UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ANÁLISE DE SISTEMAS DE TELEFONIA IP EM REDES PAR-A-PAR SOBREPOSTAS

Elaborado por:

Douglas Contente Pimentel Barbosa

Recife, Abril de 2008.

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ANÁLISE DE SISTEMAS DE TELEFONIA IP EM REDES PAR-A-PAR SOBREPOSTAS

por

DOUGLAS CONTENTE PIMENTEL BARBOSA

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de Mestre em Engenharia Elétrica.

ORIENTADOR: RAFAEL DUEIRE LINS, Ph.D.

Recife, Abril de 2008.

B238a Barbosa, Douglas Contente Pimentel.

Análise de sistemas de telefonia IP em redes par-a-par sobrepostas / Douglas Contente P. Barbosa. - Recife: O Autor, 2008.

xix, 166 folhas.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2008.

Inclui bibliografia e Apêndice.

1. Engenharia Elétrica. 2. Sistema de Telefonia - VoIP. 3. Redes Sobrepostas. I. Título.

UFPE

621.3 **CDD (22. ed.)** **BCTG/2008-121**

Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

DOUGLAS CONTENTE PIMENTEL BARBOSA

TÍTULO

"ANÁLISE DE SISTEMAS DE TELEFONIA IP EM REDES PAR-A-PAR SOBREPOSTAS"

A comissão examinadora composta pelos professores: RAFAEL DUEIRE LINS, DES/UFPE, VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE, e CARMELO JOSÉ ALBANEZ BASTOS FILHO, DCC/UPE sob a presidência do primeiro, consideram o candidato **DOUGLAS CONTENTE PIMENTEL BARBOSA APROVADO**.

Recife, 31 de março de 2008

EDUARDO FONTANA
Coordenador do PPGEE

RAFAEL DUEIRE LINS
Orientador e Membro Titular Interno

CARMELO JOSÉ ALBANEZ BASTOS FILHO
Membro Titular Externo

VALDEMAR CARDOSO DA ROCHA JÚNIOR
Membro Titular Interno

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

ANÁLISE DE SISTEMAS DE TELEFONIA IP EM REDES PAR-A-PAR SOBREPOSTAS

Douglas Contente Pimentel Barbosa

Abril / 2008

Orientador: Rafael Dueire Lins, Ph.D.
Área de Concentração: Telecomunicações.
Linha de Pesquisa: Redes de Computadores.
Palavras-chave: Telefonia, VoIP, QoS, Redes Par-a-Par, Redes Sobrepostas, Roteamento.
Número de Páginas: 186.

RESUMO: As redes de telefonia IP popularizaram-se nos últimos anos sobretudo por seu baixo custo e facilidade de utilização. Transmitir voz na forma de pacotes IP favorece o desenvolvimento de uma rede integrada, na qual diversos tipos de dados e mídia trafegam segundo um padrão único, que uniformize os sistemas de telecomunicações (Convergência IP). As redes sobrepostas par-a-par são parcial ou totalmente independentes de qualquer servidor centralizado, possuem alta escalabilidade e fornecem meios para que a comunicação atravesse obstáculos impostos por NATs e *firewalls*. Tais redes oferecem aos pacotes uma maior flexibilidade de roteamento, permitindo que novas estratégias sejam utilizadas no encaminhamento dos pacotes. Essas estratégias proporcionam uma melhor qualidade de voz ao usuário, principalmente durante falhas e congestionamentos.

Nesta dissertação são estudados os sistemas de comunicação de voz sobre IP (VoIP) arquitetados em topologias par-a-par sobrepostas. Aspectos de codificação, sinalização, roteamento e tráfego, bem como os protocolos envolvidos em tais sistemas são descritos. Alternativas para obter uma melhor qualidade de voz através do uso dessa configuração são analisadas. Como contribuições dessa dissertação, é realizada uma análise comparativa entre sistemas VoIP e apresentada uma nova forma quantitativa de medição da QoS (Qualidade de Serviço) baseada na correlação entre os sinais transmitidos e recebidos.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

AN ANALYSIS OF IP TELEPHONY SYSTEMS OVER OVERLAY PEER-TO-PEER NETWORKS

Douglas Contente Pimentel Barbosa

April / 2008

Supervisor: Rafael Dueire Lins, Ph.D.
Area of Concentration: Telecommunications.
Line of Research: Computer Networks.
Keywords: Telephony, VoIP, QoS, Peer-to-Peer Networks, Overlay Networks, Routing.
Number of Pages: 186.

ABSTRACT: IP telephony has become popular in the last years due to its low cost and ease of use. Voice over IP favors the development of an integrated service digital network, in which diverse types of data and media are transmitted according to a standard, unifying the telecommunications systems (IP Convergence). Overlay peer-to-peer networks are partially or totally independent of central servers, offer high escalability and allow that the communication goes through NATs and firewalls. Such systems provide high routing flexibility and more independence between application QoS and the state of the physical network. This characteristic allows the use of new strategies to foward packets, providing a better quality of voice to the end user, especially during congestions or network failures.

In this dissertation, the features of overlay peer-to-peer voice over IP communication systems are studied. The aspects of codification, signalling, routing and traffic, as well as the protocols involved in such systems are described. Alternatives to achieve better voice quality through the use of this configuration are sought. A comparative analysis between VoIP systems is presented and a new quantitative way of measuring the QoS based on the correlation between the transmitted and received signals is proposed.

Dedico este trabalho aos que me precederam e aos que virão a seguir.

“The best way to predict the future is to create it.”

Peter Drucker

Agradecimentos

Agradeço aos meus pais Severino Ramos Barbosa da Silva e Roseneide Contente Pimentel Barbosa, pelo incentivo e pelo esforço empregados em minha formação e educação. Ao meu irmão Diogo Contente Pimentel Barbosa e a todos os amigos e familiares que acreditaram no meu potencial, por torcer e estar sempre comigo. A Yllen Alves de Medeiros, pelas ilustrações que compõem esta dissertação, pelo incentivo, colaboração e companheirismo de todos os dias, mas acima de tudo pelo seu amor.

Sou grato ao meu orientador, professor Rafael Dueire Lins, pela confiança depositada em mim e por partilhar seu conhecimento e experiência. Obrigado pelas longas conversas, pelos bons conselhos (acadêmicos, profissionais e pessoais), pela paciência, compreensão e por todo o apoio. Aos componentes da banca examinadora, professores Carmelo José Albanez Bastos Filho e Valdemar Cardoso da Rocha Júnior pelas observações, contribuições e críticas construtivas.

Agradeço aos demais docentes do Grupo de Telecomunicações, professores Hélio Magalhães de Oliveira, Ricardo Menezes Campello de Souza, Márcia Mahon Campello de Souza e Cecílio José Lins Pimentel por despertarem em mim, com o seu exemplo, a vontade de me tornar um mestre e por serem motivo de orgulho para todos os seus alunos. A todos os professores do Departamento de Eletrônica e Sistemas da Universidade Federal de Pernambuco, em especial ao professor Mauro Rodrigues dos Santos, pelos ensinamentos e amizade. Muito obrigado a todos os funcionários do DES-UFPE pela convivência amigável e à CAPES pelo apoio financeiro.

Meus agradecimentos a todos os amigos – graduandos, mestrandos e doutorandos – com os quais compartilhei boa parte do meu tempo durante o desenvolvimento desta dissertação, pelas discussões, considerações e sugestões pertinentes, mas principalmente pela amizade que torna o trabalho e o dia-a-dia extrememente prazerosos.

Agradeço imensamente a Deus, engenheiro maior do universo, pela minha vida e por colocar em meu caminho pessoas tão extraordinárias. Muito obrigado a todos vocês.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

ACELP – Algebraic-Code-Excited Linear Prediction

ACK – Acknowledgement

ACR – Absolute Category Rating

ADPCM – Adaptative Differential Pulse Code Modulation

ADSL – Assymmetric Digital Subscriber Line

AMR-WB – Adaptive Multi-Rate Wideband

ARPA – Advanced Research Projects Agency

ARPANET – Advanced Research Projects Agency Network

AS – Autonomous System

ASAP – Autonomous System Aware Peer-Relay Protocol

ASCII – American Standard Code for Information Interchange

ATM – Asynchronous Transfer Mode

B-ISDN – Broadband Integrated Services Digital Network

BGP – Boarder Gateway Protocol

C/S – Client / Server

CCR – Comparative Category Rating

CCSS7 – Common Channel Signaling System 7

CELP – Code Excited Linear Predition

CMOS – Comparative Mean Opinion Score

CNG – Comfort Noise Generator

Codec – Coder / Decoder

Companding – Compressing / Expanding

COPS – Common Open Policy Service

CS-ACELP – Conjugate-Structure Algebraic-Code-Excited Linear Prediction

DCR – Degradation Category Rating

DMOS – Degradation Mean Opinion Score

DNS – Domain Name Server

DSP – Digital Signal Processor

DSVD – Digital Simultaneous Voice and Data

DTMF – Dual-Tone Multi-Frequency

DTX – Discontinuous Transmission

DWDM – Dense Wavelength Division Multiplexing

EGP – Exterior Gateway Protocol

ETSI – European Telecommunications Standards Institute

FEC – Forward Error Correction

FTP – File Transfer Protocol

GIPS – Global IP Sound

GK - Gatekeeper

GPRS – General Radio Packet Service

GQOS-LAN – Generic Quality of Service Local Area Network

GSM – Global System for Mobile Communications

GSM-FR – Global System for Mobile Communications Full Rate

GSTN-POTS – General Switched Telephone Network - Plain Old Telephone Service

GW - Gateway

HTTP – Hiper-Text Transfer Protocol

IAB – Internet Activities Body

ICANN - Internet Corporation for Assigned Names and Numbers

IETF – Internet Engineering Task Force

IGP – Interior Gateway Protocol

iLBC – Internet Low Bitrate Codec

IP – Internet Protocol

iPCM – Internet Pulse Code Modulation

IPv4 – Internet Protocol version 4

IPv6 – Internet Protocol version 6

iSAC – Internet Speech Audio Codec

ISDN – Integrated Services Digital Network

ISO – International Standards Organization

ITU-T – International Telecommunications Union – Telecommunications

JID – Jabber Identification

LAN – Local Area Network

LAR – Logarithmic Area Ratio

LD-CELP – Low-Delay Code-Excited Linear Prediction

LPC – Linear Predictive Coding

MC – Multipoint Controller

MCU – Multipoint Control Unit

MIPS – Million Instructions per Second

MIT – Massachusetts Institute of Technology

MOS – Mean Opinion Score

MP – Multipoint Processor

MPE-LTP – Multipulse Excited Long-Term Prediction

MP-MLQ – Multipulse Maximum Likelihood Quantization

MTU – Maximum Transmission Unit

NACK – Non-Acknowledgement

NAT – Network Address Translator

NPL – National Physical Laboratory

NS – Network Server

NSF – National Science Foundation

NSFNET – National Science Foundation Network

OC3 – Optical Carrier 3

OSI – Open Systems Interconnection

OSP – Open Settlement Protocol

P2P – Peer-to-Peer

PAMS – Peceptual Analysis / Measurement System

PBX – Private Branch Exchange

PC – Personal Computer

PCM – Pulse Code Modulation

PDA – Personal Digital Assistant

PESQ – Perceptual Evaluation of Speech Quality

PGP – Pretty Good Privacy

PLC – Packet Loss Concealment

POP-3 – Post Office Protocol 3

PSQM – Perceptual Speech Quality Measure

PSTN – Public Switched Telephone Network

PVP – Packetized Voice Protocol

QoS – Quality of Service

RAS – Register, Admission and Status

RELP – Residual Excited Linear Prediction

RFC – Request for Comments

RNP – Rede Nacional de Pesquisa

RON – Resilient Overlay Network

RPE-LTP – Regular Pulse Excitation Long Term Prediction

RR – Receiver Report

RSVP – Reservation Protocol

RTCP – Real-Time Control Protocol

RTP – Real-Time Transport Protocol

RTSP – Real-Time Streaming Protocol

RTT – Round-Trip Delay

SB-ADPCM – Sub-Band Adaptative Differential Pulse Code Modulation

SBC – Sub-Band Coding

SCTP – Stream Control Transmission Protocol

SDES – Source Description

SDP – Session Description Protocol

Servent – Server / Client

SID – Silence Insertion Descriptor

SIP – Sesion Initiation Protocol

SMS – Short Message Service

SMTP – Simple Mail Transfer Protocol

SNR – Signal to Noise Ratio

SNR_Q – Quantization Signal to Noise Ratio

SMTP – Simple Mail Transfer Protocol

SR – Sender Report

SSL – Secure Sockets Layer

SSLv3 – Secure Sockets Layer version 3

STUN – Simple Traversal of UDP through NATs

TCP – Transport Control Protocol

TCP/IP – Transport Control Protocol / Internet Protocol

TLS – Transport Layer Security

TTS – Text-to-Speech

TURN – Traversal Using Relay NAT

UA – User Agent

UAC – User Agent Client

UAS – User Agent Server

UDP – User Datagram Protocol

UMTS – Universal Mobile Telecommunications System

UPQ – User-Perceived Quality

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

VAD – Voice Activity Detector

VoIP – Voice over IP

VPN – Virtual Private Network

WAN – Wide Area Network

WCDMA – Wideband Code Division Multiple Access

WOFDM – Wideband Orthogonal Frequency Division Multiplexing

WWW – World Wide Web

XMPP – Extensible Messaging and Presence Protocol

1. Introdução

Os sistemas de voz sobre IP (VoIP) popularizaram-se nos últimos anos, ganhando força tanto entre os usuários domésticos quanto entre os corporativos, principalmente devido ao seu baixo custo e à sua facilidade de utilização. A qualidade da voz em tais aplicativos depende não somente da eficiência dos *codecs* (codificadores-decodificadores) utilizados, mas também do modo no qual os pacotes são roteados e encaminhados pela rede, sobretudo em momentos de congestionamento.

As redes de topologia par-a-par (*Peer-to-Peer*, P2P), até então utilizadas massivamente na implementação de sistemas de compartilhamento de arquivos, surgiram como uma alternativa para solucionar – ou ao menos minimizar – alguns dos aspectos que causam a degradação do sinal de voz. Tal topologia, baseada fortemente no conceito de redes sobrepostas, cria uma arquitetura em um nível mais alto de abstração, tornando mais fácil a solução de problemas que, em geral, são difíceis de tratar ao nível dos roteadores da rede.

O Skype [1], aplicativo proprietário lançado em 2003, é o primeiro sistema para comunicação de voz sobre IP baseado em redes P2P sobrepostas e um dos grandes responsáveis pela popularização da telefonia IP. Em grande parte devido ao seu sucesso, o uso de redes par-a-par em VoIP é um tema que tem sido alvo de inúmeros trabalhos científicos que buscam desvendar o funcionamento de tais redes e analisar o desempenho de aplicativos proprietários – principalmente do Skype – utilizadores dessa nova tecnologia.

Tais esforços se dão no intuito de desenvolver sistemas baseados nesses novos conceitos, que tragam lucros aos desenvolvedores de tecnologia e vantagens aos usuários finais dos serviços.

1.1. **Motivação**

A versatilidade das redes baseadas no *Internet Protocol*, conhecidas como redes IP tornou possível o tráfego de diferentes tipos de mídia (voz, áudio, video e imagens) através de uma rede inicialmente projetada para dados. Esse fato permitiu que, apenas alguns anos após a popularização da Internet, novos serviços surgissem e fossem oferecidos, tornando essa enorme rede ainda mais difundida.

O desafio de se estabelecer um padrão unificado numa rede integrada para os mais diversos tipos de aplicações é uma tendência das telecomunicações, batizado pelo termo convergência IP. Diante disso, sistemas de voz sobre IP prometem ser a evolução natural dos sistemas de telefonia, possibilitando a completa integração entre os sistemas fixos e móveis e o desenvolvimento de novas aplicações.

Dentre eles, os modernos sistemas de telefonia IP baseados em redes par-a-par oferecem uma série de vantagens relativas à melhoria da qualidade da voz transmitida e à habilidade de transpor restrições impostas por NATs (*Network Address Translators*) e *firewalls*, por meio de novas estratégias de roteamento. Tais sistemas são estruturados na forma de redes sobrepostas, ou seja, estruturas de comunicação situadas logicamente acima de uma infra-estrutura física, estabelecendo ligações virtuais entre seus integrantes.

Apesar disso, alguns desses sistemas operam de forma sub-ótima em relação a aspectos como encaminhamento dos pacotes e qualidade de serviço, indicando que ainda não foi explorada toda a potencialidade dos mesmos [2].

Através do entendimento de tais sistemas e do estudo de novos algoritmos de busca dos pares, roteamento dos pacotes e formação da rede, melhorias significativas podem ser implementadas e melhores níveis de qualidade de serviço prestado ao usuário final podem ser atingidos.

Pelo exposto, é natural esperar que, seguindo a tendência da convergência IP, comunicações em VoIP representem o futuro do sistema telefônico, possibilitando o surgimento de novas aplicações impossíveis de serem imaginadas no sistema de telefonia convencional. Neste universo, os sistemas VoIP com topologia par-a-par mostram-se extrememente promissores para a comunicação de sinais de voz de melhor qualidade através da Internet.

1.2. **Objetivo**

O objetivo desta dissertação é descrever os sistemas de voz sobre IP (VoIP), compreendendo os aspectos de codificação, sinalização, roteamento, tráfego e protocolos envolvidos nos mesmos. Também são apresentados o funcionamento e as principais características dos novos e promissores modelos de sistemas de telefonia IP arquitetados com topologia de rede par-a-par sobreposta. Além disso, são analisadas as possibilidades de implementação de estratégias alternativas para o roteamento dos pacotes em tais redes, assim como as vantagens oferecidas pelo uso dessa tecnologia em relação à melhoria da qualidade de voz.

Finalmente, esta dissertação poderá servir como referência básica, promovendo o estudo e a concepção de sistemas de telefonia IP com elevados níveis de QoS (*Quality of Service*) e que sejam mais robustos às inerentes variações das redes sobre as quais operam.

1.3. **Organização da dissertação**

Além deste capítulo introdutório, esta dissertação é composta de mais cinco capítulos e dois apêndices, cujos conteúdos são descritos a seguir:

Capitulo 2 – Voz sobre IP

Apresenta uma idéia geral do que são Sistemas de Voz sobre IP, sua topologia e suas vantagens e desvantagens frente aos sistemas de telefonia convencionais.

Capítulo 3 – Protocolos e sinalização

Apresenta os protocolos e os sistemas de sinalização que permitem um transporte em tempo real da voz em aplicações de telefonia IP.

Capítulo 4 – Telefonia IP em Redes Par-a-Par

Apresenta as principais topologias de rede encontradas nos sistemas distribuídos e introduz os conceitos, características e requisitos das redes sobrepostas (*overlay networks*) e das redes par-a-par (*peer-to-peer networks*).

Capítulo 5 – Análise comparativa de aplicativos VoIP

Apresenta uma análise comparativa entre três dos mais utilizados aplicativos de telefonia IP disponíveis na Internet. Apresenta ainda uma breve descrição do funcionamento do Skype, primeiro aplicativo comercial de telefonia IP da Internet arquitetado em uma topologia de rede sobreposta par-a-par.

Capítulo 6 – Conclusões e trabalhos futuros

Apresenta uma análise crítica a respeito do uso das redes sobrepostas par-a-par na melhoria da qualidade de voz em VoIP. Propõe possíveis linhas para trabalhos futuros.

Referências

Apresenta as referências que serviram como base para o desenvolvimento desta dissertação.

Apêndice A – Digitalização de voz

Apresenta as principais ferramentas matemáticas e processos envolvidos na digitalização da fala humana assim como as técnicas de codificação de voz utilizadas em VoIP.

Apêndice B – Qualidade de voz em VoIP

Apresenta os principais aspectos relativos a qualidade da voz percebida pelo usuário em um sistema de telefonia IP: as métricas empregadas, os fatores responsáveis pela degradação do sinal de voz e as técnicas utilizadas para minimizar os efeitos negativos causados pelos mesmos.

2. Voz sobre IP

2.1. Voz Sobre IP

Sinais de voz vem sendo transmitidos via Rede Pública de Telefonia (*Public Switched Telephone Network - PSTN*) há mais de 100 anos. Esse é um mercado que movimenta quase 100 bilhões de dólares anualmente [3].

Com o surgimento da Internet, a rede pública de telefonia foi utilizada como sub-rede de comunicação para o tráfego de dados, porém, por volta de 1999 o número de bits de dados transferidos igualou-se ao número de bits de voz que trafegava nos troncos, já que nos troncos a voz é codificada em PCM (*Pulse Code Modulation*) e, portanto, pode ser medida em bits [4]. Em 2002, o tráfego de dados já era dez vezes maior que o de voz [4]. Novas aplicações surgem todos os dias fazendo com que o volume de dados continue crescendo exponencialmente, enquanto que o de voz, permaneça quase estacionado (um crescimento de aproximadamente 5% ao ano) [4].

O aumento do uso e da infra-estrutura da Internet nos últimos 10 anos fez com que um promissor mercado surgisse para as operadoras de comutação de pacotes. Esse mercado, que consistia na exploração da comunicação de voz sobre suas redes de pacotes, levou ao aumento do interesse na tecnologia de Voz Sobre IP (VoIP) ou Telefonia IP, como também é conhecida. VoIP é a entrega em tempo real de voz entre dois ou mais participantes através de uma rede que utiliza o *Internet Protocol* (IP) [5].

Algumas vezes, o termo voz sobre Internet é utilizado equivocadamente como sinônimo de VoIP, já que em sua maioria, os sistemas de voz sobre IP operam sobre a Internet. No entanto, existem diversas redes IP que não fazem parte da Internet propriamente dita, tais

como redes corporativas, intranets e mesmo internets, fazendo com que essa denominação se refira a uma aplicação específica de VoIP usando a rede mundial de computadores, e não o sistema propriamente dito.

Do ponto de vista teórico, telefonia IP pode ser considerada como um membro da família de aplicações conversacionais de tempo real, como mostra a figura 2.1. Neste aspecto, é importante observar as diferenças entre as aplicações que desejam realizar uma mímise do sistema telefônico convencional, com todas as suas características de QoS, e aquelas que são simplesmente baseadas em melhor-esforço (*best-effort*) [5]. Os sistemas VoIP geralmente estão associados ao tráfego de voz em redes abertas (como a Internet), porém, tais sistemas também são utilizados em redes corporativas ou *Intranets*. Esse cenário é inclusive uma alternativa tecnológica utilizada atualmente pelas companhias telefônicas em seus *backbones*.

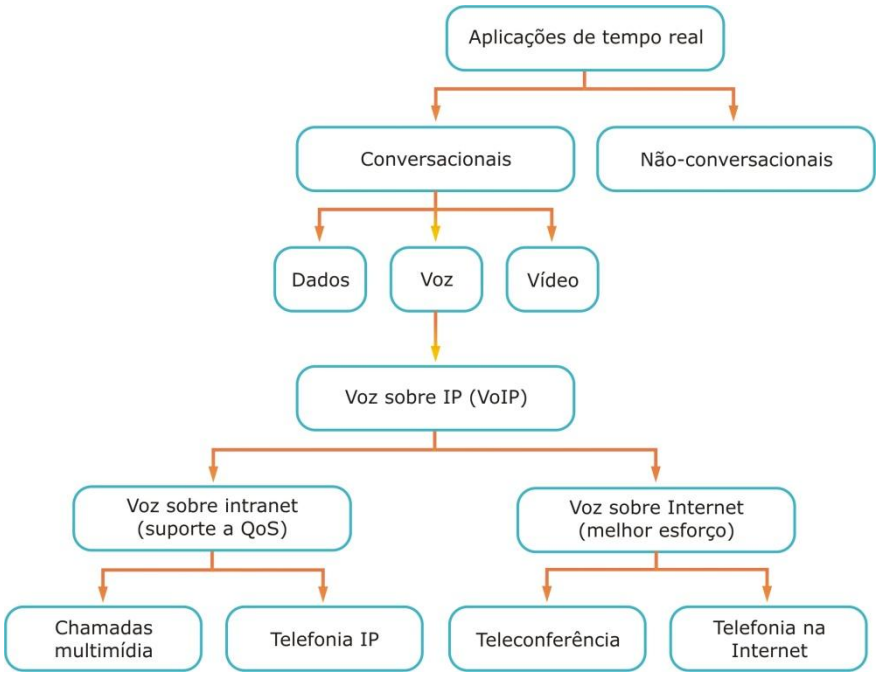


Figura 2.1 – Os *sistemas de voz sobre IP de um ponto de vista taxonômico.*

Em aplicações de telefonia IP, a voz que trafega encapsulada na forma de pacotes IP é enviada juntamente com as informações necessárias para controlar a comunicação, processo conhecido como sinalização.

Para trafegar numa rede de pacotes, a voz necessita ser processada. Para isso, o sinal analógico de voz captado no transdutor (microfone), é amostrado (discretização no tempo) e quantizado (discretização na amplitude), tornando-se um sinal digital de tempo discreto.

O sinal digitalizado é então submetido a um *codec* (codificador-decodificador) cuja função é comprimir e proteger o sinal de possíveis erros, utilizando codificação de fonte e de canal, respectivamente. Dependendo da segurança necessária, o sinal pode ser cifrado e só então encapsulado em um pacote IP para que seja transmitido através da rede.

No receptor, o processo ocorre em "sentido" contrário, ou seja, na ordem inversa em relação ao transmissor. Os pacotes recebidos são desencapsulados, decifrados, corrigidos se necessário e decodificados para que a voz original seja reproduzida no transdutor (alto-falante). Todo o processo está ilustrado na figura 2.2.

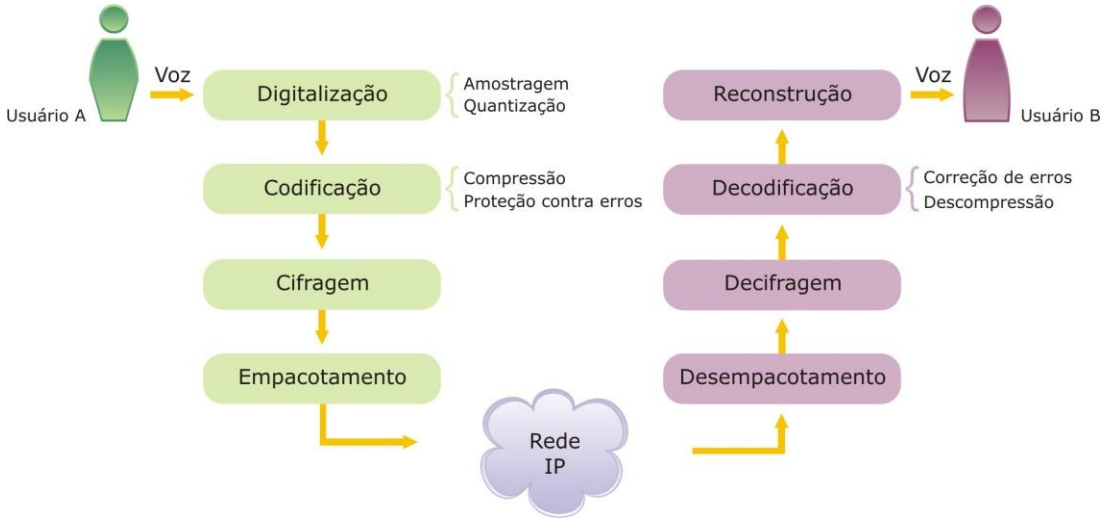


Figura 2.2 – *Etapas do processo de transmissão de voz nos sistemas de telefonia IP*.

Possivelmente, os pacotes são encaminhados segundo a estratégia de roteamento do sistema por caminhos distintos e a rede, como qualquer canal de comunicação real, introduz erros nos dados que nela trafegam. Eventualmente, alguns pacotes podem ser corrompidos, perdidos ou chegarem fora de ordem ao receptor, como mostrado na figura 2.3.

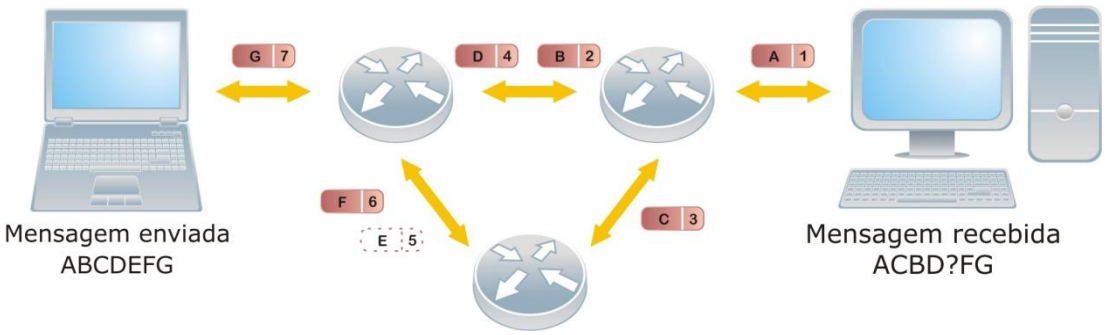


Figura 2.3 – *Roteamento dos pacotes em uma rede IP*.

2.2. **Vantagens**

Os sistemas de voz sobre IP existem desde 1995 com a apresentação do software Internet Phone, desenvolvido pela Vocaltech. Porém, foi com o lançamento do Skype em 2003, que os aplicativos de telefonia IP chamaram a atenção do mercado e dos consumidores.

A facilidade de utilização e a necessidade mínima de infra-estrutura (um software que geralmente é grátis, um PC conectado à rede e um kit multimídia) fizeram com que esses sistemas agradassem aos mais diversos clientes e tivessem uma rápida difusão. Usuários domésticos podem utilizar sua conexão à Internet (geralmente ADSL – *Asymmetric Digital Subscriber Line*) e seu computador pessoal ou *laptop* para realizar chamadas telefônicas grátis ou a baixo custo para qualquer parte do mundo. Usuários corporativos podem aproveitar toda sua infra-estrutura de rede para implantar a comunicação via VoIP, reduzindo os custos das chamadas para seus setores internos, filiais e clientes nacionais e internacionais.

O mercado de telefonia IP para produtores de hardware, desenvolvedores de software e criadores de aplicações é promissor, e ainda tem muito espaço para crescer. Os números desse mercado são divergentes, mas igualmente atraentes. Pesquisas realizadas pela *Infonetics Research* em 2005 indicam que esse mercado movimentará nos anos de 2006, 2007 e 2008 cerca de US$ 5,6 bilhões e projetam um total de 24,3 milhões de usuários residenciais de VoIP nos Estados Unidos e 27,8 milhões na Europa, conforme publicado na edição de dezembro de 2005 da revista *Info Online* [6]. Por outro lado, análises realizadas pela consultoria *Frost & Sullivan* em 2001 apontavam uma movimentação de US$ 31,8 bilhões de 2001 a 2007 [7]. O Brasil é um dos países onde o uso de VoIP cresce mais rapidamente. Em 2007, uma em cada dez empresas brasileiras optou pelo uso de VoIP em ligações de longa distância.

Algumas das vantagens dos sistemas de voz sobre IP são apresentadas a seguir.

- **Diminuição dos custos das ligações** – A adoção do sistema de voz sobre IP pode gerar ao usuário final doméstico ou empresarial uma economia média nos custos de cerca de 80% em relação às tarifas do sistema de telefonia convencional. Através do roteamento das chamadas via rede IP, estima-se que o usuário pode chegar a redeuzir em 90% os custos das ligações locais, em 60% os custos das ligações nacionais e internacionais e em 100% os custos das chamadas entre usuários VoIP.

Além disso, para conversar PC-com-PC, basta que o usuário tenha um computador dotado de caixas de som e microfone, um software específico e uma conexão IP.

- **Diminuição dos custos dos equipamentos de rede** – Os equipamentos de rede utilizados pelos sistemas de voz sobre IP, não possuem grandes diferenças em relação aos equipamentos comuns de rede (*gateways, switches, hubs,* roteadores) que são produzidos em massa para suprir o crescimento da Internet. Toda essa demanda leva a uma concorrência intensa entre os fornecedores, causando queda de preços. Por outro lado, devido ao menor número de compradores e a forte regulação do setor, os equipamentos específicos para telefonia convencional são produzidos por um número restrito de grandes fabricantes, encarecendo os preços e tornando o mercado proibitivo para empresas de pequeno porte. Embora haja uma boa interoperabilidade entre alguns desses equipamentos, tais dispositivos são normalmente desenvolvidos com tecnologia proprietária (tanto hardware quanto software), causando uma dependência do cliente com o fornecedor.
- **Facilidade de implantação devido à larga utilização do protocolo IP** - A Internet sofreu difusão exponencial nos últimos anos atingindo os mais diversos recantos do planeta. Dessa maneira, atualmente, a grande maioria das empresas possui uma rede IP, seja ela uma LAN (*Local Area Network*) ou uma WAN (*Wide Area Network*). A tendência mundial de convergência IP e a adoção da versão 6 do protocolo IP (IPv6) levará a integração dos mais variados dispositivos: *desktops, palmtops, laptops* e até eletrodomésticos. A infra-estrutura montada para acesso a Internet pode ser utilizada para telefonia IP, pois essa aplicação é considerada apenas mais um tipo de serviço, como web ou e-mail.
- **Integração entre voz e dados e novas aplicações** – Devido à operação por comutação de circuitos do sistema de telefonia convencional, para que dois usuários realizem uma comunicação de voz e troquem dados simultaneamente, são necessárias duas linhas: uma linha para acesso à Internet e outra exclusiva para tráfego de voz. Com a Telefonia IP não existe esse problema, pois voz e dados podem compartilhar a mesma conexão ao mesmo tempo, já que a rede não faz distinção entre ambos. Com uma única linha, um corretor de imóveis poderia ao mesmo tempo conversar com o cliente ao telefone, lhe enviar um e-mail com a planta-baixa de um imóvel e realizar uma busca na Internet por preços ou outras opções, por exemplo. VoIP pode permitir muito mais que isso. Possibilita, por

exemplo, que o corretor mostre em tempo real a planta 3D de um imóvel enquanto apresenta as vantagens do mesmo durante um "passeio virtual" com o cliente – integração total de voz e dados. As vantagens de se operar em um ambiente convergente de rede permitem ainda uma grande variedade de novos serviços: *e-commerce* otimizado, videoconferência, ensino a distância e outros. Tal integração é praticamente impossível nas redes de telecomunicações convencionais (telefone, fax, rádio e tv), pois as mesmas em sua forma tradicional são divergentes em termos de tecnologia, dispositivos e interface com o usuário [3].

- **Melhor aproveitamento da largura de banda** – Em uma ligação telefônica convencional é estabelecida uma conexão fim-a-fim exclusiva entre o transmissor e o receptor por onde a voz será transportada a uma taxa de 64 kbps. Toda banda disponível é utilizada por essa conexão, mesmo se ambos os usuários estiverem em silêncio simultaneamente. Através da telefonia IP, mecanismos sofisticados de compressão de voz e supressão de silêncio podem ser utilizados, permitindo a transmissão de voz a 32 kbps, 16 kbps, 6,3 kbps ou 5,3 kbps. Dessa forma, a capacidade do sistema é aumentada, já que diversos usuários podem se comunicar ao mesmo tempo compartilhando uma banda que através da telefonia convencional acomodaria uma única chamada.
- **Mobilidade** – Enquanto um terminal telefônico convencional está fisicamente "amarrado" ao local correspondente ao seu número através do cabeamento da rede de telefonia, um usuário VoIP possui mobilidade no sentido de que pode realizar chamadas com seu número em qualquer lugar do planeta com um dispositivo (PC, *laptop*, *palmtop*) conectado à Internet. Uma associação válida é a do sistema postal convencional e do *e-mail*. No sistema convencional de correio o endereçamento não possui mobilidade, pois está associado à localização física (rua, bairro, cidade), enquanto que com o *e-mail*, um usuário pode receber sua correspondência independentemente de sua localização física.
- **Mercado, lucros e empregos** – A telefonia é um dos ramos mais lucrativos do planeta. Após a popularização do VoIP, as empresas de telefonia, as concessionárias de telecomunicações e os provedores de Internet viram no mesmo uma oportunidade de obter grandes lucros. Com isso, um novo campo se abre, gerando empregos para profissionais especializados e lucros da ordem de bilhões

de dólares anuais para as empresas dos diversos segmentos que compõem o mercado de telefonia IP.

2.3. **Desvantagens**

Apesar de todas as vantagens dos sistemas de voz sobre IP, existem desafios reais que deverão ser vencidos para que o mesmo se estabeleça como o futuro da telefonia. Dentre eles pode-se destacar:

- **Falta de padronização de protocolos** – Não existe um padrão único de sinalização e tansporte de voz sobre IP. Para realizar a sinalização (estabelecimento e controle de sessões) e o transporte de mídia, são utilizados diversos padrões distintos. Dentre eles, os mais utilizados são o SIP (*Session Initiation Protocol*) e o padrão H.323 [8, 9, 10, 11]. Os dois sistemas encontram-se bastante disseminados na Internet e diferem na filosofia e no modo de operação. Embora sejam incompatíveis, existem formas de realizar a interoperação entre os mesmos enquanto não é definido um padrão geral para VoIP [5].
- **Confiabilidade e disponibilidade da rede** – Para competir em nível de igualdade com o sistema de telefonia convencional, os sistemas de voz sobre IP devem atingir o grau de disponibilidade conhecido por “cinco noves” (99,999% disponível) demandado pelo serviço de telefonia e oferecido atualmente pela telefonia convencional PSTN. As redes IP estão sujeitas a falhas, principalmente devido aos aspectos de segurança, quedas de energia e problemas em seus servidores. Além disso, não há nenhuma garantia *a priori* de que um pacote IP chegará corretamente ao seu destino. No entanto, tem sido demonstrado que é possível conceber sistemas de telefonia IP tão confiáveis quanto as plataformas comutadas a circuitos da PSTN através das técnicas de roteamento adaptativo e por múltiplos caminhos [2, 12, 13].
- **Segurança** – Devido às redes IP não serem exclusivamente dedicadas para VoIP, tais sistemas estão sujeitos às mesmas vulnerabilidades de segurança que qualquer outra aplicação. Ataques aos servidores de VoIP podem resultar na perda total do serviço telefônico, roubo de dados dos usuários e utilização ilegal por parte de terceiros. Criminosos podem utilizar artifícios como o *caller ID spoofing* (camuflagem ou ocultação da identidade de quem realiza a chamada) para cometer

fraudes. O serviço de telefonia pode ainda ser uma porta de entrada para vírus e mais um meio de proliferação de *spams*, o que pode vir a derrubar o sistema.

- **Qualidade de Voz** – O protocolo IP não foi inicialmente projetado para tráfego de voz em tempo real. Quando transportado via IP, o fluxo de voz está sujeito aos mesmos fatores degradantes que qualquer outro fluxo de pacotes de dados da rede. O ouvido humano é bastante exigente em relação à qualidade da conversação, estabelecendo restrições e limites para as perdas e atrasos dos pacotes de voz. Com isso, um dos principais desafios dos sistemas VoIP é manter uma qualidade de voz nas ligações semelhante a dos sistemas de telefonia convencional. Este aspecto encontra-se parcialmente resolvido, por meio de diversos desenvolvimentos e melhorias realizados nos *codecs*, protocolos e equipamentos. Projeções indicam que num futuro próximo, seja possível realizar uma chamada VoIP com uma qualidade similar ou mesmo superior à de uma chamada via PSTN [7].

A telefonia convencional móvel e fixa tem avançado rapidamente no sentido de usar redes de computadores comutadas a pacotes como sub-redes, substituindo a infraestrutura de comunicação por centrais digitais. A qualidade do serviço em tais redes é obtida através do uso "prioritário" de tal rede para o tráfego telefônico.

Por exemplo, a segunda geração do sistema de telefonia celular GSM (*Global System for Mobile Communications*) através do GPRS (*General Packet Radio Service*), aloca janelas de tempo (*time-slots*) ociosas do canal de voz para a comunicação de dados, liberando os mesmos caso haja voz a ser transmitida [14]. O sistema de telefonia móvel celular de terceira geração UMTS (*Universal Mobile Telecommunications System*) é, por definição, um sistema que integra voz e dados. A comunicação entre alguns dos componentes de sua sub-rede é realizada via pacotes ATM (*Assynchronous Transfer Mode*) [15, 16].

3. Protocolos e sinalização

A principal função de um sistema de comunicação é permitir que uma mensagem gerada por uma fonte de informação possa ser entregue corretamente em seu destino [17]. As redes de computadores, um dos mais importantes sistemas de comunicação modernos, são estruturadas em camadas independentes e com funções bem definidas segundo um modelo de referência.

Para executar suas funções de forma adequada, cada camada deve obedecer um conjunto de regras conhecidas e que estejam de comum acordo entre todas as máquinas que participam da rede. Tais regras, conhecidas como protocolos, definem o modo de operação do sistema, estabelecem quais procedimentos devem ser tomados a cada instante, e determinam o que deve ser feito em momentos críticos ou de conflito. Para que a comunicação ocorra, todos os elementos de um sistema devem estar em comum acordo quanto ao protocolo a ser utilizado.

Os sistemas de telefonia, por sua vez, caracterizam-se pela necessidade de inicializar, estabelecer, controlar e encerrar sessões entre usuários. Esse conjunto de procedimentos é denominado *sinalização*. O sistema de telefonia convencional possui dois tipos de sinalização: dentro da faixa (*in-band*) e fora da faixa (*out-of-band*). O sistema *in-band* possui este nome porque utiliza a mesma faixa de frequências do sinal de voz, portanto, é composto por um conjunto de tons audíveis. Desenvolvido nos anos 80 para substituir a discagem eletromecânica, esse sistema é conhecido por DTMF (*Dual-Tone Multifrequency*) e consiste em associar frequências distintas f_i a cada linha e f_j a cada coluna de uma matriz, identificando seus elementos a_{ij} através da soma $f_i + f_j$ de suas duas frequências associadas. Os tons DTMF são ouvidos quando se digita um número

telefônico num teclado. Esses tons informam ao *switch* telefônico qual número se deseja contactar [18].

O sistema de sinalização *out-of-band* (conhecido por sinalização em canal comum) CCSS7 (*Common Channel Signaling System 7*), também chamado de SS7 ou C7, foi desenvolvido pelo ITU-T para aumentar a eficiência do sistema de telefonia [19]. O SS7 é uma rede a parte cujas funções são estabelecer, configurar, monitorar, rotear e encerrar chamadas na PSTN. O SS7 serviu de base para os modernos sistemas de sinalização SIP e H.323 utilizados na telefonia IP, pois também é baseado em pacotes, implementado em software e opera independentemente do serviço de transporte propriamente dito (PSTN) [18].

Neste capítulo serão descritos o modelo de referência da rede IP, bem como os principais protocolos e esquemas de sinalização utilizados nos sistemas de telefonia IP.

3.1. **TCP/IP**

O desenvolvimento do protocolo TCP/IP está intimamente atrelado ao nascimento e desenvolvimento da Internet. No final da década de 1950, auge da Guerra Fria, o departamento de defesa dos estados Unidos sentiu a necessidade de desenvolver uma rede de comunicação distribuída, capaz de resistir a ataques nucleares. Esta rede descentralizada deveria ser flexível e capaz de se adaptar a mudanças repentinas e manter as conexões intactas enquanto as máquinas de origem e destino estivessem em funcionamento, mesmo se parte do hardware da sub-rede fosse destruído.

A razão para essa preocupação era que todo o sistema de comunicação norte-americano baseava-se em centrais telefônicas hierarquizadas com pouca redundância entre si. Tal vunerabilidade permitiria que um ataque a algumas centrais interurbanas importantes fragmentasse todo o sistema em algumas ilhas isoladas umas das outras [4].

Anos mais tarde e após alguns contratempos, o projeto foi levado adiante com a criação da ARPA (*Advanced Research Projects Agency*). Em 1967, baseado nas idéias de Paul Baran que havia em 1960 proposto um sistema distribuído de comutação de pacotes tolerante a falhas e no sucesso da implementação de uma rede com tais características no NPL (*National Physical Laboratory*) na Inglaterra, percebeu-se que a contrução de tal sistema era viável. Em 1969 entrava em operação a ARPANET, uma rede experimental ligando quatro centros de pesquisa com muitos computadores incompatíveis: Instituto de

Pesquisa de Stanford (SRI), Universidade de Utah e Universidade da Califórnia em Los Angeles (UCLA) e Santa Barbara (UCSB). A ARPANET cresceu rapidamente e em quatro anos já cobria todo o território norte-americano.

Apesar do sucesso da ARPANET, seus protocolos mostraram-se ineficientes para a execução de redes múltiplas. As intensas pesquisas sobre protocolos culminaram em 1974 no desenvolvimento do TCP/IP (*Transmission Control Protocol / Internet Protocol*), protocolo destinado a manipular a comunicação sobre inter-redes que acabou tornando-se um modelo de referência em redes de computadores e permitiu que mais redes se conectassem à ARPANET. Porém, era necessário um contrato com o Departamento de Defesa para poder se conectar a ARPANET.

No final da década de 1970, percebendo o enorme impacto nas pesquisas causado pelo compartilhamento de dados entre os centros de pesquisa através da ARPANET, a NSF (*National Science Foundation*) desenvolveu a NSFNET, com a proposta de ser uma rede aberta a todos os grupos de pesquisa universitários.

O Departamento de Ciência da Computação da Universidade de Berkeley realizou um melhoramento do sistema operacional UNIX, incorporando no mesmo a pilha de protocolos TCP/IP. A sua utilização com sucesso na ARPANET e sua distribuição gratuita nas universidades americanas fez com que a NSFNET fosse a primeira WAN puramente TCP/IP, já que seus computadores executavam desde o início este protocolo.

A NSFNET construiu inicialmente uma rede de *backbone* entre seis centros universitários equipados com supercomputadores: San Diego, Boulder, Urbana-Champaign, Pittsburgh, Ithaca e Princeton, e conectava-se a ARPANET através de um *link* na central de processamento de dados de Carnegie-Mellon. Em 1º de janeiro de 1983, o TCP/IP tornou-se o único protocolo oficial do sistema ARPANET-NSFNET, promovendo um crescimento exponencial do número de conexões de máquinas e usuários e fazendo com que os mesmos passassem a ver esse conjunto de redes como uma só inter-rede (a Internet) [4].

Pode-se dizer então que de certo modo o TCP/IP foi um dos grandes responsáveis pelo surgimento da Internet e é hoje o protocolo executado na maioria esmagadora das redes de computadores em todo o mundo. O TCP/IP é um protocolo aberto, e não existe uma instituição responsável por ele, no entanto, organismos como o IAB (*Internet Activities Board*) coordenam os esforços de pesquisa na área por meio de diversos grupos como o IETF (*Internet Engineering Task Force*) através de especificações que detalham um

conjunto de padrões de comunicação entre as máquinas descritos e apresentados ao público em RFCs (*Requests for Comments*) [20].

3.1.1. O modelo de referência TCP/IP

O modelo de referência TCP/IP surgiu como uma descrição de um conjunto de protocolos que já se encontravam em operação prática na ARPANET, modelando-se perfeitamente a esses protocolos, mas de pouca utilidade para a modelagem de redes que não utilizem o TCP/IP [4]. Por isso, neste tópico será frequentemente traçada uma relação comparativa entre o TCP/IP e o modelo de referência de sete camadas OSI (*Open Systems Interconnection*) desenvolvido pela ISO (*International Standards Organization*). Tal relação facilita o entendimento do TCP/IP, já que o OSI, ao contrário do TCP/IP, é um modelo bastante genérico e flexível que descreve a funcionalidade das camadas com bastante clareza e didática [4]. O modelo de referência TCP/IP divide-se em quatro camadas: aplicação, transporte, inter-redes e *host-rede*, sendo cada uma delas responsável por um aspecto da comunicação.

Para fins didáticos, devido à falta de especificação da camada *host-rede* do TCP/IP e da pouca utilização prática das camadas de sessão e apresentação do modelo OSI/ISO, alguns autores renomados como Tanenbaum [4] adotam o modelo de referência híbrido de cinco camadas baseado nos modelos OSI/ISO e TCP/IP mostrado na figura 3.1.

O modelo TCP/IP se divide em quatro camadas. São elas:

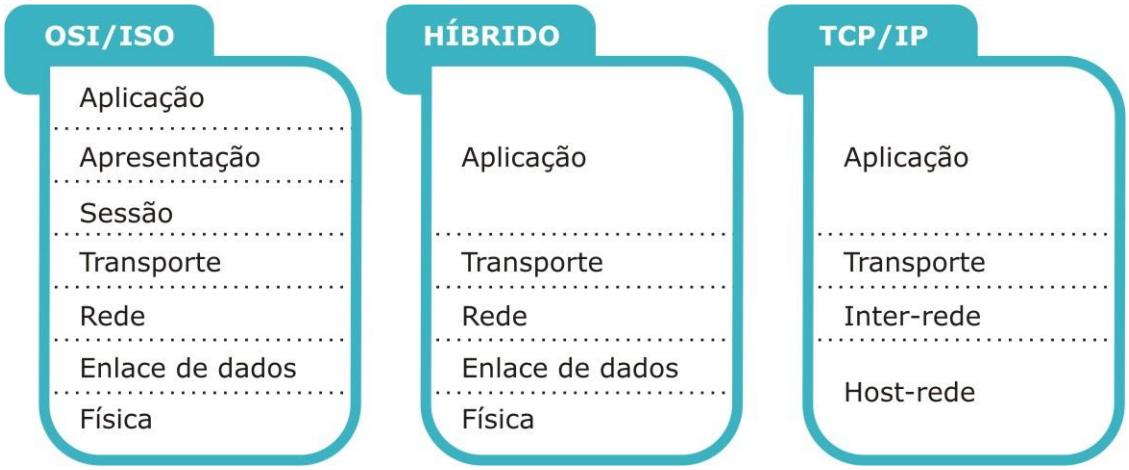


Figura 3.1 – *Modelos de referência.*

a) **Camada de aplicação** – A camada de aplicação é a camada de mais alto nível do modelo TCP/IP. Nesta camada estão os programas dos usuários e todos os protocolos de alto nível: terminal virtual – TELNET, transferência de arquivos –

FTP, correio eletrônico – SMTP, buscador de páginas na *World Wide Web* (WWW) – HTTP, DNS (*Domain Name Server*) e outros. A camada de aplicação do TCP/IP corresponde aproximadamente às camadas de aplicação, apresentação e sessão do modelo OSI/ISO. No modelo OSI/ISO a camada de sessão é responsável pelo controle de diálogo, sincronização e gerenciamento de *token*, permitindo que usuários em máquinas diferentes estabeleçam sessões entre si. A camada de apresentação trata da sintaxe, semântica e codificação dos dados que serão entregues à camada de apresentação. No entanto, a experiência prática mostrou estas camadas eram pouco usadas na maioria das aplicações e suas funções poderiam ser facilmente executadas pela camada de aplicação, portanto, não havia muita necessidade de estarem presentes no modelo TCP/IP.

b) **Camada de transporte** – A camada de transporte tem como objetivo estabelecer uma comunicação fim-a-fim entre as entidades pares dos *hosts* origem e destino exatamente como no modelo OSI/ISO. O tipo de serviço (orientado à conexão ou sem conexão) que deve ser fornecido à camada de apresentação também é de responsabilidade da camada de transporte através dos seus dois protocolos: o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*). O TCP é um protocolo orientado a conexões confiável que cuida do controle e da entrega de um fluxo de bytes ordenados e livre de erros a qualquer máquina da inter-rede, utilizado em aplicações como a transferência de arquivos. O UDP é um protocolo não-confiável sem conexão, destinado a aplicações onde não é necessário um controle de fluxo nem a manutenção da ordem das mensagens transmitidas. O UDP é utilizado em consultas cliente/servidor com esquema de solicitação/resposta e em aplicações onde a entrega imediata é mais importante que a entrega precisa como em aplicações de multimídia em tempo real (voz e vídeo). Os protocolos TCP e UDP serão detalhados nas seções 3.3 e 3.4.

c) **Camada inter-redes** – Equivalente à camada de rede do modelo OSI/ISO, a camada inter-redes do TCP/IP tem como função permitir que os *hosts* injetem pacotes na rede e garantir que estes pacotes trafegarão independentemente até o seu destino (possivelmente em outra rede). O TCP/IP define um formato padrão de pacote e um protocolo responsável pelo roteamento dos pacotes na rede e pela entrega dos mesmos ao seu destino: o IP (*Internet Protocol*). Os pacotes IP são datagramas que trafegam independentemente uns dos outros através da rede e

podem chegar ao destino fora da ordem na qual foram enviados, necessitando algumas vezes que as camadas mais altas realizem a reordenação dos mesmos. A camada inter-redes é o coração do TCP/IP, sendo responsável pelas características de integração, flexibilidade e descentralização da arquitetura. O protocolo IP será descrito com maiores detalhes na seção 3.2.

d) **Camada host-rede** – A camada *host-rede* não é bem especificada no modelo TCP/IP, deixando um grande vácuo abaixo da camada inter-redes. O modelo especifica somente que o *host* deve se conectar a rede através de algum protocolo (que não é definido) para que seja possível transmitir e receber pacotes IP [4]. No entanto, pela localização da camada *host-rede*, se pode conjecturar que ela deve comportar as funções das camadas física e de enlace de dados do modelo OSI/ISO. A camada de enlace de dados tem por função segmentar o fluxo bruto de dados em quadros, controlar o seu fluxo, corrigir os possíveis erros e gerenciar o acesso ao meio físico compartilhado de transmissão da rede (canal). A camada física é responsável pela transmissão de bits brutos através do canal de comunicação, definindo o meio físico (ar livre, fibra óptica, cabo coaxial, par trançado), os padrões elétricos (níveis de tensão, modulação, freqüência da portadora, codificação, duração do bit) e mecânicos (tipo de conector, número de pinos) para que a comunicação seja realizada.

3.2. **O protocolo IP**

O protocolo IP (*Internet Protocol*) [21] é o elemento que mantém a inter-rede unida. Projetado com a função de interligar redes, o IP deve transportar da melhor forma possível os datagramas através da rede desde a origem até o seu destino. Baseado numa estratégia de melhor esforço (*best-effort*), o IP é responsável pelo encaminhamento dos pacotes nó a nó através de um ou mais elementos de rede (roteadores, *switches* e *gateways*).

O protocolo IP oferece um serviço sem conexão, ou seja, os pacotes são encaminhados pela rede independentemente uns dos outros, podendo seguir rotas diferentes, ser recebidos fora de ordem ou mesmo perdidos. Não há mecanismo de controle para assegurar a sequência dos datagramas ou garantir a entrega dos mesmos, portanto, um pacote IP pode ser reproduzido, perdido, atrasar-se ou ser entregue com problemas sem que o serviço detecte esse fato ou reporte-o ao transmissor ou receptor [20]. O IP pode ainda realizar a

fragmentação de datagramas longos para garantir a interoperabilidade em redes que apresentam melhor desempenho com pacotes menores.

Cada elemento de rede numa rede IP é identificado por um endereço IP único de 32 bits. Diz-se que uma máquina está conectada à uma rede IP se ela executa a pilha de protocolos TCP/IP, possui um endereço IP e é capaz de trocar pacotes IP com qualquer outra máquina da rede [4]. O protocolo IP em sua versão 4 define um formato de pacote dividido em cabeçalho e texto mostrado na figura 3.2.

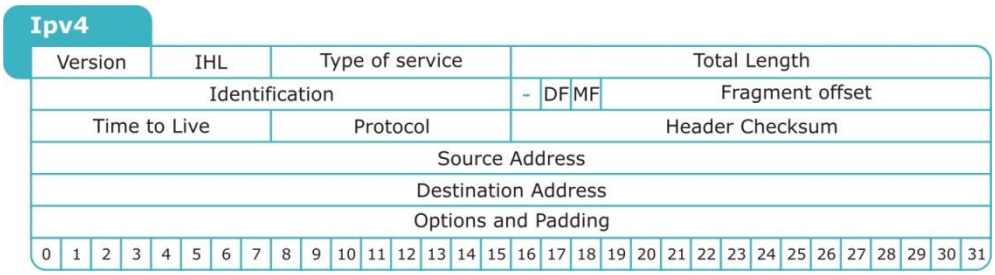


Figura 3.2 – *Cabeçalho do pacote IPv4.*

O pacote é transmitido na notação *big-endian*, da esquerda para a direita, com o bit mais significativo do campo Version aparecendo primeiro. O cabeçalho possui uma parte fixa com 20 bytes de comprimento e uma parte de tamanho variável que pode ter até 40 bytes de comprimento, perfazendo um valor máximo de 60 bytes para o cabeçalho do datagrama IP. Seus componentes serão descritos a seguir:

Version – Possui quatro bits e controla a versão do protocolo, que atualmente encontra-se em transição do IPv4 para o IPv6;

IHL – Possui quatro bits e informa o tamanho do cabeçalho em palavras de 32 bits;

Type of Service – Possui oito bits e indica a classe à qual o serviço pertence, informando ao roteador a melhor combinação entre confiabilidade e velocidade para o tipo de dado que está sendo transportado. Os roteadores atuais simplesmente ignoram este campo;

Total Length – Possui 16 bits e informa o comprimento total do datagrama até um máximo de 65.535 bytes;

Identification – Possui 16 bits e informa ao receptor à qual datagrama pertence um fragmento recém-chegado. Todos os fragmentos de um mesmo datagrama possuem um mesmo valor no campo identification;

DF – Este bit (*Don't Fragment*) na verdade é um *flag* indicando que este datagrama não deve ser fragmentado;

MF – Este bit (*More Fragments*) também é um *flag* indicando que ainda existem mais fragmentos do datagrama por serem recebidos. Todos os fragmentos com excessão do último possuem esse *flag*;

Fragment Offset – Possui 13 bits e informa a que ponto do datagrama atual o fragmento pertence;

Time to Live – Possui oito bits é um contador utilizado para limitar a vida útil dos pacotes e evitar que os mesmos fiquem vagando indefinidamente. Esse contador é decrementado a cada salto (*hop*) e enquanto o pacote é enfileirado por um longo tempo em um roteador. Este campo possui uma duração total de 255 saltos e quando seu valor chega a zero o pacote é descartado e uma mensagem de advertência é enviada ao transmissor;

Protocol – Possui oito bits e informa a qual serviço de transporte (TCP, UDP ou outros) o datagrama deve ser entregue;

Header Checksum – Possui 16 bits e contém uma soma de verificação contabilizada a cada salto para a detecção de erros no cabeçalho;

Source Address – Possui 32 bits e contém o endereço IP da aplicação que originou do datagrama;

Destination Address – Possui 32 bits e contém o endereço IP da aplicação de destino do datagrama;

Options – Possui de 0 a 40 bytes e foi criado para que versões posteriores do protocolo adicionassem funcionalidades inexistentes no projeto inicial. Algumas das opções definidas são mostradas na tabela 3.1 [4].

Tabela 3.1 – *Opções do IPv4.*

Opção	Descrição
Security	Especifica o nível de segurança do datagrama
Strict Source Routing	Mostra o caminho completo a ser seguido pelo datagrama
Loose Source Routing	Apresenta uma lista de roteadores pelos quais o datagrama obrigatoriamente deve passar
Record Route	Faz com que cada roteador anexe seu endereço IP ao datagrama
Timestamp	Faz com que cada roteador anexe seu endereço IP e seu timbre de hora ao datagrama

3.2.1. **Endereços IP e NATs**

Um endereço IP é um código hexadecimal de 32 bits representado usualmente em notação decimal com o valor de cada byte separado por pontos. Os endereços IP vão do 0.0.0.0 ao 255.255.255.255, totalizando mais de quatro bilhões de possibilidades. O

endereços IP são formados pelo par <endereço de rede | endereço do *host*>, ou pela tripla <endereço da rede | endereço da sub-rede | endereço do *host*> e dado que o espaço de endereços é finito existe um compromisso entre o número de sub-redes e a quantidade de *hosts* em cada uma delas.

Os endereços IP são atribuídos pela ICANN (*Internet Corporation for Assigned Names and Numbers*), uma instituição sem fins lucrativos e a princípio cada interface de rede possui um endereço IP que é único. Porém, com o crescimento exponencial da Internet, do número de usuários e das aplicações que operam sobre redes IP, os endereços IP tornaram-se escassos e novas estratégias tiveram que ser desenvolvidas [4].

Inicialmente, foi realizada a atribuição dinâmica de endereços na qual o usuário recebia um endereço IP enquanto se mantinha conectado e o liberava após sua desconexão. Esta solução poderia servir para usuários domésticos esporádicos mas não para usuários corporativos e servidores que desejavam manter-se conectados durante todo o dia.

Num contexto quase que emergencial, o NAT (*Network Adress Translator*) apareceu como uma solução de curto prazo e rápida implementação. O NAT é um processo de mapeamento que se aproveita da estrutura dos protocolos de transporte TCP e UDP para ampliar virtualmente o número de endereços IP disponíveis. Os pacotes TCP e UDP possuem no seu cabeçalho dois campos de 16 bits de comprimento chamados porta de origem e porta de destino**.**

Cada conexão TCP ou UDP é associada a uma porta de origem no computador local e uma porta de destino no computador remoto que servem como um canal virtual de comunicação entre cada processo (TCP ou UDP) local e o seu correspondente processo remoto. O NAT situa-se na fronteira entre a rede local da empresa e a Internet e recebe um endereço IP válido.

O funcionamento do NAT é ilustrado na figura 3.3. No processo de transmissão, um computador local transmite um pacote IP para o NAT, que o analisa e realiza a substituição do campo endereço de origem pelo seu endereço IP válido e do campo porta de origem do pacote TCP ou UDP por um índice que aponta para o local em sua tabela de mapeamento. Essa tabela guarda os campos endereço de origem e porta de origem iniciais do pacote. A seguir, as somas de verificação dos pacotes são recalculadas. O NAT transmite o pacote alterado ao processo remoto que recebe o pacote como se este fora transmitido pelo NAT e encaminha as respostas para o mesmo.

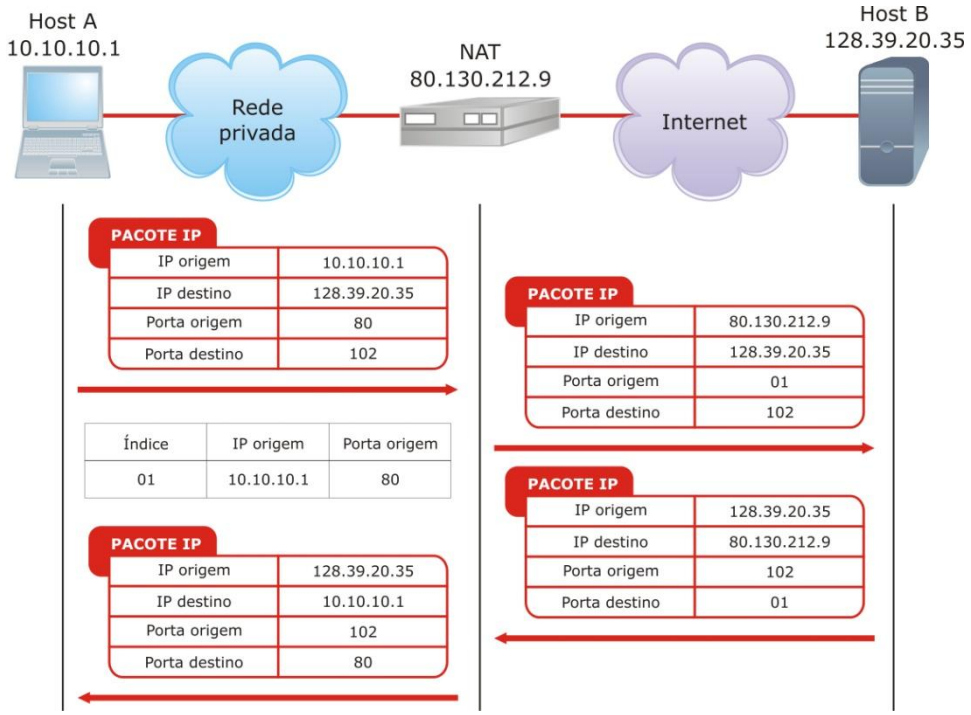


Figura 3.3 – *Funcionamento do NAT.*

No processo de recepção, o pacote IP é inspecionado pelo NAT que verifica o índice contido no campo porta de destino, identifica em sua tabela de mapeamento os valores associados armazenados como endereço de origem e porta de origem no processo de transmissão, substitui estes valores nos campos endereço de destino e porta de destino dos pacotes, recalcula as somas de verificação e transmite na rede local os pacotes para os seus verdadeiros destinos.

O NAT é uma solução muito utilizada mas também muito discutida, sendo considerada por muitos uma verdadeira aberração devido a aspectos como [4]:

a) O NAT viola o princípio da arquitetura TCP/IP no qual cada endereço IP identifica uma entidade de rede única em todo o mundo;

b) O NAT faz com que as características de rede sem conexões descentralizada sejam perdidas pela Internet, já que o não-funcionamento da caixa NAT destruirá consigo todas as conexões que estiverem sob o seu controle;

c) O NAT viola o príncípio da independência entre as camadas: alterações em uma camada não devem ser sentidas pelas demais. Caso o padrão TCP ou UDP seja alterado (por exemplo, para portas de 32 bits) o NAT deixará de funcionar;

d) O uso de NAT implica o uso de TCP ou UDP. Aplicações que utilizem outro protocolo de transporte falharão ao tentar encontrar os campos porta de origem e porta de destino;

e) Os campos porta de origem e porta de destino são de 16 bits, e as primeiras 4.096 portas estão reservadas para usos especiais. Assim o NAT pode manipular apenas 61.440 máquinas por IP válido;

f) Algumas aplicações inserem endereços IP no corpo do texto para que o receptor os extraia e os utilize. O NAT nada sabe sobre estes endereços e não os pode substituir, fazendo com que qualquer tentativa de utilizá-los no lado remoto falhe. O protocolo para transferência de arquivos FTP e o padrão de telefonia IP H.323 utilizam esse princípio e a menos que correções sejam realizadas, podem falhar na presença de NAT.

Uma solução definitiva para o problema da escassez de endereços IP promete ser o protocolo IPv6 que possui endereços de 128 bits.

3.2.2. **O protocolo IPv6**

A explosão do interesse em aplicações na Internet, o crescimento do comércio elerônico (*e-commerce*) e a proliferação dos recursos de rede levarão o espaço de endereços do IPv4 à exaustão [7], mesmo com todas as medidas paliativas, tais como o uso de NAT. Além disso, com a inevitável convergência das indústrias de informática, comunicação e entretenimento, talvez não demore muito para que cada eletrodoméstico do mundo possa se tornar acessível e comandado via Internet [4].

Para dar suporte ao ininterrupto crescimento do número de máquinas e usuários conectados à Internet e manter a atual arquitetura de endereçamento e roteamento das redes IP não basta apenas aumentar o comprimento dos endereços IP, mas permitir que sua atribuição e roteamento sejam realizadas de forma escalável. Os endereços IP de 32 bits do IPv4 permitem a manipulação de cerca de 4 bilhões de dispostivos em aproximadamente 16 milhões de redes. No entanto, o desperdício de endereços gerado pela sua divisão em classes [7] e o compromisso entre o número de sub-redes que podem ser formadas e a quantidade de *hosts* presente em cada uma delas são um incoveniente. A solução definitiva promete ser o IPv6 [22].

A questão do aumento do tamanho do endereço foi apenas uma das motivações do IPv6. O grupo que o projeta focou fortemente nesse aspecto, mas propôs também uma série de melhorias que permitem aumentar sua escalabilidade para que o problema da escassez de endereços não tenha sido simplesmente adiado para o futuro.

Este protocolo também deve se preocupar com questões ausentes no IPv4 como autoconfiguração, segurança da camada de rede, prioridade e outras. O IPv6 foi projetado para atender aos seguintes objetivos [4]:

a) Aceitar bilhões de hosts;
b) Reduzir o tamanho das tabelas de roteamento;
c) Simplificar o protocolo, permitindo que os roteadores processem os pacotes com maior rapidez;
d) Oferecer mais segurança (autenticação e privacidade) que o IPv4;
e) Dar maior importância aos aspectos de qualidade de serviço, principalmente no caso de dados de tempo real;
f) Permitir multidifusão;
g) Permitir que um *host* não precise mudar de endereço ao mudar de rede;
h) Permitir a coexistência e a compatibilidade entre protocolos novos e antigos durante vários anos;
i) Permitir que o protocolo evolua no futuro.

Após várias disputas, discussões, revisões e análises de propostas, o IETF (*Internet Engineering Task Force*) definiu o IPv6 através da RFC 2460 [22]. O protocolo desenvolvido atende a todos os objetivos propostos, preservando as boas características e descartando ou reduzindo os aspectos negativos do IP [4].

O cabeçalho do IPv6 foi otimizado para o processamento eficiente e os campos supérfluos como o de soma de verificação do cabeçalho (*Header Checksum*) foram eliminados. O campo de opções (*Options*) tornou-se mais flexível e os roteadores não podem mais fragmentar os datagramas, ou seja, apenas o *host* pode gerar fragmentos. Os roteadores e *hosts* devem determinar dinamicamente o tamanho do datagrama utilizado, e uma mensagem de erro é enviada ao *host* caso um pacote seja muito grande para ser encaminhado por um roteador. Isso se deve ao fato de ser muito mais eficiente fazer com que os *hosts* enviem pacotes com o tamanho exato que solicitar que os roteadores fragmentem os mesmos [4].

O IPv6 apresenta ainda melhorias em relação à segurança dos dados, dando suporte à aspectos de autenticidade e confidencialidade aos pacotes [7]. Em resumo, o IPv6 resolve o problema dos endereços IP, aumenta o *throughput* (desempenho do sistema como um todo), diminui o retardo no processamento devido à otimização do cabeçalho, oferece

melhor suporte para os seviços oferecidos, proporciona uma maior segurança na comunicação e uma melhor qualidade de serviço.

O cabeçalho principal do IPv6 é mostrado na figura 3.4 e composto dos seguintes campos:

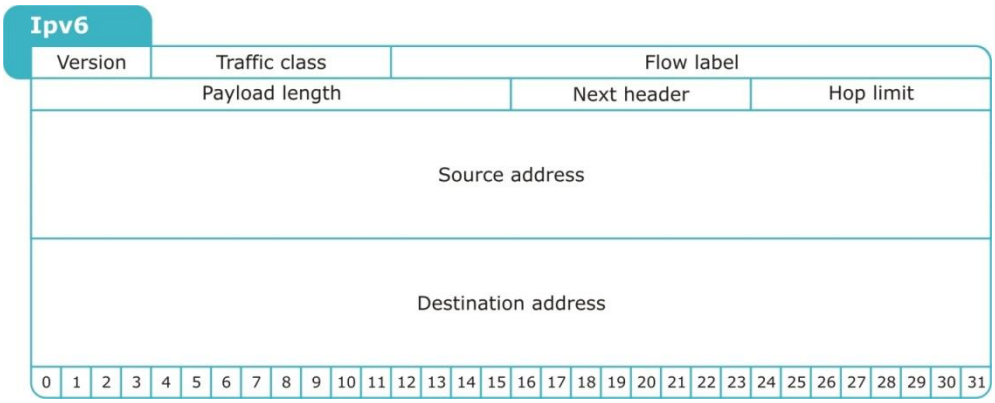


Figura 3.4 – *Cabeçalho do pacote IPv6.*

Version – Possui quatro bits de comprimento e indica a versão do protocolo (4 para o IPv4 e 6 para o IPv6). Durante o período de transição, os roteadores analisarão este campo para identificar a qual protocolo pertence o pacote.

Traffic Class – Possui oito bits de comprimento e é utilizado para fazer distinção entre pacotes com diferentes requisitos de entrega. Útil no caso de tráfego de dados em tempo real. Este campo indica o nível de prioridade relativa de um datagrama em relação aos outros. Os datagramas podem ser classificados em relação ao tipo de prioridade em dois grupos: datagramas de congestionamento controlado e datagramas de congestionamento não-controlado. Útil em momentos de congestionamento, o primeiro grupo possui oito níveis, que são indicados na tabela 3.2:

Tabela 3.2 – *Classes de tráfego do IPv6.*

Nível	Prioridade
0	Sem prioridade específica
1	Tráfego de *background* (ex. Notícias)
2	Transferência de dados não-assistida (ex. E-mail)
3	Reservado para definição futura
4	Transferência assistida em grandes quantidades (ex. Transferência de arquivos)
5	Reservado para definição futura
6	Tráfego iterativo (ex. *Login* remoto)
7	Tráfego de controle (ex. Protocolos de roteamento e gerenciamento da rede)

Os datagramas de congestionamento não-controlado não se ajustam em momentos de congestionamento da rede. Nesse tipo de tráfego estão incluídas as aplicações de

multimídia de tempo real que não podem ser atrasadas. Os níveis de prioridade de 8 a 15 são reservados para este tipo de tráfego e não possuem associações como as mostradas na tabela 3.2, mas indicam o grau de predisposição que um pacote possui para ser descartado. Como algumas aplicações são mais tolerantes à perda de pacotes que outras, o uso do indicador de prioridade auxilia a rede a tomar a decisão de qual pacote ela pode ou deve descartar em um momento crítico.

Flow Label – Possui 20 bits de comprimento. É um rótulo de fluxo que pode ser associado a fluxos de dados particulares para se garantir a qualidade de serviço (QoS) necessária. Este campo encontra-se em fase de experiência e deve permitir que a origem e o destino estabeleçam uma pseudoconexão com propriedades e necessidades específicas. Na prática é a tentativa de se ter a flexibilidade de uma sub-rede de datagramas juntamente às garantias de uma sub-rede de circuitos virtuais [4].

Payload Length – Este campo de 16 bits informa o número de bytes da carga útil do pacote IPv6, isto é, o número total de bytes do pacote menos os 40 bytes do cabeçalho.

Next Header – Este campo possui oito bits de comprimento e informa qual dos cabeçalhos de extensão opcionais (se houver algum) segue o cabeçalho principal. Os cabeçalhos de extensão já definidos são listados na tabela 3.3 [4].

Tabela 3.3 – *Cabeçalhos de extensão do IPv6.*

Cabeçaho de extensão	Descrição
Hop-by-hop options	Informações diversas para os roteadores
Destination options	Informações adicionais para o destino
Routing	Lista parcial dos roteadores a visitar
Fragmantation	Gerenciamento dos fragmentos dos datagramas
Authentication	Verificação da identidade do transmissor
Encrypted security payload	Informações sobre o conteúdo cifrado

Hop Limit – Possui oito bits de comprimento e a mesma função do campo *Time to Live* do IPv4: impedir que os pacotes fiquem vagando eternamente pela rede em caso de problemas. É um contador decrescente, decrementado a cada salto (*hop*), que ao atingir valor zero descarta o pacote.

Source Address – Endereço IP de 128 bits (16 bytes) da aplicação que gerou o datagrama.

Destination Address – Endereço IP de 128 bits (16 bytes) da aplicação à qual se destina o datagrama.

Os campos *source address* (endereço de origem) e *destination address* (endereço de destino) são a principal melhoria trazida pelo IPv6. Com o uso de endereços de 16 bytes tem-se um total de 2^{128} endereços ou aproximadamente $3 \cdot 10^{38}$. Se esse número total de endereços fosse distribuído pela área da Terra, seria possível utilizar $7 \cdot 10^{23}$ endereços IP por metro quadrado, um número maior que a constante de Avogadro ($6{,}02 \cdot 10^{23}$), muito utilizada na química. Existe, no entanto, muito desperdício no processo de alocação de endereços IP [4]. Porém, mesmo na mais pessimista das hipóteses, haverá disponibilidade de mais de 1.000 endereços IP por metro quadrado com o uso do IPv6, permitindo a conexão de eletrodomésticos à Internet e o surgimento de incalculáveis novas aplicações [4].

Apesar de todos esses aspectos, o maior desafio e talvez o grande obstáculo na implementação do IPv6 é que ele não possui retrocompatibilidade com o IPv4, fazendo com que a migração dos sistemas não seja realizada de forma direta. A principal idéia no sentido da transição do protocolo IPv4 para o IPv6 é a formação de ilhas isoladas utilizando o IPv6 comunicando-se umas com as outras através de túneis. Quando começarem a se desenvolver, essas ilhas formarão ilhas maiores e em algum momento irão se unir.

É esperado que todo o processo dure cerca de uma década [4], embora muitas redes, inclusive a RNP (Rede Nacional de Ensino e Pesquisa) já estejam aptas a operar com o protocolo IPv6 em modo nativo. Um grande número de abordagens alternativas para migração foram propostas, portanto, os sistemas de Voz sobre IP em operação devem estar capacitados para funcionar em qualquer uma das situações resultantes desse processo [5].

3.3. **O protocolo TCP**

Definido na RFC 793 [23], o TCP (*Transmission Control Protocol*) é um protocolo de transporte que oferece um fluxo de bytes fim-a-fim confiável através de uma inter-rede não-confiável. O TCP associa cada fluxo de dados a um par de portas que formam uma conexão ponto-a-ponto entre a máquina de origem e destino. Um endereço IP do host e uma porta formam um único ponto de conexão de 48 bits. Os números dos pontos terminais de origem e de destino identificam a conexão. As portas com número inferior a 1.024 são reservadas para serviços padrão e algumas delas são listadas na tabela 3.4 [4].

Tabela 3.4 – *Portas associadas às principais aplicações da Internet.*

Porta	Protocolo	Uso
21	FTP	Transferência de arquivos
23	Telnet	Login remoto
25	SMTP	Correio eletrônico
80	HTTP	Pesquisa e acesso a páginas da *World Wide Web*
110	POP-3	Acesso remoto a correio eletrônico

O TCP é capaz de se adaptar dinamicamente às propriedades da inter-rede e ser robusto diante dos muitos tipos de falha que podem ocorrer [4]. O TCP implementa serviços de transferência confiável de dados, recuperando dados perdidos, danificados ou recebidos fora da sequência e controle de fluxo dos pacotes minimizando o retardo na trasmissão dos mesmos [24].

O TCP oferece um serviço orientado à conexão, ou seja, o dispositivo de origem deve estabelecer uma conexão com o dispositivo de destino antes dos dados começarem a ser trocados. Ambos os dispositivos devem concordar com o estabelecimento da conexão. A conexão é ponto-a-ponto e opera de forma *full-duplex*, onde os dados podem fluir em ambas as direções ao mesmo tempo de forma assíncrona.

A confiabilidade no TCP deve ser completa: todos os dados que foram transmitidos devem ser recebidos exatamente na ordem em que foram enviados. O TCP opera utilizando um protocolo de janela deslizante de tamanho variável, onde cada segmento transmitido possui um número de sequência.

Ao enviar um segmento, o transmissor dispara um cronômetro (*timer*). Ao receber corretamente esse segmento, o receptor envia ao transmissor uma mensagem de confirmação de recebimento juntamente ao número de sequência do próximo dado que espera receber. Se o cronômetro expirar antes que o recebimento do dado seja confirmado ocorre uma condição de *time-out* e o pacote é retransmitido. Quando um dispositivo solicita o encerramento da conexão, o TCP garante que todos os dados que ainda estejam em trânsito sejam entregues antes do fechamento da mesma.

As entidades do TCP trocam dados na forma de segmentos, compostos por um cabeçalho (*header*) e uma carga útil (*payload*). Como mostrado na figura 3.5, o cabeçalho do segmento TCP possui uma parte fixa de 20 bytes e uma parte de comprimento variável e é composto pelos seguintes campos:

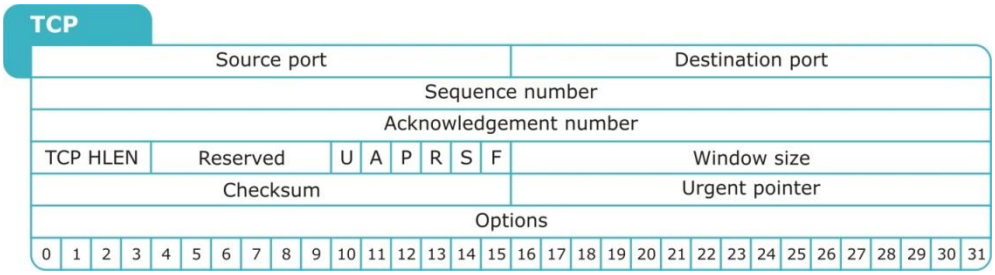


Figura 3.5 – *Cabeçalho do pacote TCP.*

Source Port – Com 16 bits de comprimento, este campo (porta de origem) é utilizado principalmente quando uma resposta deve ser devolvida à origem. Indica a porta no processo transmissor que originou a mensagem e para a qual o processo receptor deve encaminhar as possíveis respostas.

Destination Port – Com 16 bits de comprimento, este campo (porta de destino) indica para que porta do processo receptor se destina a mensagem.

Sequence Number – Possui 32 bits de comprimento e indica o número de sequência do segmento TCP.

Acknowledgement Number – Possui 32 bits e indica o número de sequência do próximo byte esperado.

TCP Header Length – Possui quatro bits de comprimento e informa o tamanho do cabeçalho em palavras de 32 bits.

URG (Urgent) – *Flag* que indica que o campo *Urgent Pointer* está sendo usado.

ACK (Acknowledgement) – *Flag* que indica que o segmento contém uma confirmação de recebimento e o campo *Acknowledgement Number* é válido.

PSH (Push) – *Flag* que indica que o receptor deve entregar os dados à aplicação assim que chegarem ao invés de acumulá-los em um *buffer* até a chegada de todos os dados da sequência.

RST (Reset) – *Flag* utilizado para reiniciar uma conexão que tenha se tornado defeituosa ou para rejeitar uma conexão ou um segmento inválido. Indica problemas na conexão.

SYN (Synchronism) – *Flag* utilizado para estabelecer conexões.

FIN (Final) – *Flag* utilizado para encerrar conexões. Indica que o transmissor não possui mais dados para enviar.

Window Size – Possui 16 bits de comprimento e é utilizado no controle eficiente do fluxo do TCP. Esse campo indica quantos bytes podem ser enviados a partir de um byte confirmado. No TCP a confirmação dos segmentos recebidos e a permissão para o envio

de novos segmentos são completamente desacopladas devido ao comprimento variável de sua janela, proporcionando ao protocolo uma flexibilidade adicional.

Checksum – É uma soma de verificação com 16 bits de comprimento utilizada para aumentar a confiablidade e detectar erros.

Urgent Pointer – No encerramento de uma conexão, indica uma quantidade de bytes a partir do número de sequência atual que devem ser urgentemente encontrados. Possui 16 bits de comprimento.

Options – Esse campo opcional composto por palavras de 32 bits oferece recursos extras ao cabeçalho do TCP. Dentre as opções definidas está o escalonamento de janela que proporciona maior flexibilidade no tamanho da janela deslizante. Outra opção bastante útil é uso de NACKs (*Non-Acknowledgements*). Normalmente, se o TCP receber um segmento defeituoso seguido de um grande número de segmentos perfeitos, o transmissor sofrerá um *time-out* e retransmitirá todos os segmentos não-confirmados, incluindo aqueles que chegaram em perfeitas condições (protocolo *go-back-N*). Através dos NACKs o receptor pode solicitar a retransmissão de segmentos específicos, reduzindo o volume de dados a serem transmitidos [4].

3.4. **O protocolo UDP**

Definido na RFC 768 [25], o UDP (*User Datagram Protocol*) é um protocolo de transporte não-confiável e sem conexão do TCP/IP, capaz de oferecer um meio para as aplicações enviarem datagramas IP encapsulados sem a necessidade do estabelecimento de conexões [4].

O UDP é utilizado em situações nas quais não é necessária uma garantia da entrega dos dados ao receptor ou quando a entrega rápida é mais importante que a entrega confiável, como nas aplicações de multimídia em tempo real (Telefonia IP, Videoconferência e outras). Como não há garantia da entrega dos dados, a aplicação que utiliza transporte sobre UDP deve assumir a responsabilidade de lidar com os problemas de perda de mensagens, duplicação, retardo, reordenação e perda de conectividade causados pela falta de confirmação de entrega dos pacotes pelo protocolo.

O UDP opera com datagramas da mesma forma do IP, sem realizar controle de fluxo, reordenação, controle de erros ou retransmissão ao detectar um segmento incorreto [4].

Segundo Comer [26]: “O UDP é um protocolo ‘fino’ porque, de modo significativo, nada acrescenta à semântica do IP”.

Na verdade, o UDP possui a vantagem de associar portas nos processos de origem e destino aos seus datagramas. As portas servem para identificar os pontos extremos nas máquinas de origem e destino, estabelecendo uma ligação entre elas através de um canal virtual de comunicação. Com o uso das portas, o UDP é capaz de multiplexar e demultiplexar diversos fluxos de dados através da rede.

Após negociar a obtenção de portas com o sistema operacional, o processo origem UDP recebe dados de diversos aplicativos, forma segmentos UDP associando números de porta diferentes a cada um desses fluxos de dados e repassa esses segmentos para que a camada de inter-redes possa transmiti-los pela rede em datagramas IP (multiplexação). Todo segmento UDP associado a uma mesma aplicação carrega um mesmo número de porta. No processo destino, o UDP aceita as mensagens recebidas pelo IP e através do número da porta encaminha os dados para o devido aplicativo (demultiplexação), como mostrado na figura 3.6.

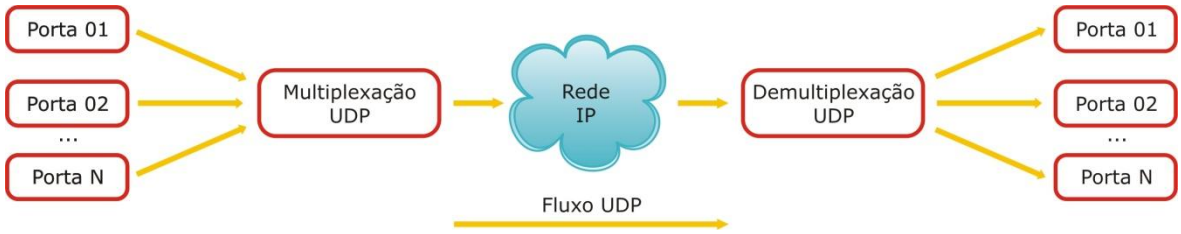


Figura 3.6 – *Multiplexação de fluxos UDP*.

É essa multiplexação baseada em porta (e não em conexão como no TCP) que permite a utilização do UDP com *multicast*, aspecto necessário para o estabelecimento de conferências de multimídia [27].

Uma messagem UDP é chamada de segmento e é composta por um cabeçalho de oito bytes seguido pela carga útil (*payload*). Conforme mostrado na figura 3.7, o cabeçalho do UDP é composto por quatro campos de dois bytes definidos como:

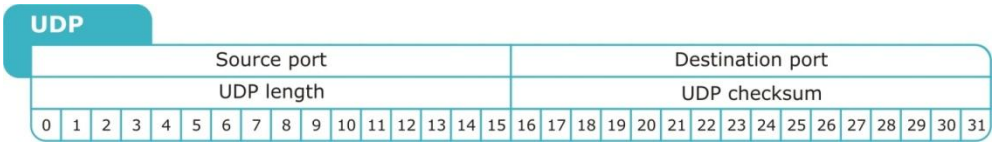


Figura 3.7 – *Cabeçalho do pacote UDP*.

Source Port – Este campo (porta de origem) é utilizado principalmente quando uma resposta deve ser devolvida à origem. Indica a porta no processo transmissor que originou a mensagem e para a qual o processo receptor deve encaminhar as possíveis respostas.

Destination Port – Este campo (porta de destino) indica para que porta do processo receptor se destina a mensagem.

UDP Length – Este campo indica o comprimento total (cabeçalho + carga útil) do segmento UDP.

UDP Checksum – É a soma de verificação do segmento UDP utilizado para a detecção de erros.

3.5. **O protocolo RTP**

O tráfego de multimídia em tempo real possui aspectos de qualidade de serviço (QoS) que o diferencia dos outros tipos de tráfego da rede e estabelece características que devem ser suportadas por um protocolo de transporte de tempo real. Dentre elas [7, 9]:

Atraso – Temporalmente, aplicações de tempo real são extremamente sensíveis e possuem fortes restrições em relação aos atrasos. O tempo total em trânsito de um pacote e o tempo de chegada entre pacotes precisa ser limitado e minimizado tanto quanto possível.

Ordenação – Os pacotes devem possuir números de sequência, permitindo que sejam reordenados em tempo real ao chegarem no destino caso cheguem fora de ordem.

Estratégia diante de perdas – Multimídia de um modo geral exige um fluxo contínuo de informação. Pacotes perdidos não podem ser simplesmente retransmitidos pois na maioria das vezes chegariam tarde demais para serem úteis [4]. A melhor estratégia nestes casos é realizar uma aproximação do valor perdido por interpolação, pois a descontinuidade gerada pela interpolação causa menos insatisfação nos usuários que a espera pela retransmissão de pacotes perdidos.

Identificação de conteúdo – Em alguns casos o reconhecimento do tipo de mídia transportada é útil para que se possa fazer ajustes em momentos críticos. Diante de congestionamentos ou em momentos em que a banda se torne escassa (a entrada de um novo usuário numa conferência que já está perto do limite de banda disponível, por exemplo), o sistema pode identificar que a codificação utilizada na mídia transportada é PCM 64 kbps e solicitar à aplicação que utilize uma codificação com taxa mais baixa como ADPCM 16 kbps.

Reconhecimento de quadro – Algumas codificações de multimídia (áudio e vídeo) são enviadas em quadros que possuem tamanhos fixos. Reconhecer o início e o fim de cada quadro é importante para que as camadas superiores consigam processá-los.

Multidifusão – O sistema deve ser fim-a-fim e dar suporte tanto a um destino único (*unicast*) quanto à multidifusão para todos os destinos (*broadcast*) ou a vários destinos (*multicast*), pois aplicações de tempo real muitas vezes envolvem interação e troca de informações entre vários usuários ou sistemas simultaneamente.

Multiplexação – O sistema deve suportar a integração de várias fontes de informação em um único fluxo de dados, pois a maioria das aplicações de multimídia agregam dados de múltiplas fontes: vídeo, áudio, legendas, dados, e outros.

Sincronismo – O sistema deve ser capaz de recontruir um ou mais fluxos de mídia simultâneos na taxa exata com a qual foram gerados, necessitando de um mecanismo que permita a sincronização dos mesmos.

Sozinhos, os principais protocolos de transporte usados na Internet (UDP e TCP) não são capazes de atender a essas restrições. Pode parecer uma boa escolha usar TCP para transportar mídia, porém existem vários problemas [3].

Em primeiro lugar, transmissão confiável não é apropriada para dados sensíveis a atrasos como áudio e vídeo de tempo-real [28]. O TCP tentará sempre reenviar um pacote perdido, causando atrasos e esperas por parte do usuário. No entanto, se mantida dentro de certos limites (até 5% aproximadamente), a perda de pacotes é aceitável e não influirá na inteligibilidade da mídia.

Em segundo lugar, o controle de congestionamento do TCP diminuirá a janela de congestionamento assim que detectar perda de pacotes, diminuindo o fluxo de pacotes recebidos. Pacotes perdidos causariam graves problemas com áudio e vídeo que necessitam de um fluxo contínuo de dados sendo enviado. Além disso TCP não suporta multidifusão [26], tornando-o inapropriado para o transporte de multimídia.

Desse modo, a melhor opção seria utilizar o UDP como protocolo de transporte e fazer com que a aplicação implementasse os aspectos e correções necessários para o tráfego de multimídia.

À medida em que as aplicações de multimídia se tornaram populares na Internet, percebeu-se que cada sistema estava reinventando aproximadamente o mesmo protocolo de tempo real. Aos poucos ficou clara a necessidade de se desenvolver um protocolo de transporte em tempo real genérico que servisse a várias aplicações [4].

Descrito na RFC 1889 [29], o RTP (*Real-Time Transport Protocol*) proporciona um serviço de entrega fim-a-fim para dados com características de tempo real tais como áudio e vídeo interativos. A função básica do RTP é multiplexar diversos fluxos de dados de tempo real em um único fluxo de pacotes UDP, proporcionando serviços de indentificação de conteúdo, numeração de sequência, monitoramento de entrega, timbre de hora e multidifusão (*multicasting*). O RTP deve agir como uma interface simples e escalável entre as aplicações de tempo real e os protocolos da camada de transporte existentes (embora seja normalmente executado sobre UDP, o RTP é independente do protocolo de transporte utilizado) [7].

A posição do RTP na pilha de protocolos é confusa e definir a qual camada esse protocolo pertence é difícil. O RTP, por definição, é inserido no espaço do usuário (camada de aplicação), porém é um protocolo fim-a-fim com a função de agregar funcionalidades que permitam que os protocolos de transporte consigam transportar dados em um ambiente de tempo real (camada de transporte). A melhor descrição do RTP o define como um protocolo de transporte implementado na camada de aplicação [4].

O RTP foi desenvolvido para operar no ambiente das redes IP onde atrasos variáveis e desordenação na entrega dos pacotes são esperados e, por mais provável que seja, nunca se tem a certeza absoluta de que um pacote chegará ao seu destino. O RTP em si não provê reserva de recursos nem mecanismos capazes de garantir a QoS dos dados de tempo real. Especificamente, o RTP não garante a entrega dos pacotes em tempo hábil nem previne a entrega desordenada dos mesmos, não realiza nenhum tipo de controle de fluxo ou de erros e não possui nenhum mecanismo de confirmação ou solicitação de retransmissão. O que o RTP efetivamente faz é proporcionar meios para que as camadas superiores identifiquem os problemas na transmissão e tomem providências para que o fluxo de dados possa ser recontruído da melhor forma possível.

Cada pacote RTP possui um número de sequência, utilizado para que a aplicação possa saber sua posição correta para inserção no fluxo de mídia e para identificar quando um pacote é perdido, realizando nesses casos uma interpolação do seu valor baseada nos conteúdos dos pacotes adjacentes. No cabeçalho do RTP existe um indentificador do tipo de codificação utilizada na mídia que ele transporta, uma vez que o conteúdo do pacote RTP pode conter amostras codificadas de várias formas possíveis pela aplicação.

O RTP permite ainda que a origem associe um timbre de hora a cada pacote transmitido em relação ao primeiro pacote do fluxo. Desse modo, o receptor pode acumular os pacotes

recebidos em um *buffer* e executá-los na cadência exata, independente do instante em que foram recebidos. O uso do timbre de hora reduz os efeitos causados pela flutuação do atraso na rede (*jitter*) e permite que diversos fluxos de mídia vindos de fontes diferentes sejam sincronizados (dois canais de áudio estereofônicos, vídeo e legendas, vídeo e áudio, dublagem em diversas línguas, etc.), mesmo que sejam transmitidos de forma bastante errática [4].

A figura 3.8 mostra o cabeçalho do pacote RTP, que é composto dos seguintes campos:

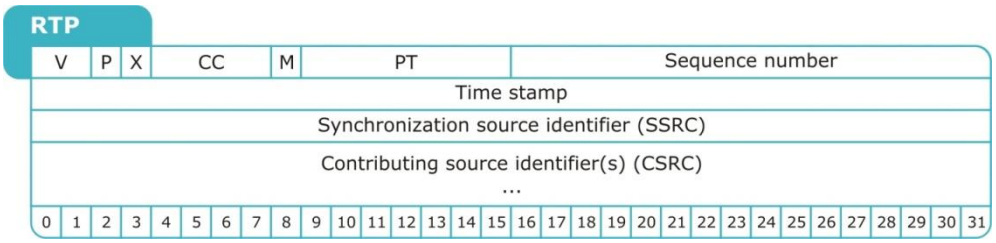


Figura 3.8 – *Cabeçalho do pacote RTP.*

Version – Versão do protocolo, possui dois bits de comprimento.

P (Padding) – Este *flag* indica que o pacote foi preenchido até possuir um comprimento múltiplo de 4 bytes. O último byte de preenchimento informa o número de bytes acrescentados.

X (Extension) – *Flag* que indica que um cabeçalho de extensão está presente.

CC (Contributing Sources Counter) – Indica o número de fontes de contribuição para o fluxo transportado, possui quatro bits de comprimento.

M (Marker) – *Flag* que introduz uma marcação para algum elemento que a aplicação reconheça. Pode indicar o início ou o fim de um quadro de áudio ou vídeo, por exemplo.

Payload Type – Este campo com 7 bits de comprimento identifica o tipo de codificação utilizada na mídia transportada, permitindo que durante a transmissão a aplicação possa alterar a codificação a qualquer momento.

Sequence Number – Esse campo de 16 bits de comprimento indica o número de sequência do pacote, permitindo que a aplicação possa reordenar e inserir o mesmo corretamente no fluxo de mídia.

Timestamp – Esse campo de 32 bits de comprimento carrega o timbre de hora gerado pela origem do fluxo de mídia, permitindo sincronização de fluxos distintos e atenuação dos efeitos do *jitter*.

Synchronization Source – Esse campo de 32 bits de comprimento informa a qual fluxo o pacote pertence, permitindo a multiplexação e demultiplexação de diversos fluxos de

dados em um único fluxo de pacotes UDP. Indica a entidade que é responsável por configurar o número de seqüência e o *timestamp* (normalmente o transmissor do pacote RTP). O identificador é escolhido randomicamente pelo transmissor e não tem qualquer vínculo com endereços de rede. Este identificador deve ser único em uma sessão e preferencialmente gerado pela aplicação.

Contributing Source (CSRC) – Usada quando os pacotes provem de um *mixer*. Indica a SSRC original que gerou a mídia e que está por trás do *mixer*. Varia de 0 a 15 entradas de CSRC em um único pacote RTP.

Os pacotes RTP possuem ainda um caractere EoP (*End of Packet*) correspondente a um byte todo nulo (0x00 em hexadecimal) para sinalizar o fim do pacote.

O RTP faz uso de dois sistemas intermediários para executar as suas funções: o misturador (*mixer*) e o tradutor (*translator*). O misturador é um dispositivo que recebe pacotes RTP de uma ou mais fontes, realiza as mudanças necessárias no formato dos dados, combinando múltiplos fluxos de mídia em apenas um. Sua principal função é ajustar a largura de banda entre os participantes de uma sessão. O misturador realiza ajustes de temporização para o fluxo de dados combinado, assim, todos os pacotes gerados pelo misturador terão esse dispositivo como fonte de sincronismo.

O tradutor é um dispositivo que possui as funções de distribuir os dados em *unicast*, *multicast* ou *broadcast*, permitir a comunicação entre usuários protegidos por *firewalls* e converter o tipo de codificação dos dados possibilitando troca de mídia entre participantes que não suportam um mesmo tipo de codificação ou taxa de transmissão [7].

3.6. **O protocolo RTCP**

O RTP possui um protocolo de controle próprio chamado RTCP (*RTP Control Protocol*) que cuida da sua sincronização, *feedback* e interface com o usuário e que também é definido na RFC 1889. Um canal RTCP é aberto sempre que se abre um canal RTP. A RFC 1889 define que O RTP utilize sempre uma porta par e o RTCP a porta ímpar imediatamente superior a que foi alocada para o RTP. Geralmente são alocadas as portas 5004 e 5005 para o RTP e o RTCP, respectivamente [30].

O RCTP é baseado na transmissão periódica de pacotes de controle entre os participantes de uma sessão, através do mesmo mecanismo utilizado para a distribuição de pacotes de dados. A principal função do RTCP é oferecer *feedback* sobre a qualidade da

comunicação, permitindo a realização de codificação adaptativa e dos controles de fluxo e congestionamento, controlando a taxa de transmissão dos participantes de modo a tornar o sistema altamente escalável.

O RTCP possui um identificador chamado CNAME (*Canonical Name*) que identifica unicamente um participante de uma sessão. Para o uso em sessões nas quais usuários entram e saem sem muito controle ou negociação de partêmetros, o RTCP possui a opção de exibir na interface do usuário (em texto ASCII, por exemplo) o nome de quem está se comunicando no momento, servindo ainda como um canal para localizar todos os participantes de uma sessão.

O RTCP reporta cinco tipos de relatório [7]:

Sender Report (SR) – Utilizado para a transmissão e recepção de estatísticas de participantes que são emissores ativos.

Receiver Report (RR) – Utilizado para a transmissão e recepção de estatísticas de participantes que não são emissores ativos.

Source Description (SDES) – Contém uma ou mais informações sobre um determinado participante de uma sessão, incluindo o *Canonical Name* (CNAME).

BYE – Indica o fim da participação de um usuário em uma sessão.

Application (APP) – Utilizado para funções específicas da aplicação.

Apesar de definidos individualmente, os pacotes devem ser enviados em um pacote composto como mostrado na Figura 3.9.

Figura 3.9 – *Pacote composto RTCP.*

A RFC 1889 define que um pacote composto deve iniciar com um pacote de relatório (SR ou RR) e deve conter um pacote SDES.

Uma exemplo de conferência de áudio utilizando *multicast* através dos protocolos RTP e RTCP é dada a seguir [3]:

Um endereço de grupo *multicast* e um par portas são alocadas para a sessão. Uma porta será utilizada para RTP e a outra para RTCP. Esse endereço e portas são distribuídos para os participantes da sessão.

A aplicação de cada participante começa a enviar áudio codificado em pequenos pedaços de, por exemplo, 20 ms de duração. Cada pedaço (*payload*) é precedido de um cabeçalho RTP. Esse pacote RTP é encapsulado dentro de um pacote UDP.

A sessão está sujeita a "largura de banda total da sessão" que é a banda utilizada pela codificação mais o excedente (*overhead*) gerado pelos cabeçalhos IP, UDP e RTP (40 bytes). Ou seja, para uma sessão que usa PCM de 64 kbps seriam necessários acrescentar mais 16 kbps, totalizando 80 kbps.

Dessa largura de banda da sessão, 5% deve ser reservado para o tráfego de pacotes RTCP que são enviados aproximadamente a cada 5 segundos [28]. O cabeçalho RTP indica o tipo de codificação (ex. PCM) que está sendo utilizada. Isto servirá para possíveis ajustes de acordo com a disponibilidade de banda do canal. Ele também possui campos que possibilitam a detecção de perda de pacotes e sua reordenação quando necessário (*timestamp* e número de seqüência).

Em uma conferência é necessário conhecer os participantes e saber se eles estão recebendo o áudio de maneira satisfatória. Para isso, periodicamente, a aplicação envia por *multicast* dados sobre cada usuários da sessão (RTCP SDES) e relatórios indicando a qualidade da sessão através de pacotes RTCP (RTCP SR e RTCP RR). Quando um usuário deseja sair da sessão, este envia um pacote RTCP BYE.

3.7. **O protocolo SCTP**

A camada de transporte do TCP/IP possui dois protocolos de transporte destinados a usos distintos. Aplicações que não necessitam de um serviço confiável e desejam uma comunicação rápida e flexível recorrem ao UDP, enquanto aquelas que precisam garantir a confiabilidade dos seus dados recorrem ao TCP. Porém, algumas aplicações que necessitam de um serviço de comunicação confiável, tornam-se menos eficientes devido ao modo de operação do TCP. Isso obriga os sistemas a adotarem artifícios como a implementação de protocolos confiáveis próprios operando sobre o protocolo não-confiável UDP, criando assim uma camada de confiabilidade entre o UDP e a camada de aplicação [31].

Um grupo de aplicações que se defrontaram com tais limitações dos protocolos de transporte do TCP/IP foram as aplicações de telefonia IP usuárias de sinalização telefônica baseada no protocolo SS7 [32].

O SCTP (*Stream Control Transmission Protocol*) é um novo protocolo da camada de transporte definido na RFC 2960 [33], desenvolvido para superar as limitações impostas pelo TCP no transporte de mensagens de sinalização nos sistemas de telefonia IP. O SCTP provê todas as funcionalidades do TCP adicionadas à capacidade de dar suporte a múltiplos fluxos de dados independentes e de múltiplos caminhos entre usuários. Outras aplicações são possíveis, mas o SCTP foi projetado para ser um protocolo confiável orientado à conexões que opera sobre uma rede de pacotes não-confiável sem-conexão, de modo a permitir que mensagens de sinalização PSTN sejam transportadas através de uma rede IP.

O SCTP é orientado a mensagens enquanto o TCP é orientado a bytes. Uma conexão SCTP possui um conceito mais amplo que uma conexão TCP e é chamada de *associação*. Uma associação é uma ligação entre dois usuários SCTP identificada unicamente pelos endereços de transporte usados pelos seus *endpoints*. Um *endpoint* é um emissor e receptor lógico de pacotes SCTP. O SCTP deve proporcionar os seguintes serviços:

a) Entrega de dados com confirmação, livre de erros ou duplicações;
b) Fragmentação dos dados de acordo com o tamanho máximo de unidade de transmissão (MTU) do caminho descoberto;
c) Entrega seqüencial das mensagens em múltiplos fluxos com a opção de entrega dos dados por ordem de chegada;
d) Inclusão opcional de múltiplas mensagens num único pacote SMTP;
e) Tolerância a falhas em nível de rede através de esquema de múltiplos caminhos;
f) Suporte à retransmissão seletiva de pacotes.

O protocolo SCTP pode ser decomposto em uma série de funções elementares [7]:

Estabelecimento de associação – Uma associação SCTP é estabelecida através da troca de quatro mensagens *four-way handshake* através de um mecanismo de *cookie*, evitando ataques do tipo negação de serviço (*Denial of Service*) comuns ao TCP.

Encerramento de associação – O SCTP permite dois tipos de encerramento da associação. No *encerramento coordenado* há uma garantia de que os dados transmitidos ou mantidos em *buffers* ainda serão entregues. No *encerramento forçado* não há qualquer garantia quanto à entrega dos dados em trânsito. Como no TCP, o SCTP não permite estados meio-abertos onde um usuário continua enviando dados para um outro usuário que já encerrou sua associação.

Entrega sequencial de fluxos – Um fluxo SCTP é uma sequência unidirecional (*simplex*) de mensagens ordenadas que devem ser entregues a um *endpoint*, podendo haver numa associação um número desigual de fluxos em cada direção. No estabelecimento da associação, cada usuário informa o número de fluxos suportado. O SCTP então associa a cada mensagem do fluxo um número de sequência e garante que elas sejam entregues ordenadamente ao receptor. Os múltiplos fluxos são conceitos lógicos fim-a-fim que concedem eficiência ao sistema. O SCTP define uma fila de ordenação e um escopo de retransmissão independentes para cada fluxo, assim, enquanto um fluxo se encontra bloqueado aguardando a chegada da próxima mensagem da sequência, a entrega de mensagens aos outros fluxos pode prosseguir normalmente.

Fragmentação de dados – Quando necessário, o SCTP pode fragmentar as mensagens para que elas possuam o tamanho máximo de unidade de transmissão (MTU) do caminho descoberto. No receptor, as mensagens são remontadas antes de serem passadas ao usuário.

Confirmação de mensagens e controle de congestionamentos – O SCTP associa um número de sequência a cada mensagem de um fluxo e o receptor confirma todas as mensagens recebidas, mesmo que exista lacunas entre elas, permitindo retransmissão seletiva. A retransmissão de pacotes que não foram confirmados em tempo hábil é controlada por uma estratégia que evita congestionamentos similar a do TCP.

Inserção de mensagens – Um pacote SCTP consiste em um cabeçalho seguido por um ou mais unidades de informação chamadas mensagens *chunk* (pedaço). Cada mensagem contém seu próprio cabeçalho e um conteúdo específico que pode ser dados ou controle. O usuário pode solicitar que uma ou mais mensagens sejam inseridas em um único pacote SCTP. Em momentos de congestionamento o SCTP pode realizar a inserção de mensagens mesmo sem a solicitação do usuário.

Validação do pacote – Os pacotes SCTP possuem um rótulo de verificação que é escolhido pelos usuários no momento da associação para garantir autenticidade aos pacotes e uma soma de verificação utilizada para a detecção de pacotes defeituosos. A recepção de um pacote com um desses campos inválidos faz com que o SCTP descarte o mesmo silenciosamente.

Gerenciamento de caminhos – Um caminho é a rota tomada por um pacote STCP entre os endereços de transporte dos *endpoints* origem e destino. O SCTP monitora a alcançabilidade dos *endpoints* destino e informa ao usuário qualquer alteração na mesma, além de identificar o conjunto de possíveis caminhos durante o estabelecimento da

associação e reportar a lista dos endereços de transporte retornados pelos *endpoints* destino ao usuário.

O pacote SCTP genérico é composto por um cabeçalho comum e uma ou mais mensagens em sua carga útil, como mostrado na figura 3.10. O cabeçalho comum do SCTP é composto dos seguintes campos:

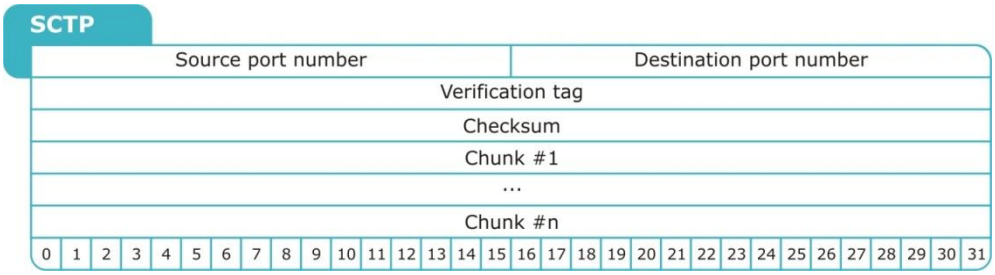


Figura 3.10 – *Pacote SCTP.*

Source Port Number – Número da porta de origem do fluxo. Possui 16 bits de comprimento.

Destination Port Number – Número da porta de destino à qual deve ser encaminhado o fluxo de dados no receptor. Possui 16 bits de comprimento.

Verification Tag – É um rótulo de verificação de 32 bits de comprimento utilizado para a validação do emissor do pacote. Garante a autenticidade da mensagem.

Checksum – É uma soma de verificação de 32 bits de comprimento utilizada para a detecção de erros no pacote.

O SCTP é um protocolo modular, o que lhe confere uma grande flexibilidade quando são necessárias alterações ou adaptações. Cada campo mensagem de um pacote SCTP possui seu próprio cabeçalho, e a sua carga útil é formatada de acordo com o tipo de conteúdo que a mensagem carrega. De um modo geral, o cabeçalho de uma mensagem SCTP é composto pelos campos mostrados na figura 3.11.

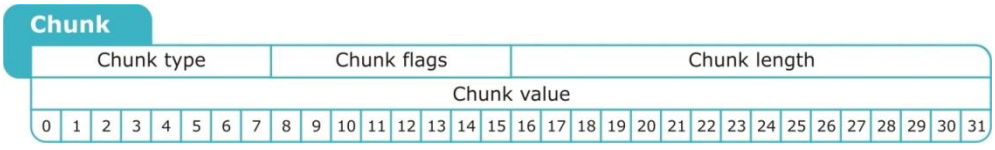


Figura 3.11 – *Cabeçalho de uma mensagem (chunk) SCTP.*

Chunk Type – Esse campo de 8 bits de comprimento identifica o tipo de informação contida na mensagem.

Chunk Flags – Esse campo de 8 bits é formatado de acordo com o tipo de conteúdo da mensagem e é definido para dar uma maior flexibilidade ao protocolo.

Chunk Length – Esse campo de 16 bits representa o comprimento total da mensagem em bytes.

Chunk Value – Esse campo possui comprimento variável e contém a informação transportada na mensagem. O uso e formatação desse campo é dependente do tipo da mensagem.

Um exemplo de mensagem SCTP é mostrado na figura 3.12. A figura traz o formato de uma mensagem de *payload* na qual podem ser visualizadas as funções específicas dadas aos campos *chunk type*, *chunk flags*, *chunk length* e *chunk value*.

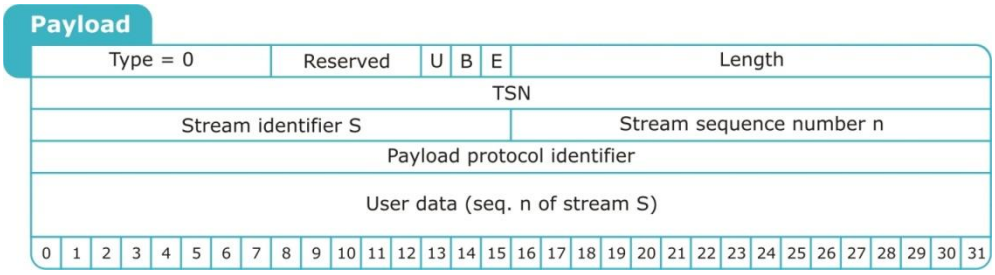


Figura 3.12 – *Formato de uma mensagem de payload do SCTP.*

3.8. A recomendação H.323

Desenvolvida pelo ITU-T em 1996 e tendo recebido diversos melhoramentos e revisões até 2006, a recomendação H.323 define terminais e outras entidades que oferecem serviços de comunicação de multimídia sobre redes comutadas a pacotes que não garantem a qualidade de serviço oferecida [34, 35]. O H.323 possui um pacote de mensagens compacto e sua sinalização é extremamente rápida, especialmente se comparado ao SIP, que em termos comparativos utiliza pacotes de mensagens bem mais longos. O projeto do H.323 foi bastante referenciado na filosofia de operação do sistema de telefonia convencional PSTN, focando o esforço nos aspectos de brevidade e disponibilidade do sistema. Os sinais do H.323 são curtos e a rede é utilizada o mínimo possível para transportar sinalização de chamadas e ao máximo para transportar voz.

Uma rede de telefonia necessita de diversos protocolos para poder funcionar, nesse aspecto, o H.323 é muito mais uma avaliação da arquitetura da telefonia IP do que um protocolo específico [4]. O H.323 é considerado um padrão "guarda-chuva" que faz referência a um grande número de protocolos específicos para codificação de voz, estabelecimento e configuração de chamadas, sinalização, transporte de dados e outros. Os principais protocolos que compõem o H.323 são descritos a seguir.

H.225 – Exerce funções de sincronização dos dados, estabelecimento e controle de chamadas através do RAS (Registro, Autenticação e Status).

Q.931 - Trata dos aspectos de sinalização, estabelecimento e encerramento de conexões, geração de tons de chamada e de discagem para a interoperabilidade com o sistema de telefonia padrão PSTN.

H.245 – Responsável pela negociação dos sistemas de codificação utilizados e da taxa de transmissão da comunicação.

G.7xx – Responsável pela codificação utilizada na mídia.

RTP – Realiza o transporte de mídia em tempo real.

RTCP – Controla o protocolo de transporte de mídia (RTP).

O H.323 também define diversos protocolos para prestação de serviços auxiliares. Dentre eles se incluem os seguintes:

T.12x – Oferecem serviços interativos de comunicação de dados para multiconferências, tais como *white-boarding*.

H.450 – Oferece serviços suplementares como chamada em espera, transferência de chamadas e outros.

H.26x – Protocolos utilizados para a codificação de vídeo.

H.246 – Protocolo utilizado para interoperação com sistemas de comutação de circuitos (RTPC).

H.235 – Protocolo que confere aspectos segurança ao sistema (autenticação, integridade e privacidade).

3.8.1. **Elementos do H.323**

Embora se possa efetuar uma chamada utilizando apenas dois terminais H.323, outros elementos são necessários quando se deseja realizar uma conferência multiponto ou interagir com outras redes de comunicação. O padrão H.323 é composto por quatro entidades principais: terminais, *gateways*, *gatekeepers* (guardiões) e unidades de controle de multiponto (MCU – *Multipoint Control Unit*).

Terminal H.323 – O Terminal H.323 (Tx) é um dispositivo que executa a pilha de protocolos H.323 no qual está implantado o serviço de interface de telefonia IP com o usuário. Esse dispositivo implementa diversos aspectos do processo de realização de chamada e atua como terminal de voz, vídeo e dados através de recursos multimídia. O Terminal H.323 pode ser uma estação multimídia (PC equipado com caixas de som e

microfone) ou um telefone IP capazes de realizar uma comunicação bidirecional com outra entidade H.323. Os terminais H.323 devem dar suporte obrigatório aos protocolos responsáveis pela codificação de áudio (G.711, G.728), sinalização, configuração e controle de chamadas (Q.931, H.245 e H.225 – RAS) e transporte de mídia em tempo real (RTP e RTCP).

Gateway H.323 – O *Gateway* H.323 (GW) é um dispositivo localizado na fronteira da rede H.323. Ele é capaz de realizar os serviços de interface e tradução bidirecional de tempo real entre terminais H.323 localizados em uma rede comutada a pacotes e outros terminais ITU pertencentes à uma rede comutada a circuitos, ou mesmo a outro *gateway* H.323. O *gateway* H.323 situa-se entre a rede IP e uma outra rede de telecomunicações (RTCP, PBX, ISDN, GSM, UMTS), realizando a compatibilização dos procedimentos de chamada e formatos de transmissão, bem como a conversão dos protocolos de sinalização e codificadores de voz das duas redes. Embora o tipo mais comum de *gateway* H.323 seja o IP/PSTN, que realiza a compatibilização entre a rede comutada a pacotes IP e a rede de telefonia convencional comutada a circuitos PSTN, também são possiveis interfaces com os sistemas correspondentes às recomendações do ITU-T H.310 (H.320 on B-ISDN), H.320 (ISDN – T1), H.321 (ATM), H.322 (GQOS-LAN), H.324 (GSTN – POTS), H.324M (*Mobile*), e V.70 (DSVD).

Gatekeeper H.323 – O *Gatekeeper* (guardião) H.323 (GK) é um dispositivo que fornece os serviços de tradução de endereços e controle de acesso dos terminais, *gateways* e MCUs à rede H.323 de forma centralizada. Tipicamente um *gatekeeper* é uma aplicação implementada em software em um PC, podendo, no entanto, ser incorporada em um *gateway* ou terminal H.323. Uma coleção de terminais, *gateways* e MCUs sob responsabilidade de um único *gatekeeper* é chamada de zona. Uma zona pode conter desde terminais separados por poucos metros de distância até terminais localizados em continentes diferentes, desde que sejam gerenciados por um único *gatekeeper*. Algumas vezes pode existir um segundo *gatekeeper* apenas para fins de *backup* ou balanceamento de carga. O *gatekeeper* é responsável pelas funções de tradução de endereços (roteamento), controle de admissão e gerenciamento de zona, podendo opcionalmente realizar as funções de controle de sinalização e autorização das chamadas, gerenciamento de largura de banda, serviços de diretório e localização de *gateways*. *Gateways* que desejam se comunicar em uma zona controlada por um *gatekeeper* precisam se registrar no mesmo para poder realizar a troca de mídia entre si [18]. Algumas mensagens passam pelo *gatekeeper*, outras

não. Fluxos de mídia nunca passam pelo *gatekeeper*, pois a função desse dispositivo é estritamente de controle. Quanto maior o número de mensagens roteadas pelo *gatekeeper* maior o seu nível de controle, sua carga e responsabilidade.

Unidade de Controle Multiponto H.323 – A Unidade de Controle Multiponto H.323 (MCU – *Multipoint Control Unit*) é o elemento de rede capaz de proporcionar a três ou mais terminais e *gateways* H.323 a capacidade de estabelecer uma conferência multiponto entre si. A MCU pode ser um dispositivo isolado (implementado num PC), ou ser incorporado a um terminal, *gateway* ou *gatekeeper*.

Uma MCU pode ser trazida para uma conferência pelo *gatekeeper*, sem ter sido explicitamente chamada pelos terminais. Tipicamente, a MCU é composta por dois elementos: o controlador de multiponto (obrigatório) e o processador de multiponto (opcional).

O controlador de multiponto (MC) é um elemento responsável pelo controle e sinalização da conferência multiponto, negociação com os usuários através dos protocolos H.225 e H.245 e da gestão dos recursos da conferência.

O processador de multiponto (MP) é um processador digital de sinais (DSP – *Digital Signal Processor*) que proporciona um processamento centralizado dos fluxos de áudio, vídeo e dados numa conferência multiponto. O MP realiza a combinação (*mixing*), chaveamento (*switching*) e transcodificação de um único ou de múltiplos fluxos de mídia da conferência multiponto.

As MCUs podem ser centralizadas (compostas por MC+MP) ou descentralizadas (compostas apenas por MC). As MCUs centralizadas manipulam sinalização e os fluxos de mídia das conferências multiponto, enquanto as descentralizadas manipulam apenas a sinalização, deixando que os fluxos de mídia fluam diretamente entre os terminais [7].

A arquitetura de rede do H.323 é mostrada na figura 3.13.

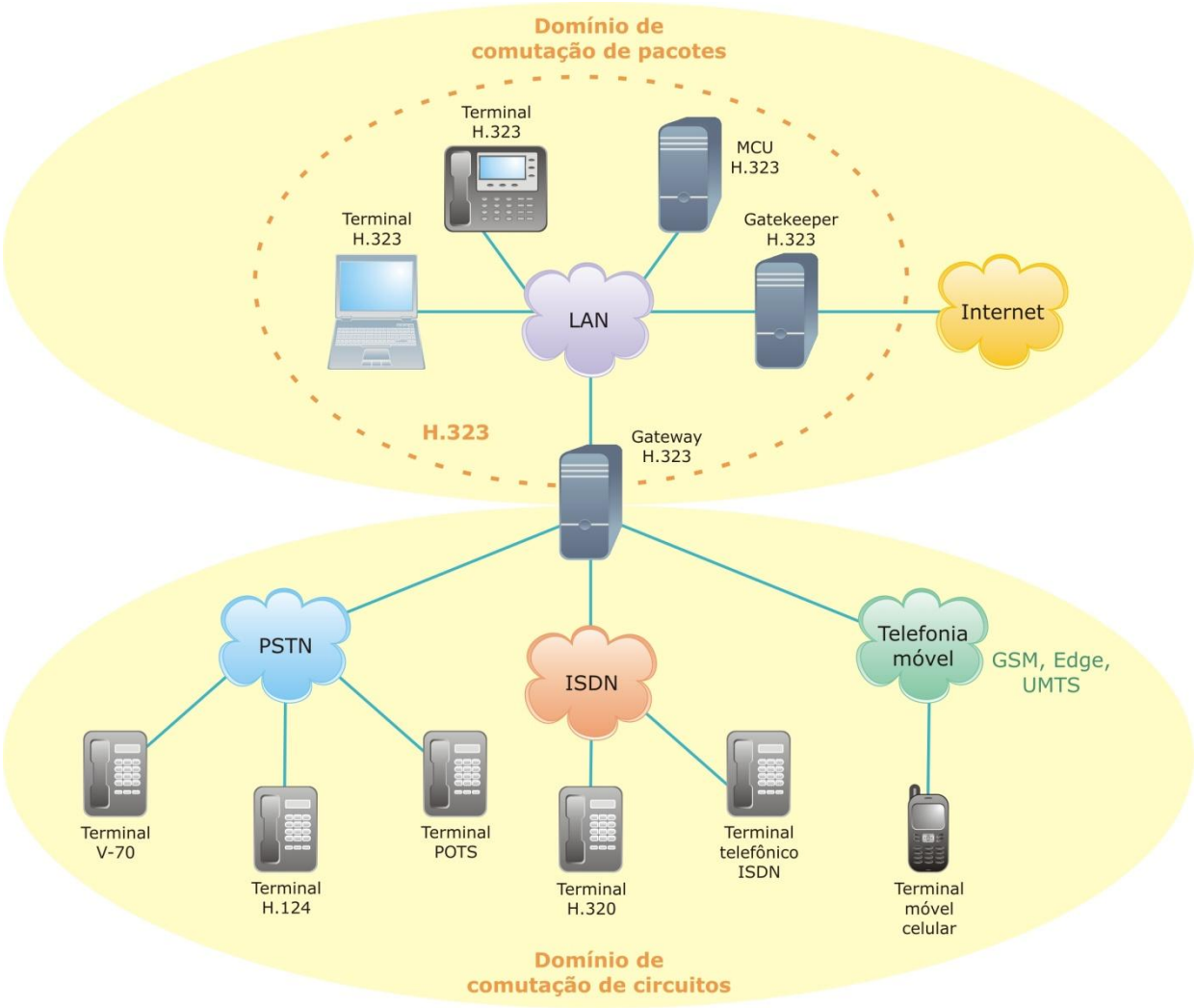


Figura 3.13 – *Arquitetura H.323.*

3.8.2. **Canais do H.323**

Os dispositivos que compõem o padrão H.323 se comunicam uns com os outros através de três canais lógicos principais definidos nas especificações H.225.0 e H.245 [36, 37].

A especificação H.225.0 descreve como informações de áudio, vídeo, dados e controle podem ser gerenciadas numa rede de pacotes para proporcionar serviços de conversação em equipamentos H.323. Esta especificação possui duas partes principais: A sinalização de chamadas e o RAS (*Registration, Admission and Status*).

O canal de sinalização é utilizado para configurar conexões entre terminais ou entre um terminal e um gatekeeper. O padrão recomenda que as mensagens que trafegam nesse canal utilizem o padrão Q.931 de sinalização e sejam transportadas através do protocolo confiável TCP.

O canal RAS do H.225.0 é utilizado para a comunicação entre um teminal ou um *gateway* com um *gatekeeper*. O RAS realiza as operações de registro, controle de admissão, ajustes de banda, *status* e desconexão e deve ser aberto em uma comunicação H.323 antes de qualquer outro canal lógico.

O canal H.245 tem a função de realizar o controle das chamadas do H.323. Ele é o responsável pelo controle de fluxo, determinação do mestre e escravo, controle de conferências, cifragem, controle de *jitte*r, negociação de capacidades e pela negociação, abertura e fechamento dos canais lógicos RTP/RTCP que carregam os fluxos de mídia. As mensagens de controle H.245 sempre são confirmadas pelo receptor. O H.245 possui a capacidade de ser tunelado em mensagens de sinalização H.225.0, o que facilita a transposição de *firewalls*.

3.8.3. **Operação do H.323**

Para serem capazes de se comunicar uns com os outros os terminais H.323 devem estar registrados em um *gatekeeper* e pertencer a uma zona. Inicialmente, o terminal precisa descobrir a qual *gatekeeper* ele deve se registrar. Isso é feito através de uma mensagem GRQ (*Gatekeeper Request*) que é lançada na rede por multidifusão (*broadcasting*). Um ou mais *gatekeepers* podem responder com mensagens GCF (*Gatekeeper Confirm*) ou GRJ (*Gatekeeper Reject*) indicando a disponibilidade de cada um deles para aquele terminal. A mensagem GCF é acompanhada do endereço de transporte do canal RAS (*Registration, Admission and Status*) para que o terminal possa se comunicar com o seu *gatekeeper*. Caso nenhum *gatekeeper* responda dentro de um período de tempo, um *timeout* será gerado e o terninal deverá realizar uma nova transmissão do GRQ.

Em seguida o terminal deve se registrar no *gatekeeper*, o que é feito através da transmissão de uma mensagem RRQ (*Registration Request*) pelo canal RAS do mesmo. O *gatekeeper* pode responder com uma mensagem RCF (*Registration Confirmation*) ou RRJ (*Registration Reject*), no primeiro caso o usuário estará registrado e poderá operar normalmente.

Tanto o usuário quanto o terminal podem solicitar o cancelamento do registro através de uma mensagem URQ (*Unregister Request*). A outra parte envolvida deverá responder com uma confirmação UCF (*Unregister Confirmation*) ou negação URJ (*Unregister Reject*) do pedido. A troca de mensagens entre um terminal H.323 e o seu *gatekeeper* é mostrada na figura 3.14.

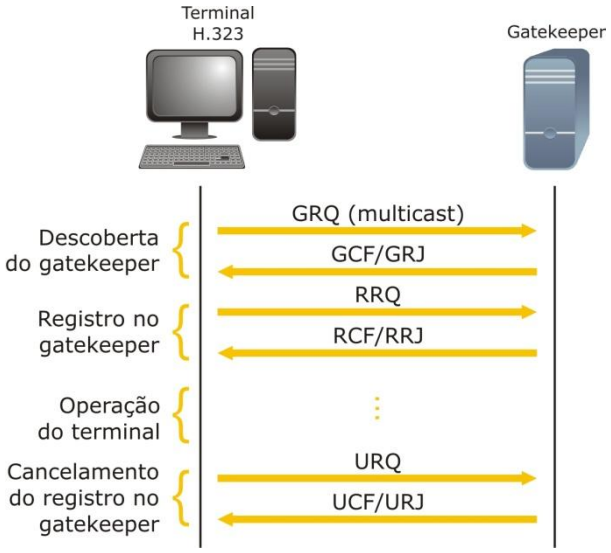


Figura 3.14 – *Comunicação entre um terminal e um gatekeeper H.323 através do canal RAS.*

O terminal H.323 pode localizar um outro terminal enviando um pedido LRQ (*Location Request*) pelo canal RAS. O *gatekeeper* no qual o terminal destino está registrado responde com uma mensagem LCF (*Location Confirmation*). Todos os demais *gatekeepers* deverão enviar um LRJ (*Location Reject*), como mostra a figura 3.15.

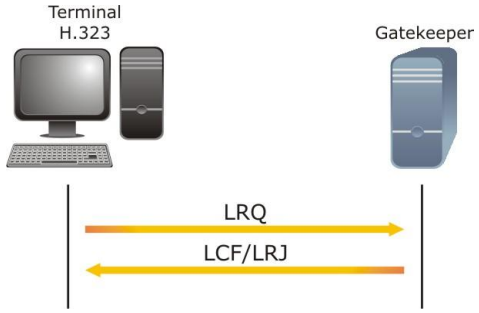


Figura 3.15 – *Localização de um usuário.*

Supondo que um terminal já localizou o *gatekeeper* da sua zona e se encontra registrado no mesmo, o processo de estabelecimento de chamadas no H.323 é mostrado na figura 3.16 e explicado a seguir.

O terminal H.323 de origem (A) solicita uma admissão ARQ (*Admission Request*) ao *gatekeeper* no qual está registrado. O *gatekeeper* retorna para A uma mensagem ACF (*Admission Confirm*) juntamente com o endereço do canal H.225 de sinalização da chamada.

Através do canal de sinalização, o terminal A envia uma mensagem de configuração (*Setup*) ao terinal H.323 de destino (B) que retorna para A uma mensagem de chamada em andamento (*Call Proceeding*). Caso aceite a chamada, o terminal B solicita uma admissão

no *gatekeeper* do mesmo modo realizado por A. Ao ser autenticado, o terminal de destino envia um tom de discagem (*Ringing*) ao terminal de origem, juntamente com uma mensagem (*Connect*) que informa o endereço do canal de controle H.245 que deverá ser utilizado na próxima fase.

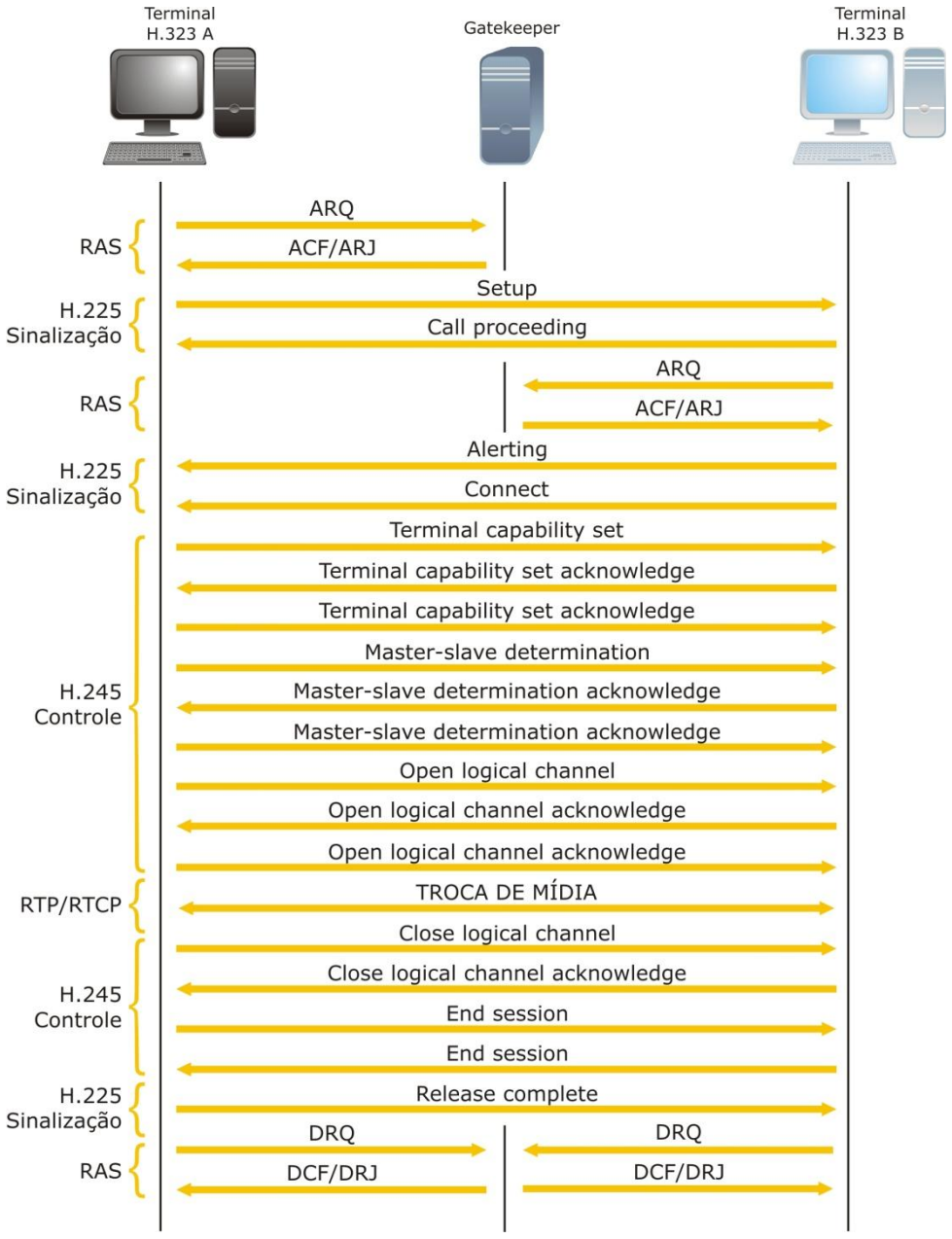


Figura 3.16 – *Estabelecimento de uma chamada H.323.*

Através do canal de controle H.245 os terminais trocam informações sobre suas capacidades, negociam parâmetros, determinam o mestre e os escravos (no caso de um conferência multiponto) e abrem os canais lógicos RTP e RTCP para a comunicação da mídia. Então, a mídia pode então ser trocada entre os terminais A e B através do protocolo RTP.

Para finalizar a chamada, qualquer um dos terminais (no caso o terminal A) envia pelo canal de controle H.245 um pedido de fechamento dos canais lógicos (*Close Logical Channel*) seguido de uma solicitação de encerramento da sessão (*End Session*) que ao ser recebida é enviada de volta pelo usuário B. O terminal A confirma o fim da sessão através de uma menagem de liberação completada (*Release Complete*). Feito isso, ambos os terminais podem se comunicar com o *gatekeeper* pelo canal RAS e desfazer seus registros através das mensagens DRQ (*Disengage Request*) que são confirmadas através de uma mensagen DCF (*Disengage Confirmation*).

Durante a comunicação, os terminais podem renegociar parâmetros como a largura de banda disponível para a chamada. O processo é mostrado na figura 3.17 e detalhado a seguir.

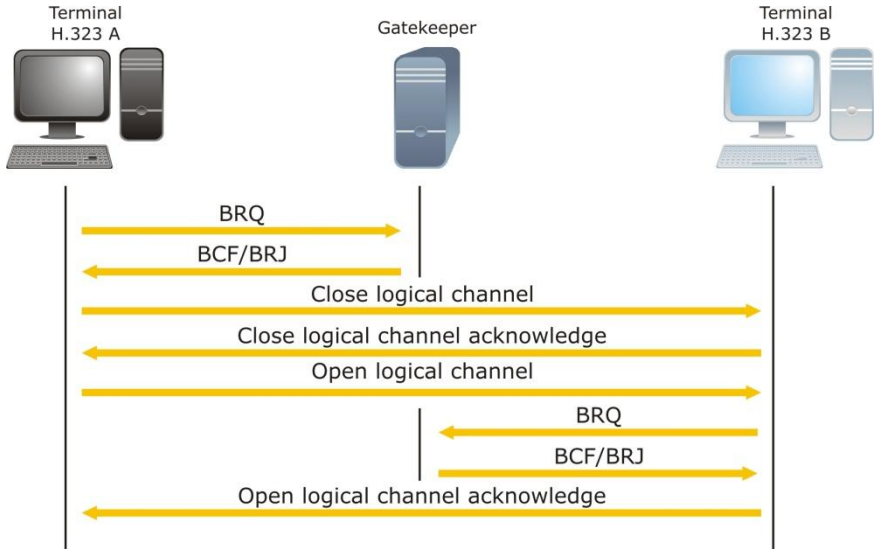


Figura 3.17 – *Renegociação de parâmetros durante uma chamada H.323.*

Inicialmente um dos terminais (A) envia ao *gatekeeper* uma mensagem BRQ (*Bandwidth Request*) que é respondida com um BCF (*Bandwidth Confirmation*) ou BRJ (*Bandwidth Reject*). Então, A envia para B uma mensagem de fechamento dos canais lógicos (*Close Logical Channel*). Após receber a confirmação do fechamento dos canais lógicos (*Close Logical Channel Acknowledge*), o terminal A solicita ao terminal B a abertura de um novo canal lógico com as características solicitadas (*Open Logical Channel*). O terminal B solicita a banda necessária ao ao *gatekeeper* (BRQ) que poderá confirmar (BCF) ou rejeitar (BRJ) o pedido. Caso o *gatekeeper* libere a banda para o terminal B, este envia uma mensagem para o terminal A confirmando a abertura do canal lógico com as novas características que foram negociadas (*Open Logical Channel Acknowledge*).

3.9. **O SIP**

Definido pelo IETF em 1999 na RFC 2543 [10] e descrito em 2002 na RFC 3261 [38], o SIP (*Session Initiation Protocol*) é um protocolo de sinalização da camada de aplicação utilizado para iniciar, modificar e terminar sessões interativas de multimídia entre usuários.

O H.323 foi visto por algumas pessoas da comunidade da Internet como um protocolo extenso, completo e inflexível, motivos esses que levaram à concepção do SIP como uma forma mais simples, modular e escalável de realizar chamadas de voz sobre IP.

Diferentemente do H.323 que é um conjunto de protocolo completo, com pacotes compactos e amigável à máquina, o SIP é um único módulo projetado para interoperar bem com as aplicações de Internet existentes, possuindo uma interface textual amigável ao usuário, baseada em protocolos conhecidos, como HTTP (*Hiper Text Transport Protocol*) e SMTP (*Simple Mail Transfer Protocol*) [4, 18].

Embora possa operar na forma par-a-par (*peer-to-peer*), o SIP é um protocolo cliente/servidor e, apesar dos sistemas de voz sobre IP serem a sua principal área de atuação, outras aplicações em sistemas de mensagens instantâneas (SMS), relatório de notícias e jogos distribuídos também são possíveis [4, 7, 20]. O SIP pode dar suporte a funções básicas de chamada (espera, encaminhamento, bloqueio, distribuição) e serviços como conferências, *click to talk*, mensagens instantâneas e siga-me.

O SIP promete ser, num futuro próximo, o protocolo das redes de comunicação convergentes. No seu desenvolvimento, foram focados os aspectos de interoperabilidade com os protocolos existentes da IETF, escalabilidade, simplicidade, velocidade, mobilidade e facilidade na implementação de características e serviços [7].

O SIP deve proporcionar os serviços de gerenciamento de participantes de uma sessão, localização de usuários, tradução de nomes e negociação de características de uma chamada.

3.9.1. **Elementos do SIP**

A arquitetura do SIP é composta por Agentes Usuários (UA – *User Agents*) e os Servidores de Rede (NS – *Network Servers*) como mostrado na figura 3.18 [7, 20].

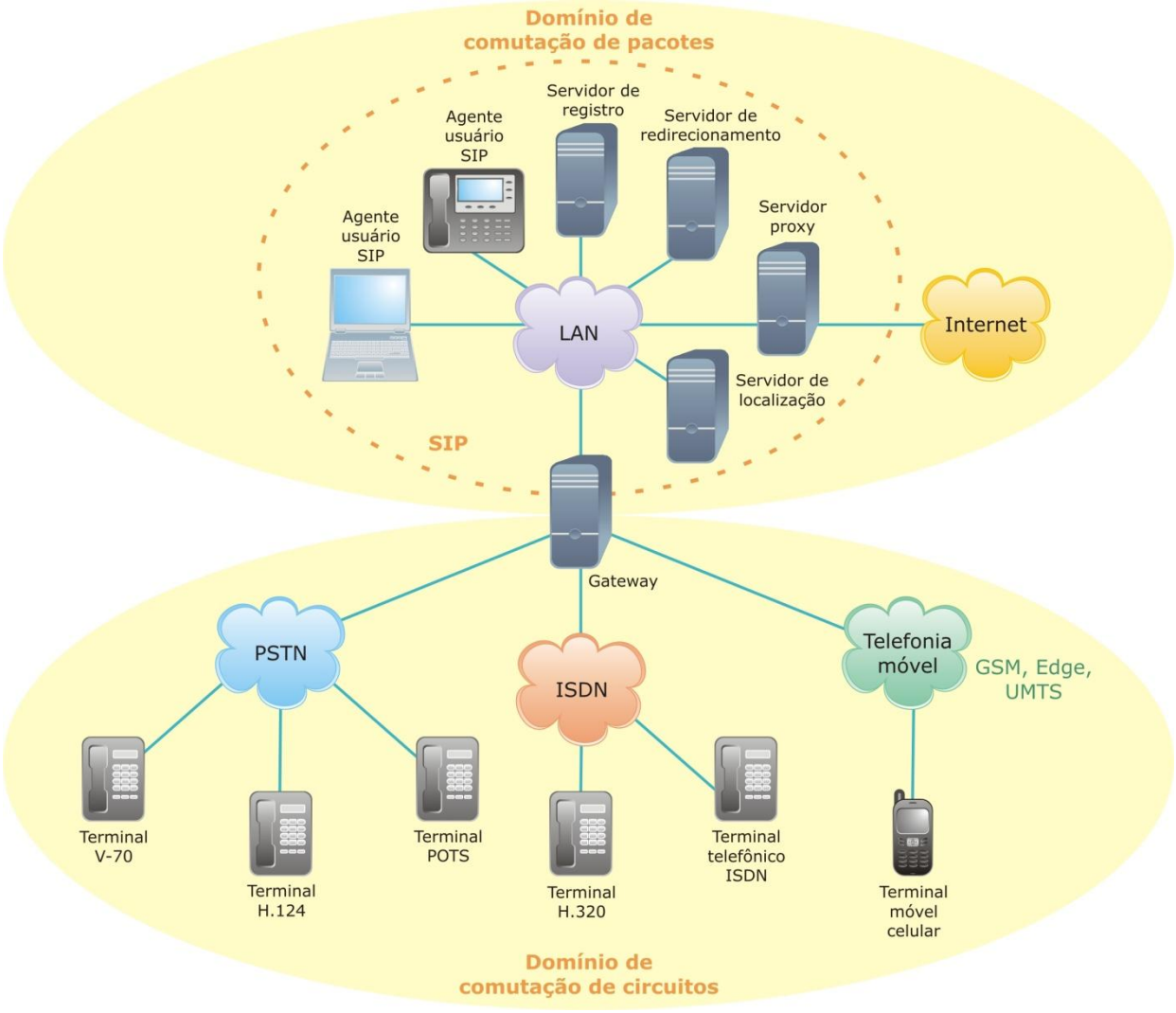


Figura 3.18 – *Arquitetura do SIP.*

Agente Usuário SIP – Implementado no dispositivo de interface humana conectado à rede (PC, *laptop*, PDA, telefone celular), o Agente Usuário SIP é efetivamente o sistema fim de uma chamada que atua em nome de um usuário e recebe uma URI (*Uniform Resource Identifier*) SIP. Uma URI é um identificador do usuário no domínio SIP, que devido ao aspecto textual do SIP, se assemelha em formato a um endereço de *e-mail* na forma: usuário@servidor. De um modo geral, uma URI segue os seguintes exemplos:

- sip: professor@ufpe.br
- sip: aluno@casa.br

O Agente Usuário SIP é composto por dois módulos: o Agente Usuário Cliente (UAC), que inicia as solicitações de chamada e o Agente Usuário Servidor (UAS), que responde às solicitações de chamada. Graças à essa arquitetura, usuários SIP podem se comunicar de uma forma par-a-par através de um protocolo cliente/servidor [38].

Servidores de Rede SIP – Os Servidores de Rede SIP são elementos de rede responsáveis pela sinalização das chamadas. As principais funções desses servidores são a determinação de nomes e a localização de usuários na rede [7]. Existem três tipos de Servidores de Rede SIP [18]:

a) *Registrar Server* (Servidor de Registro): O Servidor de Registro é um servidor que contém uma base de dados e que se comunica com os nós da rede SIP para coletar, armazenar e distribuir informações sobre o paradeiro dos usuários. Quando um usuário SIP se registra em um servidor de registro, ele informa ao mesmo como encontrá-lo e sua disponibilidade, mais especificamente, qual o seu endereço IP e porta de comunicação para a realização de futuras chamadas.

b) *Proxy Server* (Servidor Procurador): O *Proxy* é um servidor que recebe solicitações dos usuários e as trata ou as encaminha em nome dos mesmos para um ou mais domínios ou servidores. Do ponto de vista desses outros servidores, tudo se passa como se as mensagens partissem do próprio *proxy* e não dos usuários escondidos por trás dele que utilizam o *proxy* como seu procurador. O *proxy* serve como um roteador para mensagens trocadas com redes externas, não-locais. Dessa forma, se uma chamada é recebida de uma rede externa, é tarefa do *proxy* conectá-la ao usuário chamado. Algumas vezes, mesmo as chamadas locais são forçadas a passar por um servidor *proxy*, proporcionando um maior controle administrativo da rede de voz e a possibilidade de executar aplicações de telefonia centralizada como a gravação de ligações. O *proxy* também é utilizado para resolver problemas de comunicação impostos por NATs ou *firewalls*. Como o *proxy* tanto envia quanto recebe mensagens, ele age ao mesmo tempo como cliente e como servidor.

c) *Redirect Server* (Servidor de Redirecionamento): O Servidor de Redirecionamento é um servidor que recebe requisições dos usuários SIP e retorna aos mesmos o endereço da nova localização do usuário destino, ou de um servidor alternativo para o qual a requisição deve ser direcionada. Este servidor não é implementado em todas as redes SIP, sendo mais comum em redes extensas e com muitos servidores espalhados pelo mundo.

3.9.2. **Mensagens SIP**

A operação do SIP é baseada na troca de mesagens textuais de solicitação e resposta (*request-response*) de forma análoga a do protocolo HTTP (*Hyper Text Transport*

Protocol) [39]. Mensagens SIP podem ser solicitações de um cliente a um servidor ou respostas de um servidor para um cliente. Cada mensagem contém uma linha inicial seguida por zero ou mais cabeçalhos opcionalmente seguidos pelo corpo da mensagem, separado dos dois primeiros por uma linha em branco, de acordo com a sintaxe mostrada na figura 3.19.

Figura 3.19 – *Sintaxe de uma mensagem SIP*.

A linha inicial informa se a mensagem se trata de uma solicitação ou resposta, contendo a URI do agente usuário destino, a versão do SIP e o método (para as solicitações) ou o código (para as respostas) utilizado.

Dependendo da solicitação ou resposta, determinados cabeçalhos podem ser obrigatórios, opcionais ou não aplicáveis. A RFC 2543 define quatro tipos de cabeçalho cuja função é prover informações adicionais sobre a mensagem ou habilitar uma manipulação apropriada da mesma [5], são eles:

- *General Headers*: Utilizados tanto em mensagens de solicitação quanto de resposta adicionam opções gerais, como por exemplo, a memorização da rota seguida por uma mensagem.
- Request Headers: Utilizados apenas nas mensagens de solicitação, podem, por exemplo, forçar que as mensagens seguintes sigam obrigatoriamente através de um *proxy*.
- Response Headers: Utilizados apenas nas mensagens de resposta, podem, por exemplo, conter alertas sobre tipos não suportados de mídia.
- Entity Headers: Caregam informações sobre o corpo da mensagem, indicando, por exemplo, qual o tipo de codificação utilizada nos dados ou limitando o tempo de validade dos mesmos.

Alguns desses cabeçalhos são mostrados na tabela 3.5 [39, 40].

O corpo da mensagem descreve o tipo de sessão a ser estabelecida ou uma descrição dos tipos de mídia e codecs que serão utilizados durante a mesma. Como o SIP não define a estrutura ou conteúdo do corpo de mensagem [41], tal função é realizada através de outro protocolo [3], tipicamente o SDP (*Session Description Protocol*) [42].

Tabela 3.5 – *Cabeçalhos SIP*.

Cabeçalho	Função
From	Indica o usuário origem
To	Indica o usuário destino
Subject	Indica o assunto da chamada
Call-ID	Identifica univocamente a chamada
Cseq	Indica o número de sequência de uma solicitação
Contact	Lista os endereços onde o usuário pode ser contactado
Content length	Indica quantos bytes há no corpo da mensagem
Content type	Indica o tipo de informação contida na mensagem
Require	Indica o protocolo a ser negociado e utilizado como extensão do SIP
Via	Indica o caminho percorrido pela mensagem

3.9.3. **Solicitações SIP**

A RFC 2543 [10] define seis tipos de mensagens de solicitações SIP ou métodos (INVITE, ACK, BYE, OPTIONS, CANCEL, REGISTER). Duas outras foram propostas em seguida (INFO e PRACK), devendo ser adicionadas quando o SIP se tornar um padrão da Internet [5]. Após isso, mais cinco mensagens foram definidas (SUBSCRIBE, NOTIFY, MESSAGE, UPDATE e REFER). A tendência é que outras mensagens sejam criadas a medida em que o protocolo se estabeleça e novas aplicações surjam para o mesmo. Uma breve explicação sobre a função de cada um desses métodos será dada a seguir.

- INVITE: Serve para estabelecer sessões de multimídia. Seu corpo carrega uma descrição da sessão proposta e um identificador único para a mesma. Pode-se considerar que uma sessão SIP está estabelecida quando os agentes usuários trocam mensagens INVITE que são confirmadas com mensagens ACK.
- ACK: Confirma as mensagens INVITE, espelhando em seu corpo os identificadores desta para que seja associada corretamente à mensagem (sessão) que deseja confirmar.
- BYE: Mensagem que não possui corpo é trocada fim-a-fim que tem a função de finalizar uma sessão de multimídia, devendo ser trocada e confirmada por ambos os agentes usuários.

- OPTIONS: Tem a função de descobrir as capacidades de outros agentes usuários e servidores SIP. Essa mensagem pode descobrir se outro agente usuário suporta um tipo particular de mídia ou enviar informações sobre os métodos e linguagens suportados pelo transmissor da mesma.
- CANCEL: Utilizada para cancelar uma sessão que ainda está sendo negociada com outro agente usuário ou *proxy*, antes que esta seja estabelecida.
- REGISTER: Registra um agente usuário em um servidor SIP. Essa mensagem informa ao servidor onde um agente usuário pode ser encontrado.
- INFO: Utilizada para enviar informações (sinalização) entre agentes usuários durante sessões de multimídia. No entanto, as mensagens INFO não modificam sessões, o que deve ser realizado através de mensagens INVITE.
- PRACK: Funciona como um ACK provisório utilizado para notificar um agente usuário que está tentando estabelecer uma sessão complexa que o andamento da configuração da chamada está indo bem, antes da mesma ser completamente configurada e confirmada com um ACK.
- SUBSCRIBE: Indica que o usuário deseja ser notificado da ocorrência de eventos durante a chamada.
- NOTIFY: Informa a ocorrência de eventos durante uma chamada.
- MESSAGE: Indica que carrega em seu corpo uma mensagem instantânea SMS.
- UPDATE: Altera uma oferta realizada anteriormente em uma chamada ainda não estabelecida.
- REFER: Inicia uma transferência de chamadas, fazendo com que o receptor contacte uma terceira parte.

3.9.4. **Respostas SIP**

As mensagens de resposta SIP contêm em seu corpo um código (número de três dígitos) que indica o resultado de uma solicitação e uma descrição desse resultado em formato de texto. Os códigos das mensagens-resposta do SIP são agrupados em classes de acordo com a função que desempenham e serão descritos a seguir.

- 1xx – Informativas: Indicam que uma requisição foi recebida corretamente e a entidade a quem ela se destinou está dando continuidade ao processo.
- 2xx – Sucesso: Significam que a ação foi corretamente recebida, entendida e aceita.

- 3xx – Redirecionamento: Informam que alguma ação deve ser tomada para que a chamada seja completada. Geralmente essas mensagens indicam que o usuário destino não se encontra na localização informada, retornando para o usuário origem a atual localização do mesmo, para que uma nova mensagem INVITE seja enviada.
- 4xx – Erro do Usuário: Indicam que a mensagem contém erros de sintaxe ou não pode ser tratada no servidor solicitado. Essas respostas podem ainda ser uma forma dos servidores *proxy* informarem que os agentes usuários devem ser autorizados pelos mesmos antes que esses possam processar suas informações.
- 5xx – Erro do Servidor: Indicam que o servidor falhou em processar uma requisição aparentemente válida ou que a mensagem de requisição é desconhecida ou não-suportada pelo mesmo.
- 6xx – Falha Global: Indicam que a requisição não pode ser atendida por nenhum servidor do sistema.

3.9.5. **Operação do SIP**

A operação do SIP baseia-se no envio de solicitações e recebimento de respostas tanto na forma par-a-par quanto na cliente/servidor. Inicialmente, um agente usuário SIP envia uma mensagem de solicitação REGISTER a um servidor de registro que o responde com uma mensagem 200 (OK), como mostra a figura 3.20.

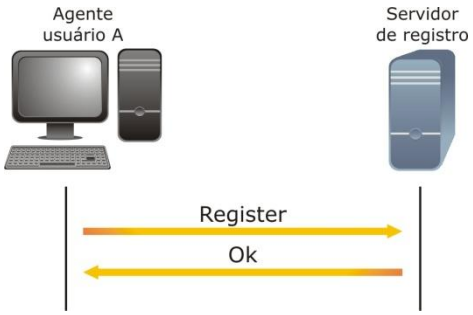


Figura 3.20 – *Registro de um agente usuário SIP*.

O processo de redirecionamento do SIP é mostrado na figura 4.21. Para localizar um usuário, o agente usuário origem A (professor@ufpe.br) envia por multidifusão (*broadcasting*) uma mensagem INVITE contendo a localização do agente usuário B (aluno@ufpe.br). Caso o agente usuário B não se encontre no local especificado, o usuário A receberá do servidor de redirecionamento uma mensagem 302 (MOVIDO TEMPORARIAMENTE) contendo a atual localização de B (aluno@casa.br). Então, o usuário A deve enviar ao servidor uma confirmação do recebimento da atual localização de

B (ACK). Feito isso, o usuário origem transmite por multidifusão uma nova mensagem INVITE contendo a nova localização do usuário B.

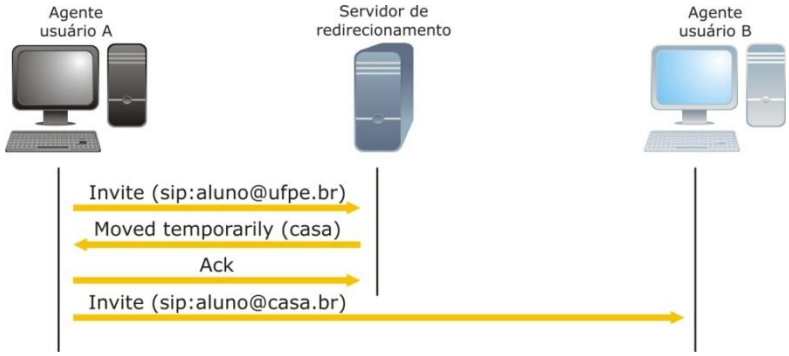


Figura 3.21 – *Redirecionamento de uma chamada SIP.*

O estabelecimento de chamadas se inicia com o usuário origem (A) enviando uma mensagem INVITE para o usuário destino (B). O usuário destino alerta o usuário origem (RINGING e TRYING) que a chamada está sendo processada e envia ao mesmo uma mensagem de confirmação 200 (OK). O usuário origem confirma o recebimento da resposta através de um ACK e o fluxo de mídia começa a ser trocado entre os usuários. Para finalizar a chamada qualquer uma das partes envia uma mensagem BYE que é respondida com um 200 (OK) e a sessão é encerrada. Todo o processo é ilustrado na figura 3.22.

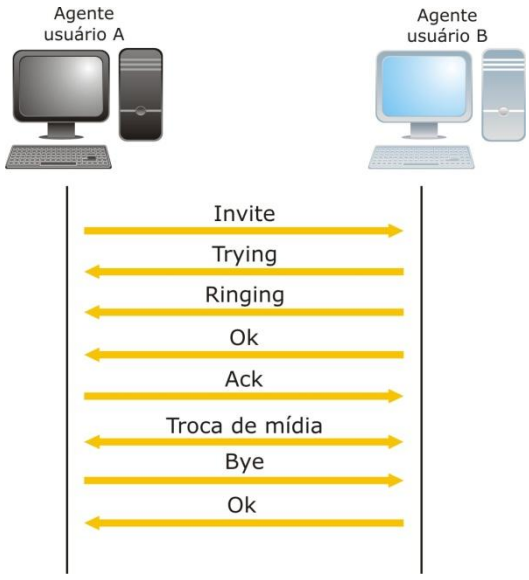


Figura 3.22 – *Estabelecimento de uma chamada SIP.*

Caso um servidor *proxy* seja utilizado, as mensagens para o estabelecimento da chamada devem ser enviadas do usuário origem para o mesmo (*proxy*) e encaminhadas pelo *proxy* ao usuário destino. Os pacotes de mídia e as mensagem para finalização da

chamada, no entanto, continuam a ser trocadas na forma par-a-par, como mostrado na figura 3.23.

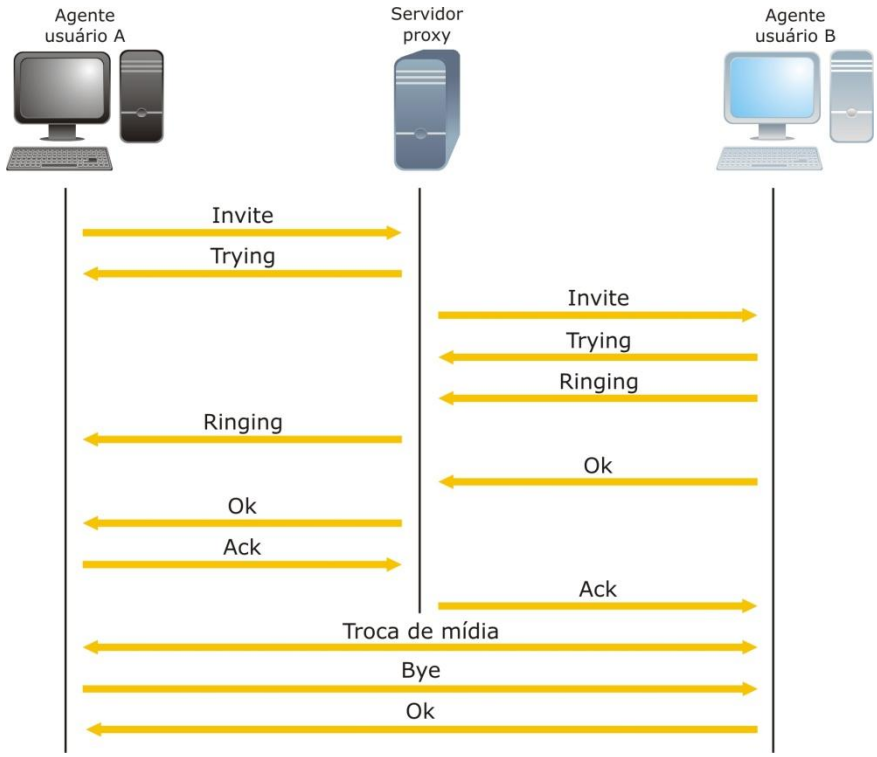


Figura 3.23 – *Estabelecimento de uma chamada SIP via servidor proxy.*

3.10. **Comparação entre H.323 e SIP**

Embora tanto o H.323 quanto o SIP sejam protocolos de sinalização para uso em sistemas de telefonia IP, permitam negociação de parâmetros, suportem multiconferências e operem sobre RTP e RTCP, existem entre eles diferenças, sobretudo na filosofia empregada no desenvolvimento de cada um. O H.323 é um padrão completo, que define com precisão o que é permitido e o que é proibido, o que o levou a ser um padrão extenso, complexo e rígido, difícil de adaptar às novas aplicações [4]. Por outro lado, o SIP possui as características típicas de um protocolo da Internet. É leve, modular, amigável e interage bem com os outros protocolos da Internet, porém não muito bem com o sistema telefônico convencional, necessitando muitas vezes do H.323 em seus *gateways*.

Algum dia, acredita-se que o SIP seja capaz de ser completamente independente e se torne o protocolo de um ambiente de rede convergente que vem se desenhando como o futuro da telefonia [18]. Enquanto isso não ocorre, alguns aspectos comparativos entre o H.323 e o SIP mostrados na tabela 3.6 podem ser analisados [4, 5, 18, 20].

Tabela 3.6 – *Comparação entre H.323 e SIP*.

Item	H.323	SIP
Filosofia	Desenvolvido para gerenciar chamadas de voz, multimídia e serviços suplementares através de recomendações específicas para cada tipo de serviço	Desenvolvido para estabelecer uma sessão entre dois terminais sem nenhuma relação especifica com algum tipo de mídia
Origens	Baseia-se na telefonia convencional PSTN. Adota protocolo de sinalização da ISDN Q.931.	Baseia-se na Internet. Adota a sintaxe das mensagens do HTTP
Clientes	Terminais inteligentes H.323	Agentes usuários inteligentes SIP
Servidores	*Gatekeeper* H.323	Servidores *registrar*, *redirect* e *proxy*
Projetado por	ITU-T	IETF
Compatibilidade com PSTN	Sim	Ampla
Compatibilidade com a Internet	Não	Sim
Arquitetura	Monolítica	Modular
Completeza	Pilha de protocolos completa	Lida apenas com a configuração
Negociação de parâmetros	Sim	Sim
Sinalização de chamadas	Q.931 sobre TCP	SIP sobre TCP ou UDP
Formato das mensagens	Binário	Texto ASCII
Transporte de mídia	RTP/RTCP	RTP/RTCP
Conferências	Suporta conferências com recursos audiovisuais e com troca de dados via especificação T.120	Suporta conferências básicas de áudio
Qualidade de Serviço	Gerenciamento de largura de banda, controle e admissão pelo *gatekeeper*. Realiza reserva de recursos	Utiliza de outros protocolos (RSVP, COPS, OSP) para implementar qualidade de serviço
Endereçamento	URL ou número de telefone (E.164)	SIP URI ou *e-mail*
Término de chamadas	Explícito ou por TCP	Explícito ou por *time-out*
SMS	Não	Sim
Cifragem das mensagens	Sim (H.235)	Sim (SSL, PGP)
Cifragem da sinalização	Sim (TLS/TCP)	Não definida
Conexão através de *firewalls*	Sim. Via *gatekeeper*	Sim. Via servidor *proxy*
Implementação	Extensa e complexa	Moderada
Escalabilidade	Não muita	Altamente escalável
Status	Extensamente distribuído	Boas perspectivas de êxito
Descoberta e admissão de usuários	Via função SIP REGISTER	Via protocolo RAS
Configuração de chamadas	Via função SIP INVITE	Via protocolo H.225
Negociação de capacidades	Via protocolo SDP (*Session Definition Protocol*)	Via protocolo H.245
Tamanho da documentação	1400 páginas	250 páginas

4. Telefonia IP em redes Par-a-Par

A recente união entre os sistemas de telefonia IP (VoIP) e as redes de comunicação par-a-par (P2P) causaram um enorme impacto na indústria de telecomunicações [43]. Tal tecnologia inovadora confere a esses sistemas a vantagem de atingir melhores níveis de qualidade de voz e a capacidade de ultrapassar os obstáculos impostos por NATs e *firewalls* [44].

O Skype, primeiro sistema comercial que implementa VoIP em redes P2P sobrepostas, é o aplicativo que apresentou o crescimento mais rápido da história da Internet, como mostrado na figura 4.1 [43]. A melhoria na qualidade do serviço prestado ao usuário final obtida pela associação desses dois sistemas fizeram o Skype ser considerado um divisor de águas dentre os sistemas de telefonia IP [45].

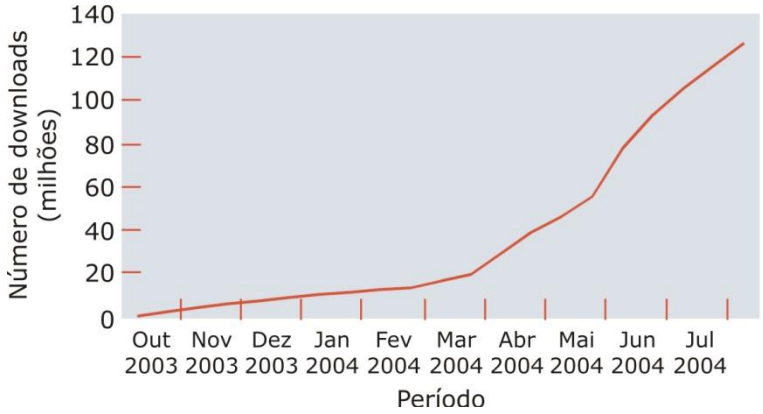


Figura 4.1 – *Número de downloads do Skype [43].*

Apesar de tudo aquilo que os sistemas de telefonia IP em redes P2P sobrepostas têm a oferecer (o que foi demonstrado pelo sucesso do Skype), os mesmos ainda operam de forma sub-ótima, indicando que muito de seu potencial ainda não foi explorado [2]. Alguns desses aspectos serão estudados no decorrer deste capítulo.

4.1. **Cenário**

A Internet é um recurso compartilhado, uma rede cooperativa composta por milhares de servidores e usuários espalhados pelo mundo [46]. No início de seu desenvolvimento, a Internet (então denominada ARPANET) possuía uma arquitetura inerentemente par-a-par [43]. Nesse cenário, um *host* podia se comunicar de igual para igual com qualquer outro *host* da rede. A rede possuia a função de servir de meio para o compartilhamento dados entre os principais *campi* universitários e centros de pesquisa do mundo.

Com o passar dos anos, no início da década de 1990, novas aplicações foram surgindo e os serviços oferecidos pela Internet (*web-browsing*, *e-mail* e transferência de arquivos) foram se distanciando do modelo P2P e se aproximando cada vez mais do modelo cliente/servidor (C/S) [43]. Tais aplicações necessitavam de um elemento centralizador de alto desempenho (o servidor) que fosse capaz de fornecer serviços e/ou recursos aos vários sistemas de baixo desempenho (os clientes) da rede. Os clientes simplesmente realizam solicitações e recebem respostas dos servidores, sem compartilhar nenhum de seus próprios recursos com os outros clientes.

Em um sistema cliente/servidor a comunicação sempre é realizada entre um cliente e um servidor, como mostrado na Figura 4.2. Apesar de apresentarem esse possível ponto vulnerável (o servidor), os sistemas centralizados são mais fáceis de gerenciar e possuem comportamento mais estável que os descentralizados.

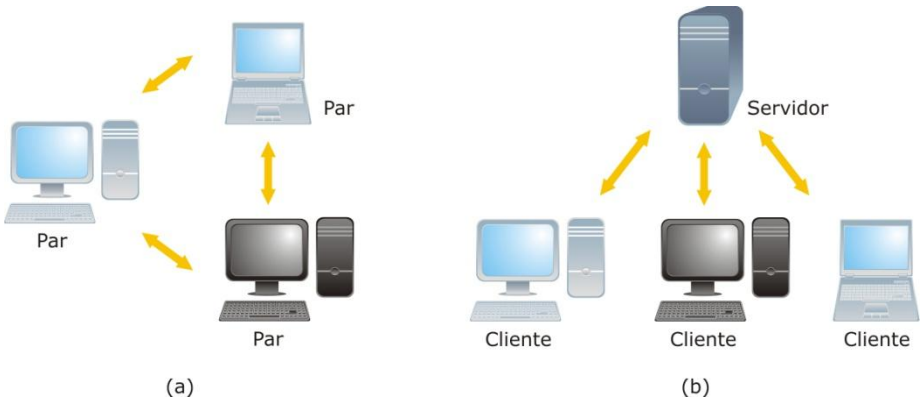


Figura 4.2 – *Topologias de rede. (a) Par-a-Par. (b) Cliente/Servidor.*

A cada instante mais pessoas unem-se à comunidade da Internet e novas aplicações são desenvolvidas. Os modernos avanços nas tecnologias de processamento, transmissão, codificação e multiplexação possibilitaram aos usuários dispor de uma imensa capacidade de processamento e transmissão. Com o crescimento do número de usuários conectados à

Internet e o aumento da capacidade dos computadores pessoais alguns papéis nesse mundo começaram a ser alterados.

Com o surgimento dos sistemas de compartilhamento de arquivos (Napster [47], KaZaa [48], eMule [49] e outros), o posicionamento do usuário como cliente ou servidor tornou-se mais sutil e os sistemas P2P voltaram a ter destaque. Os usuários tornavam-se então servidores de dados e recursos, fazendo com que suas máquinas se interconectassem diretamente uma com as outras, formando grupos de trabalho com supercomputadores virtuais, sistemas de arquivos e mecanismos de pesquisa criados pelo usuário [50].

A explosão dos sistemas P2P recordou o poder dos sistemas descentralizados, reforçando sua promessa de robustez, tolerância a falhas, liberdade na definição dos limites e alta escalabilidade. Essa mudança de paradigma se fez presente no desenvolvimento dos mais diversos sistemas de informação baseados na enorme interatividade do modelo P2P, que considera qualquer ponto da rede simultaneamente um cliente e um servidor [46].

4.2. **Topologias de rede**

Em um sistema distribuído, os dispositivos podem ser arranjados para desempenhar dada tarefa ou executar certa aplicação nas mais diversas formas. Essa flexibilidade possibilitou o desenvolvimento de várias topologias de rede, aplicadas para os mais diversos fins.

As diversas topologias de rede podem ser classificadas em [46]:

- Gerenciáveis: possuem a capacidade de administrar os nós da rede;
- Coerentes: oferecem confiabilidade no tráfego de informações entre os nós da rede;
- Extensíveis: apresentam facilidade em integrar diferentes topologias através da inclusão de nós;
- Tolerantes a falhas: Indica uma medida da quantidade de nós críticos para a operação do sistema e seus impactos na manutenção da atividade da rede;
- Seguras: apresentam facilidade no controle do acesso ao tráfego da rede;
- Escaláveis: permitem a inclusão de novos nós sem causar grandes impactos no desempenho da rede.

A seguir serão descritas algumas das topologias mais importantes da Internet.

4.2.1. **Topologia Centralizada (Cliente/Servidor)**

Os sistemas Cliente/Servidor são a forma mais tradicional de topologia, neles uma máquina é definida como servidora e as demais como clientes. Toda função e informação são centralizadas em um único servidor conectado diretamente aos clientes, como mostra a figura 4.3.

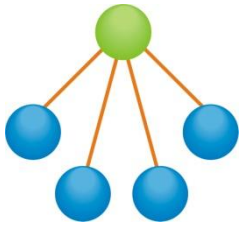


Figura 4.3 – *Topologia centralizada.*

A comunicação se dá com o processo cliente enviando solicitações ao processo servidor através da rede. A partir de então, o processo cliente deve aguardar uma resposta do processo servidor. O processo servidor, ao receber a solicitação do processo cliente, executa o que foi solicitado ou busca informações em sua base de dados e envia mensagens de resposta para as solicitações do processo cliente. Existem, portanto, dois processos distintos envolvidos, um na máquina cliente e outro na máquina servidora [4].

O processo cliente/servidor possui uma natureza centralizada, já que todas as solicitações de todos os clientes devem ser enviadas para o servidor e toda a comunicação flui sempre entre um cliente e um servidor.

4.2.2. **Topologia em anel**

Um único servidor central pode não suportar uma grande quantidade de conexões com seus clientes, assim, uma solução é usar um grupo de máquinas arranjadas em anel como mostrado na figura 4.4.

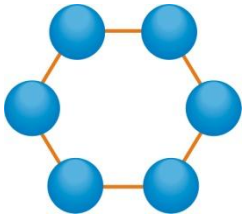


Figura 4.4 – *Topologia em anel.*

Essas máquinas se comportam como um servidor distribuído. A comunicação entre nós é coordenada para que eles operem identicamente, no entanto, suportem falhas em nós e sejam passíveis de balanceamento. Topologias em anel são geralmente criadas

considerando que todos os nós da rede estão próximos na rede ou que pertençam a uma área geográfica limitada [46].

4.2.3. **Topologia hierárquica**

Sistemas hierárquicos são considerados uma topologia distinta de sistemas distribuídos. O mais conhecido sistema hierárquico da Internet é o DNS (*Domain Name Service*) [51], no qual a autoridade sobre um nome registrado reside no servidor raiz, que repassa a autoridade para os demais servidores dos níveis inferiores. Tais sistemas se organizam em árvores, como mostrado na figura 4.5.

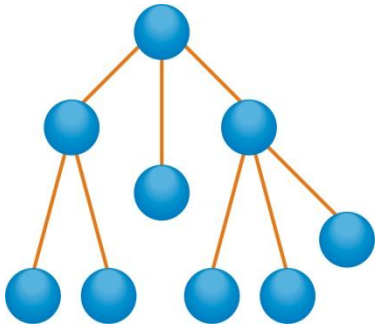


Figura 4.5 – *Topologia hierárquica.*

4.2.4. **Topologia descentralizada**

Sistemas descentralizados são aqueles em que todos os pontos se comunicam simetricamente através das mesmas regras. A própria arquitetura de roteamento da Internet é um sistema descentralizado que utiliza o *Boarder Gateway Protocol* (BGP) [52] para coordenar a conexão entre os pontos de sistemas autônomos. Alguns sistemas de compartilhamento de arquivos [53] e redes locais sem fio configuradas no modo *ad-hoc* são exemplos de topologia descentralizada. Os sistemas descentralizados podem ser estruturados ou não-estruturados, como mostra a figura 4.6.

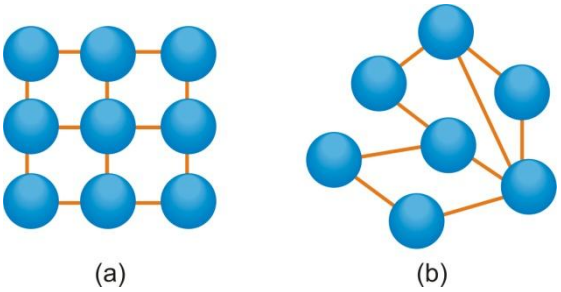


Figura 4.6 – *Topologia descentralizada. (a) estruturada. (b) não-estruturada.*

4.2.5. **Topologia híbrida (centralizada + descentralizada)**

Nesta topologia define-se uma arquitetura de sistemas centralizados aglomerados em sistemas descentralizados, como mostrado na figura 4.7. Nessa topologia existem nós com algum nível de diferenciação dos demais: os *super-nós*. Muitos nós possuem uma relação centralizada no super-nó, enviando todas as suas requisições para este de modo semelhante ao de uma topologia cliente/servidor. No entanto, os super-nós não são servidores estáticos, mas nós comuns que são promovidos dinamicamente através de algum critério pré-estabelecido (ex: largura de banda, capacidade computacional ou alcançabilidade na rede). Estes super-nós compõem uma rede descentralizada entre si, propagando as requisições dos nós ordinários.

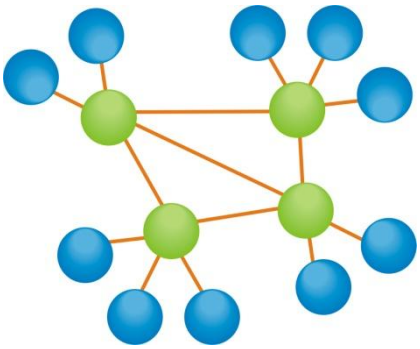


Figura 4.7 – *Topologia híbrida.*

A correspondência eletrônica é um exemplo de aplicação que utiliza esse tipo de topologia. Clientes de *e-mail* possuem uma relação centralizada com servidores de *e-mail*, mas estes compartilham os *e-mails* de forma descentralizada com o resto da rede. Esta topologia foi utilizada para o compartilhamento de arquivos no KaZaa [48] e para a comunicação de voz no Skype [1]. Um resumo das características das topologias apresentadas é mostrado na tabela 4.1 [46].

Tabela 4.1 – *Características das topologias apresentadas.*

Característica	Centralizada	Anel	Hierárquica	Descentralizada	Híbrida
Gerenciável	Sim	Sim	Parcial	Não	Não
Coerente	Sim	Sim	Parcial	Não	Parcial
Extensível	Não	Não	Parcial	Sim	Sim
Tolerante a falhas	Não	Sim	Parcial	Sim	Sim
Segura	Sim	Sim	Não	Não	Não
Escalável	Talvez	Sim	Sim	Talvez	Talvez

4.3. **Redes sobrepostas**

Numa perspectiva de alto nível, as redes par-a-par (*Peer-to-Peer*, P2P) são em sua maioria construídas utilizando o conceito de redes sobrepostas (*Overlay Networks*) [54]. De uma forma geral, as redes sobrepostas são redes que criam uma topologia virtual acima de uma topologia física [55].

Uma rede sobreposta é geralmente criada pela camada de aplicação, comportando-se como uma rede virtual localizada logicamente acima da rede física já existente. Tais redes são constituídas de enlaces virtuais que estabelecem estruturas lógicas de comunicação (caminhos virtuais) entre os nós do sistema. As ligações entre esses nós, do ponto de vista da rede física subjacente, são caminhos *multi-hop* (por múltiplos saltos) pelos quais o tráfego é roteado de forma transparente para a aplicação [56]. A figura 4.8 ilustra as relações físicas e lógicas (virtuais) existentes entre os elementos de uma rede sobreposta.

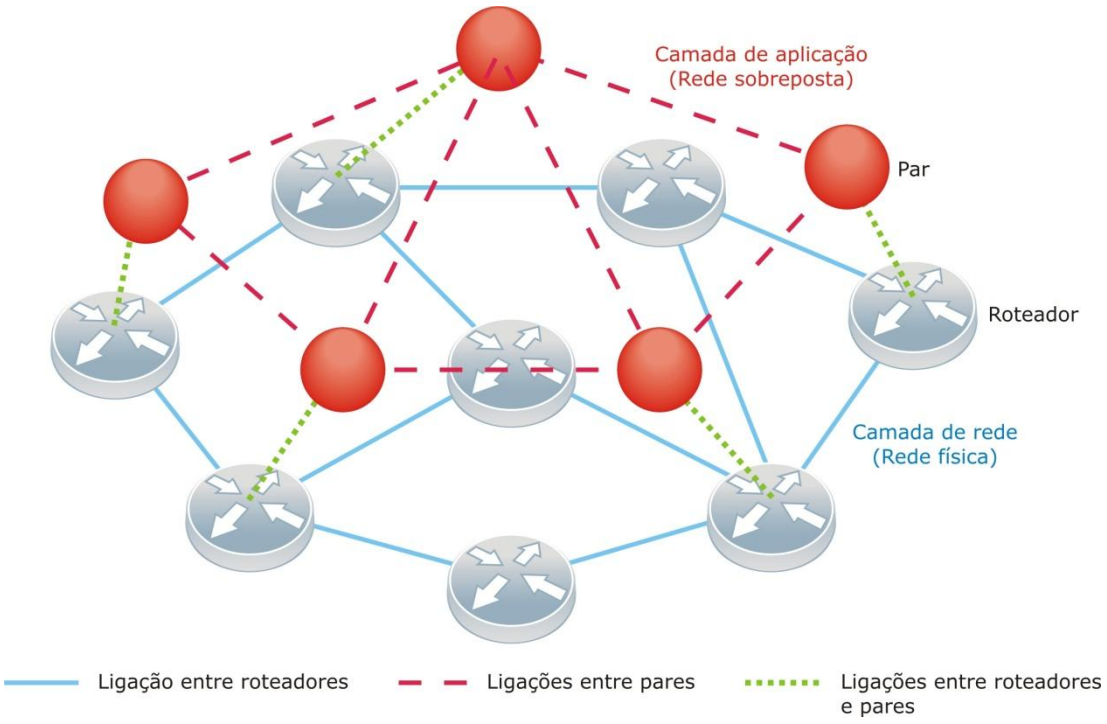


Figura 4.8 – *Estrutura de uma rede sobreposta.*

A arquitetura com nível mais alto de abstração criada pela rede sobreposta permite solucionar vários problemas que em geral são difíceis de serem tratados ao nível dos roteadores da rede física subjacente [57].

As redes sobrepostas possuem características independentes de roteamento e, em muitos casos, um esquema também independente de endereçamento. Essa flexibilidade permite à aplicação impor sua própria política de roteamento e de gerenciamento da QoS. Com isso,

o sistema pode atingir níveis de QoS superiores aos oferecidos pela Internet em relação aos parâmetros de atraso, *jitter*, taxa de entrega, perda, vazão e outros.

O conceito de rede sobreposta não é uma idéia nova, pois a própria Internet iniciou sua vida como uma rede de dados sobreposta ao sistema de telefonia, e mesmo hoje, um grande número de conexões da Internet continua sendo realizado através de linhas telefônicas. As redes sobrepostas são utilizadas para possibilitar e viabilizar o uso de protocolos, funcionalidades, serviços e aplicações não-suportados pelos roteadores da Internet atual [56]. Tais redes podem ser aplicadas em sistemas de compartilhamento de arquivos, multidifusão, telefonia IP e na criação de redes IPv6 sobre rotas que suportam apenas o IPv4.

Apesar de serem conceitos facilmente associados, uma rede sobreposta não implica necessariamente uma rede par-a-par. As VPNs (*Virtual Private Networks*), por exemplo, são redes sobrepostas que não são P2P, enquanto as redes sem fio 802.11 (WiFi) configuradas em modo ponto-a-ponto (*ad-hoc*) são redes P2P que não são sobrepostas.

4.4. **Redes Par-a-Par**

Em princípio, as redes par-a-par (P2P) são sistemas distribuídos que não possuem, nem dependem de nenhuma organização centralizada ou controle hierárquico, além de dispor a todos os seus integrantes as mesmas capacidades e responsabilidades [58, 59]. Essas redes são compostas basicamente por pares (*peers*) que são o conjunto de nós da rede que executam uma mesma aplicação. Portanto, a idéia central de um sistema P2P é a de igualdade entre seus integrantes.

Os sistemas P2P devem possuir características de auto-organização, balanceamento de carga, adaptação e tolerância a falhas distribuídas entre seus pares [60]. Além disso, devem, de maneira geral, obedecer aos seguintes requisitos [61]:

- Os nós devem possuir autonomia total ou parcial em relação a um servidor centralizado;
- Os nós devem possuir a capacidade de se comunicar diretamente uns com os outros;
- Os nós podem possuir conectividade e/ou endereços variáveis ou temporários;
- Ser escaláveis;
- Permitir que os nós estejam localizados nas bordas da rede;

- Possuir a capacidade de lidar com diferentes taxas de transmissão entre nós;
- Assegurar que os nós possuam capacidades iguais de fornecer e consumir recursos dos seus pares.

Estes requisitos caracterizam uma rede como par-a-par, mesmo que algumas das funções de controle estejam localizadas num servidor central (ponto de falha) [54].

De um modo geral, as redes par-a-par são compostas por nós que possuem um interesse comum (comunicar-se via VoIP, por exemplo) conectados através de uma mesma estrutura de comunicação. Como conseqüência dessas características tem-se que em uma rede par-a-par [54]:

- Os nós são conectados de forma aleatória e não há restrição sobre o número de nós que participam da rede;
- A conexão de um nó à rede se estabelece através de outro nó que já pertença à rede;
- Os nós podem entrar e sair da rede a qualquer momento, sem o prévio conhecimento dos demais membros.

Essas características indicam que há um movimento constante de entrada (*login*) e saída (*logoff*) de usuários numa rede par-a-par. A taxa relacionada a esse movimento é denominada *churn* [62], expressão adotada da economia que mede o nível de migração dos clientes de um fornecedor para outro, indicando seu grau de fidelidade [63]. O modelamento do *churn* é de grande importância para estimar o grau de disponibilidade dos serviços oferecidos e a taxa de falha dos nós nas redes par-a-par [62].

Em uma rede P2P os nós são frequentemente chamados de *servents*, palavra formada pela junção da primeira sílaba da palavra *server* (servidor) e da última sílaba da palavra *client* (cliente) da língua inglesa, ilustrando o fato dos mesmos serem tanto clientes quanto servidores da rede.

4.5. **Redes P2P sobrepostas em telefonia IP**

A telefonia possui inerentemente uma natureza par-a-par, já que seu objetivo final é possibilitar que usuários da rede possam realizar uma troca de voz entre si. No entanto, de acordo com o grau de centralização do sistema podemos classificar as redes telefônicas em [64]:

- Redes P2P puras: A busca por usuários, a troca de sinalização e a comunicação de voz são realizadas entre os pares do sistema.

- Redes P2P híbridas: A busca por usuários e a troca de sinalização é realizada em realação a um servidor centralizado, mas a comunicação de voz é realizada de forma par-a-par.

A rede de telefonia tradicional PSTN é um sistema de comunicação consolidado e está em operação há várias décadas, permitindo que usuários realizem chamadas de voz em qualquer parte do mundo. Essa rede foi concebida e construída de modo a trazer toda sua inteligência para o interior da rede (*switches* telefônicos) deixando os seus terminais (aparelhos telefônicos) sem qualquer responsabilidade em relação ao gerenciamento da chamada telefônica. Dessa forma, o sistema evoluiu lentamente, já que antes dos terminais fazerem uso de quaisquer melhorias, estas deveriam ser implantadas em todos os servidores centrais da rede [64]. Todo o plano de controle do sistema é realizado de uma forma *par-a-par híbrida*, ou seja, ainda são necessários servidores centrais para realizar o controle das chamadas e outros procedimentos necessários para o gerenciamento da sessão estabelecida [64].

Por outro lado, a Internet possui um conceito completamente oposto. Toda a inteligência da rede IP está localizada em seus terminais (PCs), enquanto os dispositivos da rede propriamente dita (roteadores) simplesmente encaminham pacotes o mais rapidamente possível, muitas vezes sem o menor conhecimento do conteúdo do mesmo.

Apesar das divergências conceituais entre as formas de operação da rede de telefonia convencional e da Internet, a flexibilidade das redes IP permitiu que em menos de uma década após sua popularização, os mais diversos tipos de serviço fossem oferecidos aos seus usuários, inclusive os serviços de telefonia.

As primeiras implementações de sistemas de telefonia VoIP possuíam uma topologia P2P híbrida, seguindo o modelo telefônico convencional, geralmente implementados com o uso dos protocolos H.323 [34] ou SIP [10, 41]. No entanto, com o lançamento do Skype [1], pela primeira vez na história da telefonia, este modelo foi desafiado com a introdução das redes sobrepostas par-a-par (*Overlay Peer-to-Peer Networks*). Este novo modelo possibilitava que muitas das funções necessárias para o gerenciamento das sessões telefônicas fossem distribuídas entre os usuários (pares) da rede [64].

As redes sobrepostas oferecem aos sistemas uma flexibilidade no roteamento dos pacotes que permite contornar rotas problemáticas na Internet por meio do roteamento indireto (*Relaying*) por um de seus pares. Através dessa arquitetura, os pares podem ainda implementar uma variação dos protocolos STUN (*Simple Traversal of UDP Through*

NATs) [65] ou TURN (*Traversal Using Relay NAT*) [66] para transpor os obstáculos impostos pelo NAT.

O Skype, além de ser gratuito, conseguiu através de sua arquitetura inovadora atingir níveis de qualidade de voz comparáveis aos do sistema de telefonia fixa convencional. Outros aplicativos, como o Google Talk, Yahoo! Messenger e MSN, já ofereciam serviços de comunicação de voz, mas com uma qualidade baixa e com a limitação de não operar quando os usuários estavam por trás de NATs [43].

4.6. Estratégias para melhora do desempenho de VoIP utilizando redes sobrepostas

A possibilidade de interagir no esquema de roteamento dos pacotes e de utilizar novas políticas para lidar com as perdas são algumas das maiores vantagens oferecidas pelo uso das redes sobrepostas. Várias estratégias foram desenvolvidas no intuito de melhorar o desempenho dos sistemas e a qualidade dos serviços através da utilização das redes sobrepostas. Tais estratégias visam a aplicação em cenários com aspectos críticos de QoS (como nos sistemas VoIP) e operam na tentativa de minimizar a degradação na qualidade de voz gerada por perdas e atrasos.

4.6.1. Roteamento por melhor caminho e por múltiplos caminhos

A infra-estrutura de roteamento da Internet não garante uma entrega de pacotes livre de perdas entre seus usuários. As perdas são observadas devido a fatores como congestionamentos, falhas nos enlaces e anomalias no roteamento.

Roteadores podem levar dezenas de minutos para se reestabelecerem após uma falha, causando perda de pacotes durante esse período [67]. Para lidar com a perda de pacotes, os protocolos de transporte, em geral, realizam retransmissões ou diminuem a taxa de transmissão, a um custo de diminuir o *throughput* e a aumentar a *latência*, que é definida como o tempo gasto por uma unidade de informação para transitar de um ponto da rede a outro [12].

Por outro lado, através do uso das redes sobrepostas e de otimizações na camada de rede, novas estratégias podem ser utilizadas para melhorar a probabilidade de entrega dos pacotes nas redes IP. Essas estratégias consistem em realizar medições para determinar o

melhor caminho para o encaminhamento do pacote na rede sobreposta ou enviar os pacotes de forma redundante através de múltiplos caminhos. A antiga ARPANET otimizava a seleção de caminhos, porém essa característica foi removida por razões de escalabilidade e estabilidade da rede. Os dois esquemas são ilustrados na figura 4.9.

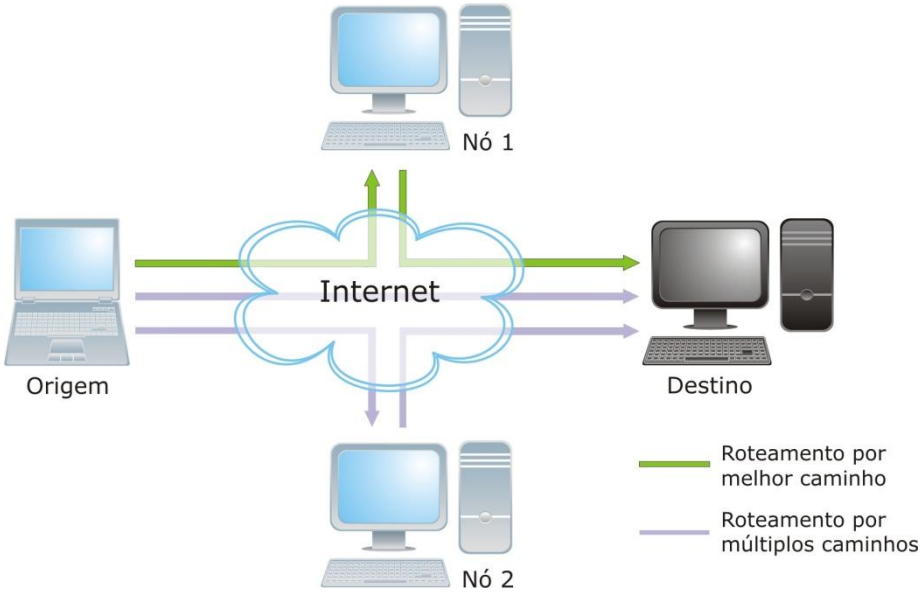


Figura 4.9 – *Roteamento por melhor caminho baseado em medições ou por múltiplos (2) caminhos.*

No esquema de roteamento por melhor caminho, o caminho indireto via nó 1 mostrado na figura 4.9 (suposto melhor caminho) é determinado a partir de medições ativas, cujo critério de escolha pode ser, por exemplo, aquele que minimiza o atraso ou a perda de pacotes. Por outro lado, no esquema de roteamento por múltiplos caminhos (ou roteamento com redundância) existe a transmissão de pacotes redundantes, ou seja, cópias de "*backup*" dos pacotes enviados são transmitidas para aumentar a probabilidade de entrega dos mesmos. Essas cópias redundantes podem ser enviadas tanto por vários caminhos distintos (como no esquema "caminho direto + caminho indireto via nó 2" mostrado na figura 4.9) quanto pelo mesmo caminho (transmitindo um pacote, e em seguida, sua cópia, ambos pelo mesmo caminho).

Tais esquemas se baseiam na hipótese que as falhas e perdas em diferentes caminhos da rede são descorrelacionadas umas com as outras. Infelizmente, a probabilidade condicional de perda de um segundo pacote, dado que o primeiro foi perdido é igualmente alta tanto no caso de se enviar ambos pelo mesmo caminho (70%) quanto por caminhos diferentes (60%) [12]. Na verdade, os caminhos na Internet são fortemente correlacionados. Andersen cita em seu trabalho um episódio em que um acidente de trem no Howard Street Tunnel,

Baltimore em 2001, prejudicou o serviço de quatro dos maiores *backbones* americanos cujas fibras ópticas passavam pela mesma localização física [12]. Em todo caso, apesar de não sanar completamente o problema das perdas, algum ganho é obtido através do uso dessas estratégias.

O roteamento por melhor caminho baseia-se em realizar medições de características como atraso ou perdas periodicamente, roteando os pacotes de forma dinâmica através do melhor caminho descoberto. Frequentemente, esta medição é realizada em escalas de tempo longas, porém, desconsiderar os efeitos das variações de curto prazo da rede resultam em uma seleção sub-ótima das rotas tomadas pelos pacotes [58, 68].

Hilt e colaboradores propuseram um esquema de roteamento por melhor caminho utilizando redes sobrepostas sem a necessidade de realizar medições ativas durante a chamada, como mostra a figura 4.10 [13].

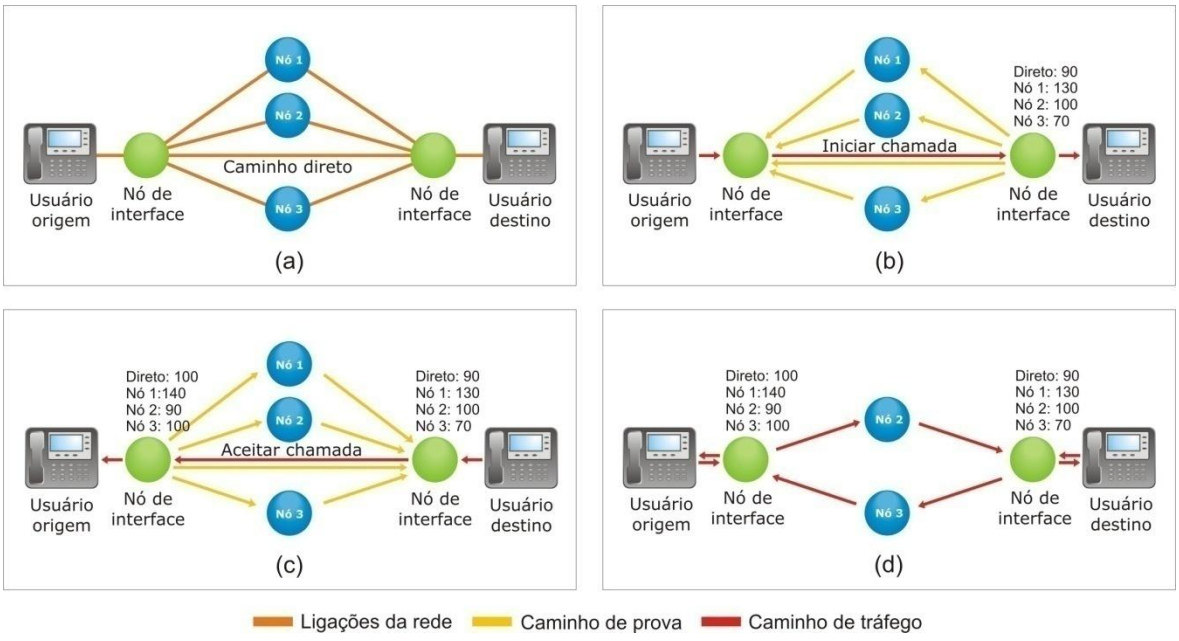


Figura 4.10 – *Estabelecimento de rotas durante a sinalização. (a) Estrutura da rede. (b) Determinação da melhor rota pelo usuário destino. (c) Determinação da melhor rota pelo usuário origem. (d) Troca de mídia pela melhor rota entre os usuários (assimetricamente).*

A figua 4.10 ilustra que o melhor caminho é escolhido de acordo com estimativas observadas pelos usuários durante a fase de sinalização que precede uma chamada VoIP. Sejam dois usuários alcançáveis pelos caminhos mostrados em (a). Durante o início da sinalização da chamada, caso a comunicação via caminho direto não ofereça um grau de QoS necessário à comunicação VoIP, o usuário destino inicia um processo de medição da qualidade dos possíveis caminhos alternativos disponíveis para a troca de pacotes de voz

(b). Se o destino aceitar a chamada, a origem também realiza uma série de medições (c). De posse desses dados, ambas as partes podem selecionar o melhor caminho para encaminhar seus pacotes de voz (d). Vale ressaltar que, em geral, devido às assimetrias da redes IP, os caminhos de ida e de volta são distintos. Como o inicio da troca de pacotes de mídia depende de uma interação do usuário (atendimento da chamada), há tempo mais do que suficiente para o sistema realizar as medições necessárias e definir a melhor rota a ser tomada pelos pacotes.

Durante o curso da chamada, relatórios periódicos são trocados entre os participantes via protocolo RTP. Esses relatórios informam ao sistema o nível de qualidade da chamada, em relação a atrasos e perdas, e podem ser utilizados como indicadores do momento em que um outro caminho deve ser escolhido para o encaminhamento dos pacotes. O sistema também pode definir esse instante através de medições passivas do atraso e *jitter* dos pacotes que estão sendo recebidos por um dado caminho [13].

Utilizando tal estratégia, o nível de qualidade da chamada aumenta, porém, também aumenta o *overhead*, embora em limites toleráveis. Os resultados obtidos no trabalho de Hilt indicam um aumento total no overhead de apenas 3,13% (1,39% pelas medições e 1,74% pelo roteamento dos pacotes) para o *codec* G.729 e de desprezíveis 0,42% (0,19% para as medições e 0,23% para o roteamento dos pacotes) para o *codec* G.711 [13].

O roteamento por múltiplos caminhos, por outro lado, baseia-se na existência de rotas redundantes ligando dois usuários da rede. O uso dessa estratégia permite que mais pacotes sejam recuperados em momentos de perda.

Andersen e colaboradores realizaram um estudo comparativo de diversos esquemas de roteamento entre 30 *hosts* dos Estados Unidos, Canadá, Holanda, Suécia, Coréia do Sul e Inglaterra. Dentre eles estavam universidades, empresas privadas e provedores de Internet de diversos tamanhos utilizando desde conexões OC3 (*Optical Carrier 3* – 155,52 Mbps) a *cable modems* e DSL. Os esquemas testados foram [12]:

- Roteamento direto (Dir): Envio de um simples pacote através de um caminho direto da Internet;
- Roteamento otimizado por mínima perda (Per): Seleção de melhor caminho baseado na minimização das perdas de pacote. Necessita de *overhead* para realizar as medições.
- Roteamento otimizado por mínimo atraso (Atr): Seleção de melhor caminho baseado na minimização do atraso. Evita enlaces completamente falhos.

- Roteamento múltiplo direto e aleatório (Dir-Rand): Roteamento por múltiplos caminhos no qual um pacote é encaminhado pelo caminho direto da Internet e uma cópia através de um nó da rede sobreposta que é selecionado aleatoriamente. Não há intervalo entre a transmissão dos pacotes.
- Roteamento múltiplo seletivo (Atr-Per): Teoricamente o melhor dos dois mundos. Roteamento por múltiplos caminhos no qual um pacote é encaminhado pelo melhor caminho baseado na minimização das perdas e uma cópia por aquele que minimiza o atraso. Não há intervalo entre a transmissão dos pacotes.
- Roteamento múltiplo direto (D-D): Roteamento múltiplo com redundância no qual tanto o pacote quanto sua cópia são encaminhados pelo mesmo caminho, um após o outro.
- Roteamento múltiplo direto com retardo de 10 ms (D-D 10 ms): Roteamento múltiplo com redundância no qual tanto o pacote quanto sua cópia são encaminhados pelo mesmo caminho, mas separados por um intervalo de 10ms.
- Roteamento múltiplo direto com retardo de 20 ms (D-D 20 ms): Roteamento múltiplo com redundância no qual tanto o pacote quanto sua cópia são encaminhados pelo mesmo caminho, mas separados por um intervalo de 20 ms.

Segundo o trabalho de Andersen e colaboradores, na estratégia de seleção de melhor caminho cada nó testa todos os outros a cada 15 segundos e o melhor caminho é selecionado após 100 testes. Na estratégia de transmissão por múltiplos caminhos são utilizados dois caminhos por transmissão. Todos os testes foram realizados em uma janela de uma hora de duração. Os principais resultados encontrados por Andersen e colaboradores foram [12]:

- As perdas em caminhos alternados não são independentes. Se um pacote é perdido a probabilidade condicional que o segundo pacote também seja é da ordem de 60%.
- A perda média de pacotes na Internet é relativamente baixa (0,42%). Em 30% do tempo a taxa de média de perda ficou abaixo de 0,1% e em 68% do tempo abaixo de 0,2%. No tempo restante a taxa média de perda foi de 13%.
- O uso de roteamento por múltiplos caminhos reduz essa taxa média de perda a 0,26%, o que diminui o número de retransmissões e a latência do sistema.
- A seleção de caminhos melhora o desempenho da transmissão por múltiplos caminhos.

A tabela 4.2 mostra o número de chamadas em cada faixa de perda em função do tipo de roteamento empregado durante o período mais altas perdas do ensaio. Percebe-se que os roteamentos por melhor caminho (medições reativas buscando minimizar perdas e/ou atrasos) é mais eficiente em baixas taxas de perda enquanto o roteamento por múltiplos caminhos apresenta seu melhor desempenho em altas taxas de perda [12].

Tabela 4.2 – *Número de chamadas em cada faixa de perda para cada um dos esquemas [12].*

	Direto	**Redundância simples**			**Reativo**		**Múltiplo**	**Combinado**
Perda (%)	**Dir**	**D-D**	**D-D 10ms**	**D-D 20ms**	**Atr**	**Per**	**Dir-Rand**	**Atr-Per**
> 0	8817	5183	4024	3832	10695	7066	3846	3353
> 10	1999	1361	1291	1275	1716	1362	1236	1134
> 20	962	799	796	783	849	791	793	757
> 30	630	585	591	575	604	573	579	563
> 40	486	480	481	465	484	468	468	451
> 50	379	377	367	359	363	359	369	334
> 60	255	251	245	249	231	219	235	215
> 70	130	130	130	128	118	106	125	114
> 80	74	73	65	64	57	59	60	56
> 90	31	31	37	30	16	31	28	16

A tabela 4.3 mostra os percentuais de perda e o atraso médio para cada esquema de roteamento analisado. A latência ***Lat*** é medida em milisegundos, ***p(P1)*** e ***p(P2)*** são as respectivas taxas de perda do primeiro pacote e do segundo pacote, ***p(P)*** é a probabilidade de perda total e ***p(P2|P1)*** é a probabilidade condicional de se perder o segundo pacote dado que o primeiro pacote foi perdido.

Tabela 4.3 – *Percentual de perdas de pacote e atraso médio fim-a-fim [12].*

Tipo de roteamento	**p(P1)**	**p(P2)**	**p(P)**	**p(P2|P1)**	**Lat (ms)**
Direto	0,42%	-	0,42%	-	54,13
Otimizado por mínimo atraso	0,43%	-	0,43%	-	48,01
Otimizado por mínima perda	0,33%	-	0,33%	-	55,62
Múltiplo (direto + aleatório)	0,41%	2,66%	0,26%	62,47%	51,71
Múltiplo otimizado (mínima perda + mínimo atraso)	0,43%	1,95%	0,23%	55,08%	46,77
Múltiplo direto sem intervalo entre pacotes	0,42%	0,43%	0,30%	72,15%	54,24
Múltiplo direto com intervalo de 10ms	0,41%	0,42%	0,27%	66,08%	54,28
Múltiplo direto com intervalo de 20ms	0,41%	0,41%	0,27%	65,28%	54,39

Nota-se que o esquema ***Otimizado por mínima perda*** reduz a taxa de perda de 0,42% para 0,33%, sem aumentar consideravelmente a latência do sistema. Utilizando o esquema

Múltiplo (direto +aleatório) se obtém uma redução de 40% nas perdas e uma diminuição na latência do sistema. Utilizando a combinação dos métodos de seleção e transmissão por múltiplos caminhos ***Múltiplo otimizado (mínima perda + mínimo atraso)*** se obtém o melhor resultado, indicando que esses métodos tiram vantagens de diferentes situações.

A tabela 4.3 mostra ainda que a probabilidade condicional de perda do segundo pacote dado que o primeiro foi perdido é menor na estratégia de enviar pacotes por caminhos ditintos que na estratégia de enviar os mesmos por um mesmo caminho. Essa probabilidade condicional é bastante superior (60 - 70%) à mais alta probabilidade de perda de pacotes da rede nesse período (4 - 6%).

No pior período da medição, cerca de 95% dos pacotes se encontram dentro do limite de 150 ms estabelecido para o atraso fim-a-fim requeridos para aplicações de telefonia IP. Com o uso da estratégia ***Múltiplo otimizado (mínima perda + mínimo atraso)*** esse percentual sobe para cerca de 98% o que mostra a viabilidade de tal estratégia para uso em sistemas VoIP [12].

Os benefícios trazidos pelo roteamento por melhor caminho são proporcionais à taxa na qual o sistema realiza as medições. Sua eficiência é limitada pelo atraso do melhor caminho, visto que em geral o caminho com menor perda não é o caminho com menor retardo [57]. Se cada um dos n caminhos ligando a origem e o destino possuem uma probabilidade de perda p_i, a probabilidade de perda utilizando roteamento por melhor caminho é dada por [12]:

$$p_{melhor\ caminho} = min_i(p_i). \quad (1)$$

O custo de realizar as medições é fixo e depende do tamanho da rede. Para uma rede com N nós cada host deve enviar e receber $O(N^2)$ dados. Esse custo pode se tornar alto ou baixo em função da quantidade de tráfego da rede [12].

Por outro lado, o roteamento por múltiplos caminhos com grau de redundância R, pode garantir a entrega do pacote mesmo com a falha de $(R-1)$ caminhos por nó, se os mesmos forem independentes. A diminuição na taxa de perda, no caso de N caminhos independentes é dada por [12]:

$$p_{múltiplos\ caminhos} = \prod_{i=1}^{N} p_i. \quad (2)$$

O custo de enviar dados por múltiplos caminhos se reflete em um aumento do tráfego proporcional ao número de caminhos utilizados para a transmissão. Esse custo não depende do tamanho da rede [12]. Além disso, o alto grau de correlação entre os caminhos

não permite uma diminuição da ordem de p^N na taxa de perdas. No entanto, melhorias significativas são observadas, sobretudo em caminhos com altas taxas de perdas [12].

4.6.2. **Retransmissão seletiva**

Com o intuito de diminuir a latência, os aplicativos de VoIP utilizam o protocolo de transporte não-confiável UDP. No entanto, a falta de confiabilidade deste protocolo expõe os pacotes às perdas e falhas que ocorrem na Internet.

Através da retransmissão seletiva em uma rede sobreposta pode-se aumentar o desempenho dos aplicativos VoIP nos instantes em que o serviço de entrega da Internet falhar, adicionando um *overhead* mínimo quando nenhuma perda ocorrer [69].

A subdivisão do caminho fim-a-fim entre a origem e o destino em um certo número de saltos na rede sobreposta traz uma série de vantagens. Primeiramente é possível recuperar pacotes perdidos mesmo com a crítica restrição temporal imposta pela interatividade do serviço VoIP. Utilizando redes sobrepostas, mesmo que o limite aceitável do atraso impeça a retransmissão do pacote perdido pelo caminho fim-a-fim, é possível realizar a recuperação localmente, apenas no salto em que ocorreu a perda [69]. Geralmente, os enlaces na rede sobreposta apresentam um menor RTT (*Round Trip Delay*) que o apresentado pelo caminho fim-a-fim.

Em segundo lugar, o algoritmo de roteamento utilizado pode ser ajustado para evitar os enlaces da rede sobreposta que se apresentem congestionados, com altas taxa de perda ou indisponíveis. Devido à especificidade do algoritmo, este pode se adaptar as condições da rede em uma escala de tempo uma ordem de grandeza menor que o da Internet, minimizando o impacto das variações da rede nos fluxos de voz [69].

A figura 4.11 mostra como a taxa de perda de pacotes influencia a qualidade da voz percebida pelo usuário. O *codec* utilizado foi o G.711 com cancelamento de perda de pacote (PLC). O gráfico é traçado para surtos de diferentes níveis. O nível de referência para voz com qualidade similar a da PSTN é 4,0.

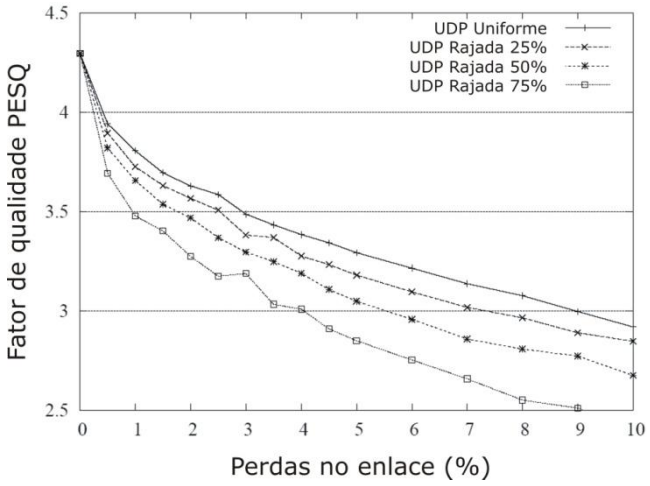


Figura 4.11 – *Índice de qualidade PESQ em função da perda de pacotes na Internet [69].*

Percebe-se que no melhor caso, quando as perdas são uniformemente distribuídas, o G.711 consegue manter o nível de qualidade apenas para taxas de perda inferiores a 0,5%.

O protocolo de retransmissão seletiva proposto por Amir e colaboradores tem como objetivo melhorar o desempenho dos sistemas de voz sobre IP sem aumentar demasiadamente o *overhead* da rede [69]. Batizado pelos autores de *Real-Time Recovery Protocol* (Protocolo para Recuperação de Pacotes em Tempo Real), este protocolo recupera os pacotes perdidos localmente no trecho em que ocorreu a perda, se e somente se, há a possibilidade de enviar o mesmo em tempo hábil para o próximo trecho. Esse protocolo opera da seguinte forma, ilustrada pela figura 4.12 [69]:

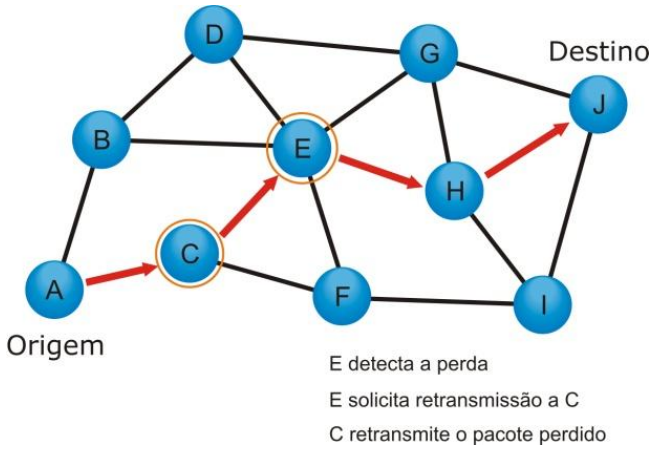


Figura 4.12 – *Funcionamento do protocolo de retransmissão seletiva.*

Supondo que o caminho traçado pelos pacotes é definido como A-C-E-H-J e a perda ocorre no trecho C-E, tem-se que:

- Cada nó da rede sobreposta possui um *buffer* circular no qual os pacotes transmitidos são mantidos por um tempo igual ao máximo atraso suportado pelo

codec. Os pacotes antigos são descartados se o tempo expirar ou o *buffer* ficar cheio;

- Os nós encaminham os pacotes assim que os recebem, mesmo fora de ordem;
- Ao detectar a perda de um pacote o nó (E) solicita ao nó anterior (C) a retransmissão do mesmo uma única vez. Através dessa estratégia de confirmação negativa (*negative acknowledgements*), limita-se a quantidade de tráfego quando pacotes não forem perdidos;
- Quando um nó da rede sobreposta (C) recebe um pedido de retransmissão, ele verifica o seu *buffer* circular. Se ainda possui o pacote o retransmite, caso contrário, nada faz. Um mecanismo de *token* regula o número máximo de retransmissões em relação ao número de pacotes enviados. Isso limita as retransmissões em enlaces com alta taxa de perda;
- Se um nó recebe o mesmo pacote duas vezes, apenas o primeiro a ser recebido será encaminhado ao seu destino.

Os testes realizados por Amir e colaboradores utilizaram o *codec* G.711 na rede sobreposta de código aberto *Spines* [70]. O protocolo não envolve confirmações positivas, portanto nenhum tráfego extra é gerado se não há perda de pacotes. Além disso, não há *time-outs* e o protocolo nunca é bloqueado para a recuperação de pacotes perdidos. Em contrapartida este não é um protocolo totalmente confiável e pode ocorrer de alguns pacotes serem perdidos [69].

Supondo que a probabilidade de perdas na rede seja uniforme e igual a p, os pacotes não são recuperados em duas situações principais:

- Um pacote é perdido (probabilidade p), o pacote seguinte é recebido (assim a perda é detectada: probabilidade $1-p$) e uma solicitação de retransmissão é realizada, mas perdida (probabilidade p). Isso ocorre com probabilidade P_1 igual a:

$$P_1 = [p \cdot (1-p) \cdot p] = (p^2 - p^3). \quad (3)$$

- Outro caso relevante ocorre quando o pacote é perdido (probabilidade p), o pedido de retransmissão é recebido (probabilidade $1-p$), mas a retransmissão em si é perdida (probabilidade p), o que ocorre com probabilidade P_2 igual a:

$$P_2 = [p \cdot (1-p) \cdot (1-p) \cdot p] = (p^2 - 2p^3 + p^4) \cong (p^2 - 2p^3), \quad (4)$$

supondo o termo p^4 desprezível.

Somando as duas probabilidades ($P_1 + P_2$), se obtém a probabilidade de perda P do protocolo:

$$P = (P_1 + P_2) = (2p^2 - 3p^3). \quad (5)$$

Isso indica que: para uma probabilidade de perdas p num *link* com retardo T e um tempo de detecção de perdas Δ, uma fração de $(1 - p)$ pacotes chegam no instante T, $(p - 2p^2 + 3p^3)$ são retransmitidos e chegam no instante $(3T + \Delta)$ e $(2p^2 - 3p^3)$ são perdidos pelo protocolo. O tempo Δ depende do intervalo entre pacotes, que é da ordem de 20 ms para a maioria dos codecs.

A figura 4.13 mostra que o protocolo reduz a taxa de perda efetiva do sistema para 2% quando há uma perda de 10% no enlace. O impacto na qualidade da voz percebida pelo usuário é mostrado na figura 4.14.

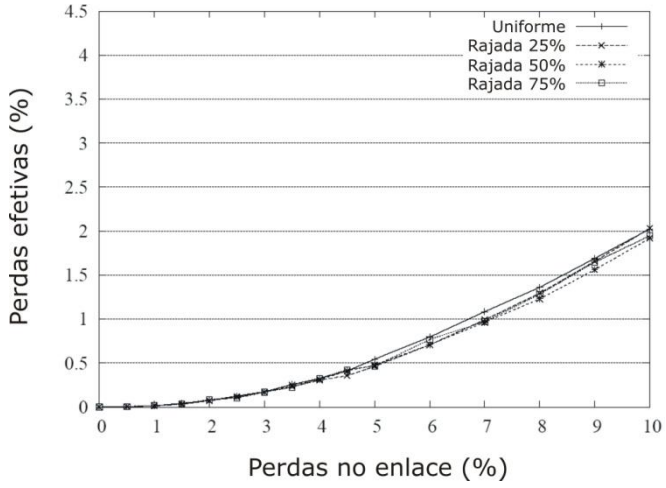


Figura 4.13 – *Perda de pacotes do protocolo versus taxa de perda do link [69].*

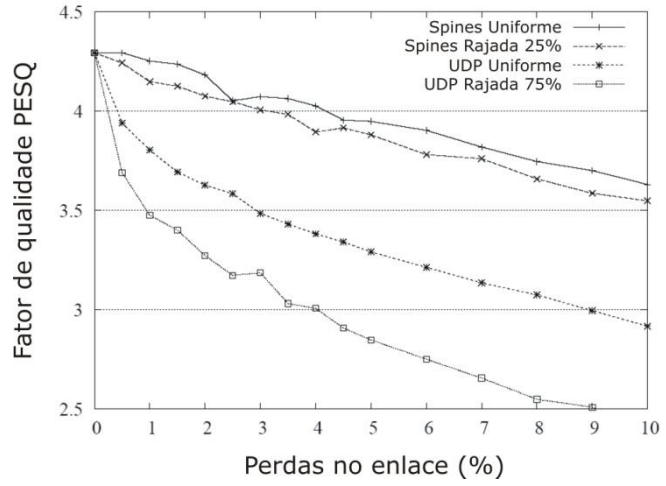


Figura 4.14 – *Comparação do índice de qualidade de voz PESQ do codec G.711 com roteamento através da rede sobreposta Spines e da Internet (UDP) em função da perda de pacotes [69].*

O *codec* G.711 que apresentava uma qualidade de voz abaixo do nível da PSTN para uma taxa de perdas na rede de apenas 0,5% (figura 4.11), agora se mantém com um alto nível de qualidade mesmo quando a rede apresenta uma perda de pacotes de 3,5%. Tudo isso se deve ao protocolo implementado na rede sobreposta que faz com que uma fração dos pacotes perdidos seja recuperada e a taxa efetiva de perda vista pelo *codec* seja diminuída.

4.6.3. **ASAP (Autonomous System Aware Peer-Relay Protocol)**

A qualidade de voz numa chamada VoIP é considerada satisfatória se apresenta um MOS acima de 3,6 e um atraso fim-a-fim abaixo de 150 ms [71, 72]. O Skype é considerado um sistema de referência de telefonia IP baseado em redes P2P sobrepostas devido ao seu pioneirismo e à alta qualidade de voz oferecida ao usuário [44]. O Skype utliza um esquema par-a-par tanto para encontrar usuários quanto para encaminhar pacotes de voz. Esse aplicativo pode encaminhar os pacotes diretamente ao destino (*major path*) ou indiretamente através de outros pares da rede, em um salto (*one-hop*) ou dois saltos (*two-hops*).

No entanto, esse aplicativo possui limitações de desempenho que podem ser associadas a três principais fatores [2]:

- Muitas de suas seleções de pares para o encaminhamento de pacotes são sub-ótimas;
- O tempo de espera para selecionar um desses nós pode ser excessivamente longo;
- É realizado um número desnecessário de medições, gerando um excesso de tráfego que compromete a escalabilidade do sistema.

Encontrar um modo eficiente de selecionar os nós da rede sobreposta que farão o encaminhamento dos pacotes é um desafio constante.

A Internet é um aglomerado de redes controladas por entidades chamadas *Autonomous Systems* (AS). Um AS é uma partição da Internet, ou seja, um conjunto de roteadores que compõem uma rede administrada tecnicamente por uma organização ou um conjunto de organizações que possuam uma política de roteamento única e bem-definida [73]. Os AS utilizam o IGP (*Interior Gateway Protocol*) para realizar o roteamento dentro de suas próprias redes e o EGP (*Exterior Gateway Protocol*) para rotear os pacotes para outros AS. Esses protocolos são variações do BGP (*Boarder Gateway Protocol*) descrito na RFC4271 [52].

O roteamento na Internet depende das relações comerciais e contratuais entre os AS [74]. Devido ao fato dos AS forçarem suas próprias políticas de roteamento, o caminho direto ligando dois usuários através da Internet não é necessariamente o caminho ótimo dentre todas as possíveis rotas entre os mesmos [2]. Encaminhar os pacotes indiretamente através de um nó de uma rede sobreposta pode ser mais rápido do que o enviar através do caminho direto da Internet. Ren e colaboradores analisaram que o roteamento indireto via rede sobreposta pode ser vantajoso em duas situações principais [2]:

- Se um AS do caminho direto estiver congestionado ou falhar;
- Se existe um AS *multi-homed* que ofereça uma rota mais rápida para os pacotes.

A figura 4.15 ilustra as duas situações. Nesses casos, se pode fazer uso das relações entre os AS para se estabelecer um critério de roteamento que seja capaz de selecionar o nó apropriado da rede sobreposta para o encaminhamento dos pacotes com uma sobrecarga mínima.

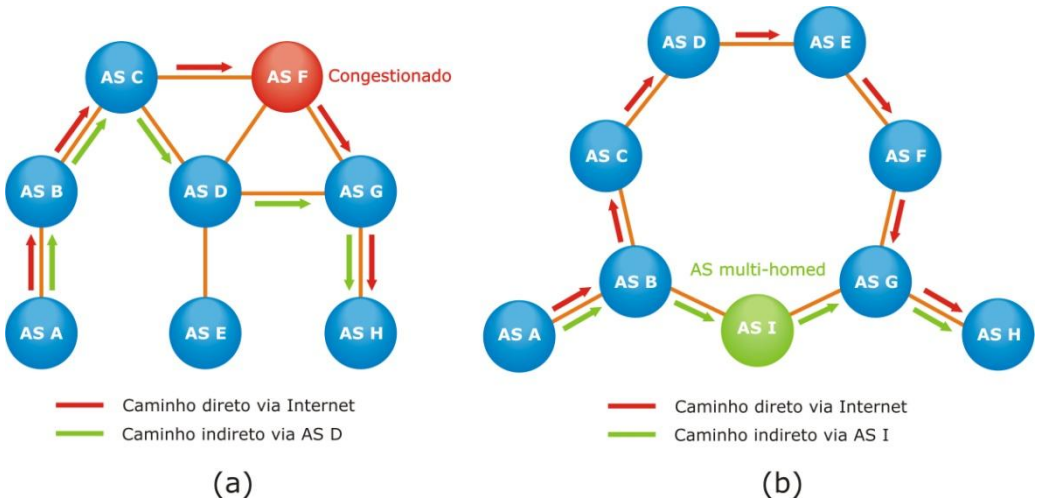


Figura 4.15 – *Dois cenários nos quais o roteamento indireto via rede sobreposta é mais rápido que o roteamento direto através da Internet.(a) AS congestionado. (b) Existência de uma rota mais curta.*

No caso (a), existe um AS congestionado (F) no caminho direto entre o AS origem (A) e o AS destino (H). Enviar pacotes de forma indireta via AS (D), faz com que a comunicação contorne o trecho defeituoso do sistema, garantindo a QoS. No caso (b), o caminho direto entre o AS origem (A) e o AS destino (H) é muito logo e cruza vários AS aumentando demasiadamente o atraso da comunicação. Através do encaminhamento indireto dos pacotes via o AS *multi-homed* (I), são necessários menos saltos, e em geral, menos tempo para que os pacotes atinjam o seu destino.

Ren e colaboradores mostraram que coletando endereços IP da rede sobreposta *Gnutella* [53] através das tabelas do BGP, foi possível estabelecer as relações entre os diversos AS e agrupar os nós pertencentes a um mesmo AS em *clusters* [75]. Já era conhecido que nós que possuem o mesmo prefixo IP geralmente pertencem a um mesmo AS e estão próximos fisicamente uns dos outros [76].

Por meio da seleção aleatória de um representante de cada um desses *clusters* e da medição do atraso entre os mesmos, é possível estimar um panorama dos atrasos entre os AS e construir uma tabela de roteamento [2]. A figura 4.16 mostra que na maioria das sessões que experimentaram um atraso de ida-e-volta (RTT – *Round-Trip Delay*) maior que 300 ms pelo caminho direto (nível de atraso acima do que é recomendado para VoIP), existia um caminho indireto cujo atraso de ida-e-volta era menor que 300 ms.

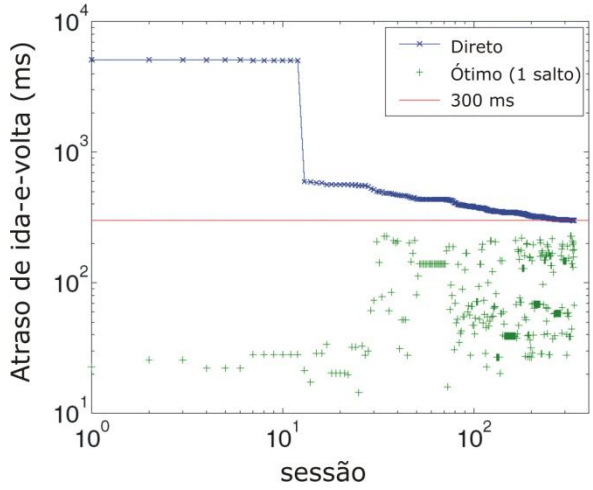


Figura 4.16 – *RTT dos caminhos direto e indireto por 1 salto (one-hop) nas piores sessões [2].*

A motivação do protocolo ASAP (*Autonomous System-Aware Peer-Relay Protocol*) proposto por Ren e colaboradores baseia-se no fato que os algoritmos de roteamento utilizados nos aplicativos de VoIP não levam em consideração as relações entre os AS [2]. As principais vantagens do ASAP são [2]:

- Nós pertencentes a um mesmo *cluster* estão fisicamente próximos uns dos outros, portanto, pode-se inferir que o atraso entre os representantes de dois diferentes *clusters* é aproximadamente o mesmo que entre quaisquer outros pares de nós pertencentes aos mesmos;
- Através da coleta das tabelas do BGP (que são publicamente disponíveis) pode-se traçar o grafo das ligações entre os AS;

- O atraso de um pacote está fortemente correlacionado com o número de saltos entre AS que ele deve realizar. Caminhos cruzando muitos AS são mais dispostos a apresentar longos atrasos [77].

O algoritmo define três tipos de nós [2]. Os Nós de Inicialização (*Bootstrap nodes*) são os nós dedicados responsáveis por guardar informações importantes do algoritmo. Dentre suas funções estão a construção e atualização do grafo de ligação entre os AS e das tabelas de mapeamento dos prefixos IP. Os nós de inicialização também devem selecionar novos nós de substituição no caso de falhas ou quedas e disseminar o grafo dos AS para os mesmos.

Os Nós de Substituição (*Surrogate nodes*) são nós com alta capacidade de processamento e largura de banda responsáveis pelo gerenciamento de um *cluster*. Devem manter uma lista dos endereços IP dos hosts presentes em seus *clusters* e contactar periodicamente os nós de inicialização para receber atualizações sobre o grafo dos AS. Eles também devem informar o conjunto de *clusters* vizinhos (com menor atraso) aos nós ordinários sob sua responsabilidade e aceitar informações dos mesmos. Um novo nó de substituição pode ser eleito dentre os nós ordinários caso ele se mostre mais capaz de executar as tarefas que o atual. O processo de troca deve ser informado aos demais nós ordinários e de inicialização.

Nós ordinários (*End nodes*): São os usuários que originam e recebem chamadas VoIP. Eles têm a função de solicitar informações sobre seus nós de inicialização e substituição, reportar suas informações nodais aos seus nós de substituição e tornar-se um deles quando necessário. De posse das informações enviadas pelos nós de inicialização e substituição os nós ordinários tornam-se aptos a selecionar a melhor rota para os seus pacotes de voz. O processo de comunicação é ilustrado na figura 4.17 e explicado a seguir.

Seja um *host* h1, num *cluster* A, que deseja se comunicar com um *host* h2, em um cluster B. Inicialmente, h1 envia uma solicitação "*join*" para o nó de inicialização que responde com a localização (endereço IP) do nó de substituição do *cluster* A. O usuário h1 solicita ao nó de substituição o seu conjunto de *clusters* vizinhos. Ao iniciar a chamada de voz com h2, h1 mede o RTT (através de algum programa como o "*ping*") e caso esteja maior que 300 ms (limite para VoIP) solicita que h2 envie o seu conjunto de *clusters* vizinhos. Através da interseção dos dois conjuntos o sistema pode escolher um nó (no caso o *host* h3 no *cluster* C) para intermediar a comunicação e permitir a troca de voz com um RTT menor que 300 ms.

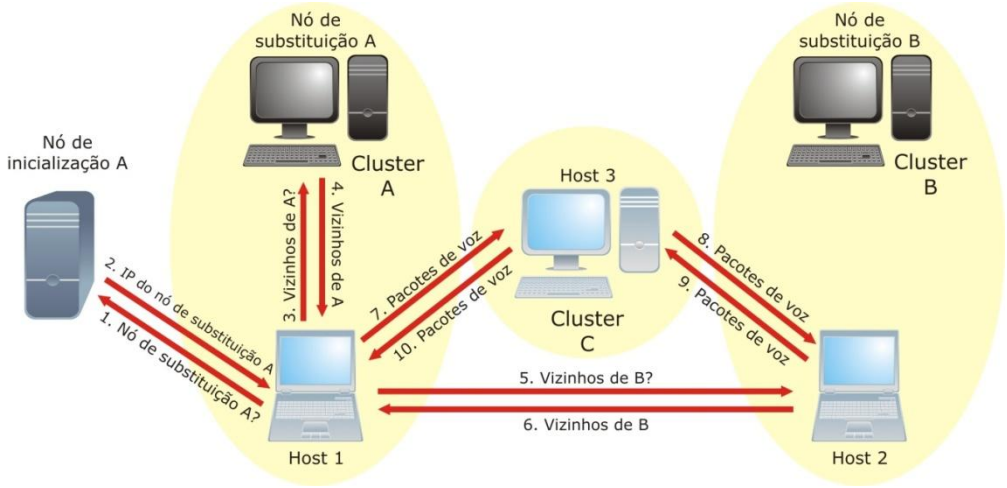
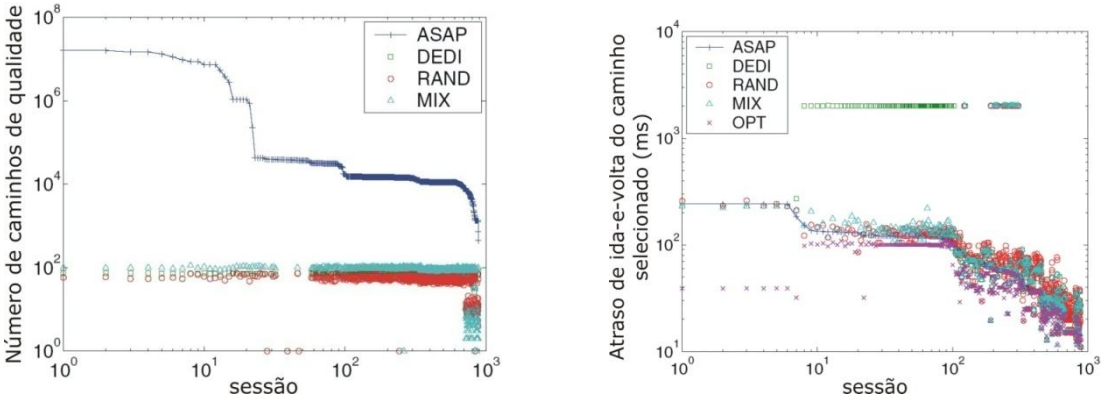


Figura 4.17 – *Funcionamento do protocolo ASAP.*

Os gráficos a seguir mostram uma comparação entre o desempenho do ASAP e o de outros algoritmos conhecidos para seleção de rotas em redes sobrepostas [2]. O **DEDI** é um algoritmo similar ao usado no RON (*Resilient Overlay Network*) [57], desenvolvido pelo MIT (*Massachusetts Institute of Technology*), que utiliza nós intermediários dedicados para realizar o roteamento dos pacotes. Nos testes, o algoritmo escolhia entre um conjunto de 80 nós pertencentes a 80 clusters distintos. O **RAND** é um algoritmo de seleção similar ao SOSR (*Scalable One-Hop Source Routing*) [78], desenvolvido pela Universidade de Washington, que seleciona aleatoriamente os nós para o encaminhamento dos pacotes. Durante os testes, o algoritmo selecionava o melhor dentre 200 nós escolhidos ao acaso. Foi ainda testado um algoritmo **MIX**, que utilizava uma combinação de ambos. Esse algoritmo selecionava o melhor entre 40 nós dedicados e 120 nós aleatoriamente escolhidos [2].

A amostra analisada era composta por 100.000 chamadas VoIP entre pares de nós coletados da rede sobreposta *Gnutella* [53]. Aproximadamente 1.000 dessas sessões apresentaram RTT superiores a 300 ms.

O ASAP apresentou um desempenho superior aos demais algoritmos testados. A figura 4.18 (a) mostra que o número de caminhos de alta qualidade (RTT inferiores a 300 ms) encontrados pelo ASAP é muito superior que os dos outros algoritmos. Os pontos nos gráficos estão organizados em ordem decrescente para facilitar a visualização. A figura 4.18 (b) mostra que o menor RTT do caminho selecionado pelo ASAP é comparável ao ótimo (**OPT**). Em todas as seções o ASAP apresentou RTTs abaixo de 115 ms, enquanto DEDI, RAND e MIX apresentaram RTTs superiores a 1 segundo em cerca de 5% das sessões [2].



(a) (b)

Figura 4.18 – *(a) Número de caminhos com RTT < 300 ms encontrados pelos algoritmos e (b) RTT do caminho selecionado pelos algoritmos [2].*

A estimativa do MOS foi realizada através do modelo-E [79] para o *codec* G.729A+VAD. A escolha da rota adequada para o encaminhamento do pacote de mídia se reflete no alto índice de qualidade de voz apresentada no ASAP (acima de 3,85 em todas as sessões), o que é mostrado na figura 4.19. Em cerca de 3% das sessões os outros algoritmos apresentaram um MOS abaixo de 2,9 o que é considerado insatisfatório.

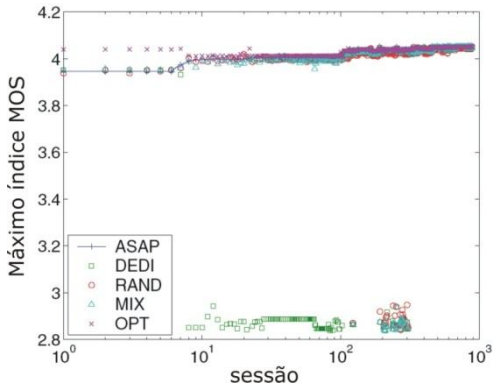


Figura 4.19 – *Qualidade de voz apresentada pelos algoritmos [2].*

De uma maneira geral, de acordo com os resultados dos trabalhos de diversos pesquisadores apresentados neste capítulo, pode-se inferir que o uso de redes sobrepostas par-a-par é capaz oferecer melhorias substanciais à qualidade do serviço de voz nos aplicativos VoIP.

5. Análise comparativa de aplicativos VoIP

Este capítulo apresenta uma análise comparativa entre três dos mais difundidos aplicativos de comunicação VoIP da Internet: *Google Talk, Yahoo! Messenger* e *Skype*. Uma nova métrica objetiva baseada em correlação será proposta para a avaliação do desempenho dos mesmos em relação à fidelidade dos sinais de voz transmitidos pelos mesmos. Os resultados aqui apresentados estão fortemente baseados no artigo intitulado "Uma Análise Comparativa da QoS do Skype, Yahoo! Messenger e Google Talk", apresentado no XVI Simpósio Brasileiro de Telecomunicações [80].

5.1. Trabalhos relacionados

A enorme difusão e sucesso do Skype, desde o seu lançamento em 2003, chamou a atenção da comunidade científica e despertou o interesse no estudo das aplicações das redes sobrepostas par-a-par em VoIP. Diversos trabalhos científicos nesta linha de pesquisa foram publicados desde então. Dentre eles incluem-se análises que confrontam os sistemas de telefonia convencional e VoIP [81] ou que comparam os sistemas par-a-par puros (Skype) e híbridos (baseados nos protocolos SIP ou H.323) de telefonia IP [64].

O Skype é um aplicativo proprietário de código fechado. Portanto, todo entendimento do seu modo de operação é fruto de investigações que se propõem a identificar, medir e analisar o tipo de tráfego gerado pelo Skype [82, 83, 84, 85, 86], compreender seu funcionamento [44, 87] e simular sua estrutura [88].

Seguindo outra linha, alguns pesquisadores se dedicaram a realizar análises comparativas entre aplicativos de telefonia IP [89, 90, 91] ou analisar o impacto causado

por diferentes tipos de ruído na qualidade de voz dos mesmos [92]. Embora algumas dessas análises tenham sido realizadas por meio de simulações e não em chamadas reais através da Internet, seus resultados são extremamente úteis e servem de base para o estudo apresentado neste trabalho.

5.2. **Método**

O objetivo do experimento foi realizar uma análise comparativa entre três dos mais utilizados e difundidos aplicativos de voz sobre IP da Internet: Skype versão 3.1.0.152 [1], Yahoo! Messenger versão 8.1 [93] e Google Talk versão Beta [94]. Os ensaios consistiram na realização de chamadas VoIP entre dois computadores situados na região metropolitana da cidade de Recife – PE. As máquinas utilizadas possuiam a seguinte configuração:

- PC Pentium IV 1,4 GHz com 512 MB de RAM.
- Laptop Intel Centrino Duo 1,8 GHz com 1 GB de RAM.

Ambos encontravam-se conectados à Internet através de linhas ADSL configuradas com taxas de 128 kBps no *uplink* e 512 kBps no *downlink*.

Em todos os testes, um dos usuários executava um arquivo padrão de áudio, cujo som era capturado pelo microfone e transmitido através de um dos aplicativos VoIP para o outro usuário. Para garantir a uniformidade dos testes, foi utilizado um único texto em língua inglesa para gerar dois arquivos de áudio: um de voz masculina, outro de voz feminina. O sintetizador utilizado foi o TTS (*Text-to-Speech*) da AT&T Labs [95].

Os arquivos gerados possuem formato *wave* (.wav), taxa de amostragem de 16 amostras por segundo e duração aproximada de 20 s. A densidade de potência espectral estimada através dos periodogramas dos mesmos é mostrada na figura 5.1.

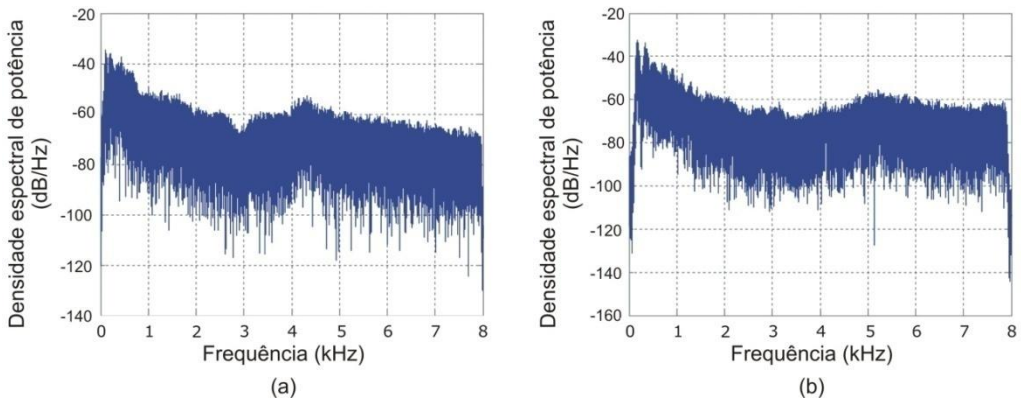


Figura 5.1 – *Densidade espectral de potência dos trechos de 20 s de voz sintetizada.(a) Masculina (b) Feminina.*

Optou-se por utilizar sinais de curta duração para viabilizar a análise através da correlação cruzada entre os sinais transmitido e recebido.

Foram realizadas 10 chamadas com voz masculina e 10 chamadas com voz feminina para cada um dos três aplicativos, perfazendo 20 chamadas por aplicativo e um total de 60 chamadas realizadas no total. As chamadas foram realizadas alternando o aplicativo em uso. Realizar todo um bloco de 20 aquisições em um mesmo aplicativo antes de passar ao próximo poderia fazer as variações naturais das condições da rede mascarar os dados coletados. Realizar os ensaios de forma alternada faz com que os aplicativos estejam submetidos aproximadamente às mesmas condições da rede e minimiza o viés introduzido pela mesma.

As chamadas foram monitoradas localmente pelos dois usuários, como mostrado na figura 5.2. Os sinais de voz transmitido e recebido era gravado em formato *wave* (.wav) com taxa de amostragem de 48 kHz pelo aplicativo *Mx Skype Recorder* versão 3.2.1 [96] nas duas extremidades da comunicação. Esse aplicativo é capaz de gravar a conversação em diversos aplicativos de voz sobre IP.

Paralelamente, todos os pacotes enviados e recebidos por cada uma das máquinas eram capturados pelo aplicativo *Wireshark* versão 0.99.5, um *sniffer* (farejador de pacotes) anunciado como substituto do *Ethereal* [97]. A ferramenta *My Trace Route* do *Fedora 6* [98], foi utilizada para traçar a rota tomada pelos pacotes.

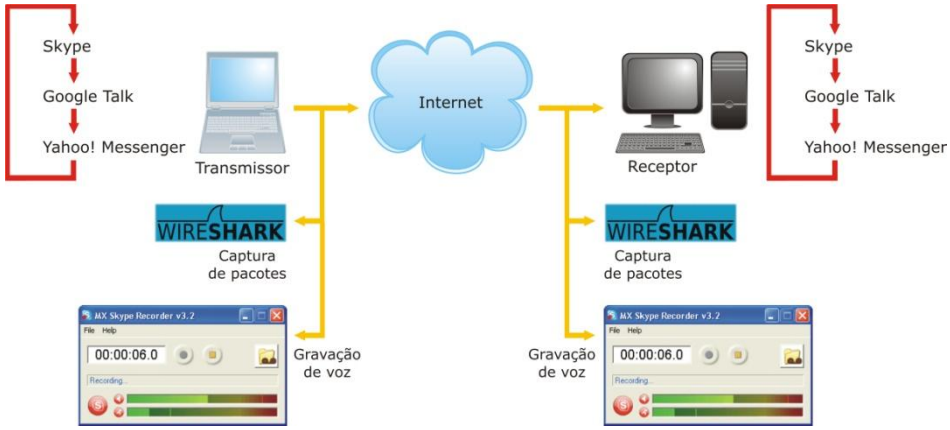


Figura 5.2 – *Esquema utilizado para aquisição dos dados.*

Para cada uma das chamadas realizadas eram gerados quatro arquivos:

- Gravação do sinal de voz transmitido pelo computador A;
- Gravação do sinal de voz recebido pelo computador B;
- Captura dos pacotes no computador A;
- Captura dos pacotes no computador B.

Perfazendo um total de 120 gravações de áudio e 120 capturas de pacotes.

A etapa de análise dos sinais de voz coletados se iniciou com o pré-processamento dos mesmos e foi realizada do seguinte modo:

Todas as chamadas possuíam longos períodos de silêncio antes e depois do trecho de conversação para permitir sua edição sem que nenhuma informação importante do sinal fosse perdida. Após serem coletadas, as gravações das chamadas de voz foram editadas no aplicativo *Nero Wave Editor* versão 3.0.0. O intuito dessa edição foi retirar os trechos inicial e final da gravacão de cada chamada. Com isso, tons de discagem e ruídos gerados pelo atendimento ou finalização das chamadas pelos aplicativos ou usuários foram eliminados e a informação útil foi deixada inalterada. Os arquivos de voz finais (já editados) possuiam 2.063 kB de tamanho e cerca de 22 s de duração.

A análise da fidelidade do sinal recebido em relação ao que foi transmitido foi computada através da função de correlação cruzada. A função de correlação cruzada é uma ferramenta de estatística que estima o grau de similaridade/diferença entre dois sinais. A definição dessa função é dada como segue: Sejam duas sequências de tempo discreto $t[n]$ (sinal transmitido) e $r[n]$ (sinal recebido) a função de correlação cruzada entre $t[n]$ e $r[n]$ é dada por [99]:

$$C_{t,r}[k] = \mathcal{E}\{t[n] \cdot r[n+k]\}, \tag{6}$$

na qual $\mathcal{E}\{x\}$ é a expectância da variável x e k o deslocamento no tempo em amostras de uma das sequências. Matematicamente, a correlação cruzada pode ser dada através da expressão [99]:

$$C_{t,r}[k] = \sum_{n=0}^{N-1} \{t[n] \cdot r[n+|k|]\}. \tag{7}$$

Para desconsiderar os viéses introduzidos pela amplificação ou atenuação dos sinais transmitidos e recebidos, os mesmos serão normalizados pela respectiva energia E de cada um deles, que é dada por [106]:

$$E_x = \sum_{n=0}^{N} |x[n]|^2. \tag{8}$$

A análise será realizada tomando o valor máximo da função de correlação cruzada, o que ocorre para um dado deslocamento k_l que alinha perfeitamente os sinais transmitido e recebido. Ou seja, a métrica utilizada para estimar o grau de distorção gerado pelos aplicativos nos sinais de voz é expressa por:

$$corr = máx_k \left\{ \frac{C_{t,r}[k]}{(E_t)^{\frac{1}{2}} \cdot (E_r)^{\frac{1}{2}}} \right\}. \quad (9)$$

O valor de $corr$ excursiona desde -1 (mínima similaridade) até +1 (máxima similaridade, sinais idênticos). Os cálculos foram realizados com o aplicativo *Matlab 7*.

5.3. **Aplicativos VoIP**

Uma breve descrição do modo de operação e das principais características dos três aplicativos analisados será apresentada a seguir.

5.3.1. **Yahoo! Messenger**

O Yahoo! Messenger é um aplicativo VoIP que utiliza um protocolo de sinalização cliente-servidor proprietário baseado em SIP [93]. A linha 102 de um trecho de captura realizada com o aplicativo *Wireshark*, apresentada na figura 5.3 mostra o uso de tal protocolo (YMSG). As colunas mostradas nas capturas informam o número do pacote, o tempo de chegada do pacote em relação ao primeiro (em ms), o endereço IP de origem, o endereço IP destino, o protocolo utilizado, os comentários e o conteúdo do pacote.

96	7.959794	192.168.15.236	68.142.233.173	SSL	Continuation Data	663 7.959794
97	8.158626	192.168.15.236	68.142.233.173	TCP	1628 > https [FIN, ACK] Seq=609 Ack=0 Win=6553	54 8.158626
98	8.180207	68.142.233.173	192.168.15.236	SSL	Continuation Data	523 8.180207
99	8.180276	192.168.15.236	68.142.233.173	TCP	1628 > https [RST, ACK] Seq=610 Ack=469 Win=0 I	54 8.180276
100	8.180316	68.142.233.173	192.168.15.236	TCP	https > 1628 [FIN, ACK] Seq=469 Ack=609 Win=65	60 8.180316
101	8.180339	192.168.15.236	68.142.233.173	TCP	[TCP ACKed lost segment] 1628 > https [RST] Se	54 8.180339
102	8.180815	192.168.15.236	216.155.193.174	YMSG	YAHOO_SERVICE_SKINNAME, YAHOO_STATUS_AVAILABLE	277 8.180815
103	8.201194	192.168.15.236	216.155.193.174	TCP	1610 > 5050 [FIN, ACK] Seq=223 Ack=0 Win=64774	54 8.201194
104	8.396239	216.155.193.174	192.168.15.236	TCP	5050 > 1610 [ACK] Seq=0 Ack=224 Win=65535 Len=0	60 8.396239
105	8.396276	216.155.193.174	192.168.15.236	TCP	5050 > 1610 [FIN, ACK] Seq=0 Ack=224 Win=65535	60 8.396276
106	8.396299	192.168.15.236	216.155.193.174	TCP	1610 > 5050 [ACK] Seq=224 Ack=1 Win=64774 [TCP	54 8.396299

Figura 5.3 – *Captura da troca de pacotes com o servidor durante o logoff no Yahoo! Messenger.*

Esse sistema oferece suporte tanto a *codecs* padrão (G.711, G.723) quanto a *codecs* da *Global IP Sound* (iSAC, *Speex*, e iLBC) [100]. A privacidade e integridade dos dados é garantida através do protocolo SSLv3 (*Secure Sockets Layer*) [101], que realiza uma autenticação das partes envolvidas e da cifragem dos dados trocados. O uso do SSLv3 no Yahoo! Messenger pode ser visualizado na figura 5.4.

21	14.109223	201.50.209.91	209.73.168.74	TCP	1529 > https [SYN] Seq=0 Len=0 MSS=1360	70 14.109223
22	14.234835	201.50.209.91	216.155.193.150	TCP	1528 > 5050 [ACK] Seq=56 Ack=140 Win=65396 Len	62 14.234835
23	14.569939	209.73.168.74	201.50.209.91	TCP	https > 1529 [SYN, ACK] Seq=0 Ack=1 Win=65535 I	68 14.569939
24	14.571738	201.50.209.91	209.73.168.74	TCP	1529 > https [ACK] Seq=1 Ack=1 Win=65535 Len=0	62 14.571738
25	14.572514	201.50.209.91	209.73.168.74	SSLv3	Client Hello	164 14.572514
28	15.102690	209.73.168.74	201.50.209.91	SSLv3	Server Hello, Certificate, Server Hello Done	914 15.102690
29	15.104872	201.50.209.91	209.73.168.74	SSLv3	Client Key Exchange, Change Cipher Spec, Encry	260 15.104872
30	15.595401	209.73.168.74	201.50.209.91	SSLv3	Change Cipher Spec, Server Hello[Malformed Pacl	129 15.595401
31	15.598592	201.50.209.91	209.73.168.74	SSLv3	Application Data	634 15.598592
32	16.186567	209.73.168.74	201.50.209.91	SSLv3	Application Data	1196 16.186567
33	16.188486	201.50.209.91	209.73.168.74	TCP	1529 > https [FIN, ACK] Seq=884 Ack=2060 Win=6	62 16.188486
34	16.189709	201.50.209.91	216.155.193.150	TCP	[TCP segment of a reassembled PDU]	889 16.189709
35	16.194574	209.73.168.74	201.50.209.91	SSLv3	Encrypted Alert	81 16.194574
36	16.195999	201.50.209.91	209.73.168.74	TCP	1529 > https [RST, ACK] Seq=885 Ack=2083 Win=0	62 16.195999
37	16.650179	209.73.168.74	201.50.209.91	TCP	https > 1529 [ACK] Seq=2083 Ack=885 Win=65535 I	68 16.650179
38	16.655546	209.73.168.74	201.50.209.91	TCP	https > 1529 [FIN, ACK] Seq=2083 Ack=885 Win=6	68 16.655546
39	16.719168	216.155.193.150	201.50.209.91	TCP	[TCP segment of a reassembled PDU]	328 16.719168
40	16.811279	201.50.209.91	216.252.107.223	TCP	1530 > http [SYN] Seq=0 Len=0 MSS=1360	70 16.811279
41	16.850121	201.50.209.91	216.155.193.150	TCP	1528 > 5050 [ACK] Seq=883 Ack=406 Win=65130 Le	62 16.850121
43	16.905758	201.50.209.91	216.155.193.150	TCP	[TCP segment of a reassembled PDU]	193 16.905758
44	16.919764	201.50.209.91	216.252.107.223	TCP	1531 > http [SYN] Seq=0 Len=0 MSS=1360	70 16.919764
46	17.064850	201.50.209.91	68.142.233.142	TCP	1532 > https [SYN] Seq=0 Len=0 MSS=1360	70 17.064850

Figura 5.4 – *Captura da troca de pacotes com o servidor durante o login no Yahoo! Messenger.*

Os pacotes capturados durante a realização das chamadas indicam que há comunicação entre o usuário e quatro servidores principais da rede Yahoo. São eles:

- **sip48.voice.re2.yahoo.com (68.142.233.145):** Servidor de sinalização SIP. Foi observada a troca de pacotes entre as máquinas dos usuários e esse servidor do início ao fim da chamada.
- **relay1.voice.vip.re2.yahoo.com (68.142.233.72):** Servidor responsável pelas solicitações de permissão para ligar (*Binding Request*).
- **stun2a.voice.re2yahoo.com (68.142.233.76):** Servidor responsável pela transposição de NATs através do protocolo STUN [65].
- **relay6.voice.re2.yahoo.com (68.142.233.79):** Servidor contactado no momento que antecede o início da troca de pacotes de voz (UDP) entre os usuários.

A relação entre um usuário do Yahoo! Messenger e os servidores do sistema é mostrada na figura 5.5.

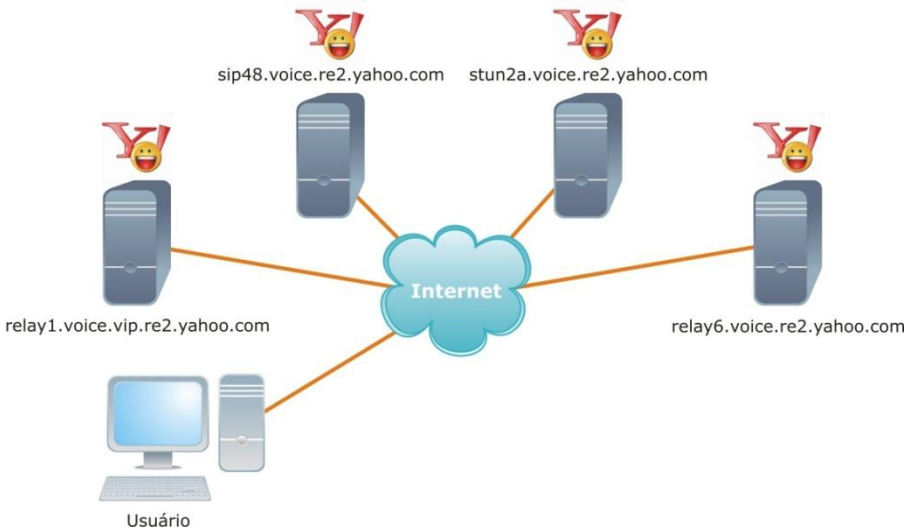


Figura 5.5 – *Principais servidores da rede Yahoo! Messenger.*

No decorrer das chamadas, em vários momentos foi observada a troca de pacotes com outros endereços IP, no entanto, o destino sempre correspondia a um servidor com as mesmas funcionalidades dos que foram descritos. Essa estrutura de vários servidores distribuídos pode ser uma forma de realizar o balanceamento da carga da rede.

5.3.2. **Google Talk**

O Google Talk [94] é um aplicativo de telefonia IP gratutito distribuído pelo Google. O Google Talk utiliza o protocolo *jabber* XMPP (*Extensible Messaging and Presence Protocol*), definido na RFC 3920 [102] como sistema de sinalização, embora existam planos para num futuro próximo operar sobre SIP. O *Jabber* XMPP, embora seja descentralizado, é um protocolo cliente-servidor, ou seja, os usuários não se comunicam diretamente uns com os outros. Cada usuário da rede recebe um identificador *Jabber ID* (JID) único, que se assemelha em sintaxe a uma URI SIP ou um endereço de *e-mail*, da forma: usuário@servidor. A comunicação entre dois usuários através do XMPP se dá por intermédio dos seus servidores como ilustrado na figura 5.6.

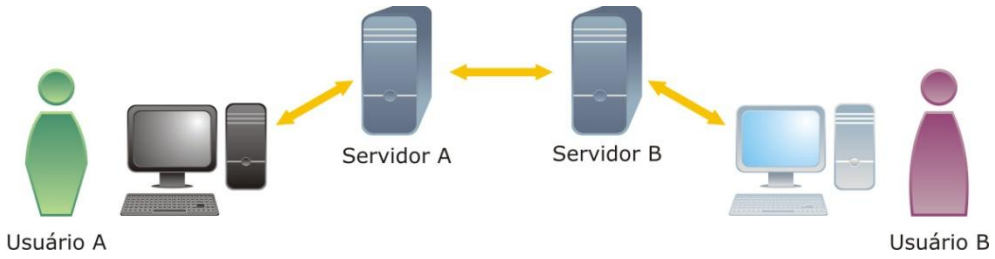


Figura 5.6 – *Comunicação cliente-servidor através do protocolo Jabber XMPP.*

O Google Talk permite a interação com outros sistemas que utilizam o XMPP e oferece suporte tanto a *codecs* padrão (G.711, G.723) quanto a *codecs* da *Global IP Sound* (iSAC, EG711, iLBC e *Speex*) [100].

Traçando a rota tomada pelos pacotes durante o login no Google Talk foi obtida a seguinte sequência de saltos, que é mostrada na figura 5.7.

- 192.168.253.254 – Endereço IP da máquina do usuário.
- 200.165.149.177 – Endereço IP válido atrelado a LAN do usuário.
- Estação Telemar Boa Vista – Recife –PE.
- Estação Telemar Botafogo – Rio de Janeiro – RJ.
- Rede Gigabit New York D.C. – USA.
- Provedor em Washington.
- Servidor Google.

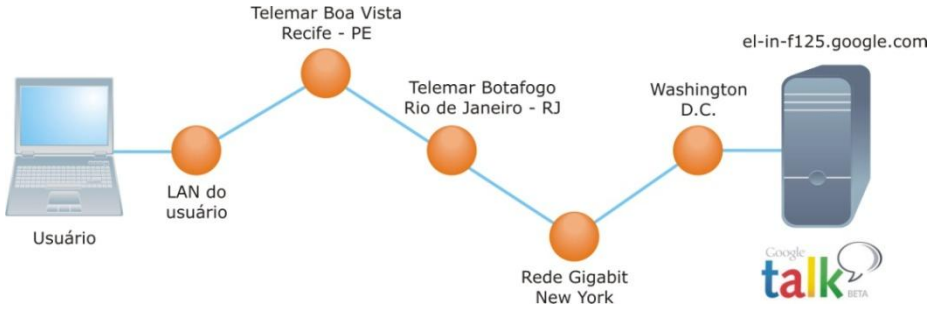


Figura 5.7 – *Sequência de saltos até o servidor de login do Google Talk.*

As figuras seguintes mostram trechos de capturas realizadas. A figura 5.8 ilustra o início do processo de autenticação. Há uma solicitação do tipo *Client Hello* (pacote 66) seguida por uma confirmação (ACK) do servidor (pacote 68). Em seguida há uma perda do pacote 69 e uma retransmissão do mesmo que traz a mensagem *Server Hello* (pacote 70).

65 3.333805	192.168.15.236	64.233.161.147	TCP	4732 > https [ACK] Seq=1 Ack=1 Win=65525 [TCP /	54	3.333805
66 3.334117	192.168.15.236	64.233.161.147	SSLv2	Client Hello	132	3.334117
67 3.550928	64.233.161.147	192.168.15.236	TCP	https > 4732 [ACK] Seq=1 Ack=79 Win=8190 Len=0	60	3.550928
68 3.575549	64.233.161.147	192.168.15.236	TLS	Server Hello,	1484	3.575549
70 3.882691	64.233.161.147	192.168.15.236	TLS	[TCP Retransmission] Server Hello,	1484	3.882691
71 3.882737	192.168.15.236	64.233.161.147	TCP	4732 > https [ACK] Seq=79 Ack=1431 Win=64105 [	54	3.882737
72 4.087389	64.233.161.147	192.168.15.236	TLS	Certificate, Server Hello Done	342	4.087389
73 4.088918	192.168.15.236	64.233.161.147	TLS	Client Key Exchange, Change Cipher Spec, Encryp	236	4.088918
74 4.289972	64.233.161.147	192.168.15.236	TLS	Change Cipher Spec, Encrypted Handshake Messag	97	4.289972
75 4.290405	192.168.15.236	64.233.161.147	TLS	Application Data	258	4.290405
76 4.529547	64.233.161.147	192.168.15.236	TCP	https > 4732 [ACK] Seq=1762 Ack=465 Win=5720 L	60	4.529547
77 4.529581	192.168.15.236	64.233.161.147	TLS	Application Data	235	4.529581

Figura 5.8 – *Captura da troca de pacotes com o servidor durante o login no Google Talk.*

A figura 5.9, mostra em destaque o uso do protocolo *Jabber* XMPP pelo Google Talk. Pode ser observada, na linha 41, uma mensagem *Jabber Request* (Requisição para falar), enviada para o servidor do Google Talk antes do início da conversação.

41 5.341797	201.32.165.30	209.85.163.125	Jabber	Request: \027\003\001\000]\207\354\276t\275\26	152	5.341797
42 5.343750	209.85.163.125	201.32.165.30	Jabber	Response: \027\003\001\000\210P\353A\204\037\0	195	5.343750
43 5.347656	209.85.163.125	201.32.165.30	Jabber	Response: \027\003\001\000\210h\215.z\213\211D	195	5.347656
44 5.347656	201.32.165.30	209.85.163.125	TCP	1585 > 5222 [ACK] Seq=3646 Ack=1364 Win=16170	54	5.347656
45 5.578125	209.85.163.125	201.32.165.30	Jabber	Response: \027\003\001\000\210\250]B\026?v#\33	195	5.578125
46 5.614257	209.85.163.125	201.32.165.30	TCP	5222 > 1585 [ACK] Seq=1505 Ack=3646 Win=18720	60	5.614257
47 5.684570	201.32.165.30	209.85.163.125	TCP	1585 > 5222 [ACK] Seq=3646 Ack=1505 Win=16029	54	5.684570
48 6.051757	200.255.242.189	201.32.165.30	UDP	Source port: 64904 Destination port: 53379	124	6.051757
49 6.967773	189.13.172.233	201.32.165.30	STUN	Message: Binding Request	98	6.967773
50 6.967773	201.32.165.30	189.13.172.233	STUN	Message: Binding Error Response	123	6.967773

Figura 5.9 – *Captura da troca de pacotes com o servidor antes do início da chamada no Google Talk.*

5.3.3. **Skype**

O Skype é um aplicativo capaz de realizar chamadas VoIP tanto entre usuários Skype quanto entre um usuário Skype e um usuário do sistema de telefonia convencional em qualquer parte do mundo [1]. A grande inovação apresentada pelo Skype é utilizar uma arquitetura baseada em redes par-a-par sobrepostas [44].

O Skype oferece suporte aos codecs iLBC, iSAC e iPCM da *Global IP Sound* e cifra seus dados com o AES (*Advanced Encryption Standard*) [1]. Não há supressão de silêncio e pacotes são transmitidos mesmo se ambos os usuários permanecem em silêncio o que

possibilita que informações sobre o ruído do ambiente sejam transmitidas [44]. A transposição de NATs e *firewalls* é realizada através de variações dos protocolos STUN [65] e TURN [66]. O tráfego de mídia flui entre os dois usuários de uma forma par-a-par se eles possuem ambos endereços IP públicos ou se apenas um deles encontra-se sob restrições de NATs ou *firewalls*. Caso ambos os usuários estejam sob tais restrições o tráfego de mídia entre eles é encaminhado indiretamente via um terceiro nó da rede que se comporta como um servidor *proxy* [44]. A rede Skype é composta por três entidades principais [44]:

- Cliente Skype: É qualquer nó (máquina) da rede executando o aplicativo Skype e que esteja habilitada a iniciar e receber chamadas de voz, enviar mensagens de texto e realizar troca de arquivos com outros usuários.
- Servidor de Login Skype: Com exceção dos servidores destinados à integração com a telefonia convencional através dos serviços de *skype-in* (PSTN-Skype) e *skype-out* (Skype-PSTN), e que não desempenham nenhuma função no cenário PC-a-PC, o Servidor de Login é o único elemento centralizado do Skype. Esse servidor tem a função de guardar os registros de nome de usuários e suas senhas, realizando a autenticação dos mesmos e garantindo que sejam únicos no espaço de nomes da rede Skype.
- Super-Nó Skype: Clientes Skype que sejam facilmente alcançáveis na rede (não apresentem restrições de NATs ou *firewalls*) e possuam disponibilidade de largura de banda podem ser promovidos a Super-Nós. Os super-nós têm a função de prover à rede funcionalidades como encaminhar pedidos aos destinos apropriados e responder às solicitações dos clientes skype ou de outro super-nó. Eles podem ainda oferecer serviços de *proxy* para possibilitar a troca de mídia entre clientes Skype que possuam acesso restrito à Internet através de NATs e *firewalls*. Os super-nós mantêm uma rede sobreposta entre si, ao passo que um cliente Skype liga-se a um único ou a um pequeno grupo de super-nós.

A estrutura da rede Skype é mostrada na figura 5.10.

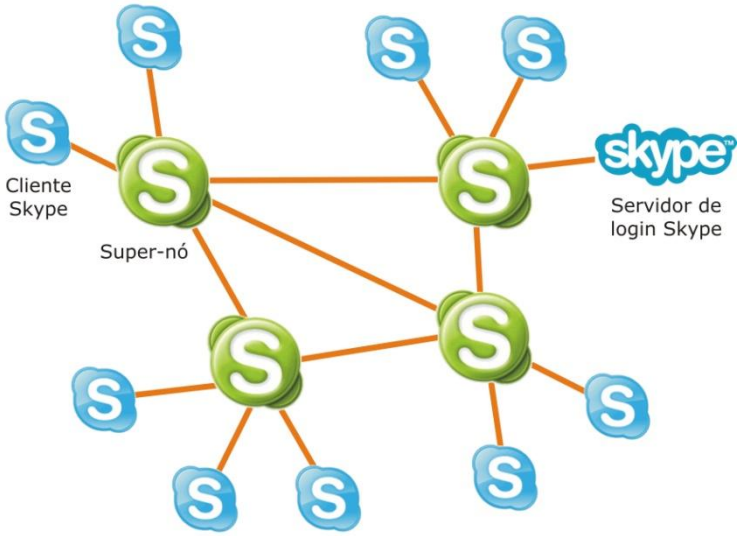


Figura 5.10 – *Estrutura da rede Skype.*

Como o Skype possui uma arquitetura par-a-par sobreposta, um usuário Skype precisa encontrar e se comunicar com um super-nó *on-line* da rede para se conectar ao sistema. Para isso, os clientes Skype mantém e atualizam periodicamente uma lista de super-nós disponíveis (*host cache*), dentre os quais alguns dedicados, chamados super-nós de inicialização (*bootstrap super-nodes*) [44]. Há um total de sete super-nós de inicialização distribuídos entre três provedores americanos e um dinamarquês, presentes na *host cache* dos clientes Skype desde a instalação do aplicativo.

Os clientes Skype enviam mensagens de *refresh* (atualização) periódicas via TCP aos seus super-nós durante o tempo em que permanecem conectados. Após a conexão com um super-nó, o cliente Skype deve realizar sua autenticação na rede.

Por ser a única etapa realizada de uma forma centralizada, o *login* é a função mais crítica da operação do Skype. Durante o login, o cliente Skype autentica seu nome e senha, informa sua presença aos outros nós da rede, determina se possui restrições de NAT ou *firewall*, descobre os usuários Skype *on-line* que possuem endereços IP públicos e verifica a disponibilidade de atualizações da versão do Skype [44]. O servidor de *login* Skype encontra-se hospedado em um provedor dinamarquês [44].

A busca por usuários utiliza a tecnologia *Global Index* [103], capaz de localizar um usuário, caso o mesmo tenha realizado *login* nas últimas 72 horas [44].

A busca de usuários e a troca de mídia fluem de um modo completamente par-a-par. Os clientes Skype origem encaminham suas mensagens diretamente aos clientes Skype destino ou indiretamente através de um terceiro cliente Skype (super-nó).

5.4. **Análises e resultados**

A análise dos dados coletados foi realizada em duas etapas: A primeira delas teve como objetivo comparar aspectos característicos dos aplicativos analisados. A segunda consistia em analisar as distorções geradas nos sinais de voz por cada aplicativo através de uma métrica objetiva (função de correlação cruzada).

A análise dos pacotes capturados trouxe à tona diversas características dos aplicativos estudados. A primeira delas se refere ao tamanho médio dos pacotes. A figura 5.11 mostra o tamanho médio dos pacotes transmitidos por cada aplicativo em cada uma das chamadas realizadas.

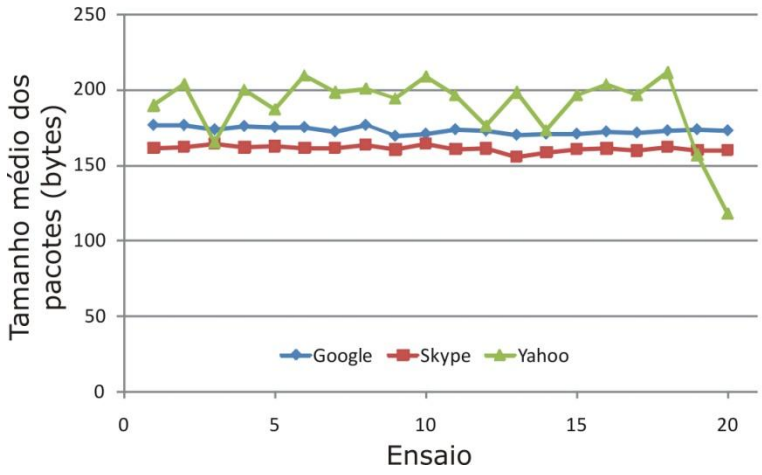


Figura 5.11 – *Tamanho médio dos pacotes transmitidos por cada aplicativo.*

A tabela 5.1 apresenta as estatísticas de média e desvio padrão dos pacotes transmitidos pelos aplicativos testados.

Tabela 5.1 – *Estatísticas do tamanho dos pacotes transmitidos pelos aplicativos.*

(Média / Desvio Padrão)	Google Talk	Skype	Yahoo! Messenger
Tamanho dos pacotes (Bytes)	(172,95 / 2,13)	(160,91 / 1,92)	(188,83 / 21,71)

Nota-se que tanto no Skype quanto no Google Talk praticamente não há variação no tamanho dos pacotes transmitidos. Esse fato ajuda a confirmar a hipótese de que o Skype não realiza supressão de silêncio e continua a enviar pacotes mesmo que um dos interlocutores esteja calado [44]. Os resultados indicam que o Google Talk pode utilizar uma estratégia semelhante. O comprimento do pacotes nesses dois aplicativos gira em torno dos 160 bytes. O Yahoo! Messenger, por outro lado, apresentou uma grande variação

no comprimento dos pacotes, certamente devido a alguma estratégia de adaptação dinâmica às condições da rede.

As figuras 5.12 e 5.13 mostram, respectivamente, o número total de pacotes enviados pelos aplicativos e a taxa na qual esses pacotes são transmitidos.

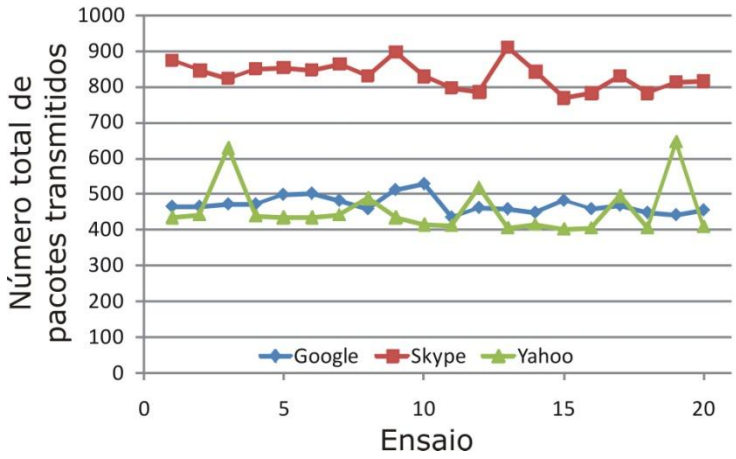


Figura 5.12 – *Número total de pacotes enviados por cada aplicativo.*

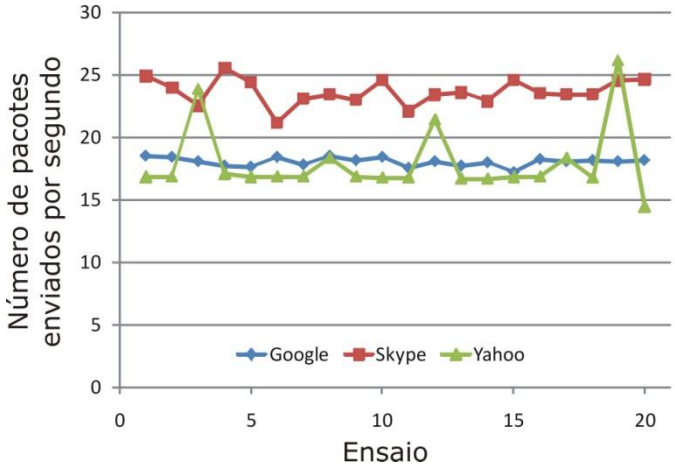


Figura 5.13 – *Número de pacotes transmitidos por segundo por cada aplicativo.*

Os gráficos mostrados figuras 5.12 e 5.13 indicam que o Skype, por utilizar pacotes de menor comprimento, deve enviar os mesmos em maior número e frequência que os outros dois aplicativos. Uma vez que a rota é estabelecida no Skype ela permenece até o fim da chamada [89]. Assim, ese aplicativo se adapta dinamicamente às condições da rede através da alteração de parâmetros como o *codec* ou a taxa de transmissão [89]. Esse fato pode explicar a variação do número de pacotes transmitidos pelo Skype e indicar que o Yahoo! Messenger utiliza alguma estratégia semelhante.

A tabela 5.2 apresenta as estatísticas obtidas de média e desvio padrão para o número médio de pacotes transmitidos e a taxa de transmissão média observadas para os aplicativos em teste. O Google Talk mantém um compromisso entre o comprimento e a quantidade de pacotes transmitidos. Esse aplicativo apresentou uma taxa de envio de pacotes praticamente constante, enviando aproximadamente 18 pacotes a cada segundo.

Tabela 5.2 – *Número de pacotes transmitidos e taxa de transmissão de pacotes apresentados pelos aplicativos.*

(Média / Desvio Padrão)	Google Talk	Skype	Yahoo! Messenger
Pacotes transmitidos	(469,00 / 23,06)	(830,25 / 37,10)	(453,85 / 68,08)
Taxa de transmissão (Pacotes por segundo)	(18,01 / 0,34)	(23,59 / 1,02)	(27,87 / 2,69)

O consumo de banda pelos aplicativos analisados é mostrado na figura 5.14 e resumido na tabela 5.3.

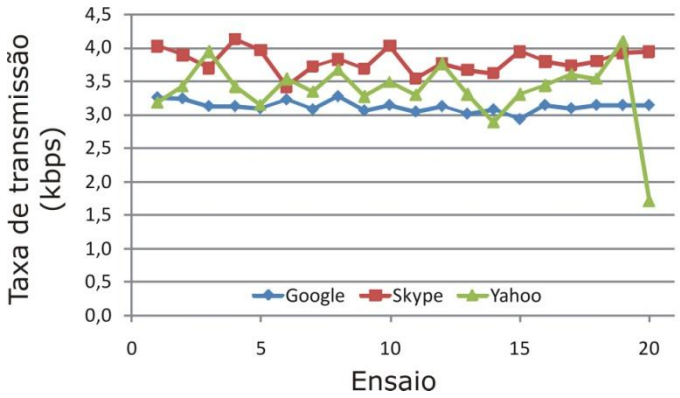


Figura 5.14 – *Taxa de transmissão dos aplicativos analisados.*

Tabela 5.3 – *Taxa de transmissão média apresentada pelos aplicativos.*

(Média / Desvio Padrão)	Google Talk	Skype	Yahoo! Messenger
Taxa de transmissão (kbps)	(24,92 / 0,64)	(30,37 / 1,39)	(26,87 / 3,74)

O Skype apresentou o maior consumo de banda entre os três aplicativos testados. Além de apresentar o mais baixo consumo médio de banda, o Google Talk mostrou ser mais uma vez o aplicativo mais estável, apresentando a menor variação na taxa de transmissão, em relação aos outros dois (cerca de 13% contra 18,4% do Skype e 65% do Yahoo! Messenger). Esse fato fortalece a observação realizada em [89] de que o Google Talk se adapta as condições de rede realizando triangulação (roteamento indireto) e não por alteração do *codec* ou da taxa de transmissão como é o caso do Skype.

Os dados relativos aos atrasos dos pacotes, não foram considerados, pois os erros introduzidos devido à falta de um modo de sincronizar os relógios das duas máquinas tornariam os mesmos sem nenhuma significância. No entanto, essa falta de sincronismo não introduz erros no cálculo do *jitter*, pois o mesmo é definido como a variância do atraso (momento estatístico de segunda ordem) e é insensível a *off-sets* nos relógios das máquinas. A falta de sincronismo entre os relógios das máquinas foi desprezado, pois como o experimento é de curta duração, esse efeito não teria tempo de se tornar relevante. A figura 5.15 mostra o *jitter* observado nos ensaios realizados com os três aplicativos.

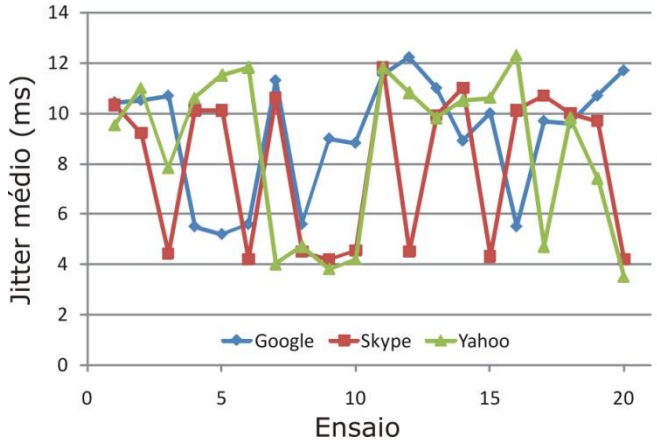


Figura 5.15 – *Jitter médio observado durante os ensaios com os três aplicativos.*

Os três aplicativos apresentaram características muito semelhantes em relação ao *jitter*, justificando o método utilizado de realizar alternadamente os ensaios entre os aplicativos. Percebe-se no entanto que o Skype apresentou valores de *jitter* ligeiramente menores que os dos outros aplicativos, como mostra a tabela 5.4.

Tabela 5.4 – *Jitter médio apresentado pelos aplicativos.*

(Média / Desvio Padrão)	Google Talk	Skype	Yahoo! Messenger
Jitter médio (ms)	(9,17 / 2,31)	(7,92 / 2,95)	(8,51 / 3,09)

A análise da qualidade do sinal de voz foi estimada através da função de correlação cruzada entre os sinais transmitidos e recebidos por cada aplicativo. Os resultados dessas análises de correlação para as amostras coletadas são apresentados no gráfico da figura 5.16.

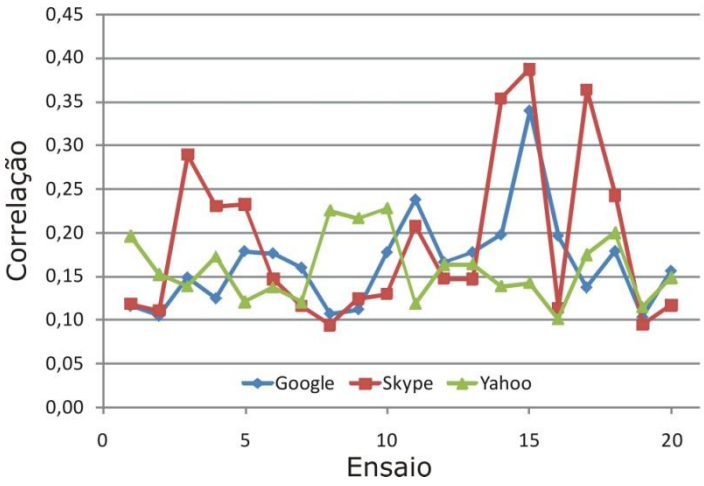


Figura 5.16 – *Correlacão entre os sinais transmitidos e recebidos.*

Nota-se que em cerca de metade dos ensaios os três aplicativos apresentaram aproximadamente a mesma correlação entre os sinais transmitidos e recebidos. No entanto, na outra metade, o Skype mostrou-se bastante superior que os outros dois aplicativos, atingindo maiores níveis de correlação entre os sinais. Os resultados obtidos em média são apresentados na tabela 5.5.

Tabela 5.5 – *Média e desvio padrão da correlação máxima medida nos aplicativos.*

(Média / Desvio Padrão)	Google Talk	Skype	Yahoo! Messenger
Correlação máxima	(0,16 / 0,05)	(0,19 / 0,09)	(0,16 / 0,04)

Em uma análise subjetiva, realizada ouvindo simplesmente os arquivos de áudio gravados percebeu-se que o Skype e o Google Talk atingem níveis de qualidade melhores que os do Yahoo! Messenger. No entanto, a diferença entre a qualidade da voz obtida no Skype e no Google Talk é muito sutil, sendo ligeiramente superior no Skype. Isso pode ser justificado pelo maior nível médio de correlação entre os sinais transmitidos e recebidos pelo Skype. Porém, de um modo geral, todos os três aplicativos oferecem voz a uma qualidade relativamente satisfatória, servindo bem ao propósito a que que foram projetados.

6. Conclusões e trabalhos futuros

Os sistemas de telefonia IP se tornaram muito populares nos últimos anos, apresentando-se como uma alternativa atraente para usuários domésticos e corporativos, em parte pela facilidade de implantação e utilização, mas principalmente por oferecer uma diminuição dos custos com chamadas telefônicas. Mais do que isso, os sistemas de voz sobre IP representam a evolução da telefonia e são um grande passo no sentido de desenvolver uma rede que integre os mais variados tipos de informação (dados, voz, vídeo, entre outros) e unifique as telecomunicações através da convergência IP.

Os grandes desafios do VoIP são transmitir voz em tempo real através de uma rede de pacotes que não foi inicialmente projetada para esse fim (rede IP) e oferecer aos seus usuários uma alta conectividade, com níveis de qualidade de voz similares aos experimentados no sistema de telefonia convencional PSTN (*Public Switched Telephone Network*).

O uso de redes sobrepostas par-a-par (*overlay peer-to-peer networks*) permite contornar alguns dos aspectos das redes IP que contribuem para a degradação do sinal de voz e diminuição de sua qualidade (atraso, *jitter*, perda, e outros). Essas redes criam um nível mais alto de abstração que permite resolver problemas que são difíceis de solucionar ao nível dos roteadores da rede. Através dessa arquitetura, torna-se possível o desenvolvimento de estratégias independentes de roteamento e endereçamento que viabilizem o uso de serviços, protocolos, funcionalidades e aplicações não-suportados pelos roteadores da Internet atual.

Nesta dissertação, além de apresentar todos os conceitos necessários ao entendimento de aplicativos de voz sobre IP, foram analisadas diversas estratégias de uso das redes

sobrepostas par-a-par visando a melhoria da qualidade de voz nos sistemas de telefonia IP. Tais estratégias buscam realizar um roteamento mais criterioso que o oferecido pelos roteadores da Internet, que leve em consideração as exigências de qualidade e interatividade requeridas pelo serviço de telefonia.

Essas novas formas de roteamento diminuem o atraso fim-a-fim e as perdas efetivas de pacotes, o que implica um aumento significativo na qualidade do serviço oferecido pelos sistemas VoIP.

O custo associado à melhoria da qualidade se traduz no aumento de fatores como carga computacional e *overhead* do sistema. No entanto, o acréscimo na carga computacional não chega a se tornar um problema, principalmente se for levado em consideração o crescente poder de processamento dos dispositivos atuais (PCs, telefones celulares, *palmtops*). Do mesmo modo, os impactos causados pelo crescimento do *overhead* não comprometem o funcionamento do sistema e são compensados pelo aumento da capacidade de alguns meios de transmissão (por exemplo, fibras ópticas) e da eficiência do uso dos mesmos através de técnicas avançadas de codificação e multiplexação (WCDMA, WOFDM, DWDM).

O crescimento e a popularização da comunicação de voz sobre IP se mostra inevitável. Num futuro próximo, com a integração e convergência promovidas pelo IPv6, a troca de pacotes de voz com alta qualidade entre indivíduos e entre indivíduos e máquinas deve se tornar realidade em um sem-número de novas aplicações. No entanto, os atuais sistemas VoIP de uso difundido (mesmo os que possuem arquitetura sobreposta par-a-par), ainda operam de forma sub-ótima, abrindo um horizonte promissor para a pesquisa e desenvolvimento de estratégias que melhorem o desempenho dos mesmos.

Nesta dissertação, foi realizada uma análise comparativa entre três dos mais difundidos sistemas de telefonia IP da Internet. Foram observados aspectos como banda consumida, *jitter* médio, número e tamanho dos pacotes transmitidos e a taxa de transmissão dos mesmos. Além disso, foi proposta uma abordagem objetiva para comparação da QoS de sistemas de telefonia IP. Tal abordagem utiliza a função de correlação para comparar o grau de distorção entre os sinais de voz transmitidos e recebidos pelos aplicativos VoIP. Essa análise pode ser facilmente estendida a novos cenários, por exemplo:

- Sistemas VoIP em enlaces intercontinentais;
- Fluxo mútuo de voz e vídeo em redes IP;
- Multiconferências e videoconferências em redes IP;

- Análise da QoS de chamadas entre telefones VoIP e telefones convencionais PSTN;
- Interação entre protocolos e sistemas (VoIP sobre WiFi, WiMax, UMTS, e outros).

Em trabalhos futuros pretende-se ainda verificar qual é a relação entre os valores obtidos via função de correlação e os níveis de qualidade percebidos pelo usuário avaliados com ferramentas de medição subjetivas.

Além disso, as informações aqui apresentadas servem como base para que novos sistemas VoIP, estratégias de roteamento ou aplicações, baseados em redes sobrepostas par-a-par, também possam ser propostos em trabalhos futuros.

A integração dos sistemas de telefonia IP com as redes sobrepostas par-a-par faz com que a aplicação possa impor sua própria política de roteamento e de gerenciamento da QoS. Usuários de sistemas VoIP que utilizam essa tecnologia podem se comunicar mesmo quando ambos se encontram por trás das barreiras impostas por NATs (*Network Address Translators*) ou *firewalls*. Além disso, os diversos novos esquemas de roteamento que são possíveis nessa arquitetura permitem que os sistemas de telefonia IP atinjam níveis de qualidade de voz compatíveis com os observados no sistema de telefonia convencional PSTN.

São vários os aspectos que mostram que as redes com arquitetura par-a-par sobreposta têm muito a oferecer aos sistemas de telefonia IP. Transposição de NATs, seleção ótima de rotas, diminuição da taxa efetiva de perdas e redução do atraso entre os usuários são alguns deles.

O uso dessa tecnologia aumenta o poder de conectividade dos sistemas VoIP e permite que os mesmos ofereçam um serviço de voz de alta qualidade aos seus usuários, tornando a telefonia IP uma realidade cada vez mais presente no cotidiano das pessoas.

Referências

[1] SKYPE. Skype homepage. Disponível em: <http://www.skype.com>. Acesso em: 27 Abr. 2008.

[2] REN S.; GUO L.; ZHANG X. ASAP: An AS-Aware Peer-Relay Protocol for High Quality VoIP. In: IEEE INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (ICDCS'06) (26. : Jul. 2006 : Lisboa, Portugal). *Proceedings*. p. 70.

[3] SILVA JUNIOR, Jucimar Maia da. *Uma Aplicação de Voz Sobre IP Baseada no Session Initiation Protocol*. Recife, 2003. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Elétrica. UFPE.

[4] TANENBAUM, A. *Redes de Computadores*, 8ª Tiragem da Tradução da Quarta Edição. Editora Campus, 2003.

[5] SWALE, R. *Voice Over IP: Systems and Solutions*. 1. ed. London: Institution of Electrical Engineers, 2001.

[6] SILVESTRE, P. **Info Online**. Reportagem publicada em meio eletrônico na revista Info Online em 01 Dez. 2005. Disponível em: <http://info.abril.com.br/aberto/infonews/122005/01122005-5.shl>. Acesso em 27 Abr. 2008.

[7] MINOLI, D.; MINOLI, E. *Delivering Voice Over IP Networks*. 2. ed. Indiana : Willey, 2002.

[8] DALGIC, I.; FANG, H. A Comparison of H.323 and SIP for IP Telephony Signaling. In: PHOTONICS EAST (Set. 1999 : Boston, Massachusetts). *Proceedings*. Massachusetts, 1999.

[9] SCHULZRINNE, H.; ROSENBERG, J. A Comparison of SIP and H.323 for Internet Telephony. In: INTERNATIONAL WORKSHOP ON NETWORK AND OPERATING SYSTEM SUPPORT FOR DIGITAL AUDIO AND VIDEO (NOSSDAV) (8. : Jul. 1998 : Cambridge, Inglaterra). *Proceedings*. p. 83-86.

[10] INTERNET ENGINEERING TASK FORCE (IETF). *SIP: Session Initiation Protocol*, RFC 2543, 1999.

[11] COLLINS, D. *Carrier Grade Voice Over IP*. USA: Mc Graw-Hill, 2001.

[12] ANDERSEN, D. G.; SNOEREN, A. C.; BALAKRISHNAN, H. Best-Path vs. Multi-Path Overlay Routing. In: ACM SIGCOMM INTERNET MEASUREMENT CONFERENCE. (3. : Out. 2003: Miami, Florida). *Proceedings*. Florida, 2003. p. 91-100.

[13] HILT, V.; HARI, A.; HOFFMANN, M. An Efficient and Robust Overlay Routing Scheme for VoIP. In: INTERNATIONAL CONFERENCE ON INFORMATION, COMMUNICATIONS AND SIGNAL PROCESSING. (5. : Dez. 2005 : Bangkok, Tailândia). *Proceedings*. p. 508-512.

[14] REDL, S. M.; WEBER, M. K.; OLIPHANT, M. W.; *GSM and Personal Communications Handbook*. Artech House, 1998.

[15] HOLMA, H. and TOSKALA, A. *WCDMA for UMTS - Radio Access for Third Generation Mobile Systems*. John Willey & Sons. 2000.

[16] PRASAD, R.; MUÑOZ, L. *WLANs and WPANs Towards 4G Wireless*. Ed. Artech House, 2003.

[17] PIMENTEL, C. J. L. *Comunicação Digital*. Ed. Brasport, 2007.

[18] WALLINGFORD, T. *Switching for VoIP*. USA : O'Reilly, 2005.

[19] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Introduction to CCITT Signaling System No. 7*, Recommendation Q.700. 1993.

[20] SILVA, João Guilherme de Moraes. *Aplicações VoIP Utilizando o Teleporto da Rede Metropolitana da Prefeitura Municipal de Manaus*. Recife, 2004. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Elétrica. UFPE.

[21] DEFENSE ADVANCED RESEARCH PROJECTS AGENCY (DARPA). *Internet Protocol*, RFC 791. USA, Set. 1981.

[22] INTERNET ENGINEERING TASK FORCE (IETF). *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460. 1998.

[23] DEFENSE ADVANCED RESEARCH PROJECTS AGENCY (DARPA). *Transmission Control Protocol*, RFC 793. USA, Set. 1981.

[24] GALLO, M. A.; HANCOCK, W. M. *Comunicação entre Computadores e Tecnologias de Rede*. São Paulo: Pioneira Thomsom Learning, 2003.

[25] INTERNET ENGINEERING TASK FORCE (IETF). *User Datagram Protocol*, RFC 768. 1980.

[26] COMER, D. E. *Interligação em Rede com TCP/IP* Volume I. 3. ed. Rio de Janeiro : Editora Campus, 1998.

[27] CROWCROFT, J.; HANDLEY, M; WAKEMAN, I. *Internetwork Multimedia.* San Francisco : UCL Press , 1998.

[28] SCHULZRINNE, H. **RTP homepage**. Homepage com diversas informações sobre o protocolo RTP. Disponível em: <http://www.cs.columbia.edu/~hgs/rtp/>. Acesso em: 27 abr. 2008.

[29] INTERNET ENGINEERING TASK FORCE (IETF). *RTP: A Transport Protocol for Real-Time Applications*, RFC 1889. 1996.

[30] INTERNET ENGINEERING TASK FORCE (IETF). *RTP Profile for Audio e Video Conferences with Minimal Control*, RFC 1890. 1996.

[31] COSTA, D. G. *SCTP: Uma alternativa aos tradicionais protocolos de transporte da Internet*. 1.ed. Rio de Janeiro : Ciência Moderna, 2005.

[32] CCSS7. SS7 News Portal. Disponível em: <http://www.ss7.com.> Acesso em: 27 Abr. 2008.

[33] INTERNET ENGINEERING TASK FORCE (IETF). *Stream Control Transmission Protocol*, RFC 2960. 2000.

[34] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service*, Recommendation H.323. 1996.

[35] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Terminals and Others Entities that Provide Multimedia Communications Services over Packet Based Networks which May Not Provide a Guaranteed Quality of Service*, Recommendation H.323. 2006.

[36] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Call Signaling Protocols And Media Stream Packetization for Packet-Based Multimedia Communication Systems*, Recommendation H.225.0. 2003.

[37] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Control Protocol for Multimedia Communication*, Recommendation H.245. 2006.

[38] INTERNET ENGINEERING TASK FORCE (IETF). *SIP: Session Initiation Protocol*, RFC 3261. 2002.

[39] HERSENT, O.; GUIDE, D.; PETIT, J-P. *Telefonia IP*. Addison-Wesley, 2002.

[40] DAVIDSON, Jonathan; PETERS, James. *Voice Over IP Fundamentals – A Systematic Approach to Understanding the Basics of Voice Over IP*. Indianapolis: Cisco Press, 2000.

[41] SCHULZRINNE, H. **Session Initiation Protocol (SIP)**. Homepage com diversas informações sobre o protocolo SIP, mantidas pelo autor do mesmo e hospedada na Columbia University. Disponível em: <http://www.cs.columbia.edu/sip/>. Acesso em: 27 Abr. 2008.

[42] INTERNET ENGINEERING TASK FORCE (IETF). *SDP: Session Description Protocol*, RFC 2327. 1998.

[43] EGELAND, G.; ENGELSTAD, P. Peer-to-Peer IP Telephony. *Telektronikk*, v. 102, n. 1, p. 54-64, Jan. 2006.

[44] BASSET, S. A.; SCHULZRINNE, H. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS (INFOCOM 2006). (25. : Set. 2004 : Barcelona, Espanha). *Proceedings*. Barcelona, 2006. p. 1-11.

[45] RAO, B.; ANGELOV, B.; NOV, O. Fusion of Disruptive Technologies: Lessons from the Skype Case. *European Management Journal*. v. 24, n. 2-3. p. 174-188. Abr.-Jun. 2006.

[46] PISA, Ivan Torres. *MIDster – Uma Arquitetura de Compartilhamento de Imagens Médicas Baseada em Modelos Peer-to-Peer (P2P) e Serviços Web*. São Paulo, 2003. Tese (Doutorado em Física Aplicada à Medicina e Biologia) – Departamento de Física e Matemática, Universidade de São Paulo.

[47] NAPSTER. Napster homepage. Disponível em: <http://www.napster.com>. Acesso em: 27 Abr. 2008.

[48] KAZAA. Kazaa homepage. Disponível em: <http://www.kazaa.com>. Acesso em: 27 Abr. 2008.

[49] EMULE. eMule homepage. Disponível em: <http://www.emule-project.net>. Acesso em: 27 Abr. 2008.

[50] TIWANA, A. Affinity to Infinity in Peer-to-Peer Knowledge Platforms. *Communications of the ACM*, v. 46, n. 5, Mai. 2003. p. 76-80.

[51] SALAMON, A. **Domain Name Service Resources Directory (DNSRD)**. Diretório com diversos documentos relacionados ao DNS. Disponível em: <http://www.dns.net/dnsrd/>. Acesso em: 27 Abr. 2008.

[52] INTERNET ENGINEERING TASK FORCE (IETF). *Boarder Gateway Protocol 4 (BGP4)*, RFC 4271. Jan. 2006

[53] KLINGBERG, T.; MANFREDI, R., **Gnutella 0.6**. Draft, Jun. 2002. Disponível em: <http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html>. Acesso em: 27 abril 2008.

[54] KAMIENSKI, C.; SOUTO, E.; ROCHA, J. et al. Colaboração na Internet e a Tecnologia Peer-to-Peer. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (SBC2005). (25. : Jul. 2005 : São Leopoldo, Rio Grande do Sul). *Anais*. p. 1407-1454.

[55] DOVAL, D.; O'MAHONY, D. Overlay Networks - A Scalable Alternative for P2P. *IEEE Internet Computing*, Dublin, Vol. 7, N. 4, p. 79-82, Jul.- Ago. 2003.

[56] MACEDO, D. F.; OLIVEIRA, L. B.; LOUREIRO, A. A. F. Integrando Redes *Overlay* e Redes de Sensores Sem-Fio. In: WORKSHOP DE COMUNICAÇÃO SEM FIO (WCSF) (5. : Out. 2003 : São Lourenço, Minas Gerais). *Anais*. p. 190-198.

[57] ANDERSEN, D.; BALAKRISHNAN, H.; KAASHOEK, F.; MORRIS, R. Resilient Overlay Networks. In: ACM SYMPOSIUM ON OPERATING SYSTEMS (18. : Out. 2001: Banff, Canada). *Proceedings*. Banff, 2001. p. 131-145.

[58] LUA, K.; CROWCROFT, J.; PIAS, M. et al. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and Tutorials*. v. 7, n. 2, p. 72-93. Mar. 2006.

[59] PARAMESWARAN, M.; SUSARLA, A.; WHINSTON, A. P2P Networking: An Information-Sharing Alternative. *IEEE Computer*, v. 34, n. 7, p. 31-38, Jul. 2001.

[60] YANG, B.; GARCIA-MOLINA, H.; Designing a Super-Peer Network. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE'03) (19. : Mar. 2003 : Bangalore, India). *Proceedings*. p. 49-60.

[61] P2P ARCHITECT PROJECT. **Ensuring Dependability of P2P Applications at Architectural Level**. Deliverable Survey. Abr. 2002. Disponível em: <http://www.atc.gr/p2p_architect/results/0101F05_P2P Survey.pdf>. Acesso em: 27 Abr. 2008.

[62] LEONARD D.; RAI, V.; LOGUINOV, D. On Lifetime-Based Node Failure and Stochastic Resilience of Decentralized Peer-to-Peer Networks. *IEEE/ACM Transactions on Networking*. San Francisco, California, v. 15, n. 3, p. 644-656. Jun. 2007.

[63] BERRY, M. J.A.; LINOFF, G. S. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. New York: John Wiley & Sons, 2000.

[64] MATTILA, E. An Analysis of Hybrid and Pure Peer-to-Peer Technologies for IP Telephony. In: SEMINAR OF INTERNETWORKING (Abr. 2005 : Helsinki, Finlândia).

[65] INTERNET ENGINEERING TASK FORCE (IETF). *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, RFC 3489. 2003.

[66] ROSENBERG, J.; MAHY, R; HUITEMA, C. **Traversal Using Relay NAT (TURN)**. IETF Draft, Set. 2005. Disponível em: <http://www.tools.ietf.org/html/draft-rosenberg-midcom-turn-08>. Acesso em: 27 Abr. 2008.

[67] LABOVITZ, C.; AHUJA, A.; BOSE, A. et al. Delayed Internet Routing Convergence. *IEEE / ACM Transactions on Networking*. Estocolmo, Suécia, v. 9, n. 3, p. 293-306. Jun. 2001.

[68] SAVAGE, S.; COLLINS, A.; HOFFMAN, E. et al. The End-to-End Effects of Internet Path Selection. In: ACM SIGCOMM CONFERENCE (Ago. 1999 : Boston, Massachusetts). *Proceedings*. p. 289-299.

[69] AMIR, Y.; DANILOV, C.; HILSDALE, M. et al. 1-800-Overlays: Using Overlay Networks to Improve VoIP Quality. In: INTERNATIONAL WORKSHOP ON NETWORK AND OPERATING SYSTEMS SUPPORT FOR DIGITAL AUDIO AND VIDEO. (15. : Jun. 2005: Washington). Washington, 2005. p. 51-56.

[70] SPINES. The Spines Overlay Network homepage. Disponível em: <http://www.spines.org>. Acesso em: 27 Abr. 2008.

[71] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Methods for Subjective Determination of Transmission Quality*, Recommendation P.800. 1996.

[72] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *One-Way Transmission Time*, Recommendation G.114. 2003.

[73] INTERNET ENGINEERING TASK FORCE (IETF). *Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*, RFC1930. 1996.

[74] GAO, L. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, New Jersey, v. 9, n.6, p. 733-745, Dez. 2001.

[75] GUO, L.; JIANG, S.; XIAO, L. et al. Fast and Low Cost Search Schemes by Exploiting localities in P2P Networks. *Journal of Parallel and Distributing Computing*, v. 65 n. 6, p. 729-742, Jun. 2005.

[76] KRISHNAMURTHY, B.; WANG, J. On Network-Aware Clustering of Web Clients. *ACM SIGCOMM Computer Communication Review*. Estocolmo, Suécia, v. 30, n. 4, p. 97-110. Out. 2000.

[77] McMANUS, P. M. A Passive System for Server Selection within Mirrored Resource Environments Using AS Path Length Heuristics. *Applied Theory Communications*. Jun. 1999.

[78] GUMMADI, K. P.; MADHYASTHA, H. V.; GRIBBLE S. D. et al. Improving the Reliability of Internet Paths with One-Hop Source Routing. In: USENIX OSDI (6. : Dez. 2004 : San Francisco, California). *Proceedings.* California, 2004.

[79] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *The E-model, a Computational Model for Use in Transmission Planning*, Recommendation G.107. 2000.

[80] BARBOSA, D. C. P.; SOUZA, R. S.; LINS, R. D. et al. Uma Análise Comparativa da QoS do Skype, Yahoo! Messenger e Google Talk. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES (SBrT). (16. : Set. 2007: Recife, Pernambuco). *Anais*. Pernambuco, 2007. p. 1-6.

[81] CHONG, H. M.; MATTHEWS, H. S. Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems. In: IEEE INTERNATIONAL SYMPOSIUM ON **ELECTRONICS AND THE ENVIRONMENT.** (Mai. 2004: Phoenix, Arizona). *Proceedings*. p. 106-111.

[82] SUH, K.; FIGUEIREDO, D. R.; KUROSE, J. et al. Characterizing and Detecting Skype-Relayed Traffic. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS (INFOCOM06). (25. : Abr. 2006 : Barcelona, Espanha). *Proceedings*. p. 1-12.

[83] COSTEUX, J.-L.; GUYARD, F.; BUSTOS, A.-M. Detection and Comparison of RTP and Skype Traffic and Performance. In: IEEE GLOBECOM (Nov. 2006 : San Francisco, California). *Proceedings*. California, 2006. p. 1-5.

[84] EHLERT, S.; PETGANG, S. Analysis and Signature of Skype VoIP Session Traffic. In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS, INTERNET AND INFORMATION TECHNOLOGY. (Jul. 2006 : St. Thomas, Virginia).

[85] YU, Y.; LIU, D.; LI, J. et al. Traffic Identification and Overlay Measurement of Skype. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND SECURITY (Nov. 2006 : Guangzhou, China). p. 1043-1048.

[86] HOβFELD, T.; BINZENHÖFER, A.; FIEDLER, M. et al. Measurement and Analysis of Skype VoIP Traffic in 3G UMTS Systems. In: INTERNATIONAL WORKSHOP ON INTERNET PERFORMANCE, SIMULATION, MONITORING AND MEASUREMENT (IPS-MOME 2006). (4. : Fev. 2006: Salzburg, Austria). *Proceedings*. p. 52-61.

[87] GUHA, S.; NEIL, R.; DASWANI, J. An Experimental Study of the Skype Peer-to-Peer VoIP System. In: INTERNATIONAL WORKSHOP ON PEER-TO-PEER SYSTEMS IPTPS (5. : Fev. 2006: Santa Barbara, California). *Proceedings*. California, 2006. p.1-6.

[88] COCKCROFT, A. Simulation of Skype Peer-to-Peer Web Services Choreography Using Occam-Pi. In: 8th IEEE INTERNATIONAL CONFERENCE ON E-COMMERCE TECHNOLOGY AND THE 3rd IEEE INTERNATIONAL CONFERENCE ON ENTERPRISE COMPUTING, E-COMMERCE AND E-SERVICES (CEC/IEEE'06). (2006: San Francisco, California). *Proceedings*. California, 2006. p. 88.

[89] BARBOSA, R.; CALLADO, A.; KAMIENSKI, C. et al. Avaliação do Desempenho de Aplicações VoIP P2P. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES (SBrC). (24. : Mai.-Jun. 2006: Curitiba, Paraná). *Anais*. Paraná, 2006.

[90] SAT, B.; WAH, B. W. Analysis and Evaluation of the Skype and Google Talk VoIP Systems. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO. (Jul. 2006 : Ontario, Canada). *Proceedings*. p. 2153-2156.

[91] SAT, B.; WAH, B. W. Evaluation of Conversational Voice Communication Quality of the Skype, Google Talk, Windows Live and Yahoo Messenger VoIP Systems. In: IEEE WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (9. : Out. 2007 : Creta, Grécia). p. 135-138.

[92] BECVAR, Z.; NOVAK, L.; ZELENKA, J. et al. Impact of Additional Noise on Subjective and Objective Quality Assessment in VoIP. In: IEEE WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (MMSP 2007). (9. : Out. 2007 : Creta, Grécia) p. 39-42.

[93] YAHOO! MESSENGER. Yahoo! Messenger homepage: Disponível em: <http://messenger.yahoo.com/>. Acesso em: 27 Abr. 2008.

[94] GOOGLE TALK. Google Talk homepage. Disponível em: <http://www.google.com/talk/>. Acesso em: 27 Abr. 2008.

[95] TTS. Text to Speech homepage. Disponível em: <http://www.research.att.com/~ttsweb/tts/demo.php>. Acesso em: 27 Abr. 2008.

[96] MX SKYPE RECORDER. MX Skype Recorder homepage. Disponível em: <http://www.skyperec.com/>. Acesso em: 27 Abr. 2008.

[97] WIRESHARK. Wireshark homepage. Disponível em: <http://www.wireshark.org/>. Acesso em: 27 Abr. 2008.

[98] MY TRACE ROUTE. My Trace Route Fedora homepage. Disponível em: <http://fedoraproject.org/wiki/>. Acesso em: 27 Abr. 2008.

[99] OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-Time Signal Processing*. 2. ed. USA : Prentice-Hall, 1999.

[100] GLOBAL IP SOUND. Global IP Sound Solutions homepage. Disponível em: <http://www.gipscorp.com/>. Acesso em: 27 Abr. 2008.

[101] Internet Draft. FREIER, A. O.; KARLTON, P.; KOCHER, P. C. The Secure Sockets Layer Protocol Version 3.0 (SSLv3). Nov. 1996.

[102] INTERNET ENGINEERING TASK FORCE (IETF). *Extensible Messaging and Presence Protocol (XMPP)*, RFC 3920. 2004.

[103] GLOBAL INDEX. Informação contida na homepage do Skype, informando que o mesmo utiliza tecnologia Global Index. Disponível em: <http://www.skype.com/help/guides/skypeexplained/>. Acesso em: 27 Abr. 2008.

[104] LINDLEY, C. A. *Digital Audio with Java.* USA : Prentice Hall, 2000.

[105] SPANIAS, A. S. Speech Coding : A Tutorial Review, *Proceedings of IEEE*, v. 82, n. 10, p. 1541-1582. Out. 1994.

[106] OPPENHEIM, A. V.; WILLSKY, A. S. *Signals and Systems*. 2. ed. USA : Prentice Hall, 1996.

[107] OLIVEIRA, H. M. de. *Análise de Fourier e Wavelets*. 2. ed. Recife : Ed. Universitária. 2006.

[108] OLIVEIRA, H. M. de. *Análise de Sinais para Engenheiros: Uma abordagem Via Wavelets*. Ed. Brasport. 2007.

[109] PELTON, G. E. *Voice Processing*. 1. ed. USA : McGraw-Hill, 1992.

[110] RAPPAPORT, T. S. *Wireless Communications – Principles and Practice*. USA : Prentice-Hall, 1996.

[111] FLANAGAN, J. L. SCHROEDER, M.; ATAL, B. et al. Speech Coding. *IEEE Transactions on Communications*, v. 27, n. 4, p. 710-737, Abr. 1979.

[112] JAYANT, N. S.; NOLL, P. *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[113] ALENCAR, M. S. *Telefonia Digital*. 3.ed. São Paulo: Érica, 1998.

[114] ABRAMSON, N. *Information Theory and Coding*. USA: McGraw-Hill, 1963.

[115] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Pulse Code Modulation (PCM) of Voice Frequencies*, Recommendation G.711. 1993.

[116] FERNANDES, Nelson Luis Leal. *Relação Entre a Qualidade das Respostas das Recomendações G.723.1 e G.729 e o Comportamento da Rede IP de Suporte*. Rio de Janeiro, 2003. Dissertação (Mestrado em Engenharia de Sistemas e Computação). COPPE, UFRJ.

[117] SMITH, J. I. Instantaneous Companding of Quantized Signals. *Bell System Technical Journal*, v. 36 p. 653-709. Mai. 1957.

[118] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *7 kHz Audio-Coding Within 64 kbit/s*, Recommendation G.722. 1988.

[119] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *40, 32, 24, 16 kbits/s Adaptive Differential Pulse Code Modulation (ADPCM)*, Recommendation G.726. 1996.

[120] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *5-, 4-, 3- and 2-bits Sample Embedded Adaptive Differential Pulse Code Modulation (ADPCM)*, Recommendation G.727. 1990.

[121] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Coding Speech at 16kbits/s Using Low-Delay Code Excited Linear Prediction*, Recommendation G.728. 1992.

[122] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Coding of Speech at 8kbits/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, Recommendation G.729. 1996.

[123] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *G.723.1 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*, Recommendation G.723.1. 1996.

[124] COX, R. V.; KROON, P. Low Bit-Rate Speech Coders for Multimedia Communication. *IEEE Communications Magazine*. USA, v. 34, n. 12, p. 34-41, Dez. 1996.

[125] INTERNET ENGINEERING TASK FORCE (IETF). *Internet Low Bit Rate Codec*, RFC 3951. Dez. 2004.

[126] VARY, P. et al. Speech Codec for the Pan-European Mobile Radio System. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (ICASSP88) (Abr. 1988 : New York, USA). *Proceedings*. p. 227-230.

[127] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Wideband Coding of Speech at Around 16kbit/s Using Adaptive Multi-Rate Wideband (AMR-WB),* Recommendation G.722.2. 2003.

[128] VALIN, J-M. *The Speech Codec Manual Version 1.2 Beta 3*. Jean-Mark Valin / Xiph.org Foundation. Dez. 2003.

[129] GLOBAL IP SOUND (GIPS). **Internet Speech Audio Codec (iSAC)**. Datasheet. 2007. Disponível em: <http://www.gipscorp.com/files/english/datasheets/iSAC.pdf >. Acesso em: 27 Abr. 2008.

[130] GLOBAL IP SOUND (GIPS). **Enhanced G.711 (eG711)**. Datasheet. 2007. Disponível em: <http://www.gipscorp.com/files/english/datasheets/EG711.pdf>. Acesso em: 27 Abr. 2008.

[131] GLOBAL IP SOUND (GIPS). **Internet Pulse Code Modulation (iPCM)**. Datasheet. 2007. Disponível em: <http://www.gipscorp.com/files/english/datasheets/iPCM-wb.pdf >. Acesso em: 27 Abr. 2008.

[132] BEURAN R.; IVANOVICI M.; *User-Perceived Quality Assessment for VoIP Applications*. Genéve: CERN, 2004.

[133] CARVALHO, L. S. G.; MOTA, E. S.; QUEIROZ, J. M. Análise Comparativa de Padrões para Medida de Qualidade de Voz. In: SUCESU (Abr. 2003. Salvador, Bahia). *Anais*. Bahia, 2003.

[134] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Objective Quality Measurement of Telephone-Band (300-3400 Hz) Speech Codecs*, Recommendation P.861. 1998.

[135] LUSTOSA, L. C. G.; CARVALHO, L. S. G.; RODRIGUES, P. H. A. et al. Utilização do Modelo E para a Avaliação da Qualidade da Fala em Sistemas de Comunicação Baseados em Voz sobre IP. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES (22. : Mai. 2004 : Gramado, Rio Grande do Sul). *Anai*s. p. 603-616.

[136] RIX, A. W.; HOLLIER, M. P. The Perceptual Analysis Measurement System for Robust End-to-End Speech Quality Assessment. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP) (Jun. 2000 : Istambul, Turquia). *Proceedings*. v. 3. p. 1515-1518.

[137] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-band Telephone Networks and Codecs*, Recommendation P.862. 2001.

[138] SHANNON, C. E. A Mathematical Theory of Information. *Bell Systems Technical Journal*, v. 27. p. 379-423 e 623-656. Jul. e Out. 1948.

[139] MASSEY, J. L. *Applied Digital Information Theory*. Lecture Notes. 1997.

[140] HARTLEY, R. V. L. Transmission of Information. *Bell Systems Technical Journal*, USA, v. 3, n. 7, p. 535-564, Jul. 1928.

[141] INTERNET ENGINEERING TASK FORCE (IETF). *A One-Way Delay Metric for IPPM*, RFC 2679. 1999.

[142] INTERNET ENGINEERING TASK FORCE (IETF). *A Round-Trip Delay Metric for IPPM*, RFC 2681. 1999.

[143] VARSHNEY, U.; SNOW, A.; McGIVERN, M. et al. Voice Over IP, *Communications of the ACM*, v. 45, n. 1, Jan. 2002. p. 89-96.

[144] INTERNATIONAL TELECOMMUNICATIONS UNION – TELECOMMUNICATION STANDARDIZATION SECTOR (ITU-T). *Network Performance Objectives for IP-Based Services*, Recommendation Y.1541. 2002.

[145] INTERNET ENGINEERING TASK FORCE (IETF). *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*, RFC 3393. 2002.

Apêndice A – Digitalização de voz

Este apêndice descreve de um modo geral o processo de digitalização de voz e os principais codificadores utilizados nos aplicativos de telefonia IP.

A.1. O som

O som é uma onda mecânica gerada a partir da vibração do ar percebida pelos ouvidos e convertida em impulsos nervosos que são enviados ao cérebro. Como uma onda mecânica, o som necessita de um meio material para se propagar e sua velocidade de propagação depende das características desse meio. No ar, o som se propaga com uma velocidade de aproximadamente 340 m/s.

A freqüência de uma onda é definida pelo número de repetições do padrão de oscilação dessa onda em um dado intervalo de tempo. Uma freqüência de um ciclo por segundo corresponde a um Hertz (Hz), em homenagem ao físico alemão Heirinch Rudolf Hertz (1857-1894). Sons simples tendem a ser periódicos enquanto os complexos tendem a não o ser, possuindo esses um comportamento aparentemente aleatório [104]. O ouvido humano é capaz de perceber sons de freqüência desde 20 Hz até 20 kHz aproximadamente, dependendo da idade, saúde e fatores externos como a exposição excessiva a sons altos [105].

As características da onda sonora são traduzidas no sistema fisiológico do ouvido humano de forma a trazerem ao cérebro diversas informações. A amplitude da onda distingue os sons de alta intensidade dos de baixa intensidade, a freqüência da onda diferencia os sons graves dos agudos e o formato da onda determina o timbre, permitindo identificar a fonte sonora (interlocutor, dispositivo, etc.).

A amplitude da onda sonora diminui gradativamente com o aumento da distância à fonte que a emitiu. O ouvido humano responde de forma não-linear à excitação em amplitude de uma onda sonora, ou seja, sons percebidos pelo ouvido com o dobro da intensidade não implicam ondas acústicas incidindo com o dobro da amplitude.

A resposta em freqüência do ouvido humano possui um comportamento logarítmico [104]. Esse comportamento não-linear do ouvido humano fez com que a utilização de uma escala logarítmica para medir as amplitudes de uma onda sonora se tornasse mais adequada, motivo pelo qual, dentre outras coisas, os potenciômetros de controle de volume dos aparelhos de som possuem uma escala logarítmica.

O Bell, batizado em homanagem ao criador da Bell Systems, Alexander Granham Bell, foi desenvolvido inicialmente para medir atenuação em cabos telefônicos, mas logo foi aplicado em muitos ramos da acústica e eletrônica. No entanto, percebeu-se que em algumas situações o Bell era uma unidade demasiadamente grande, tornando mais comum sua definição em termos de seu submúltiplo – o decibel (dB).

Embora seja uma função da frequência do som, o ouvido humano em média é capaz de perceber pressões acústicas a partir de $2 \cdot 10^{-5}$ N/m^2. Pessoas podem ser consideradas normais se o limiar mínimo de pressão acústica percebido por seus ouvidos (mínima potência de som que pode ser distinguida do silêncio) seja de até 3 dB acima desse patamar.

A.2. Análise de Fourier

A análise de um sinal consiste em realizar uma decomposição do mesmo em termos de suas componentes fundamentais e daí, extrair características importantes para o seu estudo e manipulação.

O francês Jean-Baptiste Joseph Fourier (1768-1830), matemático, físico e engenheiro, desenvolveu a Análise de Fourier, talvez a mais importante ferramenta da engenharia no processamento de sinais. A análise de Fourier desempenha um papel fundamental no desenvolvimento dos sistemas de comunicação. Fourier mstrou que um sinal periódico pode ser representado como uma combinação linear de funções de base ortonormais, no caso, uma soma de senos e cossenos harmônicos – a Série de Fourier.

Mais tarde, o conceito de Análise de Fourier foi expandido de modo a englobar também os sinais aperiódicos utilizando o artifício de que um sinal aperiódico se repetiria no

infinito. Tal ferramenta matemática conhecida como a Transformada de Fourier decompõe um sinal, em termos de suas componentes frequenciais, determinando o seu espectro (do latim *spectrum* – fantasma). A Transformada de Fourier realiza uma mudança de domínio no sinal, trazendo-o do domínio do tempo para o domínio da freqüência, onde o seu conteúdo espectral pode ser analisado.

Seja $f(t)$ um sinal de tempo contínuo e $F(j\Omega)$ sua representação no domínio da freqüência através da transformada de Fourier. As relações entre essas representações podem ser obtidas através das expressões de síntese e análise de Fourier, que são definidas por [106]:

- **Síntese:**

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(j\Omega)\, e^{j\Omega t}\, d\Omega. \tag{10}$$

- **Análise:**

$$F(j\Omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\Omega t}\, dt. \tag{11}$$

No entanto, um sinal analógico de tempo contínuo não pode ser diretamente armazenado ou processado por um computador digital. Portanto, esse sinal deve ser digitalizado, ou seja, convertido em uma sequência de amostras suficientemente espaçadas entre si, de modo a não ser perdida nenhuma característica de interesse do sinal orginal. Nesse contexto mais prático, o processamento do sinal se dá através da DFT (*Discrete Fourier Transform*).

Seja $x[n]$ uma sequência de tempo discreto e $X[k]$ sua representação no domínio da frequência via DFT, ambas de comprimento N, ou seja:$X[k] = DFT\{x[n]\}$. As expressões de síntese e análise são definidas por [106]:

- **Síntese:**

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{\frac{j2\pi}{N}kn}. \tag{12}$$

- **Análise:**

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi}{N}kn}. \tag{13}$$

A noção de que as notas musicais possuíam uma relação especial de *"multiplicidade"* repetindo-se a cada oitava (múltiplos da freqüência fundamental) e estabelecendo entre si

em alguns casos consonâncias “harmoniosas” e, em outros, dissonâncias desagradáveis aos ouvidos era conhecida desde a antiguidade clássica pelos gregos [107]. Nesse contexto, o uso da Análise de Fourier no tratamento de sinais sonoros tornou-se uma escolha natural por se mostrar concordante com os conhecimentos já existentes.

Outras ferramentas de análise de sinais são utilizadas nos codificadores modernos. As *wavelets*, por exemplo, permitem a unificação de um grande número de técnicas de análise e processamento como codificação em sub-bandas e análise em multiresolução [108]. No entanto, as idéias de Fourier continuam tendo um papel fundamental para o desenvolvimento de todos os aspectos relativos ao processamento de sinais em engenharia, servindo como suporte matemático para modelar e tratar o áudio nos sistemas de telecomunicações.

A.3. O sinal de voz

O ser humano diferenciou-se dos outros animais por possuir três elementos fundamentais: uma mão equipada com um dedo polegar opositor capaz de manipular objetos com precisão, um cérebro altamente desenvolvido e a capacidade de produzir uma linguagem articulada. A linguagem permite que os seres humanos se comuniquem uns com os outros e troquem experiências, transmindo e registrando os conhecimentos adquiridos.

A linguagem humana pode ser expressa através da escrita ou da fala. A fala é rica em informação que possibilita os ouvidos e o cérebro de outros seres humanos discriminar suas características e compreender o seu significado. Muitos dos aspectos do sinal de voz podem ser detectados pelas máquinas e utilizados durante os processos de síntese e análise de voz [109]. Um sinal de voz típico e o seu espectro de frequências calculado através da transformada de Fourier são mostrados nas figuras A.1 e A.2, respectivamente.

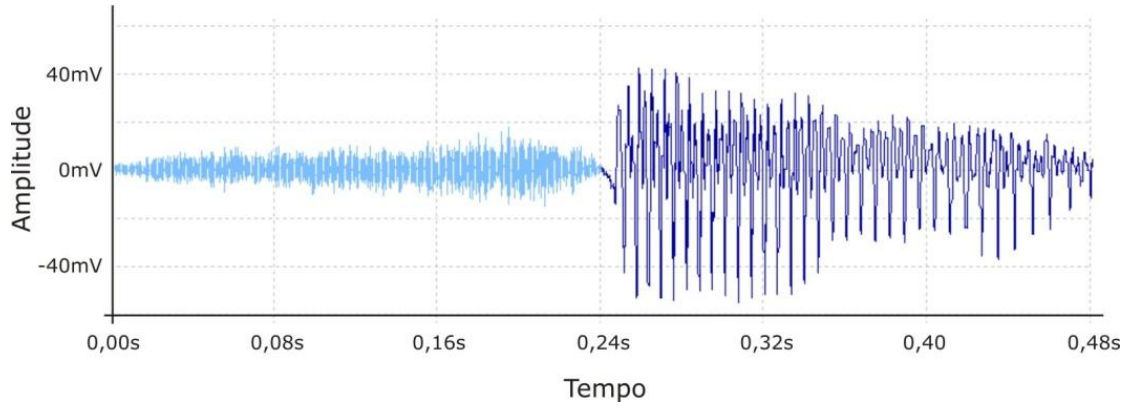


Figura A.1 – *Captura de um sinal de voz masculina pronunciando a palavra “sino” [‘sinu].*

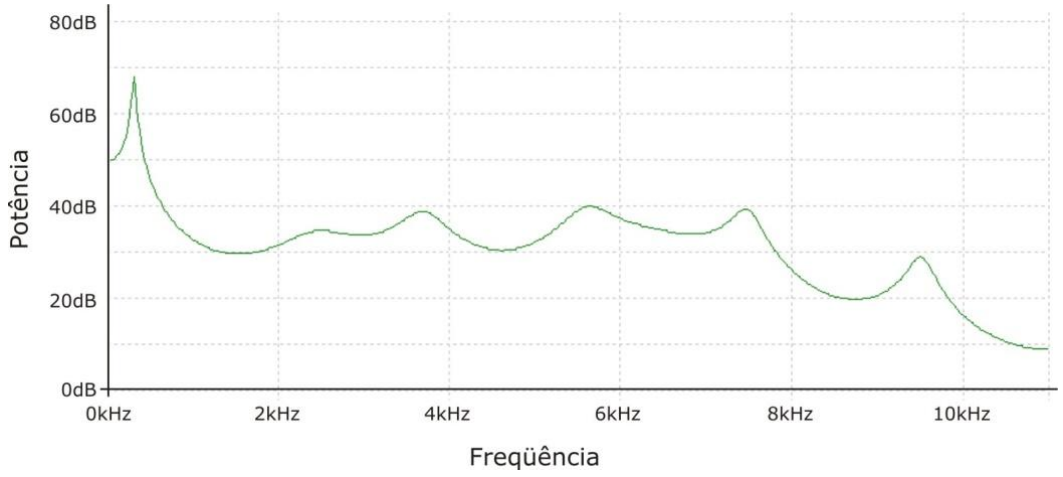


Figura A.2 – *Espectro do sinal de voz mostrado na figura A.1.*.

A.3.1. Geração da voz

O complexo sistema fonador humano baseia-se em tubos e cavidades ressonantes como mostrado na figura A.3.

O ar é pressurizado nos pulmões e enviado pela traquéia até atingir as cordas vocais fazendo-as vibrar. Esta vibração faz com que a glote, pequeno orifício localizado entre as cordas vocais, se abra e se feche repetidamente, enviando surtos de ar pressurizado às cavidades vocais com frequências que variam de acordo com as vibrações. Estes surtos de ar são refletidos nas paredes das cavidades oral e nasal e eventualmente emanam pelos lábios e pelas narinas gerando uma onda acústica de fala.

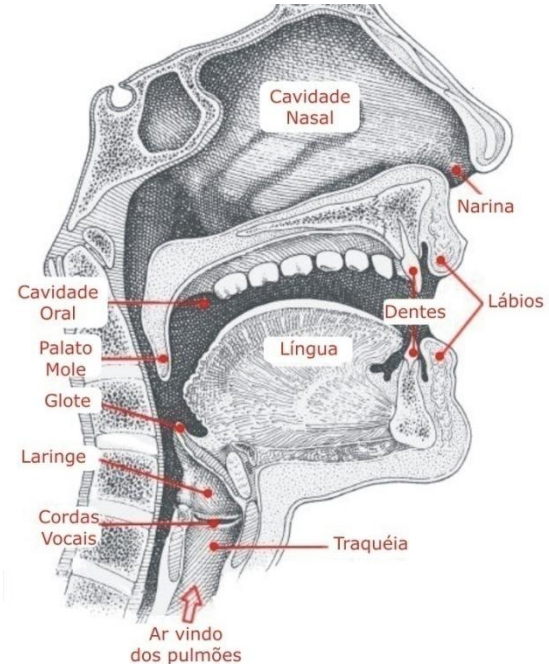


Figura A.3 – *Aparelho fonador humano.*

Diversos articuladores no trato vocal são manipulados pelo falante para produzir variados efeitos. As cordas vocais podem ser voluntariamente contraídas ou relaxadas para modificar a frequência de vibração ou simplesmente desligadas para permitir que o ar passe sem obstrução durante a respiração simples. O palato mole age como uma ligação entre as cavidades nasal e oral, podendo isolá-las ou acoplá-las, de acordo com o som a ser produzido. A língua, os dentes, a mandíbula e os lábios podem se mover de modo a alterar a configuração da cavidade oral.

A onda de som emitida depende dessa configuração assim como das características de absorção e reflexão acústicas dos diversos materiais que compõem as cavidades vocais e da fisiologia individual de cada pessoa. Desse modo, o ar impulsionado pelos pulmões encontra diversos obstáculos em seu caminho até ser emitido pelos lábios e/ou narinas. Durante o caminho, parte da energia é absorvida nas obstruções e parte é refletida nas cavidades combinando-se com outras frentes de onda. Algumas dessas ondas ressoam no trato vocal de acordo com a conformação das cavidades naquele instante, reforçando a energia das ondas com frequências ressonantes e atenuando a energia das ondas com frequências dissonantes, dando à voz as características de timbre, altura (*pitch*) e intensidade. Essas frequências ressonantes aparecem no espectro de voz como máximos locais e são chamadas de *formantes*. Para falantes adultos, os primeiros formantes encontram-se nas vizinhanças de 500 Hz, 1.500 Hz, 2.500 Hz e 3.500 Hz [110]. Os primeiros formantes da voz são os mais importantes para o modelamento do trato vocal (os demais são em geral agrupados e tratados como um único) e seu posicionamento exato varia de pessoa para pessoa.

Os sons produzidos pelo sistema fonador humano podem ser *vocálicos* ou *não-vocálicos*. Os sons vocálicos são aqueles produzidos a partir da vibração das cordas vocais em uma frequência fundamental conhecida como *frequência de pitch*. As figuras A.4 e A.5 mostram respectivamente um trecho vocálico de fala e seu espectro de frequências. Nelas podemos notar a frequência de *pitch* (período de repetição dos ciclos na figura A.4) e a localização dos formantes (picos da figura A.5).

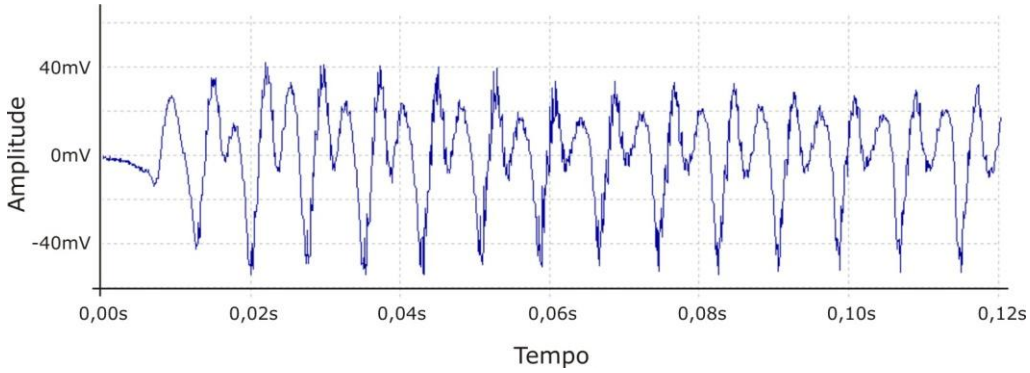


Figura A.4 – *Trecho vocálico de fala.*.

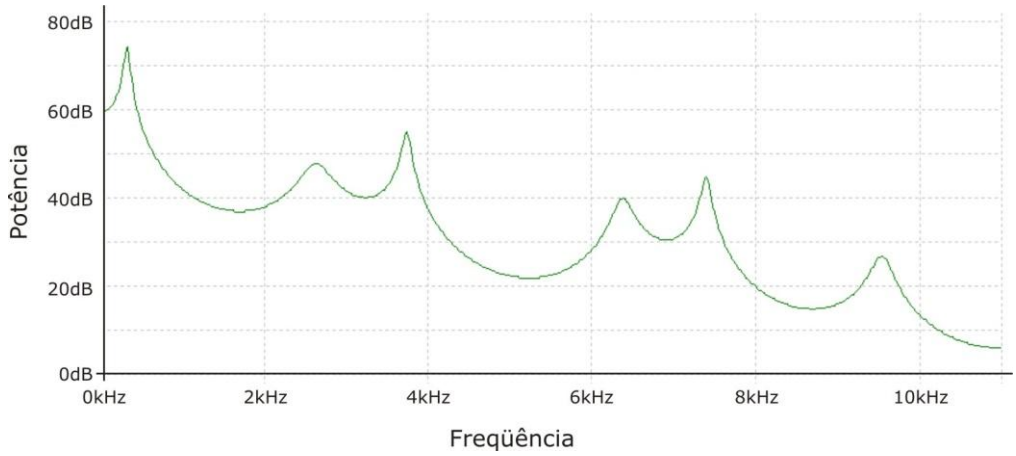


Figura A.5 – *Espectro do trecho vocálico de fala mostrado na figura A.4.*

A frequência de *pitch* de um trecho vocálico de fala é estimada através de uma técnica chamada *cepstrum*. Essa expressão foi cunhada a partir da inversão das primeiras letras da palavra *spectrum*, dando uma visão do modo de operação do processo. O *cepstrum* é definido como a transformada inversa do logaritmo do espectro de um trecho de fala, ou seja: se $x[n]$ é um trecho de fala e $X[k]$ sua DFT, temos que:

$$\hat{x}[n] = DFT^{-1}\{\log_{10} X[k]\} \quad (14)$$

é o *cepstrum* do espectro de frequências $X[k]$. Tal processamento matemático do sinal faz com que o *cepstrum* apresente um pico proeminente na frequência de *pitch*, e picos menores em seus harmônicos.

Os sons não-vocálicos são produzidos sem a vibração das cordas vocais, mantendo-as firmes e causando uma constrição dos lábios, dentes e língua. Tal constrição causa uma turbulência por onde o ar é forçado a passar. Um trecho não-vocálico de fala e seu espectro de frequências são mostrados nas figuras A.6 e A.7, respectivamente.

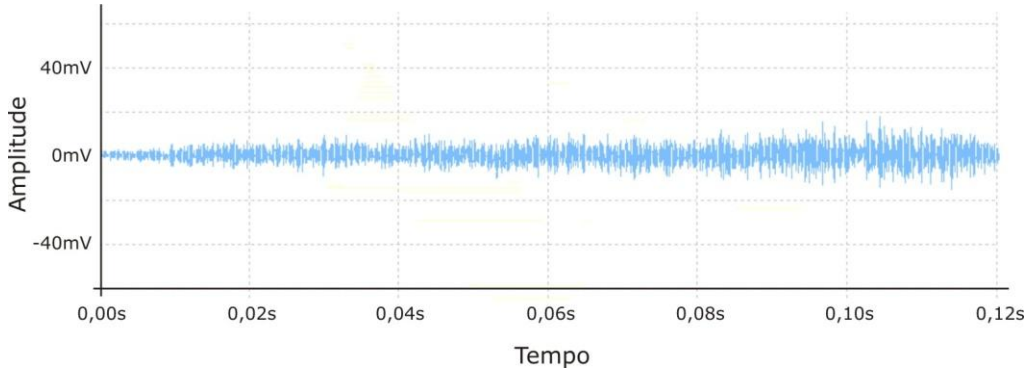


Figura A.6 – *Trecho não-vocálico de fala.*

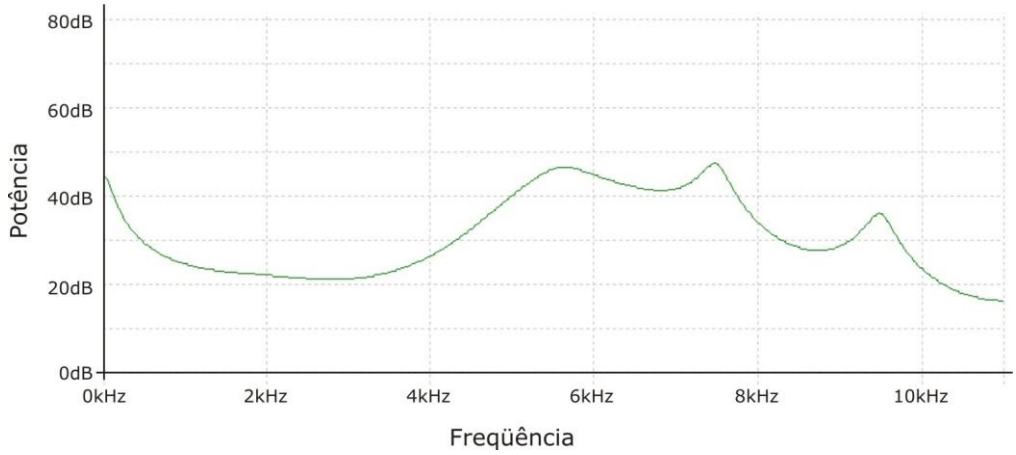


Figura A.7 – *Espectro do trecho não-vocálico de fala mostrado na figura A.6.*

De um modo geral, os trechos vocálicos são compostos por ciclos que se parecem com cópias inexatas dos ciclos vizinhos. O período de repetição de tais ciclos define o *pitch*, e sobreposto a essa frequência fundamental existe uma gama de outras componentes frequenciais responsáveis pela formação do timbre da voz. No espectro de trechos vocálicos pode-se notar facilmente o comportamento dos formantes.

Por outro lado, durante trechos não-vocálicos, se pode notar que o sinal apresenta pouca periodicidade, baixas amplitudes e componentes de alta frequência. Desse modo, o espectro de trechos não-vocálicos de fala possui um comportamento aproximadamente plano. Vale ressaltar que, embora possuam baixas amplitudes, os sons não-vocálicos podem ser diferenciados do silêncio pelo critério de número de passagens por zero.

A complexa combinação de sons vocálicos e não-vocálicos em diversas frequências e intensidades gera a fala humana que pode ser captada por um transdutor e transformada em um sinal de voz.

A.3.2. Aspectos matemáticos da voz

A voz humana possui diversas características matemáticas que podem ser utilizadas para a construção de codificadores mais eficientes [111]. Algumas dessas características tais como: não-uniformidade da função distribuição de probabilidade das amplitudes da voz; correlação não-nula entre amostras de voz sucessivas; espectro de voz não-plano; existência de segmentos vocais e não-vocais na fala e a quasi-periodicidade do sinal de voz são fortemente exploradas nesse aspecto e permitem o uso de técnicas de quantização eficientes [110].

Uma característica fundamental do sinal de voz é a de poder ser considerado de banda limitada, pois embora possua componentes de amplitude não-nula por uma faixa relativamente larga do espectro, sua energia encontra-se concentrada em frequências abaixo de 4 kHz, sendo essa frequência considerada um limite prático. De acordo com o critério de Nyquist, essa característica permite que o sinal de voz seja amostrado a uma taxa finita, correspondente ao dobro de sua máxima frequência e perfeitamente reconstruído a partir de suas amostras.

O sinal de voz é modelado através de ferramentas estatísticas. Algumas das suas principais características e definições matemáticas úteis em sua análise serão descritas aqui.

a) Função Densidade de Probabilidade

A não-uniformidade da função densidade de probabilidade (fdp) da fala é uma de suas características mais exploradas. A fdp de um sinal de voz possui amplitudes muito altas perto de zero, decrescendo monotonicamente com o aumento da freqüência até se tornar praticamente nula em frequências muito altas. A distribuição exata depende de fatores como largura de banda e condições de aquisição do sinal de entrada.

A fdp de longo prazo de um sinal com qualidade telefônica pode ser aproximada por uma distribuição exponencial bi-ladeada ou Laplaciana definida por [112]:

$$p_{lt}(x) = \frac{1}{\sqrt{2\sigma_x}} e^{-\frac{\sqrt{2}|x|}{\sigma_x}}. \qquad (15)$$

Para modelar a fdp de curto prazo é utilizada a distribuição Gaussiana definida por:

$$p_{st}(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x-m_x)^2}{2\sigma_x^2}}, \qquad (16)$$

com m_x e σ_x^2 a média e a variância das distribuições, respectivamente. Ambas as distribuições possuem um pico em zero devido à presença frequente de pausas e de segmentos de fala de baixa intensidade.

b) Função de Autocorrelação

A função de autocorrelação (fac) determina uma medida quantitativa do grau de similaridade entre duas amostras de um sinal de voz em função da separação temporal k entre as mesmas. A fac para sinais de tempo discreto é definida pela expressão [112]:

$$C[k] = \frac{1}{N} \sum_{n=0}^{N-|n|-1} \{x[n] \cdot x[n + |k|]\}. \quad (17)$$

A função de autocorrelação é geralmente normalizada com a variância do sinal de voz e excursiona entre valores do intervalo $[-1,1]$ com $C[0] = 1$. Existe muita correlação entre amostras adjacentes de um sinal de voz. Isso implica que em cada amostra existe uma grande quantidade de informação que é facilmente previsível a partir dos valores das amostras anteriores com um pequeno erro aleatório. Valores típicos da função de autocorrelação entre amostras consecutivas $C[1]$ de um sinal de voz estão entre 0,85 e 0,9.

c) Função Densidade Espectral de Potência

A função densidade espectral de potência (dep) é a representação frequencial via transformada de Fourier da função de autocorrelação, segundo o teorema de Wiener-Kintchine [99]. A função densidade de potência espectral da fala é não-plana, assim, é possível se obter uma compressão significante codificando o sinal no domínio da freqüência. Tal característica do sinal de voz é simplesmente uma manifestação no domínio da freqüência do fato da função de autocorrelação ser não-nula [110].

Típicas distribuições da função densidade de potência espectral mostram que as componentes de alta freqüência de um sinal de voz contribuem muito pouco para a energia total do sinal, no entanto, carregam informações importantes sobre o mesmo, devendo ser adequadamente representadas no sistema de codificação.

A.4. Processamento de voz

O processamento de voz pode ser dividido em duas partes: análise e síntese de voz. A análise de voz pode ser entendida como a parte do processamento de voz capaz de converter a voz humana em uma forma digital apropriada para a sua transmissão ou

armazenamento em computadores. A síntese de voz faz essencialmente o processo inverso: converte os dados de voz digital em uma forma similar a da voz original que seja capaz de ser reproduzida em um transdutor [109].

Um *codec* (codificador-decodificador) é um dispositivo capaz de realizar as funções de análise e síntese de voz. O *codec* é dividido em três partes básicas: amostragem, quantização e codificação. A amostragem e quantização são etapas da conversão analógico-digital que tem como finalidade converter o sinal de voz em uma forma que possa ser entendida e processada por um computador digital.

A função da codificação é comprimir e proteger os dados, ou seja, representar o sinal de voz digitalizado utilizando o menor número de bits possível, de forma a economizar memória durante o armazenamento e banda durante a transmissão do mesmo e possibilitar que possíveis erros sejam detectados e corrigidos. A seguir serão descritos os principais aspectos de cada uma das etapas da digitalização de voz.

A.4.1. Conversão analógico-digital

Um conversor analógico-digital (conversor A/D) é um dispositivo que recebe como entrada um sinal de tempo contínuo e amplitude contínua (sinal analógico) e produz como saída um sinal de tempo discreto e amplitude discreta (seqüência discreta). Dois processos estão envolvidos nessa tarefa: a amostragem (discretização no tempo) e a quantização (discretização na amplitude).

De um modo geral, no conversor A/D o sinal analógico é convertido em uma seqüência de amostras que o representam no domínio digital. Essas amostras são como sucessivas "fotografias" do sinal (no caso, uma onda sonora), que se colhidas a uma taxa adequada, podem representar perfeitamente o sinal contínuo original, tal como uma seqüência de quadros estáticos é capaz de representar fielmente um movimento contínuo nas telas do cinema. Uma seqüência produzida pelo conversor A/D, portanto, representa a informação de entrada com um determinado grau de precisão, o qual depende da freqüência com que o sinal é amostrado (resolução no tempo) e da quantidade de bits utilizados para representar cada amostra (resolução na amplitude).

O processo de amostragem consiste em colher periodicamente o valor instantâneo do sinal, o qual é geralmente realizado a uma taxa constante conhecida como freqüência de amostragem de acordo com o teorema de Shannon-Nyquist. Os valores das amostras em cada instante são obtidos utilizando a técnica conhecida como *sample-and-hold* (amostrar e

manter), na qual o valor do sinal em um dado instante é carregado e mantido em um capacitor até que a amostragem de um novo trecho do sinal seja realizada.

Como resultado do processo de amostragem, tem-se um sinal de tempo discreto, ou seja, cujos valores só são definidos em pontos discretos do eixo temporal, mas com uma amplitude que ainda varia no *continuum*.

A próxima etapa da conversão A/D é a discretização do eixo das amplitudes do sinal através da quantização. A quantização é um processo não-linear que tem como objetivo mapear o valor da amplitude do sinal que varia no *continuum* em um número finito de valores discretos (geralmente números binários). Existem diversas técnicas de quantização.

Quantização uniforme

A técnica de quantização mais comum é a *quantização uniforme*, na qual a máxima excursão do sinal R é dividida em 2^n segmentos iguais, sendo cada um deles representado por uma única palavra-código de n bits. O *passo de quantização* s é o comprimento de cada segmento e é definido através da expressão:

$$R = s(2^n). \tag{18}$$

A função de mapeamento da quantização uniforme é mostrada na figura A.8.

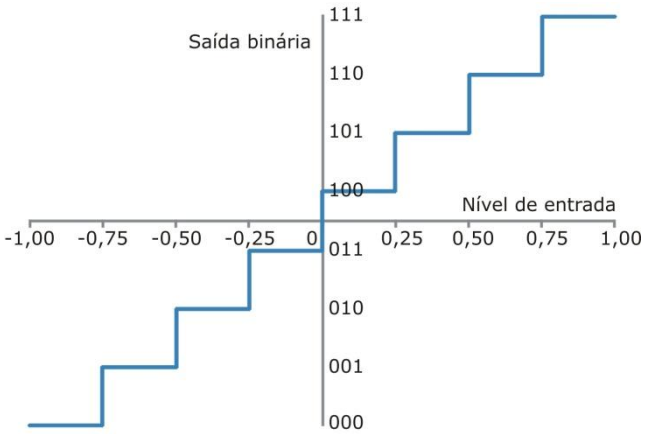


Figura A.8 – *Quantização uniforme.*

Caso o valor da amostra de entrada seja maior que R ocorrerá um ceifamento do sinal. O processo de quantização é inerentemente um processo com perdas. A diferença entre o valor real da amostra $x[n]$ e a sua representação discreta (quantizada) $Q\{x[n]\}$ produz um erro não-linear $e[n]$, conhecido como *ruído de quantização* e dado por:

$$e[n] = x[n] - Q\{x[n]\}. \tag{19}$$

Aproximações sucessivas por arredondamento ou truncamento em sistemas realimentados introduzem não-linearidades que podem gerar padrões repetitivos ou *ciclos-*

limite [99] que causam tons audíveis indesejados em sistemas de áudio. Em geral, a modelagem de efeitos não-lineares é difícil e sua análise é realizada em sua maioria através de simulações.

No entanto, para um sinal de banda larga como o de voz, que flutua rapidamente entre todos os níveis de quantização, cruzando vários deles de uma amostra para outra, a análise pode ser realizada de forma bastante precisa através de ferramentas de estatística [99]. Essa análise consiste em substituir a fonte não-linear do ruído de quantização por uma fonte estocástica linear equivalente cuja função densidade de probabilidade é uniforme no intervalo de quantização.

O quantizador pode ainda utilizar uma estratégia de aproximação por truncamento ou arredondamento da amostra. Desse modo, se m_e é a média e σ_e^2 a variância da fonte de ruído linear equivalente, para uma palavra de $(B+1)$ bits, tem-se que:

a) Para aproximação por arredondamento:

$$\begin{cases} -\frac{1}{2}2^{-B} < e[n] \leq \frac{1}{2}2^{-B} \\ m_e = 0 \\ \sigma_e^2 = \frac{2^{-2B}}{12} \end{cases} \tag{20}$$

b) Para aproximação por truncamento:

$$\begin{cases} -2^{-B} < e[n] \leq 0 \\ m_e = -\frac{2^{-B}}{2} \\ \sigma_e^2 = \frac{2^{-2B}}{12} \end{cases} \tag{21}$$

A *relação sinal-ruído* é definida como a razão entre a energia do sinal e a energia do ruído e usualmente é medida em decibéis (dB). Para um total de N amostras, a relação sinal-ruído é dada por:

$$SNR_{dB} = 10 \log \left(\frac{\sum_{n=1}^{N} x^2[n]}{\sum_{n=1}^{N} e^2[n]} \right). \tag{22}$$

Se o SNR é baixo, a voz perde inteligibilidade e o usuário sente desconforto, dificuldade para entender e é afetado pelos efeitos físicos e psicológicos adversos associados à redução da qualidade da voz [109].

O patamar aceito para voz com qualidade telefônica corresponde a um sinal de voz cuja relação sinal-ruído se mantém acima de 30 dB na maioria da sua excursão. Além disso, cada acréscimo/diminuição de 1 bit na palavra digital do quantizador faz com que a relação

sinal-ruído devido a quantização SNR_Q melhore/piore em aproximadamente 6 dB conforme a expressão:

$$SNR_Q = 6{,}02n + \alpha, \tag{23}$$

na qual n representa o número de bits e α é um escalar que assume os valores $\alpha = 0$ para SNR_Q médio e $\alpha = 4{,}77$ para SNR_Q de pico [110].

Quantização não-uniforme

Utilizando a quantização uniforme, a relação sinal-ruído é dependente da amplitude do sinal, sendo pior para sinais de entrada de baixa amplitude e melhor para sinais de entrada com maiores amplitudes. Assim, com o objetivo de minimizar os efeitos do ruído de quantização e manter uma relação sinal-ruído constante em todos os níveis de amplitude do sinal, uma técnica de *quantização não-uniforme* deve ser empregada.

Os quantizadores não-uniformes alocam mais níveis de quantização para as regiões com alta probabilidade e menos níveis para as regiões com baixa probabilidade, otimizando assim o processo de quantização.

A forma mais popular de quantização não-linear é a *quantização logarítmica*. Com o uso da quantização logarítmica ao invés de se quantizar a amostra propriamente dita $Q\{x[n]\}$, codifica-se o seu logaritmo $Q\{y[n]\}$ através de uma expressão da forma:

$$y[n] = h + k \log x[n], \tag{24}$$

sendo h e k constantes positivas. Essa expressão é válida apenas para valores positivos de $x[n]$ e uma aproximação linear por partes deve ser utilizada para abranger valores nulos ou negativos, como mostra a figura A.9.

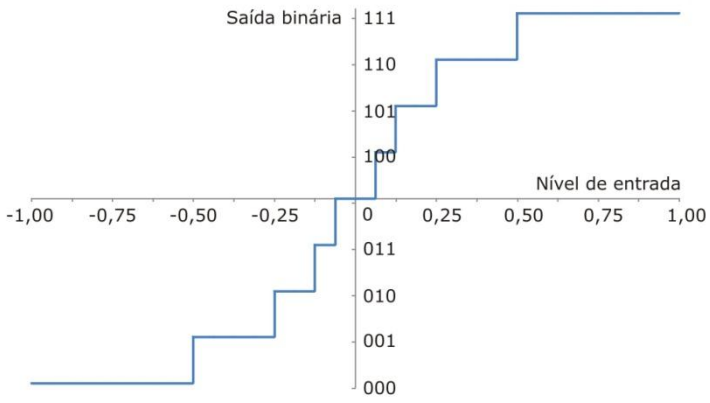


Figura A.9 – *Quantização logarítmica.*

O processo de quantização logarítmica é em si um processo que comprime o valor do sinal de entrada através da função logarítmica. Para se recuperar o sinal no receptor, uma

operação de expansão através da função exponencial é necessária. O ciclo completo é geralmente chamado de *companding* formado a partir da junção das palavras do inglês para compressão e expansão (***com****pressing* e *ex****panding***).

A figura A.10 mostra um gráfico comparativo entre a codificação uniforme e a codificação logarítmica. Percebe-se que a codificação logarítmica com 7 bits de resolução mantém uma relação sinal-ruído alta e aproximadamente constante, atingindo qualidade telefônica (30 dB) para sinais de entrada ligeiramente maiores que 1 % da excursão total. Essa característica garante a economia de aproximadamente quatro bits em relação à quantização uniforme para uma mesma relação sinal-ruído de quantização, ou de aproximadamente 32 kbps na taxa de transmissão para uma taxa de amostragem de 8 kHz.

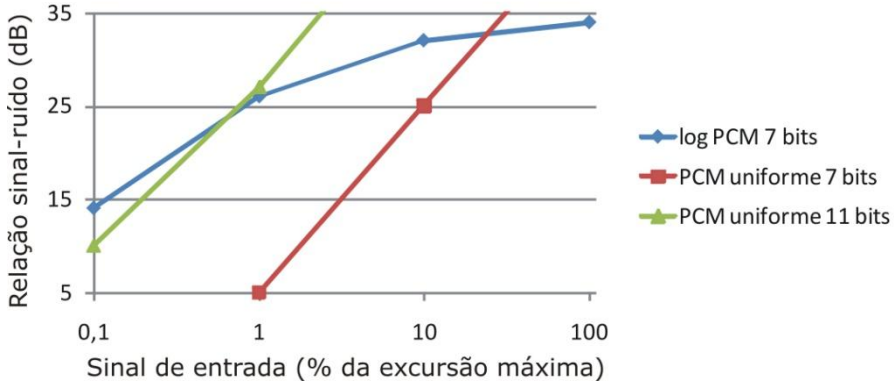


Figura A.10 – *Comparação entre quantização uniforme e logarítmica.*

Quantização adaptativa – Os sinais de voz possuem uma característica de não-estacionariedade que implica em diferenças entre suas funções densidade de probabilidade de curto e longo prazo. De fato, esse aspecto faz com que a variação de um sinal de voz possa excursionar numa escala de 40 dB ou mais [110]. Através da *quantização adaptativa* é possível explorar essa característica de forma a obter uma melhor relação sinal-ruído na voz digitalizada final. A quantização adaptativa consiste em adequar dinamicamente o passo de quantização a cada trecho do sinal, expandindo-o a medida que o sinal aumenta de intensidade e vice-versa.

Existem várias formas de quantização adaptativa que podem usar tanto quantizadores uniformes quanto não-uniformes (*adaptative PCM, feed foward adaptative PCM, feedback adaptative PCM, etc*.) e todas tentam balancear o acréscimo no passo de quantização com a diminuição na relação sinal-ruído associada. Tais quantizadores são conhecidos como *quantizadores de bloco*, pois aspectos como: energia de curto prazo, variância, faixa

dinâmica são calculados sobre um bloco de N amostras para o ajuste do passo de quantização.

Os quantizadores podem ainda ser divididos em *instantâneos* ou *silábicos*, dependendo da taxa com a qual novas informações sobre o trecho de voz são adquiridas e novos parâmetros de adaptação calculados. Os quantizadores silábicos diferem dos instantâneos porque suas características são atualizadas com uma taxa próxima a da ocorrência das sílabas em um trecho de fala. A quantização adaptativa funciona como se cada trecho do sinal de voz possuísse um quantizador próprio, feito sob medida para as suas necessidades, diminuindo o desperdício e aumentando a eficiência do sistema, como mostrado na figura A.11.

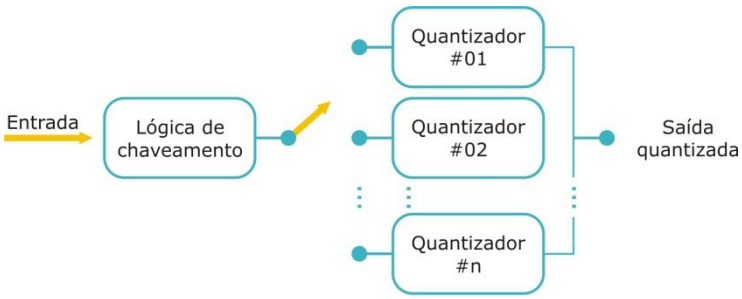


Figura A.11 – *Quantização adaptativa.*

Para sistemas com palavras com mesmo número de bits, quantizadores que utilizam a quantização adaptativa apresentam um ganho na relação sinal-ruído em comparação aos que não a utilizam de aproximadamente 6 dB, ou seja, funcionam como se possuíssem virtualmente um bit a mais na palavra de quantização [109].

Quantização vetorial

A quantização vetorial é uma técnica na qual um grupo de vetores de entrada que possuem certa "proximidade" são agrupados e mapeados em um único vetor que representa todos os membros deste conjunto, como mostra a figura A.12.

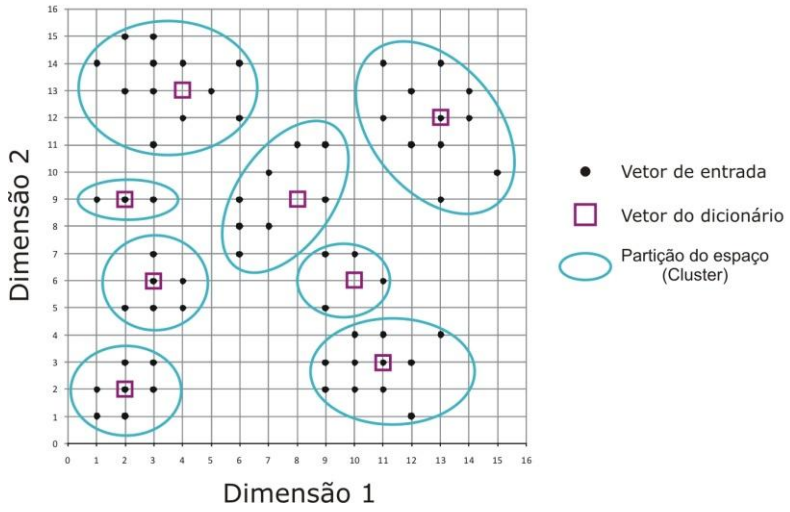


Figura A.12 – *Quantização vetorial.*

Os diversos vetores de entrada (pontos da figura A.12) são mapeados em um vetor do dicionário (quadrados da figura A.12) de acordo com a região do espaço na qual estejam localizados. Ou seja, a quantização vetorial é uma técnica de divisão do espaço composto pelos possíveis sinais de entrada (em partes geralmente disjuntas), seguida da escolha de um representante para cada uma das partições nas quais o espaço foi dividido.

Esses vetores de representação são escolhidos após sessões de treino periódicas por um algoritmo de treinamento, de forma que sejam os melhores representantes da coleção de sinais de cada conjunto através de algum critério (ex: mínimo erro quadrático ou centróide da partição do espaço). Em seguida, os vetores de representação são agrupados em um dicionário (*codebook*) e associados a um dado índice.

Durante a operação normal, após receber um vetor de entrada da fonte, o quantizador decide a qual conjunto ele pertence e o substitui pelo vetor de representação deste conjunto (ou pelo seu respectivo índice no dicionário). Dessa forma, o ruído de quantização é definido pela "distância" entre o vetor de entrada e o seu respectivo vetor de representação presente no dicionário [109]. O desempenho da quantização vetorial é função da eficiência do algoritmo de treinamento para contrução de seu dicionário.

A.4.2. Critério para reconstrução perfeita

Na conversão analógico-digital, um número discreto de amostras deve ser colhido do sinal analógico original. O bom-senso intuitivamente levaria a crer que um número demasiadamente pequeno de amostras resultaria em uma reconstrução imperfeita do sinal original devido à perda de informação entre amostras. Por outro lado, se poderia pensar

que, quanto mais freqüentemente fossem colhidas as amostras, mais perfeita seria a reconstrução, dado que os espaços entre amostras seriam menores. Esse fato se estenderia indefinidamente com a diminuição dos espaços ente as amostras, porém, sempre alguma perda seria esperada.

No entanto, o Teorema de Shannon-Nyquist mostra que, dadas algumas condições, um conjunto de amostras indexadas nos inteiros, podem sim representar perfeitamente um sinal que varia no *continuum* sem nenhuma perda de informação. O teorema é anunciado como segue:

Seja um sinal $f(t)$ de banda limitada, ou seja, cujo espectro de freqüências via transformada de Fourier $F(j\Omega)$ é nulo para freqüências acima de certa freqüência máxima Ω_M. Para que nenhuma informação seja perdida no processo de amostragem, deve-se amostrar esse sinal com uma taxa maior ou igual que o dobro da máxima frequência presente em seu espectro de Fourier [11, 113]. Essa freqüência Ω_S é chamada freqüência de Nyquist ou freqüência de amostragem do sinal. Matematicamente:

$$\text{Se: } f(t) \leftrightarrow F(j\Omega), \text{ e } \forall\ \Omega > \Omega_M \rightarrow F(j\Omega) = 0, \text{ então: } \Omega_S \geq 2\Omega_M.$$

Em 1928, o físico e engenheiro sueco Henry Nyquist (1889-1976) foi o primeiro a observar esse fenômeno cujos resultados foram posteriormente formalizados em 1948 pelo matemático e engenheiro americano Claude Elwood Shannon (1916-2001).

O critério por trás do teorema de Shannon-Nyquist é bastante sutil, mas de uma enorme aplicação prática. Ocorre que um sinal não pode ser ao mesmo tempo limitado nos domínios do tempo e da freqüência. O sinal de voz é de tempo finito, logo ele possui infinitas componentes no domínio da frequência. No entanto, a amplitude das componentes de alta frequência tornam-se cada vez menores a partir de certo ponto que define a *banda efetiva* do sinal, podendo assim ser desprezadas. Desse modo, na prática filtra-se o sinal levando suas componentes espectrais de alta freqüência tão próximas de zero quanto possível através dos filtros *anti-aliasing*. Ou seja, os filtros *anti-aliasing* forçam o sinal a ter banda limitada para que o teorema possa ser aplicado normalmente.

A voz humana possui componentes espectrais até uma freqüência de aproximadamente 12 kHz, no entanto, uma grande parcela dessas componentes não contribuem de forma essencial para a formação do sinal de voz. A maior parcela da energia do sinal de voz encontra-se até os 4 kHz, portanto, em aplicações de telefonia por exemplo, o sinal de voz é filtrado em 3,3 – 4 kHz e amostrado em PCM a uma taxa de 8.000 amostras por segundo

(de acordo com o teorema de Shannon-Nyquist). Desse modo, tem-se não só um sinal inteligível, mas também a possibilidade de reconhecimento do interlocutor.

A.4.3. Tipos de Codificadores

Em VoIP, os fluxos de áudio e vídeo são transportados em pacotes de bytes através de uma rede IP. Como qualquer outro canal de comunicação, a rede possui uma capacidade limitada de transportar informação [114].

Os *codecs* (codificadores-decodificadores) são sistemas que permitem representar sinais de voz (ou vídeo) de forma eficiente através de um número mínimo de bits, reduzindo a largura de banda necessária para sua transmissão e a quantidade de memória necessária para seu armazenamento.

Dependendo da relação entre o nível de degradação considerado aceitável e da taxa de compressão desejada, o *codec* pode extrair do sinal apenas informações inúteis, redundantes ou previsíveis (codificação sem perdas) ou mesmo eliminar aquelas que não são suficientemente relevantes e podem ser simplesmente desprezadas (codificação com perdas).

É importante notar que devido às ineficiências do protocolo IP, não é muito vantajoso que os *codecs* reduzam a taxa de transmissão abaixo de determinados limites. Isso se dá principalmente porque os cabeçalhos do IP, RTP e UDP adicionam 40 bytes de por pacote. Num sistema que opera tipicamente com pacotes de 20 ms de voz isso resulta num *overhead* de 16 kbps [5].

A figura A.13 mostra um gráfico da eficiência de compressão dos *codecs*, supondo que os mesmos operam com pacotes de 20 ms de voz, sendo eficiência de compressão definida como a redução na taxa de transmissão em relação aos 64 kbps do codificador G.711 que é utilizado na PSTN [5].

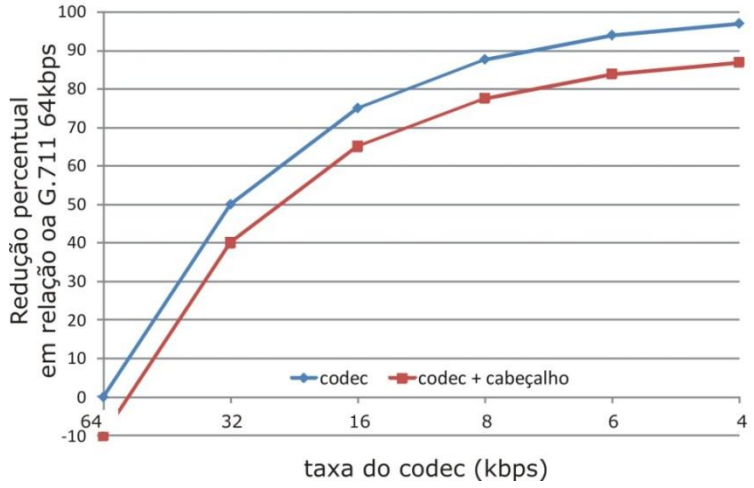


Figura A.13 – *Eficiência de compressão dos codecs. Extraído de [5].*

Existem basicamente três tipos de codificadores de voz de acordo com o modo de operação, são eles [11, 113]:

Forma de onda – Procuram reproduzir a forma de onda do sinal original, amostra por amostra, de modo que o sinal reproduzido possua a maior semelhança possível com o sinal original. São codificadores de alta qualidade, baixo atraso e pequena complexidade de implementação, porém necessitam de um uma alta taxa de transmissão (acima de 16 kbps).

Os codificadores de forma de onda são baseados quase que exclusivamente no teorema de Shannon-Nyquist, embora outras técnicas sejam usadas para minimizar o número de bits necessários para a representação do sinal. A mais comum dessas técnicas é a codificação diferencial, que ao invés de representar o valor de uma amostra, representa a diferença entre amostras suscessivas. O codificador de forma de onda mais simples é o *Pulse Code Modulation* (PCM).

Vocoders – Ao invés de tentar copiar tão fielmente quanto possível a forma de onda de um sinal de voz, os *vocoders*, codificadores de fonte ou codificadores paramétricos representam a fala através de um conjunto de características associados a um modelo simplificado de produção de voz, permitindo assim sua implementação com um custo computacional adequado. Baseados na noção de que o trato vocal muda lentamente e seu estado e configuração podem ser representados por um pequeno conjunto de parâmetros, os *vocoders* extraem e transmitem periodicamente as principais características do sinal de voz.

Um *vocoder* típico é mostrado na figura A.14. Tipicamente, os parâmetros de um *vocoder* são extraídos do espectro do sinal de voz e atualizados a cada 10-25 ms. De uma

forma simplificada, um *vocoder* caracteriza um trecho de fala através de um pequeno conjunto de parâmetros úteis que incluem: 10 ou 15 coeficientes do filtro que define as características de ressonância do trato vocal (formantes); um bit que indica se o trecho é vocálico ou não-vocálico; um valor para a intensidade do sinal e outro para a frequência de *pitch* (apenas durante trechos vocálicos).

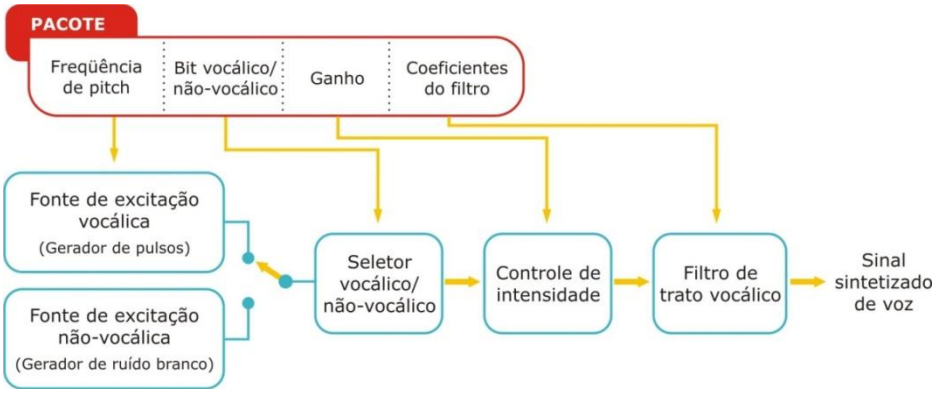


Figura A.14 – *Esquema simplificado de um vocoder típico.*

No entanto, devido à complexidade do processo de geração da voz humana, as modelagens, simplificações e aproximações utilizadas nos codificadores paramétricos introduzem perdas e distorções que acabam por tornar a qualidade da voz obtida nos mesmos inferior àquela obtida em codificadores de forma de onda.

Os *vocoders* atingem níveis de compactação consideráveis (operando normalmente entre 1,2 e 9,6 kbps), com uma taxa de transmissão média de 2,4 kbps [3]. Esses codificadores são uma solução em sistemas que possuam poucas restrições de qualidade da voz e com pouca largura de banda disponível, tais como em alguns sistemas de comunicação móvel.

Híbridos – Os codificadores híbridos combinam a qualidade dos codificadores de forma de onda com a eficiência dos codificadores paramétricos. Operam com taxas de transmissão de 4,8 a 16 kbps.

A.4.4. Técnicas de codificação de voz

G.711 – PCM

Desenvolvido pelo ITU-T em 1972, o padrão G.711 define um codificador de forma de onda que utiliza PCM (*Pulse Code Modulation*) com quantização logarítmica [115]. Duas principais formas de compressão logarítmica são utilizadas nesses codificadores: *μ-Law*, utilizada na América do Norte e Japão e *A-Law* utilizada na Europa e Brasil [116]. Nos

sistemas *μ-Law*, sinais fracos são amplificados e sinais fortes são comprimidos utilizando a seguinte expressão [117]:

$$|y[n]| = \frac{\ln(1+\mu|x[n]|)}{\ln(1+\mu)}, \tag{25}$$

com μ sendo uma constante positiva com valores tipicamente entre 50 e 300, $x[n]$ o sinal de enrada e $y[n]$ o sinal de saída. O valor de pico do sinal de entrada $x[n]$ deve ser normalizado em 1. Os sistemas *A-Law*, por sua vez, são definidos por [110]:

$$|y[n]| = \begin{cases} \dfrac{A|x[n]|}{1+\ln A}, & 0 \le |x[n]| \le \dfrac{1}{A}, \\ \dfrac{1+\ln(A|x[n]|)}{1+\ln A}, & \dfrac{1}{A} \le |x[n]| \le 1\,. \end{cases} \tag{26}$$

O PCM é uma das formas mais simples de se digitalizar voz, sendo usado em redes ISDN e na maioria dos *backbones* de telefonia digital. O espectro de voz utilizado em telefonia possui uma banda de 4 kHz, desse modo, segundo o teorema de Shannon-Nyquist, deve-se coletar 8.000 amostras por segundo desse sinal para se obter uma reconstrução perfeita. O G.711 quantiza essas amostras em 256 níveis, utilizando palavras com oito bits de comprimento. A transmissão de 8.000 amostras por segundo, cada uma com oito bits, totaliza 64 kbps, taxa de transmissão do PCM e largura de banda do canal necessária para sua transmissão.

G.722 – SB-ADPCM

O G.711 possui uma qualidade excelente, porém, parte do espectro de voz (acima de 4 kHz) é perdido. Padronizado pelo ITU-T em 1988, o G.722 foi proposto para aplicações de áudio de alta qualidade codificando a voz no espectro de 50 a 7.000 Hz através do uso do codificador de forma de onda SB-ADPCM (Sub-*Band Adaptive Differential Pulse Code Modulation*) [118].

A codificação em sub-bandas (SBC) subdivide o espectro de voz em um pequeno número de faixas de freqüência (sub-bandas) e os codifica individualmente. As melhorias alcançadas no uso desta técnica residem no fato das sub-bandas poderem ser transladadas no espectro possibilitando o uso de menores taxas de amostragem. Para ter uma idéia da melhoria alcançada com o uso de SBC, sinais codificados com SB-ADPCM a 16 kbps foram preferidos por 90% dos ouvintes submetidos a testes de comparação com o ADPCM de mesma taxa de dados [109].

A codificação adaptativa (*Adaptive Coding*) provê melhorias ajustando dinamicamente o passo de quantização ao nível de excursão do sinal, fazendo com que virtualmente cada trecho do sinal possua um quantizador próprio, dimensionado para as suas características. Os quantizadores adaptativos utilizados no G.722 são atualizados em uma taxa próxima a taxa silábica. Dado que amostras sucessivas de um sinal de voz possuem uma alta correlação entre si, permitindo estimativas com razoável nível de acerto, o ADPCM utiliza um preditor linear para estimar o valor da próxima amostra do sinal de acordo com as amostras passadas.

Por sua vez, a quantização diferencial (*Differencial Coding*) explora as características de variação temporal do sinal de voz para codificar o sinal diferença entre o sinal estimado pelo preditor e o sinal que efetivamente é recebido na entrada do codificador, utilizando um menor número de bits em relação ao necessário para codificar o sinal propriamente dito. O G.722 faz uso da combinação dessas técnicas, tornando possível a transmissão de voz de alta qualidade com taxas de 48, 56 e 64 kbps.

G.726 – ADPCM

Padronizado pelo ITU-T em 1990, o G.726 utiliza um codificador de forma de onda ADPCM (*Adaptive Differential Pulse Code Modulation*) [119]. Projetado para ser uma evolução do G.711, operando em taxas de transmissão mais baixas, o sistema ADPCM do G.726 incorpora três características em relação ao PCM do G.711: predição linear, quantização adaptativa e codificação diferencial.

No G.726 os sinais codificados em μ*-Law* ou *A-Law* são convertidos em PCM uniforme e a diferença entre este sinal e o seu valor estimado é codificada adaptativamente em uma palavra com comprimento entre cinco e dois dígitos binários, permitindo a operação em 40, 32, 24 ou 16 kbps.

O ADPCM possui um tamanho de passo e um preditor que rastreiam e se adaptam as características estatísticas variantes no tempo da fala. O preditor pode ser pós-alimentado (*feedback adaptive*) ou pré-alimentado (*feed forward adaptive*). Nos preditores pós-alimentados, cada amostra é quantizada com um passo de quantização resultante das N amostras anteriores. Após o recebimento de um bloco de N amostras, o passo de quantização é calculado em função dos seus valores e é utilizado na quantização das próximas N amostras e assim por diante. Ou seja, nesse tipo de preditor o passo de quantização é calculado a partir de um bloco de amostras, mas só é aplicado no próximo

bloco. Por sua vez, nos preditores pré-alimentados o bloco de N amostras é recebido, os cálculos para determinação do passo de quantização adequado são realizados e estas próprias amostras são quantizadas através dos parâmetros calculados. Ou seja, o passo de quantização é calculado sobre um bloco de amostras e aplicado a este mesmo bloco de amostras.

Os preditores pré-alimentados necessitam acumular o bloco de N amostras na memória a fim de realizar as computações necessárias implicando um atraso associado à acumulação das mesmas. Outra desvantagem é que informação em relação ao passo de quantização deve ser enviada juntamente com as amostras (geralmente uma vez a cada bloco).

No entanto, esse esquema de predição possui a vantagem de seus passos de quantização não serem afetados pelo ruído de quantização, já que eles são calculados a partir de amostras não-quantizadas e passados explicitamente ao decodificador. Por outro lado, os preditores pós-alimentados possuem a vantagem de serem instantâneos, porém, as estimativas dos passos de quantização sofrem influência do ruído de quantização já que o decodificador precisa computá-lo a partir de amostras já quantizadas.

G.727 – E-ADPCM

Padronizado pelo ITU-T em 1990, o G.727 utiliza um codificador de forma de onda E-ADPCM (*Embedded ADPCM*) capaz de operar a 40, 32, 24 e 16 kbps fazendo uso de 5, 4, 3 ou 2 bits por amostra respectivamente [120].

O sistema foi desenvolvido para converter um sinal PCM 64 kbps em um sinal aninhado de taxa variável e vice-versa. Os algoritmos aninhados (*Embedded Algorithms*) são algoritmos de taxa de bit variável com a capacidade de descartar bits externamente aos blocos do codificador e decodificador em qualquer ponto da rede sem a necessidade de uma coordenação entre o transmissor e o receptor. Eles consistem em uma série de algoritmos nos quais os níveis de decisão dos codificadores de taxas baixas são subconjuntos dos níveis de decisão dos codificadores de taxas mais altas.

Os algoritmos aninhados representam uma amostra de um sinal através de uma dupla (x, y), na qual x representa o comprimento total da palavra-código e y representa o seu núcleo. Por exemplo, se um quantizador é definido por (5,2), significa que ele possui dois bits em seu núcleo e um comprimento total da palavra-código de cinco bits. Os três bits de diferença servem para realçar o valor do núcleo e podem ser descartados em momentos de necessidade. O G.727 é uma extensão do G.726 e é recomendado para o uso no PVPs

(*Packetized Voice Protocol*) definido na recomendação G.764. O PVP possui a capacidade de alterar o tamanho dos pacotes de voz quando necessário.

Utilizando a propriedade de aninhamento do algoritmo, os bits menos significativos de cada palavra-código pode ser descartada em qualquer ponto da rede atingindo em momentos de congestionamento um melhor desempenho que o dos algoritmos que descartam pacotes inteiros de voz.

G.728 – LD-CELP

Definido pela ITU-T em 1992, o padrão G.728 estabelece os aspectos de codificação de voz a 16 kbps utilizando um codificador híbrido de baixo atraso com predição linear excitada a código LD-CELP (*Low-Delay Code-Excited Linear Prediction*) [121]. A essência dos algoritmos de procura em dicionários de código (*codebook searching*) CELP é mantida no LD-CELP, no entanto, este último utiliza uma abordagem adaptativa no cálculo do ganho e dos coeficientes do preditor.

O CELP é um codificador otimizado para voz [39] no qual, uma coleção de C possíveis sequências de comprimento L é armazenada nos dicionários do codificador e decodificador. Este codificador trabalha com um bloco (vetor) de cinco amostras, cada uma com um atraso de 0,125 ms totalizando um atraso total no algoritmo de apenas 0,625 ms.

Após os sinais de entrada serem convertidos de PCM μ-*Law* ou *A-Law* para PCM uniforme, segmentados em vetores de cinco amostras e passados através do filtro de síntese e da unidade de escalonamento, o G.728 os compara com cada um dos 1.024 vetores de seu dicionário. O sistema então identifica o candidato que minimiza o erro médio quadrático em relação ao sinal de entrada e transmite o seu respectivo índice de 10 bits ao decodificador (daí a taxa de operação desse sistema, de um total de 8.000 amostras coletadas por segundo são transmitidos 10 bits a cada cinco amostras, totalizando 16 kbps).

O melhor vetor-código é então passado pela unidade de escalonamento e pelo filtro de síntese para estabelecer os novos parâmetros a serem utilizados no próximo vetor-sinal de entrada. Os ganhos dos filtros são atualizados a cada vetor, mas os seus coeficientes são atualizados a cada quatro vetores transmitidos, ou seja, 20 amostras PCM ou 2,5 ms. Os parâmetros do preditor são atualizados a partir das amostras de voz que já foram quantizadas anteriormente.

A análise de desempenho do LD-CELP foi publicada em 1995 no apêndice II do G.728. Segundo esse documento, em uma transmissão livre de erros a qualidade do LD-CELP 16 kbps é inferior à alcançada com o PCM 64 kbps, mas equivalente o ADPCM 32 kbps. Em situações com taxa de erro por bit de $1 \cdot 10^{-3}$ o desempenho do LD-CELP 16 kbps é equivalente ao do ADPCM 32 kbps com taxa de erro de $1 \cdot 10^{-2}$.

G.729 – CS-ACELP

Padronizado em 1996 pelo ITU-T, o G.729 é um codificador paramétrico ou *vocoder*. O padrão descreve uma técnica para codificação de voz a 8 kbps utilizando predição linear de estrutura conjugada excitada por código algébrico CS-ACELP (*Conjugate-Structure Algebraic-Code-Excited Linear Prediction*) [122]. O sistema foi concebido para codificar voz com qualidade total a 8 kbps para uso em comunicações sem-fio e circuitos cabeados transoceânicos [20].

No G.729 o sinal de entrada é filtrado em banda telefônica (4 kHz), amostrado a 8.000 amostras por segundo (PCM 64 kbps) e convertido para PCM linear 16 kbps. Baseado no codificador CELP, o CS-ACELP opera com quadros de voz com 10 ms de duração, ou seja, um bloco 80 amostras. A cada 10 ms, o sistema extrai os parâmetros do modelo CELP (coeficientes do filtro preditor linear e índices e ganhos dos dicionários fixos e adaptativos), codificando-os e transmitindo-os.

Os coeficientes gerados por seus filtros são calculados através do método de autocorrelação por uma janela deslizante de 240 amostras de comprimento que é deslocada a cada 80 amostras (10 ms). Essa janela é composta por duas partes: a primeira (120 amostras) é formada pela metade de uma janela de Hamming e a segunda (120 amostras) por um quarto de ciclo da função cosseno. A janela possui um comprimento total de 240 amostras, divididas em 120 amostras dos quadros passados, 80 amostras do quadro atual e 40 amostras do próximo quadro em um esquema de *look-ahead* (previsão das próximas amostras), resultando num tempo de atraso total de 15 ms para o algoritmo.

Ainda em 1996, o ITU-T publicou os anexos A e B para o G.729. O primeiro reduziu a complexidade computacional e os requisitos de memória do CS-ACELP através de simplificações no modo de operação dos filtros e da forma com a qual a busca é realizada no dicionário de vetores, mantendo no entanto a interoperabilidade com o sistema original. O segundo descreve o detetor de voz ativa (VAD – *Voice Activity Detector*) e o gerador de

ruído de conforto (CNG – *Comfort Noise Generator*), utilizados na supressão e compactação de silêncio no G.729 e G.729A.

G.723.1 – ACELP/MP-MLQ

Padronizado pelo ITU-T em 1996, o G.723.1 é um codificador paramétrico com taxa dual que utiliza codificação ACELP (*Algebraic-Code-Excited Linear-Predition*) para operação em 5,3 kbps e MP-MLQ (*Multipulse Maximum Likelihood Quantization*) para operação em 6,3 kbps [123]. Definido como um padrão de codificação de voz para comunicação de multimídia com baixas taxas de transmissão (*Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*), o G.723.1 especifica um esquema de compressão de voz para meios de velocidade muito baixa como parte integrante da família de padrões H.324.

O sistema foi desenvolvido principalmente para aplicações de videotelefonia. A transmissão em 6,3 kbps garante alta qualidade e fidelidade para voz. A transmissão em 5,3 kbps provê uma boa qualidade de voz e uma maior flexibilidade na utilização do G.723.1. É possível alternar entre as duas taxas de transmissão em qualquer intervalo de 30 ms ou operar em um modo de taxa variável com transmissão descontínua (DTX – *Discontinuous Transmission*) e preenchimento de silêncio com ruído de conforto em intervalos com ausência de fala.

O G.723.1 opera com quadros de 30 ms e um esquema de *look-ahead* de 7,5 ms, resultando em um atraso total do algoritmo de 37,5 ms. O codificador baseia-se em minimização do sinal de erro perceptualmente ponderado. Através de um filtro passa-altas a componente DC é retirada de cada bloco de 240 amostras (30 ms) que é então subdividido em dua metades para a estimação do *pitch* e então em 4 sub-blocos de 60 amostras cada. Para cada sub-bloco, um filtro codificador preditor linear (LPC – *Linear Predictive Coding*) de ordem 10 é computado e um filtro perceptual ponderado de curto prazo é estimado através de uma janela deslizante de 180 amostras.

Dependendo da taxa de transmissão, os vetores que melhor aproximam o sinal são identificados através das técnicas ACELP (5,3 kbps) ou MP-MLQ (6,3 kbps). O sistema MP-MPLQ representa a sequência de excitação através de pulsos distribuídos em intervalos não-uniformes. O número de pulsos necessários para obter uma boa qualidade de voz varia de quatro a seis a cada intervalo de 5 ms. Para cada pulso devem ser

transmitidas as informações de amplitude e localização, sendo necessária a transmissão de 7 ou 8 bits por pulso [124].

O sistema ACELP realiza uma partição do espaço de excitações em um dicionário algébrico (*Algebraic Codebook*) e preenche os vetores do dicionário através de uma estrututura de multipulsos com alguns poucos impulsos de amplitude unitária, aumentando a velocidade da busca de vetores no dicionário [7]. Os quatro conjuntos de parâmetros (ganhos, coeficientes dos filtros e índices dos dicionários de cada um dos quatro sub-blocos) são então agrupados, codificados e transmitidos.

iLBC

Definido em 2004 pelo grupo GIPS (*Global IP Sound*) [100] o iLBC (*Internet Low Bitrate Codec*) é um codificador paramétrico gratuito projetado para transmissão robusta de voz sobre IP [125]. O iLBC opera com taxas de 13,3 kbps (399 bits encapsulados em 50 bytes) para quadros de 30 ms e 15,2 kbps (303 bits encapsulados em 38 bytes) para quadros de 20 ms. Este codificador apresenta qualidade de voz e complexidade na implementação similares ao anexo A do G.729, porém é mais robusto a perdas de pacotes, como mostrado na figura A.15.

O iLBC apresenta uma resposta controlada à perda de pacotes similar a técnica de cancelamento de perda de pacotes (PLC – *Packet Loss Concealment*) utilizada pelo G.711. No entanto, o G.711 utiliza PCM 64 kbps, enquanto o iLBC é um codificador de banda estreita oferecendo uma relação custo-benefício próxima ao estado da arte [125].

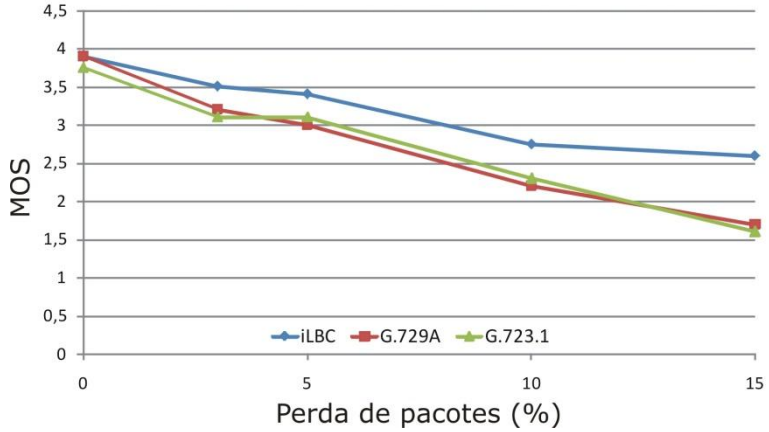


Figura A.15 – *Desempenho dos codecs iLBC, G.729A e G.723.1 [100].*

Este *codec* utiliza codificação linear preditiva de bloco independente. Após os coeficientes do filtro serem computados para um bloco de 20 ou 30 ms, o sinal de voz é filtrado e um sinal residual é obtido. A parte dominante em termos de energia deste sinal é

quantizada e utilizada como estado inicial para a construção de dicionários dinâmicos que serão utilizados para codificar os trechos restantes do sinal residual. Através deste método, se obtém independência entre os blocos, evitando a propagação da degradação devido à perda de pacotes e facilitando a operação do PLC com alta qualidade [125].

Acredita-se que o Skype utilize o codec iLBC ou alguma variante do mesmo [44].

GSM-FR – RPE-LTP

Padronizado pelo ETSI em 1988, o RPE-LTP (*Regular Pulse Excitation – Long Term Prediction*) é o vocoder utilizado no sistema de telefonia móvel europeu GSM-FR (*Global System for Mobile Telecommunications – Full Rate*) [126]. O GSM-FR realiza a amostragem da voz a uma taxa de 8 kHz com 13 bits por amostra e opera com quadros de 20 ms (160 amostras) segmentados em subquadros de 5 ms (40 amostras).

Escolhido após inúmeros testes comparativos com outros codificadores, o RPE-LTP opera a uma taxa de 13 kbps e combina as vantagens dos seus predecessores RELP (*Residual Excited Linear Prediction*) e MPE-LTP (*Multi-Pulse Excited Long-Term Prediction*). Proposto pela França, o RELP apresenta uma baixa complexidade e uma boa qualidade de voz, no entanto, seu desempenho é comprometido pelo ruído introduzido pelo processo de regeneração de altas freqüências e pelos bits perdidos durante a transmissão [110]. Por outro lado, o MPE-LTP proposto pela Alemanha, não é muito afetado pelas perdas no canal e apresenta uma excelente qualidade de voz, no entanto, possui uma alta complexidade [110]. Incorporando características do MPE-LTP no RELP a taxa de transmissão foi reduzida de 14,77 para 13 kbps sem nenhuma perda na qualidade da voz.

Após um estágio de pré-processamento, os quadros de 20 ms são submetidos a um estágio de análise de predição linear no qual são computados oito LARs (*Logarithmic Area Ratios*). Os oito LARs são codificados com números diferentes de bits (6-6-5-5-4-4-3-3) pois possuem diferentes faixas dinâmicas e funções distribuição de probabilidades. Os LARs são decodificados pelo filtro LPC inverso, no intuito de minimizar o sinal de erro.

Através da minimização do resíduo do filtro LTP, o sistema encontra o período de *pitch* e o fator de ganho e os transmite a uma taxa de 3,6 kbps. Esse resíduo é ponderado e decomposto por sub-amostragem em três possíveis sequências de excitação. A sequência de maior energia é selecionada e os pulsos nela contidos são normalizados pela maior amplitude, quantizados por um bloco PCM adaptativo e transmitidos a uma taxa de 9,6 kbps [110].

G.722.2 – AMR-WB

Definido pelo ITU-T em 2003, o padrão G.722.2 descreve um codificador de banda larga adaptativo de múltiplas taxas AMR-WB (*Adaptive Multi-Rate Wideband*) [127]. Este codificador é utilizado nos sistemas de telefonia celular de 3ª geração UMTS (*Universal Mobile Telecommunications System*) e apresenta taxas de transmissão de 6,6; 8,85; 12,65; 14,25; 15,85; 18,25; 19,85; 23,05 ou 23,85 kbps, podendo chavear entre as mesmas em qualquer intervalo de 20 ms (320 amostras). Esta capacidade permite que o codificador se ajuste dinamicamente à diminuição da banda disponível ou ao aumento da taxa de perda de pacotes, adicionando bits extras para a correção de erros quando as condições do canal se tornam adversas.

O AMR-WB foi desenvolvido para operar com sinais de voz de banda larga amostrados a 16 kHz com 14 bits de resolução. O esquema de codificação utilizado é baseado no ACELP, com a adição de melhorias como:

CNG – *Comfort Noise Generation* (Geração de Ruído de Conforto): minimiza alguns efeitos desconfortáveis causados pelo silêncio durante a conversação;

DTX – *Discontinuous Transmission* (Transmissão Descontínua): controla a transmissão de forma que os amplificadores e a bateria não sejam desnecessariamente usados durante períodos de silêncio em que não há voz para ser transmitida;

VAD – *Voice Activity Detector* (Detetor de Atividade de Voz): determina se cada quadro de 20ms contém ou não sinais que devam ser transmitidos.

Speex

O *Speex* é um *codec* de código aberto, grátis e livre de patentes. Ao contrário de muitos outros codificadores de voz, o *Speex* não foi desenvolvido para telefonia móvel, e sim para redes de pacotes e aplicações de telefonia IP [128].

O *Speex* foi desenvolvido para ser flexível e oferecer suporte a largas faixas de qualidade de voz e largura de banda. Para transmitir voz de alta qualidade o *Speex* codifica voz em banda larga (16.000 amostras por segundo).

Devido a sua concepção ser voltada para aplicações em VoIP, esse *codec* é robusto à perda de pacotes, mas não a corrupção dos mesmos. Isso é baseado na suposição que em VoIP ou os pacotes chegam inalterados ou simplesmente não chegam. O *Speex* possui uma complexidade modesta (ajustável) e um baixo consumo de memória.

Todos esses fatores levaram à escolha do CELP como técnica de codificação nesse sistema. Uma das principais razões dessa escolha é que o CELP provou operar confiável e escalarmente tanto em baixas taxas de transmissão (4,8 kbps) quanto em altas taxas de transmissão (16 kbps) [128].

O Speex utiliza técnicas de VAD, DTX e CNG e cancelamento de eco para melhorar o consumo de banda e garantir um maior conforto ao usuário. O atraso imposto por esse algoritmo é de 30 ms para codificação em banda estreita e de 34 ms para codificação em banda larga. Além disso, o Speex implementa um esquema de *buffer* dinâmico de correção de *jitter*. Suas taxas de transmissão variam de 2,15 a 44 kbps [128].

iSAC

O iSAC (*Internet Speech Audio Codec*) é um *codec* adaptativo especialmente desenvolvido para ser capaz de realizar comunicações de som em banda larga por meio de conexões com altas ou baixas taxas de transmissão. Mesmo em conexões *dial-up*, o iSAC consegue trasmitir som com qualidade superior à da rede PSTN, ajustando as taxas de transmissão de modo a oferecer ao ouvinte a melhor experiência possível naquela velocidade [129].

O *codec* ajusta automaticamente a taxa de transmissão desde 10 kbps até 32 kbps com pacotes de 30 a 60 ms de comprimento. Sua complexidade (6-10 MIPS) é comparável e sua qualidade é superior a do codificador G.722. Tal flexibilidade torna o iSAC adequado para aplicações em VoIP, multimídia de tempo real, conferências, tele-aulas e jogos cooperativos através de redes IP [129].

O iSAC não é um codificador de voz e sim um codificador de áudio. Essa característica permite que sejam manipulados sinais não-conversacionais de áudio, como música ou ruído de fundo, excepcionalmente bem. Esse algoritmo opera bem em cenários com altas taxas de perdas de pacotes e *jitter*. Seu atraso é de 3 ms somado ao comprimento do pacote. Esse *codec* é disponível em versões de baixa complexidade, o que permite o uso em conferências e dispositivos como telefones celulares e PDAs.

O iSAC é indicado para aplicações e dispositivos que operem em tempo real e que possuam restrições de banda disponível ou necessidade de alta qualidade de som.

EG711

O EG711 (*Enhanced G.711*) é uma versão modificada do codificador G.711 para uso em redes IP. Esse codificador apresenta uma excelente robustez à perda de pacotes e um alto desempenho, mesmo em redes altamente congestionadas [130]. Caso nenhum pacote seja perdido, esse *codec* opera identicamente ao G.711 de forma transparente, codificando o sinal em PCM *A-Law* ou *μ-Law*. No entanto, se ocorre perda de pacotes, o EG711 oferece uma dramática melhora na qualidade em comparação ao G.711, o que é realizado através de recodificação dos pacotes PCM.

O EG711 é capaz de transmitir voz com qualidade similar a da PSTN mesmo quando opera com taxas de perda de até 10%, apresentando apenas pequenas degradações quando essa taxa chega a 30%. Isso é alcançado através de um eficiente esquema de cancelamento de perda de pacotes integrado (PLC), que permite atingir níveis de qualidade superiores aos dos outros algoritmos [130]. Esse codificador possui ainda deteção de atividade de voz (VAD), o que reduz a taxa de transmissão à metade para silêncio e sinais de baixa intensidade [130].

O EG711 opera com pacotes de 10, 20, 30 e 40 ms com taxas de transmissão variáveis, em média, similares ou ligeiramente menores que o G.711. A voz é codificada em banda estreita (8.000 amostras por segundo) por um algoritmo que consome 4,8 MIPS, com um atraso de processamento igual ao tamanho do pacote utilizado [130].

iPCM

O iPCM (*Internet Pulse Code Modulation*) é um *codec* de banda larga que oferece uma qualidade de som nas comunicações de telefonia fim-a-fim significantemente superior que a da PSTN. Esse codificador mantém uma alta qualidade de áudio mesmo em redes fortemente congestionadas [131].

O iPCM possui um baixo atraso em comparação às soluções alternativas e é extremamente tolerante à perda de pacotes – uma característica crítica em aplicações fim-a-fim. Esse codificador opera com pacotes de 10, 20, 30 e 40 ms a uma taxa de transmissão de 80 kbps. Sua taxa de amostragem é de 16 amostras por segundo, proporcionando áudio em banda larga e com alta qualidade. Um algoritmo de cancelamento de perda de pacotes (PLC) garante robustez à perda de pacotes e o detector de atividade de voz (VAD) permite reduzir a taxa de transmissão a aproximadamente 36 kbps para silêncio e níveis baixos do sinal [131].

A.4.5. Avaliação dos principais codificadores de voz

Descrito com maiores detalhes no Apêndice B desta dissertação, o MOS (*Mean Opinion Score*) é um indicador de qualidade baseado numa pontuação média dada por um grupo de ouvintes a um *codec*, numa escala de 1 (péssimo) a 5 (ótimo). A tabela A.1 apresenta a pontuação MOS dos *codecs* aqui apresentados [20, 100, 110].

Tabela A.1 – *Índice MOS dos principais codecs utilizados em VoIP.*

Referência	Codificador	Taxa de bits (kbps)	Tamanho do quadro (ms)	MOS
G.711	PCM	64	0,125	4,2
G.722	SB-ADPCM	56	0,125	3,9
G.726	ADPCM	32	0,125	4,0
G.727	E-ADPCM	32	0,125	4,1
G.728	LD-CELP	16	0,625	3,6
G.729	CS-ACELP	8	10	3,7
G.723.1	ACELP/MP-MLQ	6,3	10	3,9
iLBC	iLBC	15,2	20	3,9
GSM-FR	RPE-LTP	13	20	3,6
G.722.2	AMR-WB	16	20	3,9
Speex	Baseado em CELP	12,8	30	3,8
iSAC	iSAC	32	30	4,0
EG711	EG711	64	20	4,2
iPCM	iPCM	80	20	4,1

Apêndice B – Qualidade de voz em VoIP

Telefonia IP é uma das mais importantes aplicações de rede em que a percepção do usuário desempenha um papel fundamental. O sucesso de uma aplicação de VoIP, depende fortemente das opiniões dos usuários quanto à qualidade da voz recebida e do resultado da inevitável comparação entre os níveis de qualidade de uma chamada VoIP e uma chamada via PSTN.

A rede, como qualquer canal de comunicação real, inerentemente introduz erros corruptivos nos pacotes de voz que nela trafegam. Esses erros possuem um comportamento aleatório e podem ser modelados a partir de ferramentas de estatística. Vários estudos documentam os efeitos introduzidos pela rede na qualidade da voz recebida e determinam faixas nas quais esses efeitos são considerados: (1) desprezíveis, (2) perceptíveis mas toleráveis, ou (3) intoleráveis [7].

Os efeitos considerados desprezíveis, são fáceis de serem contornados ou mesmo ignorados. Os efeitos perceptíveis, mas toleráveis, causam algum incômodo aos usuário, mas não comprometem a comunicação, a menos que perdurem por longos períodos de tempo. Os efeitos classificados como intoleráveis degradam a voz e incomodam o usuário de tal forma que, em alguns casos, a comunicação torna-se de qualidade inaceitável ou mesmo impossível de ser estabelecida.

Estimativas das fronteiras entre essas faixas encontram-se disponíveis na literatura, mas maiores investigações sobre os impactos de efeitos combinados ainda são necessárias [7].

Voz transmitida em pacotes IP pertence à categoria de tráfego de tempo real, e como tal, possui restrições em relação a erros, atrasos e perda de pacotes devido à tolerância limitada

do ouvido humano a tais perturbações. Essas restrições baseiam-se nos efeitos psicológicos causados pelas perturbações introduzidas pela rede e serão descritas a seguir.

B.1. Qualidade percebida pelo usuário

A qualidade da voz percebida pelo usuário (UPQ – *User Perceived Quality*) é um importante critério para a determinação da eficiência de um sistema de telefonia IP. Devido à subjetividade intrínseca à essa percepção, inúmeras figuras de mérito distintas foram criadas para a determinação da mesma. Serão expostas a seguir as principais medidas existentes para este fim, de acordo com o relatório técnico de Beuran e Ivanivici [132], o qual realiza uma estimativa da qualidade de voz percebida pelo usuário em aplicações VoIP.

B.1.1. MOS

Em 1996, o ITU-T definiu a metologia para determinar o quão satisfatória a qualidade de uma chamada telefônica pode ser. O método MOS (*Mean Opinion Score*) é subjetivo, baseado em experimentos humanos, onde usuários pontuam a qualidade de uma chamada numa escala de 1 a 5, com o seguinte critério de conforto [71]:

- 1 – Péssima: ininteligível, apesar de qualquer esforço empregado;
- 2 – Ruim: inteligível, mas muito esforço é necessário;
- 3 – Regular: inteligível, mas algum esforço é necessário;
- 4 – Boa: inteligível com alguma atenção, mas sem muito esforço;
- 5 – Ótima: inteligível com relaxamento completo, nenhum esforço é necessário.

A recomendação P.800 define o MOS como a pontuação obtida pela chamada telefônica no processo de classificação por categoria absoluta (ACR). Duas outras categorias são definidas na recomendação P.800: A classificação por categoria de degradação (DCR), cujo resultado é a pontuação de opinião média de degradação (DMOS) e a classificação por categoria de comparação (CCR), cujo resultado é a pontuação de opinião média de comparação (CMOS).

Na classificação DCR, a idéia é que os métodos sejam aplicados com as mais variadas formas de degradação (perdas, interferências, erros de transmissão, ruído ambiente, eco, distorção, etc.), enquanto na classificação CCR a pontuação é dada realizando comparações entre diversos sinais de voz.

Na classificação ACR, cujo resultado é o índice MOS, os ouvintes-avaliadores escutam os sinais a serem avaliados e os pontuam sem compará-los com nenhum sinal de referência, baseando-se no esforço exercido para a compreensão dos mesmos.

O procedimentos de avaliação devem seguir normas rígidas tais como: número suficiente de avaliadores; imparcialidade e desconhecimento prévio do conteúdo das amostras por parte dos avaliadores; diversidade das amostras em relação a sexo, idade e sotaque dos falantes; realização dos testes em diversos países de línguas distintas por autoridades competentes e rígido controle das condições do experimento (volume físico da sala, isolamento de ruídos externos, condições dos equipamentos utilizados e outros) [20, 133]. A tabela B.1 mostra comparativamente a qualidade de uma chamada e o seu respectivo índice MOS.

Tabela B.1 – *Níveis de qualidade relacionados aos índices MOS.*

Qualidade	MOS	Descrição
Alta	4 a 5	Similar ou melhor que a experiência do uso de uma chamada ISDN.
Telefônica	3,5 a 4	Similiar a obtida com com o uso do codificador G.726
Boa	3,0 a 3,5	Boa, mas com degradação facilmente audível
Militar	2,5 a 3	Comunicação ainda possível mas exige atenção

B.1.2. PSQM

Em 1998, o ITU-T padronizou um método objetivo para a medição da qualidade de voz dos codecs telefônicos chamado PSQM (*Perceptual Speech Quality Measure*) [134]. Após inúmeros testes, o ITU-T concluiu que o índice PSQM é bem-correlacionado com os resultados subjetivos para os codecs de voz. O PSQM pode ser utilizado para codecs de taxas superiores a 4 kbps, porém há informações insuficientes a respeito de seu desempenho na presença de fatores como erros de canal e atrasos, por exemplo.

O PSQM utiliza representações psicofísicas que seguem as sensações humanas relativas à uma conversação tão fielmente quanto possível. Um valor zero de PSQM indica que nenhum problema está sendo observado na comunicação, enquanto um valor de 6,5 indica um canal completamente não-utilizável. Os valores do PSQM podem ser mapeados na escala MOS, obtendo assim um valor estimado para a qualidade subjetiva da conversação.

B.1.3. O modelo-E (E-Model)

O Modelo-E surgiu em 2000 como um modelo computacional para planejar transmissões, buscando atingir altos níveis de qualidade e desempenho [79]. Desenvolvido pela ITU-T e adotado pelo ETSI (*European Telecommunications Stardards Institute*), o

modelo integra uma série de fatores que degradam a qualidade de uma comunicação tais como atrasos e o uso de *codecs* de banda estreita.

A degradação é computada a partir dos fatores de perda de entrada e valores aceitáveis para o estabelecimento da comunicação são estimados. Esses valores podem ser utilizados quando um nível de degradação similar for encontrado.

O modelo-E associa um valor numérico denominado fator de perda a cada elemento de degradação. Esses fatores de perda são levados em consideração no cálculo do fator de avaliação (Fator-R), que graduado numa escala de 0 a 100 pode ser mapeado no índice MOS.

O fator-R é obtido através da seguinte expressão [79, 135]:

$$R = R_0 - I_s - I_d - I_e + A, \tag{27}$$

na qual:

- R_0 representa os efeitos da relação sinal-ruído (SNR);
- I_s representa as perdas simultâneas do sinal de voz;
- I_d representa as perdas associadas ao atraso fim-a-fim;
- I_e representa as perdas associadas ao equipamento utilizado e
- A representa o fator de vantagem ou fator de expectativa.

O termo R_0 é determinado pelo ruído gerado pelo circuito e pelo ruído ambiente nos lados do transmissor e receptor e possui um valor padrão de 94,77 [79]. O termo I_s possui um valor padrão de 1,41 e é associado às perdas causadas pelo ruído de quantização, pela interferência da voz do falante em seu fone de ouvido e à queda de qualidade por uma conexão de volume demasiadamente alto [135]. O termo I_d é composto pelas perdas geradas pelo eco no lado do transmissor, pelo eco no lado do receptor e pelo atraso absoluto da voz, se maior que 100 ms. O termo I_e é um meio flexível para computar as perdas devido ao uso de *codecs* de baixa taxa de transmissão. Seus valores são tabelados e obtidos a partir de exaustivos experimentos MOS realizados sob diversas taxa de perda de pacotes. O fator de vantagem A é empregado para definir o grau de tolerância do usuário a uma determinada tecnologia. Este fator possui valor zero para telefonia fixa, cinco para telefonia celular *indoor*, e 20 para enlaces de satélite em localidades de difícil acesso [135]. Para VoIP, o termo A é considerado com valor zero.

De uma maneira geral, a expressão que determina o fator-R pode ser reduzida a:

$$R = 93{,}36 - I_d(T_a) - I_e(codec, perda), \tag{28}$$

na qual $I_d(T_a)$ são as perdas em função do atraso fim-a-fim e $I_e(codec, perda)$ é uma função do *codec* utilizado e da taxa de perda de pacotes.

Os valores MOS na escala de 1 a 4,5 são ilustrados na figura B.1 e foram obtidos a partir do Fator-R, usando as seguintes expressões:

$$\begin{cases} R < 0: MOS = 1 \\ 0 \leq R \leq 100: MOS = 1 + 0{,}035R + R(R-60)(100-R)\,7 \cdot 10^{-6} \\ R > 100: MOS = 4{,}5 \end{cases} \quad (29)$$

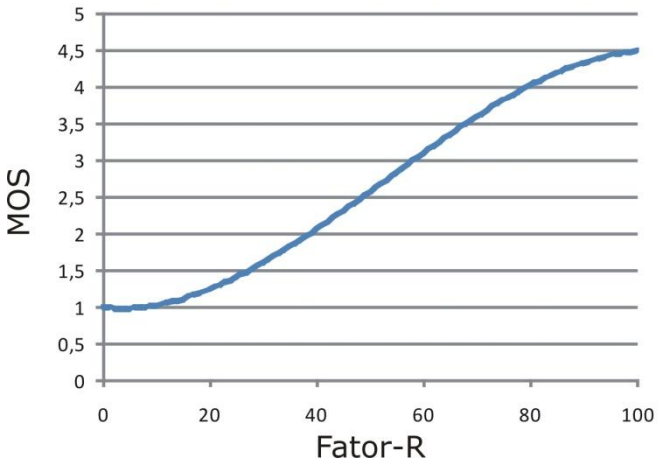


Figura B.1 – *Índice MOS em função do fator-R.*

Observa-se que o máximo da classificação MOS obtido é 4,5. Isso reflete a média da pontuação dada pelos usuários completamente satisfeitos com a qualidade da ligação (4 ou 5). A tabela B.2 mostra a relação entre a satisfação dos usuários e o Fator-R.

Tabela B.2 – *Relação entre o fator-R e o índice MOS.*

Fator-R (mínimo)	MOS (mínimo)	Satisfação dos usuários
90	4,34	Muito satisfeitos
80	4,03	Satisfeitos
70	3,60	Alguns usuários insatisfeitos
60	3,10	Muitos usuários insatisfeitos
50	2,58	Praticamente todos os usuários insatisfeitos

B.1.4. PAMS

O PAMS (*Perceptual Analysis/Measurement System*) é um método desenvolvido pela *British Telecom* (consequentemente sujeito a questões de propriedade e patente) para determinar a qualidade da voz de um sistema de transmissão através de qualquer rede, incluindo aquelas sujeitas a perdas de pacote e atrasos [136]. O processo de computação do PAMS utiliza um modelo que combina uma descrição matemática das propriedades psicofísicas da audição com uma técnica de análise que considera o erro e a subjetividade no sinal recebido no ponto de vista da percepção humana.

O PAMS compara o sinal original e degradado e determina, numa escala de 1 a 5, uma predição do MOS para a qualidade do áudio e esforço do ouvinte. O índice PAMS é típicamente metade do índice MOS obtido num teste subjetivo controlado em laboratório. Extensivos testes subjetivos em humanos, inclusive com o uso *codecs* de banda estreita foram realizados para a validação do PAMS.

B.1.5. PESQ

Com o objetivo de predizer a qualidade subjetiva dos *codecs* de voz e dos sistemas de telefonia de banda estreita a ITU-T, em fevereiro de 2001, definiu o índice PESQ (*Perceptual Evaluation of Speech Quality*) [137]. O PESQ combina o melhor do PSQM e PAMS, levando ainda em consideração filtragens, atrasos variáveis, distorções de codificação e erros de canal. Seu processo-chave é uma transformação dos sinais original e degradado em uma representação de áudio análoga àquela do sistema auditivo humano.

Os sinais de entrada (sinal de referência) e saída (sinal degradado) são alinhados, normalizados e convertidos através de um modelo interno, no qual as distorções mais significativas ao ouvido humano são ponderadas com pesos maiores que aquelas quase imperceptíveis. Em seguida, os sinais são comparados e uma pontuação é gerada.

Os índices PESQ são mapeados em índices MOS numa escala de -0,5 a 4,5. No entanto, seus valores geralmente oscilam entre 1 e 4,5, faixa de valores normalmente obtidos nos experimentos subjetivos do MOS. As relações entre o índice PESQ e a qualidade da voz são mostradas na tabela B.3.

Tabela B.3 - *Índice PESQ.*

PESQ	Qualidade	Comentário
$3,0 < PESQ \leq 4,5$	Boa	Qualidade aceitável, sendo 3,8 o índice referência da PSTN.
$2,0 < PESQ \leq 3,0$	Baixa	Algum esforço é necessário para a compreensão.
$PESQ < 2,0$	Inaceitável	A degradação tornou a comunicação impossível.

O PESQ apresenta um bom desempenho quando a clareza da voz é afetada por aspectos como [116]: taxa de transmissão e tipo (forma de onda, paramétrico ou híbrido) do codificador; transcodificações (conversões entre formatos digitais); *jitter*; ruído ambiente no lado do transmissor; nível do sinal de entrada; erros e perdas de pacotes no canal de transmissão.

O PESQ não avalia o impacto de aspectos como atraso, eco ou atenuação do sistema devido aos processos de alinhamento temporal e de normalização de nível realizados.

As desvantagens do PESQ residem no fato do modelamento de como o cérebro humano julga a qualidade de voz ainda não estar completamente definido [133], na desconsideração dos efeitos do atraso e na dificuldade de acesso às duas pontas de um canal de comunicação real [116].

Sendo uma das métricas mais recentes, o PESQ se mostrou eficiente nas tarefas de seleção de *codecs* e testes de rede tanto em simulações quanto em tempo real, superando os seus antecessores.

B.2. Fatores que afetam a qualidade da voz

O sistema telefônico convencional oferece aos seus usuários um alto nível de qualidade de serviço, graças ao uso de canais dedicados. Os atuais usuários de telefonia IP são, em sua maioria, pessoas que migraram do sistema de telefonia comutada e que estão acostumadas com o nível de QoS oferecida pelo mesmo [20]. O tráfego em tempo real de pacotes de voz digital através de uma rede não-confiável se depara com algumas restrições inerentes ao fato da mesma não ter sido inicialmente projetada para este fim específico. Nesta seção serão apresentadas algumas dessas restrições e as medidas que podem ser tomadas no sentido de minimizar os impactos negativos causados pelas mesmas.

B.2.1. Codecs

Os canais de comunicação possuem por natureza uma capacidade limitada para o tráfego de dados [114]. Largura de banda é um recurso limitado e muitas vezes escasso, principalmente quando muitos usuários compartilham um mesmo canal. No intuito de se enviar voz com a melhor qualidade possível, utilizando a menor quantidade de banda, foram criados os *codecs*. Um *codec* (codificador-decodificador) é um dispositivo capaz de codificar um sinal de voz, comprimindo-o para que seja armazenado ou transmitido de forma eficiente. A telefonia convencional utiliza o codificador G.711 [115] que possui MOS de 4,3, porém com uma taxa de transmissão de 64 kbps. Os *codecs* modernos possuem taxa de transmissão de cerca de 8 kbps, porém com MOS que variam de 3,7 a 4,0.

A degradação do sinal de voz causada pelos *codecs* é um efeito colateral da função a que eles se propõem: a compressão dos dados.

Considerado o pai da teoria da informação, Shannon, em 1948, definiu a grandeza chamada *entropia* [138], permitindo pela primeira vez que a engenharia lidasse de maneira quantitativa com o conceito de informação [139]. O conceito de entropia já havia sido

reconhecido por Hartley em 1928 [140], que percebeu a relação que existia entre informação e a distribuição de probabilidades dos possíveis resultados de uma variável aleatória. A entropia proposta por Shannon e por Hartley diferem simplesmente na base dos logaritmos usados em sua definição.

Seja uma variável aleatória X que assume com probabilidade não-nula um conjunto de N valores cujas probabilidades associadas são: $p_1, p_2, ..., p_N$. A entropia $H(X)$ desta variável é definida por:

$$H(X) = \sum_{i=1}^{N} \left(p_i \cdot \log \frac{1}{p_i} \right), \tag{30}$$

e medida em *bits* ou *Hartleys* conforme a base do logaritmo seja 2 ou 10, respectivamente. No caso de VoIP, a variável aleatória X pode representar o conjunto das possíveis palavras-código geradas na saída do codificador de voz, por exemplo.

Do mesmo modo que determina limites para a capacidade de um canal, a teoria da informação permite constatar que também há um limite para a compressão sem perdas de um sinal. Tal limite é igual à entropia do sinal [139] e define o número mínimo de dígitos capaz de representar fielmente o mesmo.

A taxa de bits dos *codecs* de banda estreita disponíveis atualmente variam de 1,2 kbps a 64 kbps. Isso afeta a qualidade da voz reconstruída visto que o processo de compressão ocasionalmente gera perdas de informação que se refletem em uma degradação do sinal de voz. Deve haver por parte do *codec*, um compromisso entre a qualidade e a largura de banda utilizada para transmissão da voz. Devido aos limitantes impostos pela compressão do sinal, o desafio é realizar a comunicação com um nível de qualidade "*just-good-enough*" (apenas bom o suficiente) que seja satisfatório e realizável em uma dada taxa de transmissão [7].

Além disso, o mapeamento de um sinal contínuo em um conjunto finito de valores discretos realizado pelo processo de quantização (que é por natureza um processo de aproximação), também introduz perdas na qualidade no sinal que são proporcionais a intensidade do ruído de quantização e inversamente proporcionais ao número de níveis utilizados pelo quantizador. Esses efeitos causam uma mudança de timbre na voz tornando-a mais artificial e com um aspecto "metálico".

Durante uma ligação telefônica típica, a cada instante apenas um dos participantes fala enquanto o(s) outro(s) ouve(m). Isso significa que em média 50% da banda do canal bidirecional simultâneo (*full-duplex*) é desperdiçada, fazendo com que o mesmo se

comporte efetivamente como um canal bidirecional alternado (*half-duplex*). Somando-se a esse fato as pausas naturais entre sílabas, palavras, sentenças e para a formulação dos trechos do diálogo pelos participantes, o canal chega a ser efetivamente utilizado em apenas 40% do tempo total da ligação [40]. Baseados neste comportamento estatístico do tráfego de conversação e visando minimizar o desperdício de largura de banda, alguns sistemas como o VAD, DTX e CNG foram desenvolvidos para otimizar a operação dos codecs e melhorar a qualidade da voz percebida pelo usuário.

O VAD (*Voice Activity Detector*) inspeciona os quadros de voz digitalizada, ientificando se o mesmo carrega voz ativa ou silêncio. No primeiro caso, um fluxo normal de bits é transmitido. No segundo caso, pode-se transmitir uma descrição compacta do ruído ambiente denominada SID (*Silence Insertion Descriptor*) ou simplesmente não transmitir nada, poupando energia da fonte de alimentação através da transmissão descontínua DTX (*Discontinuous Transmission*) [39].

Porém, o silêncio total durante uma conversação pode causar nos usuários desconforto ou a sensação que a chamada foi interrompida. Assim, no intuito de tornar a ligação mais agradável aos usuários, mecanismos de geração de ruído de conforto CNG (*Comfort Noise Generator*) são utilizados. Durante os períodos de silêncio da comunicação, o CNG recebe um modelo do ruído de fundo do ambiente onde o transmissor se encontra e o reproduz no receptor [20]. Na transição de um quadro de fala para um de silêncio, um quadro SID é transmitido ao receptor, após isso, o DTX monitora a necessidade de transmitir atualizações dos parâmetros do CNG através da inspeção das características do ruído ambiente, realizando-as quando necessário.

A deteção de atividade de voz, embora otimize o uso do canal de comunicação, introduz outra fonte de degradação no sinal de voz: o *clipping*. O *clipping* é um corte nas primeiras sílabas de uma sentença que acontece porque a deteção de voz após um período de silêncio não é instantânea. Melhorias nos algoritmos VAD podem minimizar os efeitos do *clipping*.

B.2.2. Atraso

O atraso **fim-a-fim**, ou **atraso de ida** (*mouth-to-ear delay*) é definido como o intervalo de tempo entre o instante em que a voz é capturada no transmissor e o momento em que é reproduzida no receptor [141], podendo ser descrito pela expressão [7]:

$$D(t) = V + h + d(t) + B, \tag{31}$$

na qual o atraso total no instante t, $D(t)$, é dado em termos de:

V – Atraso devido ao processo de digitalização (amostragem, quantização e codificação) do sinal de voz nos DSPs dos equipamentos do sistema. Seu valor exato depende do hardware e dos *codecs* utilizados;

h – Atraso de empacotamento devido à formação e reprodução dos pacotes. É o tempo necessário para que o sinal de voz seja inserido nos pacotes (RTP,UDP, IP, *ethernet*,...) no transmissor e extraído dos mesmos no receptor. Tais atrasos são geralmente fixos e da ordem de 20 a 30 ms para a formação e 50 ms para a reprodução dos pacotes [20];

d(t) – Atraso introduzido pela rede no instante t. É a fonte de atraso que mais pode comprometer o sistema. É um fator variante no tempo que corresponde à soma dos tempos que o pacote leva para ser encaminhado através dos roteadores e *proxys*, interfaceado através dos *gateways* e verificado pelos *firewalls* da rede.

B – Tempo de espera da aplicação destino devido ao tempo no qual o pacote fica retido no *buffer* do receptor para supressão da variação do atraso (*jitter*).

Os termos V e B se devem à fatores locais como capacidade, complexidade e eficiência dos algoritmos, *codecs* e hardware utilizados. Logo, a soma $D_{hw} = V + B$ pode ser considerada como o atraso introduzido pelos equipamentos do usuário e seu valor exato depende da configuração dos mesmos. Da mesma forma, pode-se considerar a soma $D_{net}(t) = h + d(t)$ como o atraso introduzido unicamente pela rede. O atraso introduzido pela rede é ilustrado na figura B.2.

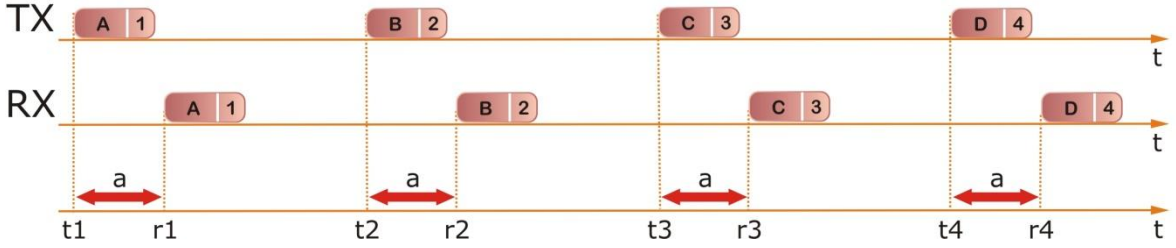


Figura B.2 – *Atraso introduzido pela rede.*

Observa-se da figura B.2 que os pacotes são transmitidos em instantes t_i e são recebidos nos instantes r_i no destino separados por um atraso de "a" unidades de tempo.

Outra forma de medir o atraso é através do **atraso de ida e volta** (*Round-Trip Delay*), que corresponde ao tempo que uma mensagem leva para sair do transmissor atingir o receptor, ser devolvida por este e finalmente recebida de volta pelo transmissor. É importante notar que devido às assimetrias da rede TCP/IP o atraso de ida e volta não corresponde obrigatoriamente ao dobro do atraso fim-a-fim [142].

O atraso em si não altera a qualidade do sinal de voz, mas altera o nível de iteração entre os participantes da conversação. No sistema telefônico convencional, o atraso fim-a-fim tipicamente não excede os 150 ms (exceto no uso de canais via satélite), o que é imperceptível para o ouvido humano [20]. No entanto, existem diversas opiniões a respeito do limite de tolerância do atraso em aplicações VoIP.

De um ponto de vista comercial, considera-se que o atraso fim-a-fim não deve ultrapassar os 200 ms [143], que é o comercialmente aceitável. Quando esse atraso atinge 800 ms, fatores psicológicos adversos impedem a comunicação telefônica. Atrasos na faixa de 200 a 800 ms são condicionalmente aceitáveis se ocorrem em poucos trechos da conversação, com curta duração e suficientemente espaçados entre si. Percebe-se que existe uma larga faixa de valores aceitáveis para o atraso se as perturbações ocorrem com baixa probabilidade e curta duração, no entanto, algumas aplicações específicas podem ser mais restritivas.

Por outro lado, de acordo com análises técnicas do ITU-T, atrasos na faixa entre 0 e 100-150 ms garantem um alto nível de iteração entre os participantes da chamada, atrasos na faixa entre 100-150 ms a 400 ms possibilitam um nível aceitável de iteração entre os usuários e atrasos acima de 400 ms não são aceitáveis para a comunicação em VoIP [72, 144].

Na presença de eco, o atraso tolerável fica restrito a cerca de 25 ms, restrição que seria dificilmente atendida pelos aplicativos de telefonia IP [7]. Por isso sistemas VoIP devem utilizar métodos de cancelamento de eco [5]. Em aplicações de multiconferência que necessitam de uma unidade de controle multiponto (MCU) o atraso máximo aceitável é de 100 ms, pois a mesma deve decodificar cada fluxo de dados, combinar os mesmos em um único fluxo e recodificar o sinal combinado. Todo esse processo acaba duplicando o atraso e reduzindo ainda mais a qualidade percebida pelo usuário [7].

Na prática, tenta-se minimizar o atraso tanto quanto possível. O objetivo dos aplicativos comerciais tipicamente é tentar manter os valores do atraso entre 100 e 200 ms.

O atraso pode ser controlado através do uso de *codecs* com baixa complexidade computacional e de técnicas de roteamento eficientes, nos quais o melhor caminho possível seja utilizado para o encaminhamento dos pacotes.

B.2.3. Jitter

As variações das condições da rede, a perda de pacotes e as diferentes rotas pecorridas por cada pacote fazem com que os mesmos cheguem desordenados e em intervalos irregulares. Porém, os pacotes de voz são gerados e transmitidos em uma dada taxa e necessitam ser reproduzidos nessa mesma taxa no receptor.

A figura B.3 ilustra o efeito do *jitter* e mostra pacotes sendo transmitidos regularmente em instantes de tempo t_i e recebidos em instantes irregulares r_i, após um intervalo de tempo composto por uma componente constante de atraso a e uma componente variável de *jitter* j_i.

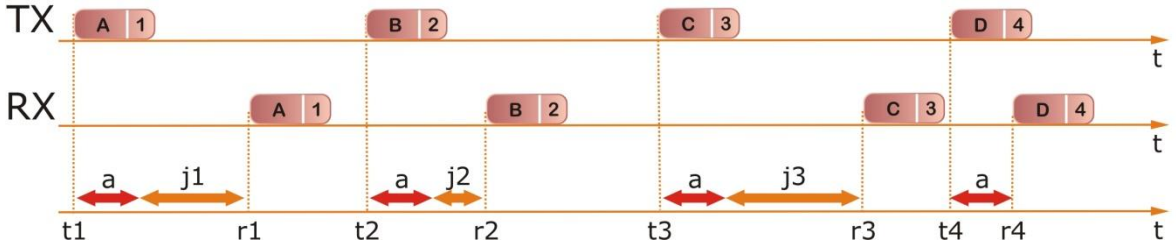


Figura B.3 – *Jitter introduzido pela rede.*

O *jitter* é definido como o momento de primeira ordem (variância) do atraso entre pacotes consecutivos. O *jitter* afeta os pacotes alterando a distribuição dos instantes de chegada no receptor em relação a distribuição dos instantes de envio no transmissor.

Em VoIP, a distribuição dos instantes de envio é determinística, e os pacotes são enviados de forma equiespaçada entre si. Dessa forma, dado que os pacotes deixam o transmissor com uma taxa constante, a variância do tempo de chegada entre pacotes é igual a variância do atraso dos pacotes do transmissor ao receptor [145]. Assim, o *jitter* médio da comunicação pode ser obtido pela expressão:

$$J = \frac{1}{N-1}\sum_{i=2}^{N}|D_i - D_{i-1}|, \tag{32}$$

na qual o *jitter* J, é dado em função do número total de pacotes transmitidos N e dos atrasos dos pacotes D_i, com $i = 0,1,\ldots,N$.

Em um sistema VoIP, para que a distribuição original seja recuperada, há a necessidade de armazenar os pacotes que chegam em um acumulador (*jitter buffer*), reordená-los, e só então entregá-los à aplicação, ao custo de um atraso adicional introduzido na conversação. O tamanho do *buffer* é dimensionado em função do *jitter* estimado da comunicação, dessa

forma, é importante conhecer a função distribuição de probabilidade do atraso, ou pelo menos, sua média e variância (atraso médio e *jitter*, respectivamente).

Os *jitter buffers* podem ser estáticos ou dinâmicos. Nos estáticos, o tamanho do acumulador é fixo enquanto nos dinâmicos o mesmo varia de acordo com as estimativas realizadas pelo sistema através da análise da variação do atraso dos pacotes recebidos em tempo real [5].

B.2.4. Perda de pacotes

As falhas geradas pela perda de pacotes (*glitch*) são as lacunas geradas no fluxo de comunicação devido às mais diversas causas, tais como: flutuação do atraso, ruído e interferência em redes sem-fio, transbordamento do *buffer* (*buffer overflow*), erros de endereçamento e descarte pelos protocolos. Em outras palavras, o *glitch* é o efeito causado pela interrupção no fluxo de pacotes de voz. A perda de pacotes na rede é ilustrada na figura B.4.

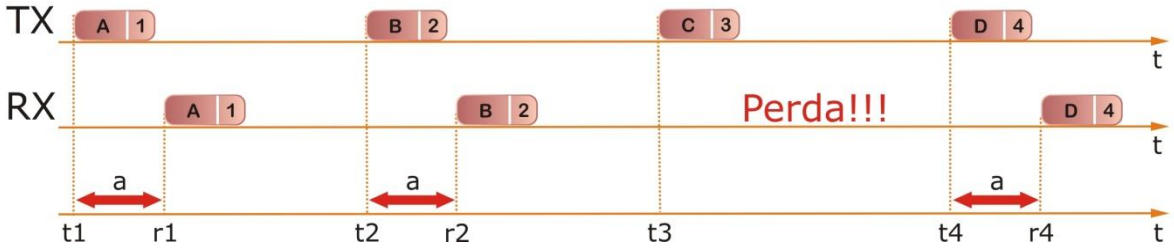


Figura B.4 – *Perda de pacotes na rede.*

Em sistemas de transferência de arquivos são utilizados protocolos ARQ (*Automatic Repeat Request*) para a recuperação de dados perdidos via retransmissão [20]. No entanto, em aplicações de fluxo contínuo de tempo real como VoIP, tal solução torna-se impraticável, pois os tempos de espera pela retransmissão tornariam-se inaceitáveis e provavelmente o pacote chegaria tarde demais para que a aplicação pudesse aproveitá-lo. Além disso, de um modo geral, uma taxa de perdas de 5% é considerada aceitável [40].

Em razão da alta correlação entre janelas curtas de trechos próximos de um mesmo sinal de voz, a taxa de perda de pacotes aceitável é função do comprimento do pacote. A inteligibilidade atinge valores muito baixos (aproximadamente 10%) quando o comprimento do pacote se aproxima de 250 ms, no entanto, utilizando pacotes de aproximadamente 20 ms, a inteligibilidade chega a 80% [7]. Tipicamente os sistemas de telefonia IP utilizam pacotes de 10 a 30 ms de comprimento. O uso de pacotes demasiadamente curtos tornaria o sistema ineficiente enquanto pacotes longos demais

estão sujeitos a atrasos que os impossibilitariam de ser usados em aplicações de tempo real [5].

Para uma mesma taxa de perda, pacotes maiores perdidos comprometem mais severamente a qualidade da conversação que a perda de pacotes menores, devido à alta quantidade de redundância existente entre trechos próximos do sinal de voz. Pelo mesmo motivo, erros em rajadas ou em surtos são mais danosos que aqueles distribuídos de uma forma mais uniforme no tempo. Se os erros são esparsos, pacotes curtos (20 ms) conseguem manter níveis aceitáveis de degradação mesmo com uma taxa de 50% de perda [7].

Em geral, quando um pacote isolado é perdido, o codificador repete o último pacote válido recebido, sem prejuízo para a inteligibilidade da comunicação (caso mais de um pacote consecutivos sejam perdidos repete-se o útimo pacote válido uma única vez, ficando em silêncio até a recepção do próximo pacote) [40].

As principais estratégias utilizadas para minimizar os efeitos das perdas são: o desenvolvimento de esquemas de roteamento mais confiáveis para evitar que os pacotes sejam perdidos; o uso de pacotes de comprimento adequado; o entrelaçamento (técnica na qual as informações são misturadas e divididas entre vários pacotes, descorrelacionando pacotes próximos e os tornando mais robustos aos erros em surtos). Uma forma refinada de se lidar com as perdas é através da FEC (*Forward Error Correction*), técnica que adiciona redundâncias aos pacotes, permitindo que através da informação contida nos pacotes recebidos o receptor consiga interpolar o valor daqueles que foram perdidos.

Apesar de aumentar o consumo de banda, a FEC possibilita que o sistema opere com uma qualidade de voz aceitável, em canais com taxas de perda de 10 a 20% [20].