815985	4.1960	0	0	1
0.75 %	0.072284272784	4.0784	0.081317661482	4.1087	0	0	1
1.00 %	0.067916073969	3.9836	0.072711087867	4.0028	0	0	1
1.25 %	0.056753844368	3.9110	0.066415273136	3.9196	0	0	1
1.50 %	0.080930827566	3.8778	0.083687877972	3.8648	0	0	1
1.75 %	0.063265804894	3.7717	0.065469173231	3.7981	0	0	1
2.00 %	0.056769548256	3.7423	0.079894535367	3.7466	0	0	1
2.25 %	0.080018790896	3.7056	0.056307458520	3.7157	0	0	1
2.50 %	0.045402364615	3.6552	0.058545433776	3.6648	0	0	1
2.75 %	0.075099627315	3.6153	0.063061263771	3.6276	0	0	1
3.00 %	0.059746677876	3.5793	0.056789194779	3.5672	0	0	1
3.25 %	0.049095684198	3.5646	0.079763479385	3.5678	0	0	1
3.50 %	0.067838084783	3.5398	0.076858012093	3.5258	0	0	1
3.75 %	0.055715038519	3.4582	0.057404433326	3.5112	0	0	1
4.00 %	0.062886323275	3.4460	0.073729670878	3.4537	0	0	1
4.25 %	0.070452201127	3.4120	0.059666515764	3.4259	0	0	1
4.50 %	0.052480363048	3.3991	0.067207210888	3.4241	0	0	1
4.75 %	0.042157613402	3.3766	0.048174598638	3.3809	0	0	1
5.00 %	0.057359814119	3.3393	0.066525675624	3.3483	0	0	1
5.25 %	0.054991963885	3.3257	0.061690216068	3.3228	0	0	1
5.50 %	0.049229710214	3.2904	0.069364645338	3.3241	0	0	1
5.75 %	0.074182161625	3.2772	0.055301898701	3.2919	0	0	1
6.00 %	0.060238081284	3.2638	0.045366705093	3.2940	0	0	1
6.25 %	0.060281942932	3.2484	0.064758800937	3.2592	0	0	1
6.50 %	0.054843182918	3.2354	0.053222683350	3.2300	0	0	1
6.75 %	0.047406302020	3.2137	0.047333847334	3.2103	0	0	1
7.00 %	0.067143490197	3.2024	0.051394138491	3.1927	0	0	1
7.25 %	0.051299481299	3.1628	0.067624053735	3.1614	0	0	1
7.50 %	0.053817059210	3.1380	0.066835739630	3.1604	0	0	1
7.75 %	0.058529499390	3.1152	0.054470323038	3.1225	0	0	1
8.00 %	0.062317725011	3.1084	0.042727955262	3.1223	0	0	1
8.25 %	0.071957434736	3.0853	0.057951823035	3.0860	0	0	1
8.50 %	0.056985388326	3.0799	0.052340353874	3.0859	0	0	1
8.75 %	0.049828222165	3.0629	0.071711338656	3.0739	0	0	1
9.00 %	0.072774758792	3.0448	0.043773975368	3.0356	0	0	1
9.25 %	0.056813872354	3.0463	0.067220549676	3.0284	0	0	1
9.50 %	0.060241553993	3.0277	0.062831731129	3.0146	0	0	1
9.75 %	0.070453995750	2.9954	0.062440417576	3.0134	0	0	1
10.00 %	0.065715417460	2.9940	0.060803659901	2.9761	0	0	1

Tabela 5.2: Informações de média, desvio padrão e resultados dos testes de hipótese propostos obtidas a partir de limitações na taxas de transferência de dados variando de 0 a 70kbps

Taxa de transferência	**SIP**		**IAX**		SIP > IAX	IAX > SIP	IAX = SIP
	Desvio Padrão	**Média**	**Desvio Padrão**	**Média**	TH1 H_1	TH2 H_1	TH1 TH2 H_0
0 a 32.5kbps	0.000000000000	-0.5000	0.000000000000	-0.5000	0	0	1
33.0kbps	0.000000000000	-0.5000	0.000000000000	-0.5000	0	0	1
33.5kbps	0.000000000000	-0.5000	0.451257022106	-0.3573	0	0	1
34.0kbps	0.000000000000	-0.5000	0.000000018330	0.9270	0	1	0
34.5kbps	0.000000000000	-0.5000	0.000000018330	0.9270	0	1	0
35kbps a 47kbps	0.000000018330	0.9270	0.000000018330	0.9270	0	0	1
47.5kbps	0.000000018330	0.9270	0.000000018330	0.9270	0	0	1
48.0kbps	0.000000018330	0.9270	0.000000000000	0.9640	0	1	0
48.5kbps	0.000000018330	0.9270	0.000000000000	0.9640	0	1	0
49.0kbps	0.011700427342	0.9307	0.000000000000	0.9640	0	0	1
49.5kbps	0.015600569790	0.9344	0.000000000000	0.9640	0	0	1
50kbps a 55.0kbps	0.000000000000	0.9640	0.000000000000	0.9640	0	0	1
55.5kbps	0.000000000000	0.9640	0.023611673195	0.9528	0	0	1
56.0kbps	0.000000000000	0.9640	0.000000000000	0.9640	0	0	1
56.5kbps	0.000000000000	0.9640	0.000000019245	0.9643	0	1	0
57.0kbps	0.000000000000	0.9640	0.000000019245	0.9643	0	1	0
57.5kbps	0.000000000000	0.9640	0.000000000000	0.9640	0	0	1
58kbps a 62.0kbps	0.000000010511	0.9080	0.000000010511	0.9080	0	0	1
62.5kbps	0.000000010511	0.9080	0.000000010511	0.9080	0	0	1
63.0kbps	0.000000000000	0.9010	0.000000010511	0.9080	0	1	0
63.5kbps	0.000000000000	0.9010	0.000000010511	0.9080	0	1	0
64kbps a 65.0kbps	0.000000000000	0.9010	0.000000000000	0.9010	0	0	1
65.5kbps	0.000000000000	0.9010	0.000000000000	0.9010	0	0	1
66.0kbps	0.011877148928	0.8918	0.014008330854	0.8867	0	0	1
66.5kbps	0.011877148928	0.8918	0.012227473619	0.8838	0	0	1
67.0kbps	0.000000000000	0.8780	0.000000000000	0.9070	0	1	0
68.0kbps	0.000000000000	0.8780	0.000000000000	0.9070	0	1	0
68.5kbps	0.000000000000	0.8780	0.000000000000	0.9070	0	1	0
69.0kbps	0.000000000000	0.9070	0.000000000000	0.9070	0	0	1
69.5kbps	0.000000000000	0.9070	0.000000000000	0.9070	0	0	1
70.0kbps	0.000000000000	0.9070	0.000000000000	0.9070	0	0	1

Tabela 5.3: Informações de média, desvio padrão e resultados dos testes de hipótese propostos obtidas a partir de limitações na taxas de transferência de dados variando de 70.5kbps e 100kbps.

Taxa de transferência	**SIP**		**IAX**		SIP > IAX	IAX > SIP	IAX = SIP
	Desvio Padrão	**Média**	**Desvio Padrão**	**Média**	TH1 H_1	TH2 H_1	TH1 TH2 H_0
70.5kbps	0.000000000000	0.9070	0.000000000000	0.9070	0	0	1
71.0kbps	0.000000000000	0.9070	0.000000000000	0.9070	0	0	1
71.5kbps	0.000000000000	0.9070	0.000000000000	0.9070	0	0	1
72.0kbps	0.000000000000	0.9070	0.000000010511	0.9080	0	1	0
72.5kbps	0.000000000000	0.9070	0.000000010511	0.9080	0	1	0
73.0kbps	0.000000000000	0.8730	0.000000014634	0.8740	0	1	0
73.5kbps	0.000000000000	0.8730	0.000000014634	0.8740	0	1	0
74.0kbps	0.000000000000	0.9010	0.017708754896	0.8986	0	0	1
74.5kbps	0.000000000000	0.9010	0.017708754896	0.8734	0	0	1
75.0kbps	0.000000000000	0.8760	0.000000012940	0.8770	0	1	0
75.5kbps	0.000000000000	0.8760	0.000000012940	0.8770	0	1	0
76.0kbps	0.000000000000	0.8600	0.000000000000	0.8600	0	0	1
76.5kbps	0.000000000000	0.8600	0.000000000000	0.8600	0	0	1
77.0kbps	0.000000000000	0.8650	0.006196773353	0.8372	0	0	1
77.5kbps	0.003373096170	0.8666	0.006196773353	0.8348	0	0	1
78.0kbps	0.001549193338	0.8488	0.007705697747	0.8224	0	0	1
78.5kbps	0.000948683298	0.8473	0.009295160030	0.8262	0	0	1
79.0kbps	0.000000000000	0.8350	0.004594682917	0.8060	0	0	1
79.5kbps	0.000948683297	0.8347	0.009720539536	0.8084	0	0	1
80.0kbps	0.005059644256	0.8324	0.087366660307	3.6116	0	1	0
80.5kbps	0.006196773353	0.8348	0.094732899365	3.6809	0	1	0
81.0kbps	0.007648529270	0.8325	0.134050115006	3.5991	0	1	0
81.5kbps	0.008694826047	0.8316	0.113171502105	3.7467	0	1	0
82.0kbps	0.013155480480	0.8152	0.021536532270	1.2504	0	1	0
82.5kbps	0.008451824260	0.8109	0.015606622525	1.2643	0	1	0
83.0kbps	0.068197099970	3.5898	0.000000000000	4.5000	0	1	0
83.5kbps	0.069484530652	3.5801	0.000000000000	4.5000	0	1	0
84.0kbps	0.125826335346	3.6856	0.000000000000	4.5000	0	1	0
84.5kbps	0.033304821139	3.7339	0.000000000000	4.5000	0	1	0
85.0kbps	0.832206331253	1.5524	0.000000000000	4.5000	0	1	0
85.5kbps	0.707983686252	1.5293	0.000000000000	4.5000	0	1	0
86.0kbps a 1000kbps	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1

Tabela 5.4: Informações de média, desvio padrão e resultados dos testes de hipótese propostos obtidas a partir de variações de atraso.

Atraso	**SIP**		**IAX**		SIP > IAX	IAX > SIP	IAX = SIP
	Desvio Padrão	**Média**	**Desvio Padrão**	**Média**	TH1 H_1	TH2 H_1	TH1 TH2 H_0
0 a 39.25ms	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1
39.50ms	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1
39.75ms	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1
40.00ms	0.000000000000	4.5000	0.912870929175	4.3333	0	0	1
40.25ms	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1
40.50ms	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1
40.75ms	0.000000000000	4.5000	0.000000000000	4.5000	0	0	1
41.00ms	0.084056829435	4.4676	0.000000000000	4.5000	0	0	1
41.25ms	0.064673228130	4.1901	0.000000000000	4.5000	0	1	0
41.50ms	0.088374100330	3.8148	0.128368744181	4.3395	0	1	0
41.75ms	0.058207407594	3.6010	0.102425981123	4.0727	0	1	0
42.00ms	0.103366772490	3.3570	0.082915162292	3.7258	0	1	0
42.25ms	0.033948727398	3.2883	0.074614018281	3.5123	0	1	0
42.50ms	0.045349333721	3.2403	0.099268525877	3.3166	0	0	1
42.75ms	0.042699174936	3.1117	0.016094405396	3.2699	0	1	0
43.00ms	0.212088368049	2.7668	0.057749001086	3.1854	0	1	0
43.25ms	0.162912602618	2.4103	0.054608154837	3.0655	0	1	0
43.50ms	0.104420810890	2.0088	0.051885251375	2.9283	0	1	0
43.75ms	0.125142291426	1.7006	0.212546479027	2.6278	0	1	0
44.00ms	0.234806263990	1.4501	0.104227514182	2.2087	0	1	0
44.25ms	0.932897880355	0.8146	0.118953912644	1.8923	0	1	0
44.50ms	0.000000000000	-0.5000	0.141422660721	1.6211	0	1	0
44.75ms	0.000000000000	-0.5000	0.350263933325	1.4647	0	1	0
45.00ms	0.000000000000	-0.5000	0.610620491842	-0.2661	0	0	1
45.25ms	0.000000000000	-0.5000	0.000000000000	-0.5000	0	0	1
45.50ms	0.000000000000	-0.5000	0.000000000000	-0.5000	0	0	1
45.75ms a 100ms	0.000000000000	-0.5000	0.000000000000	-0.5000	0	0	1

Tabela 5.5: Informações de média, desvio padrão e resultados dos testes de hipótese propostos obtidas a partir do número de requisições HTTP geradas por segundo pelo *Web Traffic Generator*.

Requisições por segundo	**SIP**		**IAX**		SIP > IAX	IAX > SIP	IAX = SIP
	Desvio Padrão	**Média**	**Desvio Padrão**	**Média**	TH1 H_1	TH2 H_1	TH1 TH2 H_0
1	0.283730801669	3.9649	0.235655576158	4.0163	0	0	1
2	0.333547528881	3.7696	0.393311243778	3.7847	0	0	1
3	0.501893087484	3.5748	0.492662250536	3.5971	0	0	1
4	0.331896481922	3.4243	0.278952245325	3.4263	0	0	1
5	0.470024352511	3.1989	0.504886462712	3.2200	0	0	1
6	0.501035048215	2.9058	0.625523895369	2.9908	0	0	1
7	0.550739921305	2.8235	0.530187470639	2.8502	0	0	1
8	0.459738411378	2.7688	0.409722200532	2.7781	0	0	1
9	0.454920613258	2.5268	0.532537516891	2.6270	0	0	1
10	0.481820513154	2.3326	0.654281289467	2.4288	0	0	1
11	0.473111170139	2.3486	0.566589125188	2.3957	0	0	1
12	0.637447947841	2.0581	0.472891716846	2.2181	0	0	1
13	0.483346052944	1.9598	0.705124217474	2.0149	0	0	1
14	0.506659036542	1.7860	0.489255855308	1.8704	0	0	1
15	0.510619602209	1.6040	0.507355923719	1.6580	0	0	1
16	0.463838087336	1.5287	0.444394601754	1.5383	0	0	1
17	0.421529215251	1.4455	0.655129237052	1.6124	0	0	1
18	0.365009414144	1.3683	0.577234771656	1.4164	0	0	1
19	0.257849274336	1.2624	0.615676173466	1.3465	0	0	1
20	0.252714160913	1.1710	0.312933030549	1.1965	0	0	1

Capítulo

6

Conclusões e Trabalhos Futuros

.

6.1 Considerações Iniciais

Este trabalho apresentou uma comparação entre os protocolos SIP e IAX através de uma descrição detalhada de seus funcionamentos e também da análise da qualidade do áudio em chamadas VoIP simuladas pelo *Network Simulator*.

Para a obtenção dos resultados e viabilização de uma análise dos mesmos, foi necessário o desenvolvimento de uma infra-estrutura que possibilitasse as ações desejadas. Para tanto, inicialmente foram efetuados estudos detalhados sobre o funcionamento dos protocolos e sua inserção no contexto da ferramenta NS. Na seqüência, tornou-se indispensável o entendimento e as investigações sobre métodos de avaliação de qualidade de áudio em ligações VoIP, o que determinou a escolha do método PESQ. Por fim, para tornar as simulações fiéis às situações reais, foi necessária a compreensão de algoritmos de *buffer* utilizados em dispositivos e *softwares* VoIP tradicionais.

Finalizando este estudo, este capítulo exibe na seção 6.2, conclusões obtidas a partir dos resultados apresentados pelo capítulo anterior. Em seguida, as contribuições e limitações verificadas por este trabalho são relatadas nas seções 6.3 e 6.4 respectivamente.

Finalmente, idéias sobre ampliação e desenvolvimento de trabalhos futuros são apresentados na seção 6.5.

6.2 Conclusões

A partir dos problemas de perda de pacotes, limitação da taxa de transferência, atraso e adição de ruído de fundo, foi possível a comparação das ligações simuladas pelos protocolos SIP e IAX.

Desta maneira, apesar do protocolo IAX ter conseguido notas maiores de qualidade em determinadas áreas de resultados nos problemas de atraso, limitação da taxa de transferência e *jitter*, as análises estatísticas consideraram que o áudio transportado pelos protocolos tiveram a mesma qualidade na maioria dos experimentos. Estas pequenas vantagens do IAX foram atribuídas ao uso dos *mini-frames*, estes empregam um cabeçalho 8 *bytes* menor que o RTP, protocolo de transporte utilizado pelo SIP.

De forma geral o protocolo SIP tem a mesma eficiência de transporte de voz que o IAX em termos de qualidade de voz. Apesar das dificuldades relacionadas ao uso de NAT com o RTP, o protocolo SIP apresenta uma especificação e documentação mais detalhada o que facilita o seu uso comercial e acadêmico.

6.3 Contribuições

Basicamente, este trabalho teve como principais contribuições:

- Uma ampla revisão bibliográfica sobre o funcionamento dos protocolos VoIP SIP e IAX, incluindo a busca por estudos que tinham como objetivo compará-los.
- Integração de três áreas de estudos distintas: protocolos VoIP, simulação e métodos de avaliação de qualidade de voz.
- Implantação dos protocolos VoIP SIP e IAX no contexto do simulador *Network Simulator*.
- Desenvolvimento de um aplicativo que representa o funcionamento de um cliente VoIP, associando pacotes de áudio trocados nas simulações NS a fragmentos de arquivos de áudio codificados em G.711 (*nsTraceVoip*).
- Criação de uma infra-estrutura modularizada, flexível à manipulação de cenários e parâmetros, capacitada para modificar, executar e analisar simulações NS baseadas

nos protocolos VoIP SIP e IAX. Desta forma, outras variáveis podem ser empregadas e estudadas.

6.4 Limitações

As principais limitações encontradas no desenvolvimento do projeto estão na área experimental.

Primeiramente, cita-se que a estrutura de *scripts* criada para execução das simulações não possibilita execução paralela, o que torna necessária a espera do ciclo completo para execução da próxima simulação.

Outro fator limitante empreendido aos experimentos foi torná-los dependentes do banco de dados ao *PostgreSQL*, sendo indispensável o seu uso para a realização das análises.

Por fim, a restrição encontrada na adição dos protocolos no *Network Simulator* baseia-se na não inclusão de métodos de criptografia para autenticação dos agentes SIP e IAX.

6.5 Trabalhos Futuros

Como esta pesquisa abrange várias áreas de estudo, diversas perspectivas são criadas para o desenvolvimento de trabalhos futuros.

Dentro do contexto de simulação, a elaboração e empreendimento de cenários que possibilitem a comparação dos protocolos sobre outros aspectos, inclusive a utilização de outros arquivos de áudio, seriam facilitados pela infra-estrutura que gerencia a execução das simulações. A experiência adquirida através da implantação dos protocolos SIP e IAX no *Network Simulator* possibilita a inclusão de outros protocolos VoIP e futuras análises comparativas.

Na área de qualidade de ligações, através de modificações de parâmetros é possível a comparação dos diversos tamanhos de pacotes empregados pelo codificador G.711. A adição de outros tipos de codificações ao *nsTraceVoIP* o capacitaria para a análise de codificadores de voz. Adicionalmente, a implementação de outros algoritmos de *jitter-buffer* e emprego de outras técnicas para mascarar o efeito dos pacotes perdidos também originariam novas pesquisas.

Outros métodos de medição de qualidade de áudio poderiam ser avaliados a partir de poucas mudanças no ambiente de simulação proposto, o que possibilitaria a comparação do método PESQ com outras maneiras de mensurar a qualidade do som.

Referências

[ast, 2006] (2006). Asterisk. http://www.asterisk.org/.

[Ns, 2006] (2006). The network simulator homepage. http://www.isi.edu/nsnam/ns/.

[rea, 2006] (2006). Real overview. http://www.cs.cornell.edu/skeshav/real/overview.html.

[vin, 2006] (2006). Vint project. http://www.isi.edu/nsnam/vint/index.html.

[int, 2008] (2008). Internet traffic report. http://www.internettrafficreport.com/.

[Abbasi and Prasad, 2005] Abbasi, T. and Prasad, S. (2005). A comparative study of the SIP and IAX voip protocols. In *Canadian Conference on Electrical and Computer Engineering 2005*, Saskatoon Inn.

[Baratvand et al., 2008] Baratvand, M., Tabandeh, M., Behboodi, A., and Ahmadi, A. (2008). Jitter-buffer management for voip over wireless lan in a limited resource device. *Networking and Services, 2008. ICNS 2008. Fourth International Conference on*, pages 90–95.

[Camarillo et al., 2003] Camarillo, G., Kantola, R., and Schulzrinne, H. (2003). Evaluation of transport protocols for the session initiation protocol. *Network, IEEE*, 17(5):40–46.

[Casson, 2004] Casson, H. N. (2004). *The History of the Telephone*.

[Cuervo et al., 2000] Cuervo, F., Greene, N., Rayhan, A., Huitema, C., Rosen, B., and Segers, J. (2000). Megaco Protocol Version 1.0. RFC 3015 (Proposed Standard). Obsoleted by RFC 3525.

[De Rango F., 2006] De Rango F., T. M. a. a. (2006). Overview on voip: Subjective and objective measurement methods. *International Journal of Computer Science and Network Security*, 6(1B):140–153.

[Ding and Goubran, 2003] Ding, L. and Goubran, R. (2003). Assessment of effects of packet loss on speech quality in voip. *Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. The 2nd IEEE Internatioal Workshop on*, pages 49–54.

[Fasciana, 2003] Fasciana, M. L. (2003). Ns2 - sip module. *Universidade de Palermo*.

[Franks et al., 1999] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Stewart, L. (1999). HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard).

[Furuya et al., 2003] Furuya, H., Nomoto, S., Yamada, H., Fukumoto, N., and Sugaya, F. (2003). Experimental investigation of the relationship between ip network performances and speech quality of voip. *Telecommunications, 2003. ICT 2003. 10th International Conference on*, 1:543–552 vol.1.

[Group et al., 1996] Group, A.-V. T. W., Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (1996). RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (Proposed Standard). Obsoleted by RFC 3550.

[Handley and Jacobson, 1998] Handley, M. and Jacobson, V. (1998). SDP: Session Description Protocol. RFC 2327 (Proposed Standard). Obsoleted by RFC 4566, updated by RFC 3266.

[ITU-T, 1988] ITU-T (1988). ITU-T recommendation G.711. Recommendation G.711, International Telecommunication Union.

[ITU-T, 1996a] ITU-T (1996a). ITU-T recommendation P.861. Recommendation P.861, International Telecommunication Union.

[ITU-T, 1996b] ITU-T (1996b). P.800.1 - mean opinion score (mos) terminology. Technical report, International Telecommunication Union.

[ITU-T, 1998a] ITU-T (1998a). ITU-T recommendation G.107. Recommendation G.107, International Telecommunication Union.

[ITU-T, 1998b] ITU-T (1998b). SERIES Q: Switching and Signalling Digital Subscriber Signalling System No. 1. Technical report, International Telecommunication Union.

[ITU-T, 2001] ITU-T (2001). ITU-T recommendation P.862. Recommendation P.862, International Telecommunication Union.

[ITU-T, 2004a] ITU-T (2004a). Definition of next generation network. Technical report, International Telecommunication Union.

[ITU-T, 2004b] ITU-T (2004b). ITU-T recommendation P.562. Recommendation P.562, International Telecommunication Union.

[ITU-T, 2006a] ITU-T (2006a). H.225.0 : Call signalling protocols and media stream packetization for packet-based multimedia communication systems. Technical report, International Telecommunication Union.

[ITU-T, 2006b] ITU-T (2006b). SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services. Technical report, International Telecommunication Union.

[ITU-T, 2007] ITU-T (2007). ITU-T recommendation G.729. Recommendation G.729, International Telecommunication Union.

[ITU-T, 2008] ITU-T (2008). H.245 Control protocol for multimedia communication. Technical report, International Telecommunication Union.

[Kaliski and Staddon, 1998] Kaliski, B. and Staddon, J. (1998). PKCS #1: RSA Cryptography Specifications Version 2.0. RFC 2437 (Informational). Obsoleted by RFC 3447.

[Kessler, 1990] Kessler, G. C. (1990). *ISDN Concepts, Facilities, and Services*. McGraw-Hill Communications Series.

[Kevin Fall, 2007] Kevin Fall, K. V. (2007). The ns manual. `http://isi.edu/nsnam/ns/doc/ns_doc.pdf`.

[Lulling and Vaughan, 2006] Lulling, M. and Vaughan, J. (2006). A simulation-based comparative evaluation of transport protocols for sip. *Computer Communications*, 29(4):525–537.

[Rekhter et al., 1996] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and Lear, E. (1996). Address Allocation for Private Internets. RFC 1918 (Best Current Practice).

[Rivest, 1992] Rivest, R. (1992). The MD5 Message-Digest Algorithm. RFC 1321 (Informational).

[Rix et al., 2001] Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P. (2001). Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *ICASSP '01: Proceedings of the Acoustics, Speech, and Signal Processing, 200. on IEEE International Conference*, pages 749–752, Washington, DC, USA. IEEE Computer Society.

[Rosenberg et al., 2002] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002). SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard). Updated by RFCs 3265, 3853, 4320, 4916, 5393.

[Rosenberg et al., 2003] Rosenberg, J., Weinberger, J., Huitema, C., and Mahy, R. (2003). STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). RFC 3489 (Proposed Standard). Obsoleted by RFC 5389.

[Roychoudhuri et al., 2002] Roychoudhuri, L., Al-Shaer, E., Hamed, H., and Brewster, G. (2002). On studying the impact of the internet delays on audio transmission. *IEEE Workshop on IP Operations and Management*, pages 208–213.

[Schulzrinne et al., 2003] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (2003). RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard).

[Schulzrinne et al., 1998] Schulzrinne, H., Rao, A., and Lanphier, R. (1998). Real Time Streaming Protocol (RTSP). RFC 2326 (Proposed Standard).

[Spencer and Miller, 2006] Spencer and Miller (2006). Iax2: Inter-asterisk exchange version 2. `http://www.ietf.org/internet-drafts/draft-guy-iax-02.txt`.

[Tanenbaum, 1999] Tanenbaum, A. S. (1999). *Computer Networks*. Prentice Hall, 4th edition edition.

[W.C.Shefler, 1988] W.C.Shefler (1988). *Statistics: Concepts and Applications*. The Benjamin/Cummings.

[Zeadally and Siddiqui, 2004] Zeadally, S. and Siddiqui, F. (2004). Design and implementation of a sip-based voip architecture. In *AINA 04: Proceedings of the 18th International Conference on Advanced Information Networking and Applications*, page 187, Washington, DC, USA. IEEE Computer Society.

[Zhang, 2002] Zhang, Y. (2002). Sip-based voip network and its interworking with the pstn. *Electronics & Communications Engineering Journal*, 14(6):273–282.