

UNIVERSIDADE FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Um Novo Modelo para Geração de Tráfego VoIP

Dissertação submetida à Universidade Federal do Paraná
como requisito parcial à obtenção do grau de

Mestre em Engenharia Elétrica

por

Carlos Ignácio de Mattos

Curitiba, 18 de Junho de 2011

Um Novo Modelo para Geração de Tráfego VoIP

Carlos Ignácio de Mattos

Esta tese foi julgada adequada para a obtenção do título de **Mestre em Engenharia** na especialidade **Engenharia Elétrica**, área de concentração **Telecomunicações e Redes de Computadores**, e aprovada em sua forma final pelo curso de Pós-Graduação em Engenharia Elétrica.

Curitiba, 18 de Junho de 2011.

Prof. Dr. Carlos Marcelo Pedroso, orientador

Prof. Dr. Evelio Martín García Fernández
coordenador do curso de Pós-Graduação em Engenharia Elétrica
da Universidade Federal do Paraná.

Banca Examinadora

Prof. Dr. Carlos M. Pedroso, orientador Prof. Dr. Eduardo P. Ribeiro, co-orientador

Prof. Dr. Evelio Martín García Fernández Prof. Dra. Keiko Veronica Ono Fonseca

Dedico esse trabalho a todos que contribuíram, tecnicamente ou pessoalmente, com o mesmo. A todos vocês, um forte abraço.

Agradecimentos

Primeiramente agradeço a Deus, por ter me dado a vida, saúde e perseverança para cumprir com meus objetivos.

Agradeço a minha namorada Deise Fernanda. Pelos beliscões que ela me deu quando estava com preguiça. Pelas cócegas quando estava desanimado. Por escutar por 2 anos eu falando sobre tráfego VoIP. Por fim, por ser uma companheira por todas as horas, por sempre estar disposta a me dar carinho e pelo belo amor que sentimos um pelo outro.

Agradeço a minha família, por estar sempre ao meu lado, apoiando-me. Agradeço à minha mãe, Josmari, por sempre acreditar em mim, pelo amor incondicional e pelo infinito apoio. Agradeço ao meu pai de criação (e de coração), Ronaldo, por me aceitar como filho, pelas conversas de pai e filho que tivemos e pelo incondicional amor. Agradeço ao meu pai, José Carlos, pelo amor incondicional e apoio. Agora agradeço a minha irmãzinha, Clarrisa. Ela me apoiou muito, sempre esteve ao meu lado, demonstrou sempre um grande amor de irmãos e teve paciência em me aturar quando estava mal humorado.

Agradeço ao professor Dr. Carlos Marcelo Pedroso. Ele procurava sempre estar presente para me auxiliar no desenvolvimento da pesquisa. Teve paciência em me explicar alguns conceitos teóricos repetidamente quando não conseguia entendê-los. Por não ter ficado bravo, pelos menos externamente (risos), quando arranjei um emprego e quase atrasei a entrega da dissertação. Em suma, agradeço ao Pedroso pela orientação, mas principalmente agradeço a ele pela amizade.

Agradeço ao professor Dr. Eduardo Parente Ribeiro por co-orientar a pesquisa e fornecer ótimas dicas para o desenvolvimento do modelo de tráfego VoIP. Agradeço ao professor Dr. Evelio Martín García Fernández por me aceitar e conceder a bolsa de estudos no programa de mestrado. Adicionalmente agradeço ao Evelio e ao professor Dr. César Augusto Dartora por terem escrito as cartas de recomendação.

Resumo

Neste trabalho propõe-se um novo modelo para geração de tráfego VoIP. A inovação do modelo consiste em modelar o comportamento do usuário ao invés do tráfego agregado. O comportamento do usuário é retratado pela análise do tempo de retenção da chamada e pelo intervalo de tempo entre as mesmas. Enquanto a chamada está ativa, há geração de pacotes. O modelo da geração de pacotes consiste em caracterizar o tamanho dos pacotes e o intervalo de tempo entre os mesmos. As variáveis do modelo proposto foram caracterizadas por distribuições de probabilidades e modelos de séries temporais. Os dados utilizados para caracterizar o comportamento do usuário foram coletados no backbone de uma grande operadora de telecomunicações no Brasil, consistindo unicamente de chamadas VoIP. Para modelar a geração de pacotes, foram gerados dados em ambiente laboratorial. Este processo foi necessário para caracterizar unicamente a natureza dos dados. Um simulador de tráfego VoIP foi implementado e os resultados deste foram comparados com dados reais. A similaridade entre o tráfego real e sintético indica que o modelo funciona adequadamente e pode ser utilizado para estudos de redes VoIP e geração de carga. Adicionalmente foi desenvolvida uma aplicação que analisa a saída do simulador e torna mais eficiente o processo de modelamento de redes VoIP.

Abstract

In this master thesis a new model for VoIP traffic generation is proposed. The novelty of this proposal consists in modeling the user behavior instead of the aggregated traffic. The user behavior was modeled by the analysis of the call holding time and by the time interval between calls. The packet generation model consists in characterize the packet length and the packet inter arrival time. The variables of the proposed model were characterized by probability distributions and time series models. The data used to characterize the user behavior were collected in a backbone of a large Brazilian telecommunications carrier. This traffic consists uniquely of VoIP calls. The packet generation model was based on data that were generated in a laboratorial environment. This process was necessary in order to characterize properly the data behaviour. A VoIP traffic simulator was implemented and its results were compared with real data. The similarity between real and synthetic data indicates that our model works properly and can be used for VoIP modeling and workload generation. Additionally, it was developed an application that make an analysis of the simulator output to make it easier the process of VoIP network modeling.

Sumário

1	Introdução	1
1.1	Visão Geral	1
1.2	Trabalhos Relacionados	2
1.3	Objetivo	5
1.4	Metodologia	6
1.4.1	Coleta dos Dados	6
1.4.2	Simulador e Aplicativo para Auxílio à Análise de Desempenho	7
1.4.3	Análise dos Dados	8
1.5	Organização do Trabalho	8
2	Conceitos Fundamentais	9
2.1	Telefonia VoIP	9
2.1.1	Protocolo SIP	10
2.1.2	Protocolo RTP	13
2.1.3	Codecs de Voz	16
2.1.3.1	G.711	17
2.1.3.2	G.729	17
2.1.3.3	iSAC	20
2.1.3.4	SILK	21
2.2	Conceitos Estatísticos e Processos Estocásticos	22
2.2.1	Auto-Similaridade	23
2.2.1.1	Processos com Dependência de Longa e Curta Duração	24
2.2.1.2	Identificação de Um Processo Auto-Similar	25
2.2.2	Distribuições de Cauda Pesada	25
2.2.3	Testes de Aderência	26
2.2.3.1	Teste de Kolmogorov-Smirnov	26
2.2.3.2	Quantil-Quantil Plot (QQ-Plot)	28
2.2.4	Função de Auto-Correlação	28

2.2.5	Modelo Auto Regressivo	30
2.2.6	Modelo Média Móvel	30
2.2.7	Modelo ARMA	31
2.3	Modelos de Tráfego	32
2.3.1	Cadeias de Markov	32
2.3.2	Modelo ON-OFF	33
2.3.3	Modelo SURGE	33
3	Descrição do Modelo	35
3.1	Visão Geral	35
3.1.1	Tempo de Retenção da Chamada (TRC)	37
3.1.2	Intervalo de Tempo entre Chamadas (ITC)	40
3.1.3	Tamanho do Pacote(TDP)	41
3.1.3.1	G.711	41
3.1.3.2	G.729	41
3.1.3.3	iSAC	42
3.1.3.4	SILK v.3	45
3.1.4	Intervalo de Tempo entre Pacotes (ITP)	49
3.1.4.1	G.711	49
3.1.4.2	G.729	49
3.1.4.3	iSAC	49
3.1.4.4	SILK v.3	49
3.1.5	Erro do Tempo entre Pacotes (ETP)	50
4	Simulação	52
4.1	Gerador de Tráfego Sintético	52
4.1.1	Cenários de Validação	53
4.1.2	Geradores - Variáveis Aleatórias e Modelo ARMA(2,1)	55
4.2	Um Aplicativo para Auxílio à Análise de Desempenho	57
4.2.1	Interação Simulador - Aplicativo	58
4.2.2	Cenários Propostos	59
4.2.3	Resultados do Aplicativo	60
5	Análise de Resultados	64
5.1	Avaliação da Qualidade da Geração do Tráfego	64
5.2	Análise do Comportamento de Fila	66
6	Conclusão e Trabalhos Futuros	69

Lista de Figuras

2.1	Encapsulamento da Voz.	10
2.2	Exemplo de uma comunicação utilizando o protocolo SIP	11
2.3	Comunicação SIP intermediada por um Servidor Proxy	12
2.4	Estrutura do Pacote RTP	15
2.5	Exemplos de códigos da Lei- μ e Lei-A.	18
2.6	Comparação da qualidade do sinal de voz utilizando o método MUSHRA para diversos codecs presentes no mercado.	21
2.7	Comparação do codec SILK com outros codecs comuns no mercado. . .	22
2.8	Conjunto de Cantor.	23
2.9	Floco de Neve Koch.	23
2.10	Função de Distribuição Acumulada de Pareto tipo 2.	27
2.11	Teste de Aderência QQ-Plot	29
2.12	Validação do Modelo AR - Análise do Ruído.	31
2.13	Estados de uma Chamada Telefônica - Cadeia de Markov.	32
2.14	Modelo ON-OFF.	33
3.1	Tráfego de Pacotes de duas chamadas diferentes com a Identificação das Variáveis do Modelo Proposto.	36
3.2	Tempo de Retenção das Chamadas Coletadas no <i>Backbone</i> da Operadora de Telecomunicações.	38
3.3	Gráfico da Distribuição Complementar em Escala Logarítmica do <i>Tempo de Retenção de Chamadas</i>	38
3.4	<i>Quantile-Quantile Plot</i> entre os Dados Empíricos e a Distribuição Teórica de Pareto tipo 2.	39
3.5	<i>Quantile-Quantile Plot</i> entre os Dados Empíricos e a Distribuição Teórica Exponencial.	40
3.6	Comportamento do <i>Tamanho do Pacote</i> para o codec G729.	42
3.7	Periodicidade dos Pacotes para o codec G729.	42

3.8	Variação do <i>Tamanho do Pacote</i> para o codec iSAC ao longo de uma Chamada VoIP.	43
3.9	Análise do Resíduo do modelo ARMA(2,1) para o codec iSAC.	44
3.10	Comparação das Estruturas de Auto-Correlação para os Dados Reais e os Dados Gerados pelo modelo ARMA(2,1).	45
3.11	Análise do Resíduo do Conjunto 1 para o codec SILK.	46
3.12	Análise do Resíduo do Conjunto 2 para o codec SILK.	47
3.13	Comparação das Estruturas de Auto-Correlação para os Dados Reais e os Dados Gerados pelo modelo ARMA(2,1) com Parâmetros do Conjunto 1.	48
3.14	Comparação das Estruturas de Auto-Correlação para os Dados Reais e os Dados Gerados pelo modelo ARMA(2,1) com Parâmetros do Conjunto 2.	48
3.15	Comparação da Distribuição Empírica com a Distribuição Normal por codec.	50
4.1	Topologia Implementada no Simulador.	53
4.2	Análise dos Resíduos para os Dados Gerados no Simulador.	56
4.3	Comparação da Função de Auto-Correlação dos Dados Reais com os Dados Gerados.	57
4.4	Diagrama de Fluxo de Dados da Interação entre o Aplicativo e o Simulador.	58
4.5	Funcionamento Interno da Interação entre o Aplicativo e o Simulador.	59
4.6	Diagrama de Fluxo de Dados do Cenário 1.	60
4.7	Diagrama de Fluxo de Dados do Cenário 2.	60
4.8	Gráfico do Tempo de Fila por Pacote.	61
4.9	Gráfico do eMOS em função da Taxa de Transmissão.	62
5.1	Função de Auto Correlação do Tráfego Agregado. Decaimento Lento da ACF para o tráfego sintético e para o tráfego real.	65
5.2	Número de Elementos na Fila - Codec SILK v.3.	67
5.3	Número de Elementos na Fila sem Intervalo de Confiança - Codec SILK v.3.	68

Lista de Tabelas

2.1	Modos de Operação do codec SILK.	22
2.2	Matriz de Probabilidade de Estados - Markov.	32
3.1	Dados Utilizados na Caracterização do Comportamento do Usuário. . .	37
3.2	Sumário das Estatísticas sobre o <i>Intervalo de Tempo entre Chamadas</i> . .	41
3.3	Coefficientes para o modelo ARMA(2,1) do codec iSAC.	43
3.4	Coefficientes para os modelos ARMA(2,1) do codec SILK.	45
3.5	Parâmetros Obtidos da Distribuição Normal para Modelar o ETP dos Codecs Estudados.	51
4.1	Resultado da Validação do Modelo com a Configuração do Cenário 1. .	54
4.2	Resultado da Validação do Modelo com a Configuração 2.	55
4.3	Comparação de Parâmetros - Dados Reais e Modelo Implementado. . .	57
5.1	Dados Utilizados para Validar o Modelo/Simulador Desenvolvido. . . .	64

Lista de Abreviações

ACELP - Algebraic Conjugated Excited Linear Prediction
ACF - Auto Correlation Function
ACK - Acknowledgement
ADSL - Assymetric Digital Subscriber Line
AMR WB - Adaptative Multi Rate Wide Band
AR - Auto Regressive
ARIMA - Auto Regressive Integrated Moving Average
ARMA - Auto Regressive Moving Average
CBR - Constant Bit Rate
CC - CSRC Count
CD - Compact Disk
CELP - Conjugated Excited Linear Prediction
codec - coder/decoder
CS ACELP - Conjugated Structure ACELP
CSRC - Contributing Source Identifiers
eMOS - Electronic Mean Opinion Score
ETP - Erro do Tempo entre Pacotes
FEC - Foward Error Correction
FIFO - First In First Out
GSM AMR - Global System for Mobile Communications Adaptative Multi Rate
H - Hurst
HMM - Hora de Maior Movimento
HTTP - Hiper Text Transport Protocol
ID - Identificação
IETF - Internet Engineering Task Force
iLBC - Internet Low Bitrate Codec
IP - Internet Protocol
IPTV - Internet Protocol Television
iSAC - Internet Speech Audio Codec
ITC - Intervalo de Tempo entre Chamadas
ITP - Intervalo de Tempo entre Pacotes
ITU - Internation Telecommunications Union
KS - Kolmogorov Smirnov
LAN - Local Area Network

LCP - Linear Predictive Coding
LRD - Long Range Dependence
M - Mark
MA - Moving Average
MOS - Mean Opinion Score
MUSHRA - Multiple Stimulus with Hidden Reference and Anchor
NF - Número de Elementos na Fila
OSI - Open Systems Interconnection
P - Padding
PC - Personal Computer
PCM - Pulse Code Modulation
PT - Payload Type
QoS - Quality of Service
QQ - Quantil Quantil
RFC - Request for Comment
RTCP - Real Time Control Protocol
RTP - Real Time Protocol
SCTP - Stream Control Transmission Protocol
SDP - Session Description Protocol
SIP - Session Initiation Protocol
SMTP - Simple Mail Transport Protocol
SRD - Short Range Dependence
SSRC - Synchronization Source Identifier
SURGE - Scalable URL Reference Generator
SVOPC - Sinusoidal Voice over Packet Coder
TCP - Transmission Control Protocol
TDP - Tamanho do Pacote
TF - Tempo na Fila
TLS - Transport Layer Security
TRC - Tempo de Retenção de Chamada
UDP - User Datagram Protocol
UE - User Equivalent
URI - Uniform Resource Identifier
URL - Uniform Resource Location
VBR - Variable Bit Rate
VoIP - Voice over Internet Protocol
X - Extension

Capítulo 1

Introdução

1.1 Visão Geral

O planejamento de redes em telecomunicações é uma prática de suma importância, pois através deste é possível alocar recursos de maneira precisa, realizar tarifação justa, prever probabilidade de falhas da rede, horários de congestionamento, entre outros. A alocação de recursos está diretamente relacionada a aplicação em questão. Enquanto uma aplicação *Web* demanda largura de banda e baixo *Round Trip Time*, aplicações em tempo real demandam baixa perda de pacotes e um pequeno valor de *jitter*, além das demandas *Web*.

O planejamento de redes se baseia em modelos de tráfego, pois através destes é possível verificar a eficácia e os parâmetros da rede planejada. Um modelo é definido como a representação do sistema com o propósito de estudá-lo [Banks et al. 2001]. Um bom modelo precisa representar a realidade com simplicidade e precisão, para torná-lo de fácil entendimento e prover resultados confiáveis [Menascé and Almeida 1998].

As tradicionais redes de telefonia comutada já foram extensivamente estudadas e possuem modelos de desempenho bem conhecidos. Esses modelos utilizam a distribuição exponencial tanto para modelar o intervalo entre chamadas quanto a duração das mesmas [Flood 1997]. A utilização da distribuição exponencial para modelar o tempo de duração da chamada está diretamente relacionada com o comportamento do usuário. As redes de telefonia comutada são tarifadas por tempo, portanto o usuário geralmente faz ligações de curta duração e, por consequência, as chamadas de longa duração são raras. Gradualmente os sistemas telefônicos estão migrando para VoIP (*Voice over Internet Protocol*), como mostra [Teleco 2010]. Os serviços VoIP não são tarifados por tempo, mas por uma tarifa periódica, usualmente mensal, em relação à largura de banda fornecida. Os usuários da telefonia VoIP possuem um compor-

tamento diferente dos usuários da telefonia tradicional, pois as chamadas de longa duração não são mais raras. Por consequência, os modelos telefônicos tradicionais falham ao modelar o tráfego VoIP, como mostram [Chen et al. 2007], [Pedroso et al. 2009] e [de Mattos et al. 2010].

1.2 Trabalhos Relacionados

No início dos anos 90 foi comprovado por [Leland et al. 1994] que a natureza do tráfego agregado numa Ethernet LAN é estatisticamente auto-similar. A auto-similaridade é reportada quando um processo mantém parte de suas características estocásticas por uma certa faixa de escalas de observação, como definido na seção 2.2.1. Esse resultado contrastou com trabalhos tradicionais que afirmavam que o tráfego de dados seguia o modelo markoviano [Heffes and Lucsmtoni 1986] ou podia ser descrito por um trem de pacotes [Jain and Routhier 1986]. Essa nova percepção da natureza do tráfego de dados implicou em sérias reformulações no processo de planejamento, controle e análise de redes. Para gerar um tráfego sintético auto-similar, Leland et al. primeiramente propuseram o uso do Ruído Fracional Gaussiano e o modelo ARIMA fracional (*Auto-Regressive Integrated Moving Average*). Todavia, de acordo com eles, esse método é apropriado apenas para amostras pequenas, cerca de 1000 observações. Para gerar grandes tráfegos auto-similares sintéticos, com mais de 100.000 observações, os autores utilizaram um modelo baseado em teoria de filas e outro baseado na agregação de processos AR(1), auto-regressivo de primeira ordem. A auto-similaridade dos tráfegos gerados foi medida com o uso do parâmetro de Hurst, descrito na seção 2.2.1.1.

O trabalho de [Crovella and Bestavros 1997] provê uma explicação para a natureza do tráfego de dados na Internet ser auto-similar. Para gerar o tráfego, eles utilizaram o modelo ON-OFF (vide seção 2.3.2 para maiores detalhes) com tempo de permanência descrito por uma distribuição de cauda pesada, descrita no tópico 2.2.2, para ambos os estados. O tempo ON representa o tempo de transmissão dos arquivos e o tempo OFF o intervalo entre as transmissões. A natureza, diversidade de aplicações e tamanhos, dos arquivos disponíveis na Internet consiste no principal fator para o tempo ON ser modelado por uma distribuição de cauda pesada. O tempo OFF foi dividido em duas parcelas, uma representando o tempo de processamento da máquina e outra o intervalo de tempo entre ações, ou cliques, do usuário. Essa última parcela é o principal fator para o tempo OFF também ser modelado por uma distribuição de cauda pesada. Comparando os dois estados do modelo, Crovella et al. chegaram a conclusão que o tempo ON possui uma cauda mais pesada que o tempo OFF e, por consequência, possui maior influência na natureza auto-similar do tráfego de dados. O trabalho de Crovella

foi complementado por [Gong et al. 2005], o qual fornece explicações adicionais sobre a auto-similaridade e propõe novos modelos para gerar o comportamento de cauda pesada.

O trabalho de [Chen et al. 2007] mostrou que o uso da distribuição exponencial para modelar a duração da chamada VoIP é inadequado. Nesse trabalho foi analisado o tráfego de uma companhia de telefonia celular VoIP em Taiwan. Foi ajustada uma mistura de duas distribuições log-normal aos dados. Em [Pedroso et al. 2009] foi analisado o tráfego VoIP de duas grandes operadoras de telefonia do Brasil. A análise foi feita a nível do comportamento do usuário. Em ambas as operadoras, o tempo de duração de chamada foi caracterizado por uma distribuição de cauda pesada. Uma operadora oferecia o serviço de VoIP puro, inicia e finaliza a ligação com VoIP, e a outra era uma operadora de telefonia celular comum que convertia os dados internamente para facilitar e baratear o transporte dos dados. Os resultados desse trabalho mostram que a presença da cauda pesada é mais intensa no tráfego VoIP puro. O trabalho de [Pedroso et al. 2009] foi complementado por [de Mattos et al. 2010]. Nesse último trabalho foi modelado o tráfego de pacotes dos codecs G.711 [ITU-T 1988] e G.729 [ITU-T 1996a]. Esse modelo foi estabelecido com o uso das distribuições de probabilidade Gamma, Normal e Exponencial.

Os estudos do comportamento dos codecs de áudio em relação ao tráfego VoIP foram inicialmente realizados nos trabalhos de [Jiang and Schulzrinne 2000] e de [E.Casilari et al. 2002]. Em ambos os trabalhos foram propostos modelos para caracterizar codecs com detecção de silêncio. Eles propuseram o uso do modelo ON-OFF, sendo que o estado ON representa o tempo de fala e o estado OFF o tempo de silêncio. Os dados para o estudo de [Jiang and Schulzrinne 2000] foram coletados de conversas telefônicas, entre telefonia VoIP e comutada, gravadas. No trabalho de [E.Casilari et al. 2002] os dados utilizados eram proveniente de uma videoconferência. Nos dois trabalhos os estados de ON e OFF não aderiram à distribuição exponencial, mas sim a uma distribuição com decaimento mais lento, como a log-normal. O modelo dos estados varia se a técnica de *hangover* estiver ativada. Essa técnica evita que ocorram cortes abruptos na fala decorrentes de silêncios de pronúncia, como em algumas consoantes por exemplo.

Uma abordagem mais aprofundada para modelar os codecs de áudio foi proposta por [Menth et al. 2009]. Nesse trabalho, as fontes de áudio utilizadas consistiam em amostras de sinais de voz utilizadas no desenvolvimento de codecs [Schiel 1996], as quais foram codificadas pelos codecs abordados no estudo com a utilização do software Pico-phone [Vitez 2009]. Os fluxos de pacotes provenientes da codificação foram estudados e modelados a partir de processos estocásticos. Eles analisaram codecs CBR (*Constant*

Bit Rate), codecs com detecção de silêncio e codecs VBR (*Variable Bit Rate*).

Os codecs CBR analisados por [Menth et al. 2009] foram o G.711 e o G.729, e o modelo proposto para estes é bastante intuitivo. Pelo fato de eles terem analisado a saída do codec diretamente, sem tempos de processamento e filas, o modelo CBR é somente uma tabela com as características de geração de pacotes, como intervalo de tempo entre pacotes, o tamanho do pacote e as informações de controle. O modelo ON-OFF foi utilizado para modelar o comportamento dos codecs com detecção de silêncio, G.723.1 [ITU-T 1996b] e iLBC [Andersen et al. 2004]. O estado ON representa o tempo de fala e o estado OFF o tempo de silêncio. O reconhecimento dos estados, ON e OFF, não é uma tarefa trivial, dado que existem pequenos intervalos de silêncio durante a fala, os quais podem ser interpretados como silêncio.

Os autores propuseram três métodos de janelamento para identificar os períodos de ON e OFF. O primeiro método, W0, consiste em observar a ausência ou presença de pacotes após um período de fala ou silêncio, respectivamente. Dessa forma, a transição do período de fala para silêncio ocorre após a ausência de um pacote e a transição de silêncio para fala ocorre após a detecção de um pacote. O método W1 consiste num aprimoramento do método W0. Nesse, a janela de pacotes necessária para ocorrer a transição de estados é ajustável. [Menth et al. 2009] perceberam que janelas pequenas conduziam a uma interpretação equivocada entre as pequenas pausas que ocorrem durante a fala com os períodos de silêncio. Por outro lado, janelas grandes ignoram quantidades significativas do período de fala. Esse problema foi solucionado com o método W2, o qual possui tamanhos de janelas distintas para as duas transições de estados. Baseados em observações e testes, os autores identificaram que o valor ideal de pacotes ausentes para a transição do estado de fala (ON) para o estado de silêncio (OFF) é de 50 pacotes. Já para a transição do estado OFF para ON, o valor estipulado para a janela foi de 15 pacotes.

O modelo para os codecs VBR, GSM AMR [3rd Generation Partnership Project (3GPP)] e iSAC [GIPS], foram feitos em redes atuando em perfeitas condições. O codec GSM AMR possui um método de detecção de voz diferente do G.723.1 ou do iLBC, pois durante o tempo de silêncio são enviados pacotes para sincronização e para gerar um ruído de fundo. Este ruído cria uma sensação de conforto, pois dá a sensação de que a chamada continua ativa. Portanto, o fluxo de pacotes do GSM AMR não pára completamente nos períodos de silêncio, fato que dificulta a utilização do modelo ON-OFF, como sugerido por [Menth et al. 2009]. O codec iSAC possui um comportamento puramente VBR, com tamanho de pacotes variando de 21 bytes a 166 bytes. Entretanto, devido ao fluxo e tamanho dos pacotes, os tempos de fala e de silêncio continuam sendo detectados. Os autores utilizaram uma Cadeia de Markov

com Memória [MMC 1999] para modelar o tráfego VBR. Foram testados diferentes parâmetros para a Cadeia de Markov até atingir uma boa performance. Para o codec iSAC, os melhores resultados foram obtidos com a remoção do tráfego de controle.

No trabalho de [Huang et al. 2010] foi realizado um estudo do comportamento dos codecs em relação à perda de pacotes. O tráfego de voz foi gerado utilizando o software Skype, o qual utiliza o mecanismo FEC (*Forward Error Correction*) para cancelamento de erros baseado em redundâncias. Nesse estudo foram utilizados os codecs G.729, iSAC e SVOPC [Lindblom 2004]. Todos os codecs estudados no trabalho aumentam a taxa de transmissão, ajustando o tamanho do pacote, para se adaptarem à perda de pacotes. O codec G.729 varia o tamanho do pacote de acordo com níveis discretos. O codec iSAC possui grande variabilidade do tamanho dos pacotes, mas possui apenas dois níveis médios para se adaptar às perdas. Uma variabilidade menor de pacotes foi visualizada no tráfego do codec SVOPC. Todavia, esse também possui apenas dois níveis médios de tamanho de pacotes.

1.3 Objetivo

Neste trabalho é apresentado um novo modelo para o tráfego VoIP. O modelo proposto é de fácil parametrização, cujas métricas podem ser obtidas através de simples medições na rede, como a média do intervalo e da duração das chamadas. A parametrização da duração das chamadas precisa do parâmetro de forma e de escala da distribuição de Pareto. Todavia, caso esses parâmetros não sejam conhecidos, o parâmetro de escala é calculado com base na média e num parâmetro de forma aceito pela comunidade acadêmica, [Pedroso et al. 2009] e [de Mattos et al. 2010].

O modelo procura caracterizar o comportamento do usuário, ao invés de modelar o tráfego agregado. Essa abordagem conduz a diversos benefícios, como maior simplicidade e precisão quando comparados aos modelos existentes. O modelo proposto se divide em duas partes principais, o comportamento do usuário e o comportamento dos dados. O comportamento do usuário corresponde à duração da chamada e o intervalo entre as mesmas, enquanto que o comportamento dos dados está relacionado com o padrão de geração de pacotes. Para poder caracterizar as chamadas, foram coletados dados em uma grande operadora de telecomunicações no Brasil. Essa operadora fornece um serviço de VoIP puro, de forma que o usuário origina e recebe ligações usando somente VoIP. O tráfego de pacotes está diretamente relacionado com o codec (*coder-decoder*) utilizado. Foram gerados vários fluxos de pacotes, em laboratório, para cada codec estudado, de forma a modelar o comportamento dos mesmos. O modelo proposto é mais simples que o desenvolvido por [Menth et al. 2009], pelo fato de uti-

lizar distribuições de probabilidades e modelos de séries temporais ao invés de processos de janelamento e outros métodos heurísticos. Adicionalmente, o presente trabalho complementa o trabalho de [Pedroso et al. 2009] com uma análise mais profunda. A precisão do modelo proposto é confirmada através de simulação computacional e pela comparação dos dados simulados com os dados reais.

1.4 Metodologia

A primeira etapa do trabalho consistiu em realizar a coleta de dados numa grande operadora de telecomunicações do Brasil. A partir da análise desses dados, foi proposto um modelo estatístico para o comportamento da sessão VoIP. Para caracterizar o comportamento dos dados numa sessão, foram gerados fluxos de pacotes para diversos codecs em ambiente laboratorial. Em laboratório é possível controlar o uso da rede e a carga de processamento das máquinas utilizadas, de forma que o fluxo de pacotes capturado seja o mais fiel possível com aquele gerado pelo codec. Para verificar a precisão e eficiência do modelo, este foi implementado em um simulador. Os resultados gerados foram analisados com ferramentas estatísticas, como a função de auto-correlação e o parâmetro de Hurst.

1.4.1 Coleta dos Dados

A coleta dos dados foi realizada no *backbone* da rede, o qual consiste numa rede Ethernet não congestionada. Os pacotes referentes à mesma sessão foram identificados pelo endereço IP, fonte e destino, e pelo número de sequência do protocolo RTP (*Real Time Protocol*). Este último foi necessário porque o mesmo usuário pode possuir conversas com diferentes destinos simultaneamente, como numa conferência.

O serviço VoIP oferecido pela operadora de telecomunicações utiliza o protocolo SIP (*Session Initiation Protocol*) [Rosenberg et al. 2002] para sinalização e o RTP [Schulzrinne et al. 2003] para transporte dos dados. O protocolo SIP foi projetado para inicializar, modificar e terminar sessões independentemente da mídia ou aplicação. O tópico 2.1.1 apresenta maiores detalhes sobre o SIP. Depois de inicializada a sessão, a voz é codificada/decodificada por um codec e transmitida pelo RTP. Este protocolo possui funcionalidades para comunicações em tempo real, como mostra o tópico 2.1.2. Os codecs utilizados pela operadora foram o G.711 [ITU-T 1988] e o G.729 [ITU-T 1996a]. O codec G.729 foi utilizado em 93% das sessões e o G.711 nas sessões remanescentes. Maiores informações sobre esses codecs podem ser encontradas no tópico 2.1.3.

A rede analisada possuía cerca de 10.000 usuários na época da coleta, em Setembro de 2007. Em termos gerais, a rede de acesso é formada por enlaces ADSL (*Asymmetric Digital Subscriber Line*). O tráfego VoIP é gerado pelos usuários e transportado por uma rede Gigabit Ethernet não congestionada. Para realizar a coleta dos dados, as portas do *switch* Ethernet, que servem de *backbone*, foram espelhadas de forma que o tráfego total VoIP fosse capturado utilizando o software livre Wireshark [Orebaugh et al. 2006].

A geração dos dados em laboratório foi realizada com o uso dos softwares Ekiga [Sandras 2010] e Skype [Limited 2011], sendo esse último nas versões 3.1 e 4.2. No Ekiga foi realizada a coleta de dados para o codec G.711 lei A, cuja lei é utilizada por vários países, menos Estados Unidos e Japão. O Skype versão 3.1 foi utilizado para gerar o tráfego do codec iSAC e a versão 4.2 foi utilizada para obter o tráfego do codec SILK. Quando versões do Skype com codecs padrão diferentes tentam se comunicar, utiliza-se um codec em comum, sendo esse o G.729. Dessa forma foi obtido o tráfego para esse codec. O áudio utilizado nas sessões VoIP que geraram os pacotes são arquivos disponibilizados pela ITU-T para a comunidade acadêmica para o desenvolvimento de codecs [ITU-T]. O fluxo de pacotes foi capturado com o uso do Wireshark. Toda a análise dos dados foi feita no software estatístico R Cran [Team 2009].

1.4.2 Simulador e Aplicativo para Auxílio à Análise de Desempenho

O modelo proposto consiste em caracterizar o comportamento do usuário e do tráfego de pacotes. O tráfego VoIP sintético é gerado a partir de vários usuários funcionando paralelamente. Os tráfegos convergem em um único canal e a fila de pacotes obedece a disciplina FIFO (*First In - First Out*) com tamanho infinito. Utilizou-se a linguagem de programação Java para torná-lo independente da plataforma em que o simulador será executado. Cada cliente no sistema é uma *thread*, que possui acesso às rotinas do sistema controlado por um semáforo. O modelo proposto se utiliza de distribuições de probabilidade para caracterizar o tráfego VoIP. Por essa razão, foram desenvolvidos geradores de variáveis aleatórias para as distribuições utilizadas.

O simulador desenvolvido pode ser utilizado para o planejamento de redes de telecomunicações. Todavia, para interpretar os dados de saída, o usuário precisa possuir um bom conhecimento técnico, de forma a calcular alguns parâmetros de planejamento de redes, como *jitter*, MOS (*Mean Opinion Score*) [ITU-T 1996c], fator de ocupação e taxa de transmissão. Para suprir essa necessidade foi desenvolvido um aplicativo que torna essa interpretação automática, fornecendo ao usuário as métricas da rede.

1.4.3 Análise dos Dados

O modelo proposto caracteriza o comportamento dos clientes VoIP. Todavia, a análise foi realizada com o tráfego agregado. A primeira análise consistiu em verificar se o tráfego agregado sintético possui um comportamento de dependência de longa duração. Essa verificação foi realizada com a função de auto-correlação. Adicionalmente foi quantificado essa dependência com o uso do parâmetro de Hurst. Para comprovar o comportamento de cauda pesada, foi realizada uma análise do número de elementos na fila ao longo do fator de ocupação. Junto a esse gráfico foi inserida outra curva que mostra como seria a previsão do tráfego se fosse utilizado o modelo tradicional, o qual se baseia na distribuição exponencial.

1.5 Organização do Trabalho

O trabalho está organizado da seguinte forma. O capítulo 2 trata sobre os conceitos fundamentais presentes nesta dissertação. O capítulo 3 apresenta o modelo proposto. O simulador e o aplicativo que interage com o mesmo estão descritos no capítulo 4. No capítulo 5 apresentam-se resultados da simulação de tráfego e a análise destes. Conclusão e trabalhos futuros são apresentados no capítulo 6.

Capítulo 2

Conceitos Fundamentais

2.1 Telefonia VoIP

Na telefonia comutada, um circuito é disponibilizado para cada ligação. Essa característica da telefonia tradicional garante a qualidade do serviço, mas ocasiona em uma baixa eficiência na utilização de recursos. Isso porque, uma ligação não ocupa toda a capacidade de um circuito, além de existirem momentos de silêncio durante uma conversa, os quais tornam a ocupação ainda menor. Esses fatores que impactam na baixa eficiência da telefonia comutada impulsionaram o desenvolvimento de um novo tipo de telefonia, a telefonia VoIP.

Na telefonia VoIP, a comutação é por pacotes. Nessa abordagem, a voz é segmentada em pequenas partes e cada parte é encapsulada em um pacote. Cada pacote é encaminhado ao destino de forma independente dos outros pacotes. Dessa forma, os circuitos são utilizados de maneira dinâmica, aumentando a eficiência. Esse ganho em eficiência resulta numa demanda menor de infra-estrutura, fato que torna a telefonia VoIP mais interessante economicamente. Na telefonia VoIP, não há necessariamente reserva de circuito para uma ligação. Todavia, com a diminuição do custo da largura de banda e o desenvolvimento das técnicas de QoS (*Quality of Service*), as operadoras de telecomunicações realizam reservas de recursos para o tráfego VoIP de forma a garantir qualidade na ligação.

A estrutura fundamental do pacote VoIP pode ser vista na figura 2.1. O protocolo de transporte utilizado é o UDP (*User Datagram Protocol*) [Postel 1980]. A principal característica do UDP que o faz ser utilizado no VoIP é o fato desse ser simples e não garantir a entrega dos pacotes. Comunicações em tempo real geralmente não possibilitam a retransmissão de dados perdidos, pois um atraso de propagação maior que 250ms pode prejudicar a qualidade da chamada VoIP. O protocolo RTP [Schulzrinne et al. 2003]

possui funcionalidades que garantem maior controle dos dados em aplicações de tempo real. Na figura 2.1 pode se ver como a voz é transmitida por uma rede de dados, quando a sessão já está estabelecida. Para iniciar uma sessão VoIP, geralmente utiliza-se o protocolo SIP [Rosenberg et al. 2002]. Por fim, a digitalização e codificação da voz são realizadas pelos codecs. As seções 2.1.1, 2.1.2 e 2.1.3 fornecem uma descrição mais detalhada dos protocolos SIP, RTP e dos codecs respectivamente.

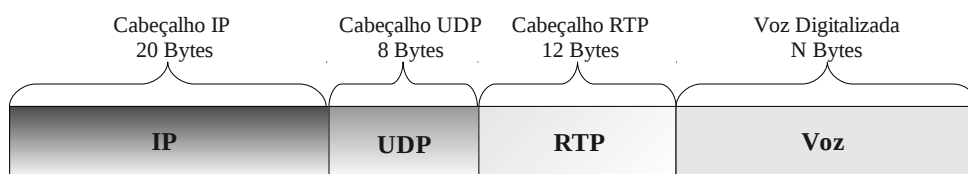


Figura 2.1: Encapsulamento da Voz.

2.1.1 Protocolo SIP

O protocolo de sinalização SIP foi desenvolvido pela IETF (*Internet Engineering Task Force*) com o objetivo de inicializar, controlar e finalizar sessões com um ou mais participantes. A RFC 3261 (*Request for Comment*) [Rosenberg et al. 2002], que padroniza o SIP, define as sessões como chamadas telefônicas, distribuições multimídia ou conferências multimídia. De acordo com [Johnston 2004], as principais funções do protocolo SIP são:

- (i) Localização dos terminais;
- (ii) Verificação da disponibilidade dos terminais para estabelecer uma conexão;
- (iii) Troca informações sobre mídia e estabelecimento a conexão;
- (iv) Enviar mensagens de controle durante a sessão de forma a gerenciá-la;
- (v) Finalizar as Sessões.

Uma função secundária do SIP consiste em informar se os usuários da lista de contato estão conectados. Portanto, o SIP publica e envia mensagens sobre a presença dos usuários, assim como solicita e entrega essas mensagens. A estrutura do protocolo possui semelhanças com dois protocolos amplamente utilizados, o HTTP (*Hyper Text Transport Protocol*) e o SMTP (*Simple Mail Transport Protocol*). A arquitetura de cliente-servidor e o uso de URL (*Uniform Resource Location*) e URI (*Uniform Resource*

Identifier) provém do HTTP e o sistema de codificação de textos e o estilo do cabeçalho do SMTP.

A figura 2.2 ilustra uma comunicação entre dois usuários utilizando o protocolo SIP. Para iniciar a comunicação, o usuário A envia uma mensagem INVITE ao usuário B. Essa mensagem contém várias informações pertinentes ao início da sessão, sendo as principais o formato de sessão a ser estabelecida e o *codec* a ser utilizado. Ao receber a mensagem, o usuário B responde com um 180 RINGING para avisar o usuário A que a mensagem INVITE foi recebida. Para aceitar a comunicação com os parâmetros propostos no INVITE, uma mensagem de 200 OK é enviada pelo usuário B. Ao receber o 200 OK, o usuário A envia a confirmação ACK e estabelece a ligação. Para finalizar a ligação, um dos lados envia a mensagem BYE. Entretanto, a sessão só é finalizada com a confirmação do BYE, com uma mensagem 200 OK. Nesse exemplo identifica-se a utilização da arquitetura cliente-servidor proveniente do HTTP e a troca de mensagens de texto semelhante ao SMTP. Todos os parâmetros negociados numa ligação, como portas, codecs e protocolos, são enviados pelo SDP (*Session Description Protocol*) [Handley et al. 2006], o qual é transportado no corpo do protocolo SIP.

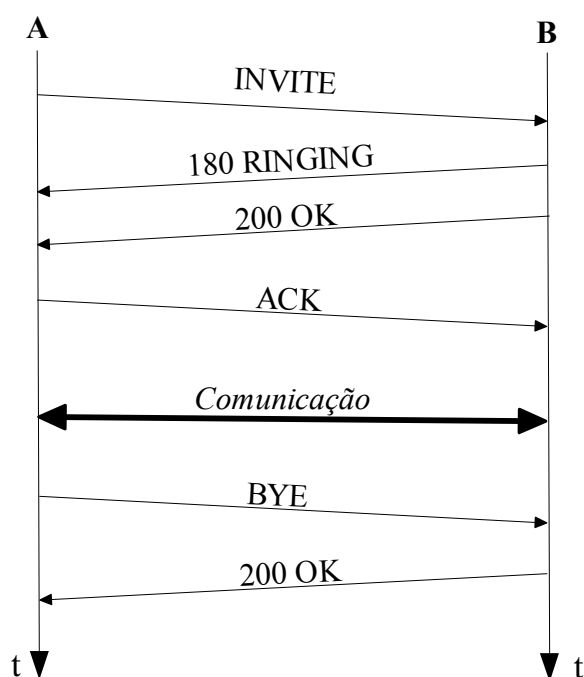


Figura 2.2: Exemplo de uma comunicação utilizando o protocolo SIP

O exemplo da figura 2.2 mostra uma comunicação na qual os endereços IP são conhecidos. Todavia, essa condição não é muito comum. Um usuário pode querer

acessar serviços de comunicação, baseados em SIP, de várias localidades diferentes, tais como no trabalho, em casa, numa viagem, entre outros. Para facilitar a identificação desse usuário são utilizadas as URIs, que identifica o usuário independente do endereço IP. A tradução da URI para um endereço IP é realizada no Servidor Proxy. Uma URI é composta pelo nome do usuário e pelo Servidor Proxy a ser acessado: sip:exemplo@servidor.com. Nesse exemplo de URI, identifica-se a comunicação SIP pelo usuário *exemplo* no Servidor Proxy *servidor.com*. Antes de iniciar a comunicação, os usuários envolvidos devem estar previamente registrados no servidor. A figura 2.3 ilustra a comunicação SIP por intermédio de um Servidor Proxy. A mensagem 100 TRYING avisa o remetente da chamada que o servidor está tentado entrar em contato com o destinatário da chamada. Após o servidor identificar as partes da ligação, a comunicação ocorre diretamente.

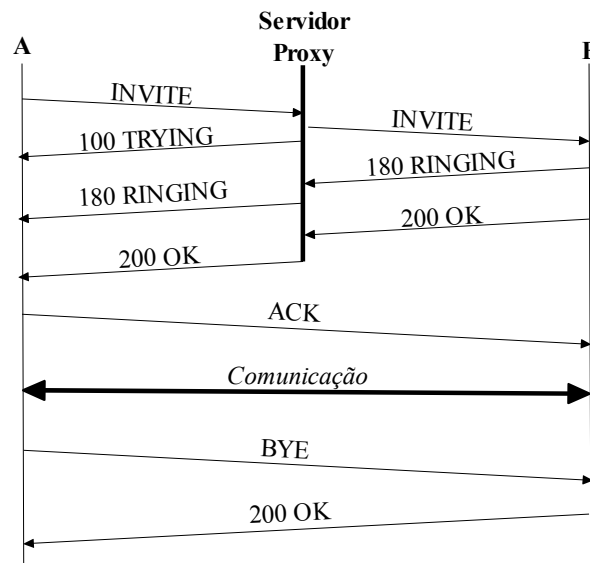


Figura 2.3: Comunicação SIP intermediada por um Servidor Proxy

As mensagens SIP podem ser agrupadas em seis classificações distintas, como evidenciado nos tópicos abaixo. As mensagens informativas indicam como está o progresso da chamada. As mensagens de sucesso indicam que uma requisição foi recebida ou aceita. Caso ocorra alguma alteração sobre a localização dos agentes SIP, uma mensagem de redirecionamento é enviada. Os grupos restantes sobre mensagens SIP se referem a possíveis falhas que podem ocorrer.

- (i) 1xx - Mensagem Informativa. Exemplo: 180 RINGING;
- (ii) 2xx - Resposta de Sucesso. Exemplo: 200 OK;

- (iii) 3xx - Resposta de Redirecionamento. Exemplo: 302 MOVED TEMPORARILY;
- (iv) 4xx - Resposta de Falha de Requisição. Exemplo: 404 NOT FOUND;
- (v) 5xx - Resposta de Falha do Servidor. Exemplo: 503 SERVICE UNAVAILABLE;
- (vi) 6xx - Resposta de Falha Global. Exemplo: 600 BUSY EVERYWHERE.

O SIP é um protocolo de aplicação e pode ser transportado pelo TCP (*Transmission Control Protocol*) ou pelo UDP (*User Datagram Protocol*). Implementações recentes permitem o transporte de mensagens SIP utilizando protocolo SCTP (*Stream Control Transmission Protocol*) [Johnston 2004]. O método mais utilizado atualmente para o transporte de mensagens SIP consiste em utilizar o UDP, o qual não garante a entrega dos dados. Em contrapartida, o TCP garante confiabilidade e controle de congestionamento, ao custo de uma maior complexidade na comunicação. Uma outra forma de transportar mensagens SIP é utilizando o TLS (*Transport Layer Security*) sobre o TCP. Nessa configuração, as mensagens transportadas são criptografadas e faz-se necessário um processo de autenticação no Servidor Proxy.

A tecnologia VoIP gradualmente está sendo incorporada nos sistemas telefônicos. Por consequência, existem usuários tanto da tecnologia VoIP quanto da telefonia comutada tradicional. Para tornar possível a comunicação entre os usuários de tecnologias diferentes, faz-se necessário a presença de um *Gateway* SIP. Este equipamento realiza a conversão da sessão de uma rede de dados para a rede de telefonia pública e vice-versa.

2.1.2 Protocolo RTP

O protocolo RTP foi desenvolvido para transportar dados em tempo real sobre uma rede IP, independente da mídia em questão. A detecção de perdas de pacotes, *jitter* e entrega fora de sequência são as principais funcionalidades do RTP. A IETF formaliza as características e os detalhes de implementação do RTP na RFC 3550 [Schulzrinne et al. 2003]. Apesar de ser desenvolvido para aplicações em tempo real, o protocolo RTP não garante nenhum tipo de qualidade de serviço.

O RTP transporta aplicações, como vídeo e voz, fato que sugere definir o RTP como protocolo de transporte. Por outro lado, o RTP possui interface direta com o usuário que escreve o código de aplicação e que envia os pacotes RTP para o socket UDP, o qual realiza o transporte. Uma definição aceita sobre a classificação do RTP no modelo OSI (*Open Systems Interconnection*) consiste em que este é um protocolo de transporte implementado na camada de aplicação.

Para entender como funciona o RTP, será analisada uma aplicação CBR (*Constant Bit Rate*). Uma aplicação CBR envia dados de forma constante, como por exemplo um fluxo de dados com pacotes gerados a cada 10ms com um tamanho fixo. Esse fluxo é encapsulado em um pacote RTP, o qual contém um número de sequência, um (*timestamp*), identificação da codificação utilizada e o cabeçalho RTP. O pacote RTP é transportado pelo UDP, o qual pode ser transmitido para um ou vários destinos. O número de sequência do RTP é necessário para remontagem dos dados no lado do receptor, pois os pacotes podem chegar em ordens diferentes após trafegarem pela rede IP. Caso o sequenciamento identifique a falta de um pacote, a aplicação terá que interpretar os dados sem o mesmo ou solicitar outro pacote, desde que o tempo entre pacotes seja inferior a 250ms. O *timestamp* consiste no horário que o pacote foi transmitido. Para o receptor, o que importa é a diferença desses horários, pois o mesmo verifica essa diferença e decide se deve reproduzir o dado assim que este chegou ou bufferizá-lo para reproduzir no tempo correto. Dessa forma, a aplicação é reproduzida de forma semelhante à aplicação original e é possível realizar sincronização entre duas aplicações paralelas e dependentes, como o áudio e vídeo de uma aplicação de IPTV. A informação sobre o codificador serve para informar o receptor como deverá ser feito o processo de decodificação.

A estrutura do pacote RTP está ilustrada na figura 2.4. A versão do protocolo RTP, cuja atual é 2, é indicada no primeiro campo. O tamanho do pacote RTP tem que ser múltiplo de 4 bytes. Caso isso não ocorra naturalmente, a aplicação adiciona octetos de forma a garantir essa multiciplidade. O campo P (*Padding*) indica se foram ou não adicionados octetos. A identificação da quantidade de octetos adicionados está no último octeto. Se houver algum cabeçalho de extensão entre o cabeçalho normal e o campo de dados, este é identificado no campo X (*Extension*). Na prática, de acordo com [Tanenbaum 2002], o campo X atua como um campo reservado para necessidades futuras. A quantidade de fontes é identificada no campo CC (*CSRC Count*). O bit M (*Mark*) serve para indicar o início de um quadro de vídeo, início de uma fala numa conversa com supressão de silêncio ou qualquer outro elemento que a aplicação reconheça. O algoritmo de codificação utilizado é informado no campo PT (*Payload Type*). O número de sequência, utilizado para reordenamento de pacotes e detecção de perdas, possui um comprimento de 16 bits. A diferença entre os valores do campo *timestamp* é utilizado pelo receptor para remover o *jitter* da transmissão. O campo SSRC (*Synchronization Source Identifier*) identifica o terminal que envia os pacotes RTP. Esse número é único e é sorteado no início da transmissão. O número SSRC serve para multiplexar e desmultiplexar os vários fluxos de pacotes RTP contidos num fluxo UDP. Se um multiplexador receber vários pacotes RTP de origens diferentes

e transmití-los de forma combinada, este multiplexador é identificado no campo CSRC (*Contributing Source Identifiers*). A quantidade de multiplexadores pode variar de 0 até 15 e é informada no campo CC.

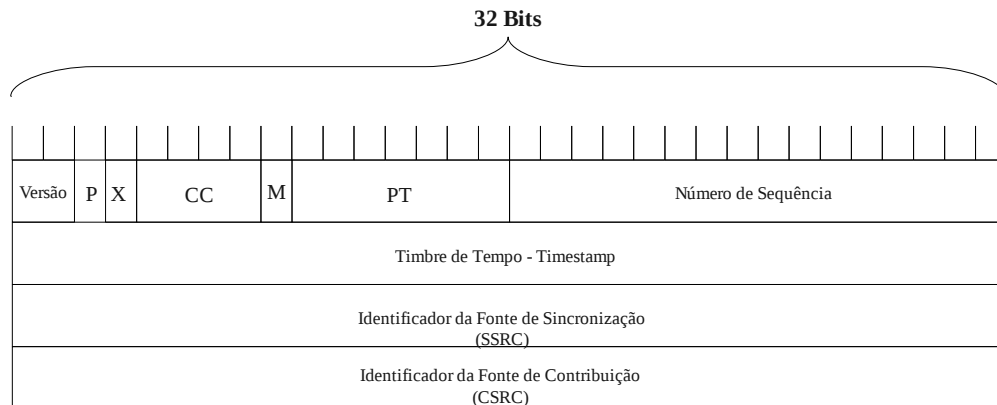


Figura 2.4: Estrutura do Pacote RTP

Um outro protocolo atua conjuntamente com o RTP, o RTCP (*Real-time Transport Control Protocol*), que também está definido na RFC 3550. O RTCP não transporta dados, entretanto atua na realimentação, sincronização e na interface com o usuário. A realimentação serve para informar os codificadores sobre a situação da rede, como por exemplo, a largura de banda, o retardo e o congestionamento. Baseado nessas informações, o codec tenta se adequar às condições da rede para fornecer uma transmissão de dados com a maior qualidade possível. Diferentes fluxos de dados podem utilizar relógios distintos. O RTCP atua de forma a manter a sincronização entre esses fluxos. A identificação das diversas origens também é feita no RTCP, de forma a indicar à aplicação quem está se comunicando no momento. O RTCP sempre é transmitido para o endereço de *multicast*, ou seja, todos os participantes da sessão recebem o pacote de controle.

Para cada fluxo de pacotes RTP recebidos, o receptor gera relatórios de recepção. Esses são agregados em um único pacote RTCP e podem informar:

- (i) O número de sequência do último pacote recebido;
- (ii) O jitter;
- (iii) O SSRC do gerador dos relatórios;
- (iv) Pacotes perdidos.

Da mesma forma que o receptor, o transmissor gera relatórios para cada fluxo de pacotes RTP transmitido. As principais informações desses relatórios são:

- (i) O SSRC do transmissor;
- (ii) O *timestamp* entre pacotes e o relógio de sincronização;
- (iii) O número de pacotes enviados no fluxo;
- (iv) O número de bytes enviado no fluxo.

2.1.3 Codecs de Voz

O codec é um dispositivo que realiza a conversão do sinal de analógico para digital, transformando-o num código para posteriormente, após uma transmissão dos dados por exemplo, decodificar o sinal e convertê-lo novamente para analógico.

A natureza e comportamento dos sinais de vídeo, música e voz são bastante distintas, por essa razão existem codecs dedicados para cada aplicação. O sinal de vídeo é bastante redundante, com poucas diferenças entre quadros subsequentes. Dessa forma, o codec de vídeo envia um quadro de referência e depois envia somente as diferenças em relação a esse quadro. A necessidade do envio de um novo quadro de referência depende de cada codec em particular. Já um codec de música precisa atuar em toda a frequência audível, 20Hz a 20kHz, e implementa conceitos da psico-acústica para tratar o sinal. Em contrapartida, um codec de voz geralmente procura identificar os tempos de silêncio e padrões na fala. Para cada sinal, vídeo, música ou voz, existem diversos codecs com abordagem de codificação-decodificação distintas. Estes são diferenciados pela largura de banda demandada, admissão ou não de perdas, complexidade computacional e a qualidade final do sinal.

Os sinais de voz de qualidade telefônica possuem dados no intervalo de 300Hz a 3400Hz, salvo algumas exceções como o G711.1 [ITU-T 2008]. Essa restrição diminui a qualidade da voz, porém esta continua inteligível e com menor quantidade de informação. O áudio de CD possui dados em toda banda média audível pelo ser humano e o áudio profissional possui banda de 20Hz a 48kHz. Perante todas essas características, existem várias abordagens possíveis para tratar o sinal de voz. As próximas sessões, de 2.1.3.1 a 2.1.3.4, abordam o princípio de funcionamento dos codecs de voz presentes neste trabalho.

2.1.3.1 G.711

O G.711 é um tradicional codec de áudio desenvolvido em 1972 para linhas telefônicas digitais, posteriormente padronizado pela ITU-T em 1988 [ITU-T 1988]. A conversão do sinal analógico para digital implementada pelo G.711 consiste no PCM (*Pulse Code Modulation*). No PCM, o sinal analógico é amostrado periodicamente, resultando no equivalente digital. A precisão da transformação está relacionada com os níveis de quantização, os quais dependem da quantidade de bits utilizados. Por exemplo, com a utilização de 16 bits, obtém-se 2^{16} níveis de quantização.

O espaçamento entre os níveis de quantização não é necessariamente linear. No caso do G.711, esse espaçamento é logarítmico. Uma quantização linear garante igual precisão ao longo de toda a onda amostrada, enquanto que uma quantização logarítmica garante maior precisão em amplitudes mais baixas. A escolha do padrão dos níveis de quantização é feita baseado na natureza do sinal a ser amostrado. Níveis de quantização espaçados logaritmicamente são interessantes para sinais que possuem maior quantidade de informação nos baixos níveis de amplitude, como o áudio. O codec G.711 possui 8kHz de frequência de amostragem com 8 bits para níveis de quantização, sendo essa logarítmica, resultando numa taxa de 64kbps. O sinal de áudio resultante possui alta qualidade, com MOS (*Mean Opinion Score*) de 4,1 [CISCO], fato que torna o codec G.711 um padrão de referência. O MOS consiste numa avaliação subjetiva da qualidade, variando numa escala de 1, ruim, a 5, excelente.

O codec G.711 pode empregar duas técnicas de codificação distintas, a Lei- μ e a Lei-A. A Lei- μ é utilizada nos Estados Unidos e Japão, enquanto que a Lei-A é utilizada nos outros países. Ambas as leis codificam os sinais entre níveis de -127 a +127, necessitando então de 8 bits. O bit mais significativo, bit da posição 1 de acordo com a figura 2.5, indica se o número é positivo, bit 1, ou negativo, bit 0, e os bits restantes representam o valor. A Lei- μ inverte todos os bits e a Lei-A somente aqueles de posição par, vide figura 2.5. A Lei- μ provê um range dinâmico da voz ligeiramente maior que a Lei-A a um custo de menor relação sinal-ruído para sinais de baixa amplitude.

2.1.3.2 G.729

O codec G.729, padronizado em 1995 pela ITU-T [ITU-T 1996a], oferece uma boa relação entre qualidade da voz e largura de banda demandada, com MOS de 3,92 [CISCO] e taxa de 8kbps. Entretanto, essa baixa taxa de transmissão resulta numa alta complexidade computacional na codificação da voz. O codec G.729 pode operar também com taxas de 6,4kbps, para os anexos D, F, H, I, C+, e 11,8kbps, de acordo com os anexos E, G, H, I, C+.

Valor Numérico	Número Bit																
	Lei A								Lei μ								
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	
127	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0
96	1	0	1	1	0	1	0	1	1	0	0	1	1	1	1	1	1
64	1	0	0	1	0	1	0	1	1	0	1	1	1	1	1	1	1
32	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	1
0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1
-32	0	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	1
-64	0	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1
-96	0	0	1	1	0	1	0	1	0	0	0	1	1	1	1	1	1
-127	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Figura 2.5: Exemplos de códigos da Lei- μ e Lei-A.

O anexo A, G.729a, propõe uma extensão do G.729 com menor complexidade computacional, porém com uma pior qualidade de voz. O G.729a possui uma frequência de amostragem de 8kHz com 16 bits para níveis de quantização. O algoritmo possui um atraso de 15ms, os quais são somados a 5ms de atraso *look-ahead*. Este último serve para obter amostras de voz suficientes para executar o algoritmo de codificação. O anexo b, G.729b, implementa a detecção de atividade da voz. Dessa forma, a transmissão do sinal de áudio ocorre somente enquanto há fala. Nos períodos de silêncio, o codec G.729b envia quadros menores que imitam um ruído de fundo, o qual gera conforto ao usuário e assegura que a ligação continua ativa. Uma outra extensão do codec G.729 é o G.729.1. Este codec permite taxas de 8-32kbps, com frequência de amostragem de 8kHz ou 16kHz. As taxas de transmissão de 8kbps e 12kbps possuem banda em frequência de 50-4000Hz e as taxas de 14kbps até 32kbps, com espaçamento de 2kbps, possuem banda de 50-7000Hz.

O algoritmo de codificação implementado no codec G.729 é o CS-ACELP (*Conjugate-Structure Algebraic-Code-Excited Linear Prediction*), o qual é baseado no LPC (*Linear Predictive Coding*). O LPC é um método para prever um sinal de voz baseado nas amostras anteriores. Esse método estima os parâmetros da voz com precisão e com alta eficiência computacional. O princípio de funcionamento do LPC consiste em dividir, filtrar (*Source Filter Model of Speech*), o sinal de voz em duas partes: a-) Função de Transferência, que contém a intensidade da voz e b-) Excitação, que descreve o tom e o som da voz. As cordas vocais atuam como a excitação e o percurso percorrido pelo ar, da corda vocal até a boca, são representados pela função de transferência. A equação 2.1 descreve matematicamente como é previsto o sinal de voz. Nessa equação, o n-ésimo sinal previsto \hat{s}_n consiste na soma ponderada de p sinais passados somado ao resíduo $e(n)$. O termo ponderador é representado pela

variável a e o sinal pela variável s . Note que a equação 2.1 é basicamente um método Auto-Regressivo (AR) [Box et al. 1994].

$$\hat{s}_n = \sum_{k=1}^{k=p} a_k s_{n-k} + e(n) \quad (2.1)$$

O atraso de *look-ahead* serve para acumular amostras de voz de forma a executar o cálculo da equação 2.1. A quantidade de amostras, variável p na equação 2.1, obtidas designa a ordem do algoritmo, a qual usualmente está entre 10 e 20. Este intervalo de ordens permite uma boa relação entre a qualidade do sinal de voz previsto e o custo computacional. O cálculo do resíduo consiste na diferença entre o sinal de voz real, s_n , e o previsto, \hat{s}_n , como mostra a equação 2.2. Tomando a transformada Z da equação 2.2, obtem-se a equação 2.3. Ajustando os termos, chega-se numa relação que engloba o sinal, $S(z)$, a função de transferência, $A(z)$, e o resíduo, $E(z)$, como mostra a equação 2.4.

$$e(n) = s_n - \hat{s}_n = s_n - \sum_{k=1}^{k=p} a_k s_{n-k} \quad (2.2)$$

$$E(z) = S(z) - \sum_{k=1}^{k=p} a_k S(z)z^{-k} = S(z)\left(1 - \sum_{k=1}^{k=p} a_k z^{-k}\right) = S(z)A(z) \quad (2.3)$$

$$S(z) = \frac{E(z)}{A(z)} \quad (2.4)$$

O espectro de $E(z)$ possui diferença entre os sons vocalizados, produzidos pelas cordas vocais, e não-vocalizados, produzidos pela boca. Os sons vocalizados são periódicos e com uma frequência fundamental, enquanto que os sons não-vocalizados não possuem frequência fundamental nem estrutura harmônica, podendo ser representado por um ruído branco. A amplitude do sinal de voz decresce com a frequência na transmissão, deixando as componentes de alta frequência mais suscetíveis à ruídos. Para deixar a resposta em frequência mais plana, aplica-se pré-ênfase, na codificação, com de-ênfase, na decodificação. O LCP precisa escolher se a codificação será baseada em sons vocalizados ou não. Esta escolha rígida acaba por prejudicar a qualidade da voz, pois várias palavras possuem ambos tipos de som. Ambientes ruidosos também afetam bastante a qualidade do LPC.

A codificação CELP (*Coded Excited Linear Prediction*) utiliza o mesmo filtro que o LPC para dividir o sinal de voz em duas partes. Entretanto, o modelo da excitação da codificação LPC é substituído por um modelo mais sofisticado. O CELP possui armazenado em memória dois *codebooks* de excitações possíveis. O *codebook* adaptativo é

responsável pela identificação do tom da voz e o *codebook* fixo corresponde às excitações possíveis. Para cada entrada, são testadas todas as excitações presentes nos *codebooks* e aquela que apresentar menor erro, calculado pela técnica de erro médio quadrático com frequência ponderada, é utilizada. A busca nos *codebooks* implica num alto custo computacional que geralmente inviabiliza a implementação do codec. Entretanto, o CELP atinge taxas de transmissão tão baixas, 4,8 kbps [Schroeder and Atal 1985], que essa metodologia de codificação da voz se torna viável.

O ACELP altera a estrutura dos *codebooks* de numérica (CELP) para algébrica. A modificação da estrutura permite buscar no *codebook* somente excitações candidatas, ou seja, aquelas que estatisticamente se adequam ao sinal de entrada. Dessa forma é possível reduzir o número de buscas e por consequência reduzir o custo computacional em até 80% [F.-K.Chen et al. 2002]. Algumas técnicas para tornar mais eficiente o processo de busca no *codebook* para o ACELP foram propostas em [Ramirez and Gerken 1998], [Ramirez and Gerken 1999] e [Ha 1999]. Tanto no CELP como no ACELP, as informações obtidas em ambos os *codebooks* e os ganhos, pré-ênfase, são enviados de maneira independente ao codificador. No CS-ACELP, as quatro informações são enviadas juntas, como mostra a equação 2.5. Nessa equação o sinal enviado C_i é composto pela soma do termo do *codebook* fixo C_{sub1} e do *codebook* adaptativo C_{sub2} . As variáveis θ_1 e θ_2 representam o ganho de pré-ênfase. Essa metodologia aumenta a robustez do algoritmo, reduz requisitos de memória e por consequência reduz a complexidade computacional.

$$C_i = \theta_1 C_{sub1} + \theta_2 C_{sub2} \quad (2.5)$$

2.1.3.3 iSAC

O codec iSAC (*internet Speech Audio Codec*) foi desenvolvido pela empresa GIPS (*Global IP Solutions*) [GIPS 2011] para aplicações VoIP. A frequência da amostragem do sinal de voz pode ser ajustada, dependendo do modo de operação. Operação em banda larga utiliza 8kHz e a operação em ultra banda larga pode utilizar 12kHz ou 16kHz. O ajuste entre os modos de operação é feito automaticamente de acordo com a qualidade da rede. O iSAC é um codec VBR e a taxa de transmissão pode variar de 10kbps a 32kbps. O intervalo de tempo entre pacotes pode ser de 30ms ou 60ms. O codec iSAC é robusto a perda de pacotes, podendo operar com qualidade em cenários com *jitter* elevado e alta perda de pacotes [Global-IP-Solutions 2010]. Pelo fato do codec iSAC ser proprietário, o algoritmo de codificação do mesmo não é público.

A empresa GIPS realizou um teste comparativo entre os principais codecs de voz presentes no mercado. Esse teste utilizou o método MUSHRA (*MUltiple Stimulus*

with Hidden Reference and Anchor) definido na recomendação da ITU-R BS.1534-1 [ITU-R 2003]. Nesse teste a qualidade da voz codificada é quantificada numa escala de 0, pobre, a 100, excelente. O resultado da comparação é apresentado na figura 2.6 [Global-IP-Solutions 2010]. As taxas de transmissão apresentadas na figura correspondem a média e o resultado possui intervalo de confiança de 95%. Analisando a figura, nota-se que para operação em ultra banda larga, o codec iSAC apresentou maior qualidade. Todavia, para operação em banda larga, o codec AMR-WB possui uma maior qualidade.

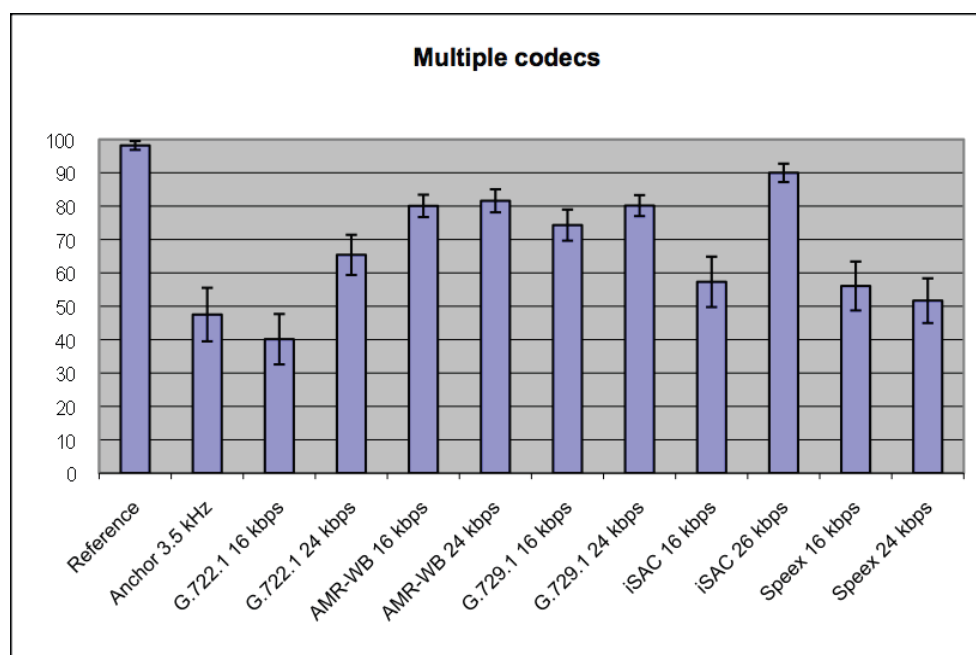


Figura 2.6: Comparação da qualidade do sinal de voz utilizando o método MUSHRA para diversos codecs presentes no mercado.

2.1.3.4 SILK

O codec SILK foi desenvolvido pela equipe do *Skype Limited* para substituir o codec SVOPC (*Sinusoidal Voice Over Packet Coder*) [Lindblom 2004]. O SILK nasceu de um ramo de desenvolvimento do codec SVOPC. As frequências de operação e as taxas de transmissão do SILK estão relacionadas na tabela 2.1. Analisando os dados dessa tabela é possível notar que a operação do SILK é VBR. O atraso entre pacotes é constante e de 20ms.

O primeiro rascunho (*draft*) do codec SILK foi submetido à IETF em Julho de 2009. Atualmente esse rascunho está na terceira versão [Vos et al. 2010] e, por essa razão, essa versão do codec é conhecida como SILK v.3. Um outro rascunho foi registrado na

Tabela 2.1: Modos de Operação do codec SILK.

	Frequência de Amostragem (kHz)	Taxa de Transmissão (kbps)
Banda Estreita	8	6-20
Banda Média	12	7-25
Banda Larga	16	8-30
Banda Ultra-Larga	24	12-40

IETF para explicar o conteúdo no RTP para o codec SILK [Spittka et al. 2009]. Em janeiro de 2009, o codec SILK foi implementado na versão 4.0 beta do Skype.

A figura 2.7 foi retirada do *datasheet* do codec SILK [Limited 2010]. Essa figura contém uma comparação de desempenho com outros codecs bastante comuns no mercado. O gráfico da esquerda ilustra a evolução do MOS em função da taxa de transmissão. Já o gráfico da esquerda mostra o decaimento do MOS em função do aumento da perda. Em ambos os casos, o codec SILK possui o melhor resultado.

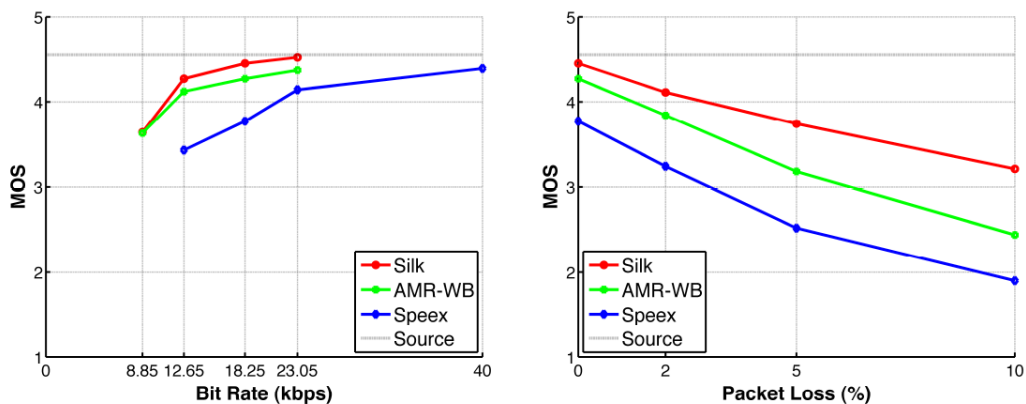


Figura 2.7: Comparação do codec SILK com outros codecs comuns no mercado.

2.2 Conceitos Estatísticos e Processos Estocásticos

O modelo proposto caracteriza o tráfego VoIP com o uso de distribuições de probabilidade, as quais foram definidas pelos testes de aderência, explicados em 2.2.3, e pelo modelo de série temporal ARMA, explicado em 2.2.7. Dentre as distribuições utilizadas, encontra-se uma distribuição de cauda pesada, que será explicada na seção 2.2.2. Trabalhos tradicionais, vide 1.2, e o modelo proposto recorrem ao conceito de auto-similaridade, abordado na seção 2.2.1. Depois de desenvolvido o modelo, a etapa subsequente consiste na validação do mesmo. O uso da função de auto-correlação ou, em inglês, *Auto-Correlation Function* (ACF), cuja explicação encontra-se em 2.2.4,

permite avaliar o comportamento dos dados testados, de forma a verificar se estes possuem dependência de longa duração e a intensidade desta. Essa avaliação é realizada pela análise do decaimento da ACF. Portanto, com a ACF é possível comparar o comportamento do tráfego real com o tráfego sintético gerado com base no modelo proposto.

2.2.1 Auto-Similaridade

O matemático George Cantor estudou o modelo auto-similar, o qual posteriormente foi aprimorado à teoria de fractais, por Benoit B. Mandelbrot [Park and Willinger 2000]. A auto-similaridade é reportada quando se identifica semelhanças entre as escalas de observação de um processo. O conjunto de Cantor, ilustrado na figura 2.8, é um exemplo de um processo auto-similar. Nesse conjunto, o segmento de reta é dividido em três partes iguais, com remoção da parte central, a cada iteração vertical. Note que se for ampliada a visão de uma iteração do conjunto de Cantor, existirão semelhanças com a forma original e com ampliações ainda menores. Um outro exemplo de processo auto-similar é o floco de neve de Koch [Kaplan and Glass 1998], que pode ser visto na figura 2.9. Esse conjunto é construído dividindo-se o segmento de reta em 3 partes iguais, removendo-se a parte do meio e inserindo 2 segmentos de reta com comprimento igual ao removido conectados em suas extremidades e na outra extremidade conectado aos segmentos de reta restantes.



Figura 2.8: Conjunto de Cantor.



Figura 2.9: Floco de Neve Koch.

A dimensão fractal de um objeto é definida pela equação 2.6. A variável N representa o número de cópias de si mesmo criados com a alteração de escala e a variável ε

indica o tamanho relativo de cada cópia em relação ao processo original. Na figura 2.8 a dimensão fractal é 0,631, pois são realizadas 2 cópias com tamanho 3 vezes menor. Já na figura 2.9, são realizadas 4 cópias com $\frac{1}{3}$ do tamanho original, resultando numa dimensão de 1,262.

$$d = \frac{\log(N)}{\log(\varepsilon)} \quad (2.6)$$

A relação matemática que descreve um processo auto-similar pode ser vista na equação 2.7 [Kaplan and Glass 1998]. O processo re-escalado X_{mt} é estatisticamente semelhante, em termos da distribuição de probabilidade, em relação ao processo original X_t , ajustado por um fator de escala m^H , sendo H o parâmetro de Hurst, explicado em 2.2.1.2.

$$X_{mt} \cong_d m^H X_t, m > 0 \quad (2.7)$$

2.2.1.1 Processos com Dependência de Longa e Curta Duração

O fenômeno da dependência de longa duração, ou em inglês, *long range dependence* (LRD) é reportado quando as amostras possuem alta correlação por um longo período de observação. Dessa forma, processos com LRD apresentam lento decaimento da função de auto-correlação, vide seção 2.2.4. A equação 2.8 apresenta a definição formal de um processo LRD [Park and Willinger 2000]. Analisando a equação 2.8 nota-se que a função de auto-correlação (ACF) possui lento decaimento, cuja intensidade é maior para valores de Hurst próximos à unidade. A variável c é uma constante positiva.

$$ACF(t) \sim ct^{2-2H}, c > 0, t > 0, 0,5 < H < 1 \quad (2.8)$$

Na seção 2.2.1.2 mostra-se que o grau de auto-similaridade e a presença de LRD são quantificados pelo parâmetro de Hurst. Um processo auto-similar geralmente apresenta dependência entre diversas escalas de tempo, fato que resulta num processo LRD. Todavia a presença de LRD em um processo auto-similar pode não existir se essas dependências existirem em poucas escalas de tempo. Já um processo LRD pode também não apresentar auto-similaridade, pois a definição deste processo está atrelada à função de auto-correlação [Park and Willinger 2000].

Um processo apresenta dependência de curta duração, ou *short range dependence* (SRD), quando o mesmo possui rápido decaimento da função de auto-correlação. Quando um processo SRD não é estacionário, possui tendências, a ACF possui lento decaimento. Dessa forma, o mesmo pode ser confundido com um LRD. Para evitar essa confusão, deve-se realizar a diferenciação do processo, o qual remove tendências, e aplicar a ACF ao resultado. A diferenciação pode ser necessária mais de uma vez. Se o

resultado apresentar rápido decaimento, então o processo original possui dependência de curto alcance.

2.2.1.2 Identificação de Um Processo Auto-Similar

A identificação de um processo auto-similar basicamente consiste em encontrar o parâmetro de Hurst. Este parâmetro foi desenvolvido pelo hidrólogo Harold Edwin Hurst quando o mesmo estudava o padrão de cheias do Rio Nilo. O parâmetro de Hurst pode não ser único para um mesmo processo, podendo variar ao longo da escalas de observação. Os tópicos abaixo mostram as interpretações do parâmetro de Hurst.

- (i) $0,5 < H < 1$: Processo Auto-Similar e LRD. Quanto mais próximo de 1, maior o grau de auto-similaridade;
- (ii) $H = 0,5$: Processo SRD e sem a presença de auto-similaridade.

Vários métodos podem ser utilizados para estimar o parâmetro de Hurst. Os mais comuns estão listados nos tópicos abaixo, dentro os quais está a Transformada de *Wavelet* que é o método mais confiável e utilizado [Abry and Veitch 1998]. Também é possível estimar o parâmetro de Hurst pela sua relação com a dimensão fractal, $d = 2 - H$.

- (i) Estatística R/S [Leland et al. 1994];
- (ii) Método da Variância [Crovella and Bestavros 1997];
- (iii) Estimador de Whittle [Abry and Veitch 1998];
- (iv) Transformada de *Wavelet* [Abry and Veitch 1998].

2.2.2 Distribuições de Cauda Pesada

Uma variável aleatória possui comportamento de cauda pesada se essa obedecer a relação da equação 2.9 [Park and Willinger 2000]. O parâmetro α é o parâmetro de forma e c é uma constante positiva. A equação 2.9 diz que existe uma probabilidade da variável aleatória x existir para valores extremos. Devido a essa probabilidade, uma característica das distribuições de cauda pesada é a variabilidade extrema exibida.

$$P(X > x) \sim cx^{-\alpha}, x \rightarrow \infty \quad (2.9)$$

A distribuição de Pareto é uma distribuição de cauda pesada, cuja função de distribuição acumulada pode ser vista na equação 2.10. A variável β representa o

parâmetro de localização. A equação 2.11 define a esperança e a equação 2.12 a variância para uma variável aleatória da distribuição de Pareto. Note que para $\alpha \leq 1$ a esperança não converge e que para $1 < \alpha \leq 2$ a variância não converge. A análise das equações 2.10, 2.11 e 2.12 permite concluir que quanto menor for o valor de α , maior é a variabilidade e maior a concentração de pontos na cauda da distribuição.

$$P(X \leq x) = 1 - \left(\frac{\beta}{x}\right)^\alpha, x > 0, \alpha > 0, \beta > 0 \quad (2.10)$$

$$E[x] = \frac{\alpha\beta}{\alpha - 1}, \alpha > 1 \quad (2.11)$$

$$V[x] = 1 - \frac{\beta^2\alpha}{(\alpha - 1)(\alpha - 2)}, \alpha > 1 \quad (2.12)$$

O modelo proposto utilizou a distribuição de Pareto tipo 2 [Kleiber and Kotz], ou distribuição Lomax, porque a distribuição de Pareto tradicional, vide equação 2.10, não admite valores menores que o parâmetro de escala. A equação 2.13 descreve a função de distribuição acumulada de Pareto tipo 2. Na figura 2.10 está ilustrado duas distribuições acumuladas de probabilidade de Pareto tipo 2 com parâmetros de forma diferentes. Note que a curva com $\alpha = 1$ demora mais para convergir porque possui maior quantidade de pontos na cauda da distribuição.

$$P(X \leq x) = 1 - \left[1 + \left(\frac{x}{\beta}\right)\right]^{-\alpha} \quad (2.13)$$

2.2.3 Testes de Aderência

Os testes de aderência permitem verificar se uma distribuição de probabilidade teórica pode ser utilizada para modelar um conjunto de dados. As seções 2.2.3.1 e 2.2.3.2 explicam os dois testes de aderência utilizados no presente trabalho.

2.2.3.1 Teste de Kolmogorov-Smirnov

O teste de Kolmogorov-Smirnov (K-S) consiste num teste de verificação de aderência, entre dois conjuntos de dados, baseado na maior diferença entre as distribuições acumuladas de probabilidade, matematicamente definido na equação 2.14. A variável D_n representa a maior diferença entre as distribuições acumuladas empírica $F_n(x)$ e teórica $F(x)$. Esse teste possui vantagens sobre testes tradicionais, como o Chi-Quadrado, pois necessita de uma menor quantidade de amostras e não depende

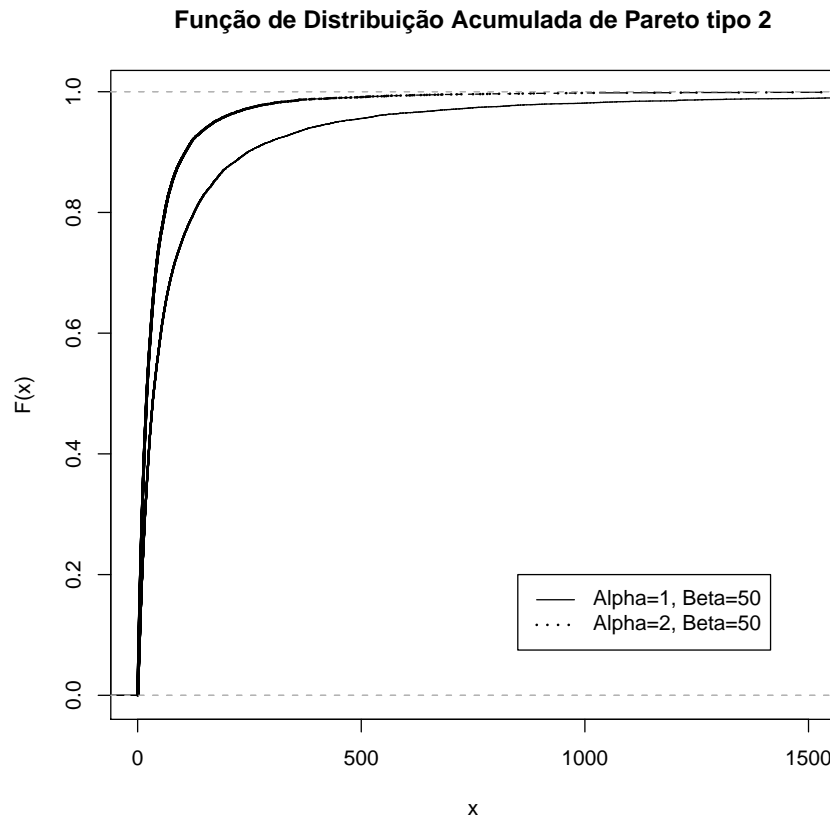


Figura 2.10: Função de Distribuição Acumulada de Pareto tipo 2.

da distribuição de probabilidade em teste, como o teste de t-Student. Todavia, por se basear na maior diferença, não se recomenda o uso do teste K-S para distribuições de cauda pesada. Isso porque essas distribuições possuem grande variabilidade, com ocorrência de eventos extremos, e podem induzir o teste ao erro.

$$D_n = \sup |F_n(x) - F(x)| \quad (2.14)$$

Para realizar o teste, deve ser obtida a máxima diferença (D_n) e escolher o nível de significância, o qual serve para definir a confiabilidade do teste. Com esses valores em mão, compara-se com os valores críticos da distribuição D. A distribuição D é tabulada em função da quantidade de amostras. Se a diferença calculada for maior que o valor crítico, descarta-se a aderência entre as distribuições. Uma outra forma de verificar aderência com o teste de Kolmogorov-Smirnov é pelo valor p . Esse valor está relacionado com o nível de significância. Se o valor p estiver abaixo do nível de significância, a aderência é rejeitada [Jain 1991]. Por exemplo, para um nível de significância de 5% (95% de confiabilidade), a aderência é rejeitada caso o valor p seja

inferior a 0,05.

2.2.3.2 Quantil-Quantil Plot (QQ-Plot)

O teste de aderência QQ-Plot é um teste gráfico no qual dois conjuntos de dados são traçados um contra o outro. Se houver aderência entre os dados, os pontos do gráfico ficarão em torno da linha $y = x$. A figura 2.11 ilustra um QQ-Plot. Nesse um conjunto de dados, distribuição empírica, é testado para ver se há aderência com a distribuição exponencial (gráfico superior). Apesar de alguns pontos não estarem sobre a linha $y = x$, fato comum em dados reais, pode-se observar uma ótima aderência entre as distribuições. Apesar de ser um teste gráfico, com inspeção visual de aderência, o QQ-Plot é uma ferramenta poderosa, pois conjuntos de dados com distribuições diferentes são facilmente identificados, como pode ser visto no gráfico inferior da figura 2.11, onde o mesmo conjunto de dados é testado com a distribuição normal.

A construção de um QQ-Plot está baseada na definição de *quantil*. Essa diz que um *quantil* é uma fração de pontos para um dado valor. Os valores dos eixos da abscissa e da ordenada da figura 2.11 representam os valores de x de uma distribuição acumulada $F(x)$. Para cada valor $x \pm$ um intervalo, são quantizados os *quantis*. Para um mesmo valor de x , se os *quantis* forem iguais, o ponto ficará em cima da reta $y = x$. Todavia, se o *quantil* da distribuição da abscissa for maior que o *quantil* da ordenada, o *quantil* do gráfico ficará acima da reta $y = x$.

2.2.4 Função de Auto-Correlação

A função auto-correlação é utilizada para verificar o nível de correlação existente entre os pontos de uma função, espaçados por um passo previamente definido. Essa função é largamente utilizada para detectar periodicidades em sinais periódicos alterados pelo ruído, descobrir a frequência fundamental, entre outros. Entretanto, a propriedade mais interessante dessa função para o presente trabalho consiste na verificação da velocidade de decaimento da correlação de uma série temporal ao longo dos *lags* (defasagem). Isso porque um rápido decaimento significa um comportamento SRD, enquanto que um decaimento lento implica num comportamento LRD.

A definição da função de auto-correlação pode ser vista na equação 2.15. Essa equação pode ser aplicada a qualquer conjunto de dados, inclusive dados não estacionários. O termo X_t representa o ponto de referência e o termo X_s representa o dado analisado. O dividendo da equação 2.15 realiza o cálculo do valor esperado $E[]$ para o produto da diferença entre os termos X_t e X_s e suas respectivas médias μ_t e μ_s . Quanto maior for o valor do dividendo, maior é o grau de correlação do ponto

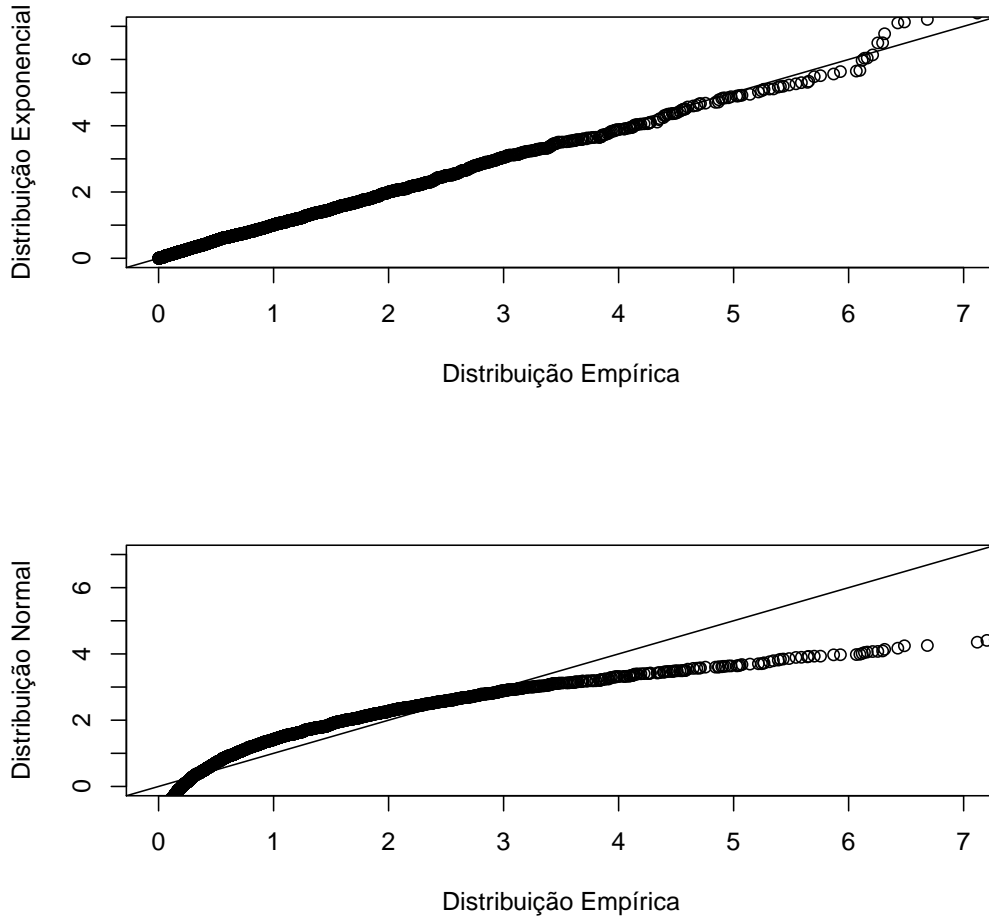


Figura 2.11: Teste de Aderência QQ-Plot

analisado ao ponto de referência. O termo divisor serve para normalizar o resultado no intervalo $[-1, 1]$. Esse termo é composto pelo produto dos desvios padrões dos dois intervalos de tempo, σ_t e σ_s . Para um processo estacionário de segunda ordem, aquele no qual a média e a variância são independentes do tempo, a definição da ACF pode ser simplificada, resultando na equação 2.16. Nessa equação a média e o desvio padrão possuem o mesmo valor nos tempos t e $t + \tau$.

$$ACF(s, t) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s} \quad (2.15)$$

$$ACF(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{\sigma^2} \quad (2.16)$$

2.2.5 Modelo Auto Regressivo

O modelo auto-regressivo (AR) serve para prever amostras futuras baseado na ponderação das amostras passadas. A equação 2.17 mostra a definição do modelo AR [Box et al. 1994]. A variável ϕ consiste no termo ponderador. O termo a_t , chamado de resíduo, representa um ruído branco, que segue a distribuição Normal com média zero e variância qualquer. Na implementação do modelo AR, o resíduo é gerado com a variância dos dados existentes. A ordem do modelo AR está relacionada com a quantidade de termos anteriores x_{t-p} para prever o termo futuro \hat{x}_t . O modelo AR de ordem 1, AR(1), utiliza apenas um termo anterior, $x_t = \phi_1 x_{t-1} + a_t$. O modelo AR(2) utiliza dois termos e assim por diante. Métodos para estimar os parâmetros do modelo AR estão descritos em [Box et al. 1994].

$$\hat{x}_t = \sum_{p=1}^{p=n} \phi_p x_{t-p} + a_t \quad (2.17)$$

A figura 2.12 ilustra uma análise do resíduo, cujo resultado comprova que o modelo AR definido descreve corretamente o evento estudado. O gráfico superior direito mostra que os resíduos não apresentam tendência são estacionários. Não deve haver correlação entre os pontos, fato que pode ser investigado com o uso da ACF, vide gráfico superior direito. Para verificar se o ruído segue a distribuição Normal, pode ser utilizado o QQ-Plot, ilustrado no gráfico inferior esquerdo. O gráfico inferior direito apresenta o teste de Ljung-Box, o qual avalia o grau de correlação e a aleatoriedade dos pontos [Ljung and Box 1998]. Quanto mais próximo da unidade os pontos estiverem, melhor o resultado.

2.2.6 Modelo Média Móvel

O modelo média móvel (MA - *Moving Average*) prevê amostras futuras com base na soma ponderada dos resíduos. A definição formal do modelo MA encontra-se na equação 2.18. Nessa, o termo ϕ_p serve para ponderar os resíduos passados a_{t-p} . A média e o resíduo a_t são somados ao resultado do somatório da ponderação dos resíduos passados.

$$\hat{x}_t = \mu + a_t + \sum_{p=1}^{p=n} \phi_p a_{t-p}, \mu > 0 \quad (2.18)$$

As técnicas para a validação do modelo MA são as mesmas utilizadas para o modelo AR. A ordem do modelo MA está relacionada com a quantidade de resíduos passados utilizados, MA(1) para ordem um, MA(2) para ordem dois, e assim sucessivamente.

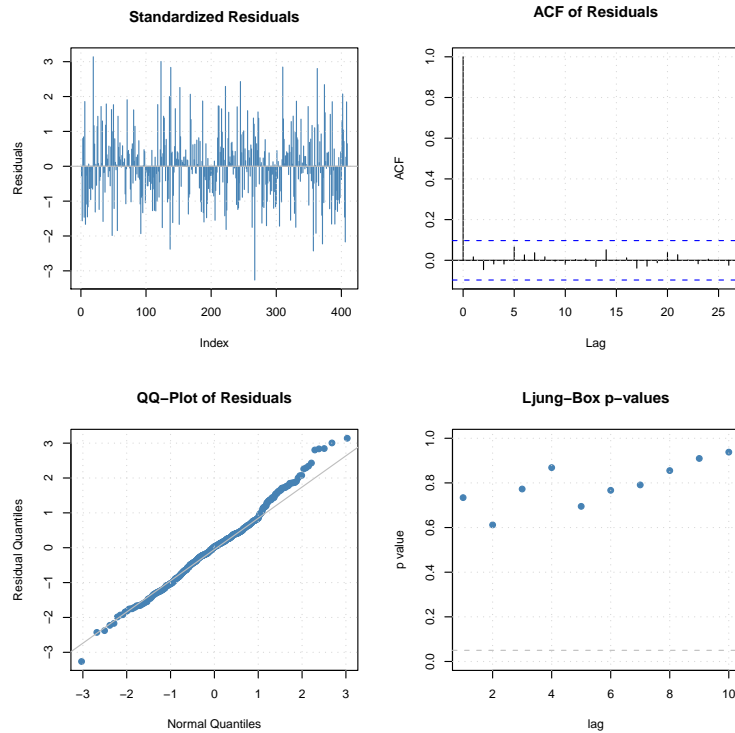


Figura 2.12: Validação do Modelo AR - Análise do Ruído.

Em [Box et al. 1994] estão descritas técnicas para estimar os parâmetros de ponderação (ϕ).

2.2.7 Modelo ARMA

O modelo ARMA advém dos modelos AR e MA, cuja definição matemática é a soma desses modelos, como mostra a equação 2.19. Os termos ϕ_p e θ_k ponderam os termos x_{t-p} e os resíduos a_{t-k} passados respectivamente. A média μ foi omitida porque a mesma está contida nos termos x_{t-p} . O termo previsto \hat{x}_t consiste na soma do modelo AR, primeiro somatório, com o modelo MA, segundo somatório.

$$\hat{x}_t = a_t + \sum_{p=1}^{p=n} \phi_p x_{t-p} + \sum_{k=1}^{k=n} \theta_k a_{t-k}, \mu > 0 \quad (2.19)$$

A ordem do modelo ARMA é separada para cada modelo que o compõe. Por exemplo, ARMA(2,3) consiste na soma do modelo AR(2) e do MA(3). Um modelo ARMA (2,0) não possui a parte referente ao modelo MA. Os métodos para estimação dos parâmetros ponderadores são mais complexos para o ARMA. Estes envolvem a resolução de equação não lineares [Box et al. 1994]. O processo para validação do modelo ARMA segue o mesmo padrão daquele utilizado para os modelos AR e MA.

2.3 Modelos de Tráfego

Vários modelos de tráfego de telecomunicações foram desenvolvidos nos últimos anos, como visto na seção 1.2. Esse estudo intenso sobre o assunto ressalta a importância do mesmo e a necessidade de se estabelecer um modelo eficiente e confiável para caracterizar o tráfego de telecomunicações. Nas seções 2.3.1, 2.3.2 e 2.3.3 segue uma explicação dos modelos abordados ao longo do presente documento.

2.3.1 Cadeias de Markov

A figura 2.13 ilustra a concepção de uma cadeia de Markov. Nesta figura estão dispostos os dois estados possíveis de uma chamada telefônica: a) ocupado e b) livre. Dado um estado qualquer, há um tempo de permanência no mesmo, o qual pode ser descrito por uma distribuição de probabilidade, e as possíveis mudanças de estados. Os possíveis estados de destino são sorteados de acordo com a probabilidade de transição.

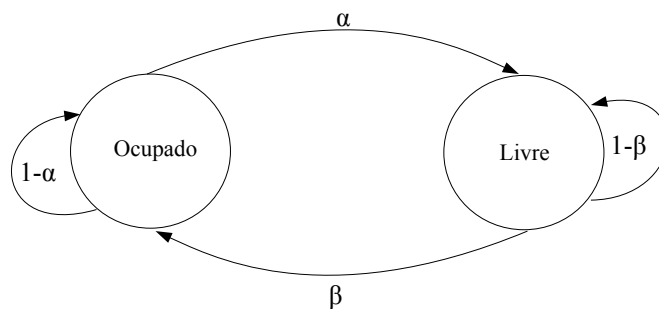


Figura 2.13: Estados de uma Chamada Telefônica - Cadeia de Markov.

A tabela 2.2 descreve todas as probabilidades de transição entre estados. Note que, para o caso da figura 2.13, a probabilidade de um estado permanecer nele mesmo é o complemento da probabilidade do mesmo trocar de estado. Uma propriedade interessante da matriz de probabilidade de estados é que a soma dos termos de uma linha tem que ser unitário, pois uma linha contém todas as mudanças possíveis para cada estado. As cadeias de Markov consistem num modelo sem memória, pois a mudança de estado depende unicamente do estado presente.

Tabela 2.2: Matriz de Probabilidade de Estados - Markov.

	Ocupada	Livre
Ocupada	$1-\alpha$	α
Livre	β	$1-\beta$

2.3.2 Modelo ON-OFF

O modelo ON-OFF é um modelo simples e bastante utilizado para geração de tráfego [Adas 1997]. A estrutura do modelo ON-OFF, vide figura 2.14, consiste no estado ON, ligado, e OFF, desligado. O tempo de permanência num estado é descrito por uma distribuição de probabilidade. Se pelo menos um estado for caracterizado por uma distribuição de cauda pesada, várias fontes de tráfego ON-OFF paralelas resultam num tráfego agregado auto-similar. A relação entre o grau de auto-similaridade, parâmetro de Hurst, e o parâmetro de forma da distribuição de Pareto (α) é dada por: $H = \frac{3-\alpha}{2}$ [Crovella et al. 1998].

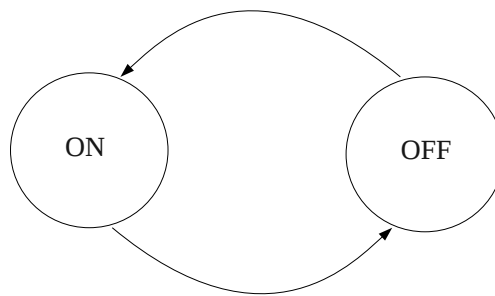


Figura 2.14: Modelo ON-OFF.

2.3.3 Modelo SURGE

O modelo SURGE (*Scalable URL Reference Generator*) é um modelo de desempenho para servidores *Web*. A abordagem de [Barford and Crovella 1998] no desenvolvimento do modelo SURGE foi bastante interessante, pois nesse modelo podem constar o conteúdo do tráfego e o comportamento do usuário que o gera. Dessa forma, podem ser gerados tráfegos bastante aproximados com o tráfego real. O funcionamento do modelo SURGE é ditado pelo comportamento das variáveis abaixo:

- (i) *Tempo OFF*: Tempo de inatividade, usuário pensando. Usualmente modelado por uma distribuição de cauda pesada.
- (ii) *Tamanho dos Arquivos*: Usualmente modelado por uma distribuição de Pareto;
- (iii) *Número de Referências*: Quantidade de arquivos transmitidos pelo usuário durante uma sessão, modelado pela distribuição de Pareto;
- (iv) *Tempo active-off*: Intervalo de tempo entre os arquivos transmitidos, modelado pela distribuição de Weibull;

- (v) *Popularidade*: Descreve o número relativo de acessos a um arquivo individual. Essa variável em servidores *Web* geralmente segue a Lei de Zipf. Essa lei diz que se os arquivos forem ordenados do mais para o menos popular, o número de referências a um arquivo P , tende a ser inversamente proporcional a sua posição r , $P = kr^{-1}$, sendo k uma constante qualquer.
- (vi) *Localidade Temporal*: Essa variável diz que se um arquivo for requisitado, a probabilidade deste ser novamente requisitado no futuro aumenta. Essa variável é modelada a partir de uma análise em pilha, e a distância entre as requisições de um mesmo arquivo é geralmente descrita por uma distribuição log-normal.

Para gerar o tráfego de servidores a partir do SURGE, seja HTTP 1.0 ou 1.1, devem ser ativadas várias fontes de tráfego funcionando paralelamente. O tráfego resultante, agregado, se assemelha muito com o tráfego real. Estudos quantitativos sobre o modelo SURGE podem ser encontrados em [Barford 2001].

Os conceitos apresentados nesse capítulo servem para facilitar o entendimento do modelo proposto no capítulo 3. Foram abordados conceitos da telefonia VoIP e informações sobre os codecs utilizados no estudo. Os conceitos de processos estocásticos e testes de aderência utilizados no modelo proposto também foram explicados. Por fim, foram apresentados os três principais modelos de tráfego utilizados em VoIP, sendo que o modelo SURGE foi utilizado como base para o modelo do presente trabalho.

Capítulo 3

Descrição do Modelo

3.1 Visão Geral

A idéia geral do modelo proposto foi baseada no modelo SURGE [Barford and Crovella 1998]. O SURGE difere de outros modelos pois o desenvolvimento deste foi baseado no comportamento do usuário e na natureza dos dados. A geração de tráfego do modelo SURGE utiliza o conceito de *User Equivalents* (UEs). Um UE é definido como um único processo, usuário, em um laço infinito que alterna entre requisições de arquivos Web e a condição inativa. Diferentemente do SURGE, o modelo proposto não precisa definir o número de usuários ativos no sistema. A geração de tráfego se baseia no tempo de retenção das chamadas e no intervalo de tempo entre as mesmas, os quais definem a condicção do usuário. Esses tempos podem ser obtidos a partir da análise das mensagens de sinalização VoIP. O modelo proposto utiliza cinco variáveis, as quais estão relacionadas nos tópicos abaixo.

- (i) Tempo de Retenção da Chamada (TRC);
- (ii) Intervalo de Tempo entre Chamadas (ITC);
- (iii) Tamanho do Pacote (TDP);
- (iv) Intervalo de Tempo entre Pacotes (ITP);
- (v) Erro do Tempo entre Pacotes (ETP).

Na figura 3.1 é possível analisar os significados das variáveis do modelo proposto. As variáveis TRC e ITC caracterizam o comportamento do usuário, enquanto que as variáveis TDP, ITP e ETP modelam a natureza dos dados. A variável ITC permite

estabelecer a frequência com que novas chamadas são geradas. A duração dessas chamadas é descrita pelo *Tempo de Retenção da Chamada*. Enquanto a chamada está ativa, existe a geração de pacotes, a qual é caracterizada pelo *Tamanho do Pacote* e pelo *Intervalo de Tempo entre Pacotes*. Todavia, a variável ITP sofre variações devido ao processamento de máquina e formação de filas internas. Dessa forma, foi necessário modelar o *Erro do Tempo entre Pacotes* para realizar um modelo mais fiel ao tráfego real. As variáveis para a geração de pacotes estão fortemente relacionadas com o codec em uso. A caracterização das cinco variáveis do modelo descrevem o comportamento do usuário da telefonia VoIP. O tráfego agregado será produzido pela superposição do tráfego de diversas fontes (usuários).

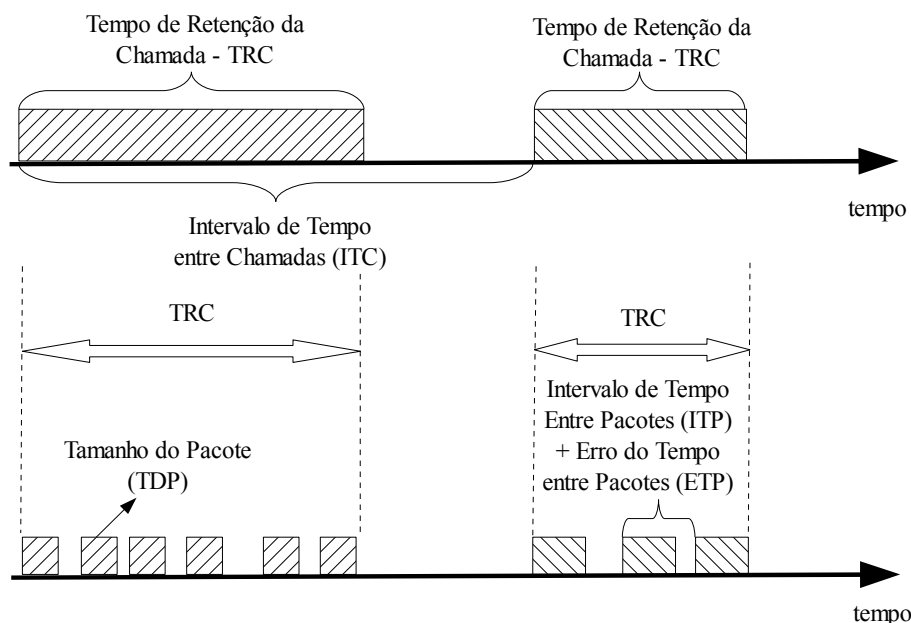


Figura 3.1: Tráfego de Pacotes de duas chamadas diferentes com a Identificação das Variáveis do Modelo Proposto.

Na tabela 3.1 encontram-se informações sobre os dados utilizados para modelar o comportamento do usuário. A coleta foi realizada no horário de maior movimento (HMM) da rede e quando o tráfego encontrava-se estável. Os dados foram coletados na HMM porque as redes de telecomunicações são projetadas para suportar o maior nível de tráfego. A estabilidade do mesmo é necessária para modelar exatamente o HMM, e não períodos de transições da intensidade do tráfego.

Tabela 3.1: Dados Utilizados na Caracterização do Comportamento do Usuário.

Coleta 1	17/09/2007
Coleta 2	18/09/2007
Coleta 3	19/09/2007
Coleta 4	20/09/2007

Os dados utilizados para modelar o comportamento dos dados, tráfego de pacotes, foram gerados em um ambiente laboratorial. Nesse ambiente é possível evitar que os dados sofram distorções, como atrasos e perdas. Essas distorções podem fazer que o modelo não caracterize unicamente o codec, mas sim um comportamento particular da rede. Para cada codec estudado foram geradas 64 chamadas. Esse número de chamadas corresponde à quantidade de arquivos de áudio utilizados da ITU-T [ITU-T]. Cada chamada foi modelada separadamente e um modelo geral para cada codec em estudo foi proposto.

3.1.1 Tempo de Retenção da Chamada (TRC)

O *Tempo de Retenção da Chamada* foi obtido com a identificação do intervalo de tempo entre as mensagens SIP INVITE e BYE referente a uma mesma chamada. As chamadas que não foram finalizadas graciosamente, com um BYE, não foram analisadas. Para identificar uma chamada foram analisados os endereços IPs das partes envolvidas e o número de sequência dos pacotes gerados. Esse último parâmetro foi necessário devido à presença de conferências nas chamadas analisadas.

A figura 3.2 mostra que existem chamadas com duração extrema, evidenciando um comportamento de cauda pesada. Já a figura 3.3 mostra o gráfico da distribuição complementar em escala logarítmica. Nesse tipo de gráfico, distribuições de curta e média duração apresentam uma reta aproximadamente paralela ao eixo y no decaimento da função complementar. Note que na figura 3.3 a reta apresenta um decaimento lento, confirmando a presença de cauda pesada nos dados.

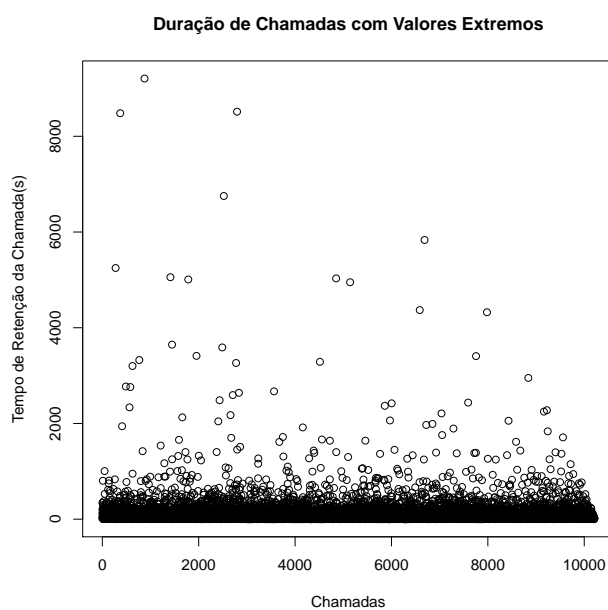


Figura 3.2: Tempo de Retenção das Chamadas Coletadas no *Backbone* da Operadora de Telecomunicações.

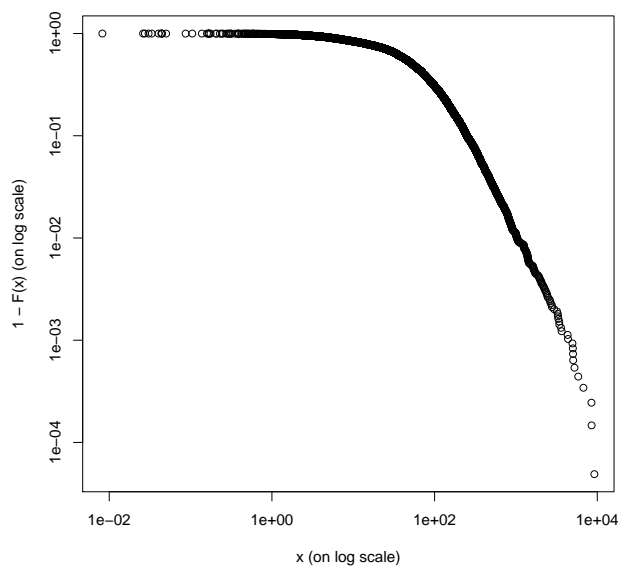


Figura 3.3: Gráfico da Distribuição Complementar em Escala Logarítmica do *Tempo de Retenção de Chamadas*.

Inicialmente foi utilizada a distribuição de Pareto para modelar os dados. Todavia, a distribuição de Pareto tradicional, definida na equação 2.10, não foi utilizada porque

essa não representa a realidade pelo fato de não gerar valores inferiores ao parâmetro de escala (β). A distribuição de Pareto tipo 2 resolve esse problema, proporcionando uma boa caracterização do *Tempo de Retenção de Chamada*. A eficiência dessa caracterização é comprovada pelo QQ-Plot dos dados empíricos e da distribuição de Pareto tipo 2, ilustrado na figura 3.4. A partir do valor 1500 os dados empíricos passam a não apresentar um cauda tão pesada, fato evidenciado nas figuras 3.3 e 3.4. O teste de Kolmogorov-Smirnov, seção 2.2.3.1, não pode ser aplicado, pois o mesmo não é recomendado para distribuições de cauda pesada devido à alta variabilidade dos dados.

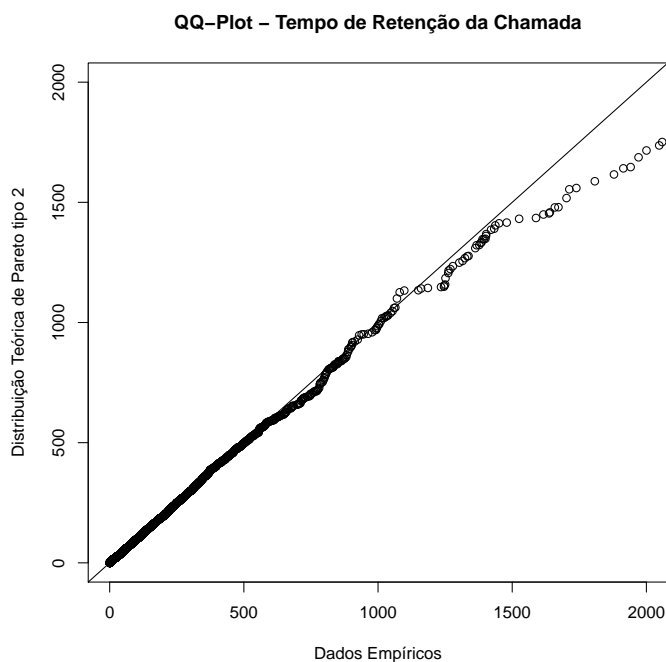


Figura 3.4: *Quantile-Quantile Plot* entre os Dados Empíricos e a Distribuição Teórica de Pareto tipo 2.

Os parâmetros α e β , que caracterizam a distribuição de Pareto tipo 2, foram obtidos com o uso do método da máxima verossimilhança [Aldrich 1997]¹. Os valores encontrados foram de 2,16 para α e 63,43 para β . Na distribuição de Pareto, a variância não converge para $1 < \alpha \leq 2$. Pelo fato de α estar próximo da região de não convergência da variância, foi identificada uma grande variabilidade no tempo de duração das chamadas. O tempo médio de TRC foi de 143,7 segundos com um desvio padrão de 490,4. A caracterização do *Tempo de Retenção de Chamadas* foi feita com a análise

¹Esse método parte do princípio que a distribuição de probabilidade é conhecida. O método consiste em obter os parâmetros da distribuição de probabilidade que maximizem a semelhança entre os dados gerados e os reais

do comportamento de 10.210 sessões VoIP.

3.1.2 Intervalo de Tempo entre Chamadas (ITC)

A identificação do *Intervalo de Tempo entre Chamadas* foi feita pela quantificação do intervalo de tempo entre as mensagens SIP INVITE sucessivas. De acordo com [Flood 1997], o *Intervalo de Tempo entre Chamadas* da telefonia comutada é descrito por uma distribuição de decaimento exponencial. O teste de Kolmogorov-Smirnov foi realizado para comparar os dados empíricos com a distribuição exponencial. Foi obtido um valor de p de 0,84. A rejeição da hipótese de aderência ocorre quando esse valor é inferior a 0,05, como explicado na seção 2.2.3.1. Portanto, o resultado da telefonia tradicional para o *Intervalo de Tempo entre Chamadas* também é válido para a telefonia VoIP. A figura 3.5 ilustra graficamente, com o uso do QQ-Plot, a aderência dos dados empíricos com os dados reais. A tabela 3.2 contém a média e o desvio padrão dos dados utilizados para modelar a variável ITC. A distribuição exponencial é parametrizada somente pela média².

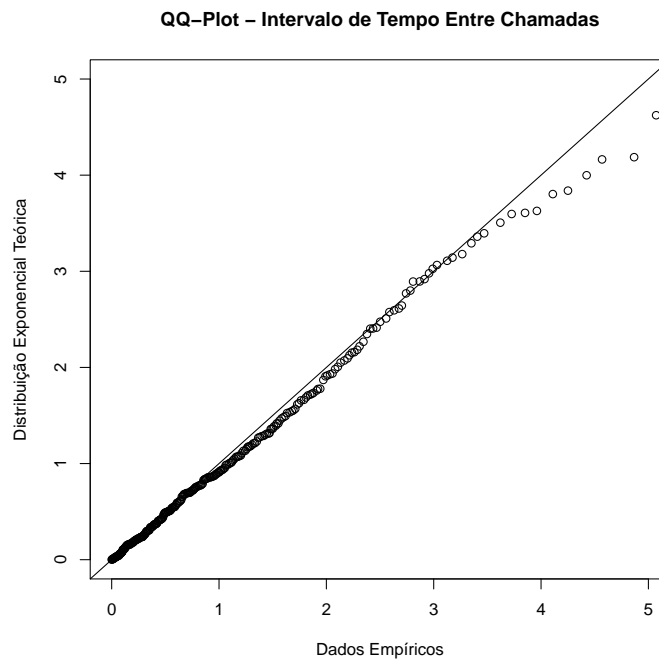


Figura 3.5: Quantile-Quantile Plot entre os Dados Empíricos e a Distribuição Teórica Exponencial.

² $F(x)=1-e^{-x/\mu}$, $x > 0$ e μ sendo a média

Tabela 3.2: Sumário das Estatísticas sobre o *Intervalo de Tempo entre Chamadas*.

Média	1,127s
Desvio Padrão	1,16

3.1.3 Tamanho do Pacote(TDP)

A variável TDP foi caracterizada de maneira distinta para cada codec estudado. Os resultados obtidos serão apresentados separadamente, organizados por codec. Os tamanhos dos pacotes apresentados nesse tópico já contêm os cabeçalhos Ethernet (26 bytes), IP (20 bytes) e UDP (8 bytes). O cabeçalho do RTP (12 bytes) não foi considerado. Isso porque o Skype [Limited 2011], onde parte dos dados foram gerados, não utiliza o SIP como protocolo de sinalização em ligações de PC para PC. Dessa forma, o software que captura o tráfego de pacotes, Wireshark, não consegue descobrir qual protocolo está rodando sobre o UDP. O fato de não utilizar o cabeçalho do RTP não influencia na qualidade do modelo, pois o mesmo possui um valor fixo. Caso deseje-se utilizar o cabeçalho do RTP, basta adicionar o valor de 12 bytes ao valor da variável TDP. O modelo apresentado para caracterizar o *Tamanho do Pacote* leva em consideração que não há perda de pacotes na rede. Esse fator é importante porque alguns codecs, como o iSAC e o SILK, adaptam-se às condições da rede e diminuem a qualidade da codificação caso a rede apresente perdas [Huang et al. 2010]. Essa alteração da qualidade em função das perdas altera a característica do tráfego de dados. Detalhes sobre a geração dos dados para modelar o comportamento do codec estão descritos na seção 1.4.1.

3.1.3.1 G.711

Os dados utilizados para modelar o *Tamanho do Pacote* o codec G.711 foram gerados no software Ekiga [Sandras 2010]. Escolheu-se por utilizar o G.711 Lei-A devido ao fato dessa versão ser utilizada na maior parte do mundo, inclusive no Brasil. Por ser um codec CBR, a variável TDP é constante e o valor desta é de 214 bytes.

3.1.3.2 G.729

O comportamento do tamanho dos pacotes para o codec G.729 está ilustrado na figura 3.6. Nessa figura é possível ver que a maioria dos pacotes possuem 70 bytes, mas periodicamente são enviados pacotes de 74 bytes. Essa periodicidade possui um intervalo de 100 pacotes, ou 2s, como mostra o diagrama ilustrado na figura 3.7.

A taxa de transmissão de 6,4kbps produz um pacote de 16 bytes, os quais somados ao 54 bytes do cabeçalho resulta em 70 bytes. Dessa forma, conclui-se que o Skype utilizou 6,4kbps na comunicação com o G.729. Para a taxa de transmissão de 8kbps mais o cabeçalho, o tamanho do pacote fica 74 bytes. Possivelmente, o Skype implementa uma transmissão paralela para transmitir informações de controle.

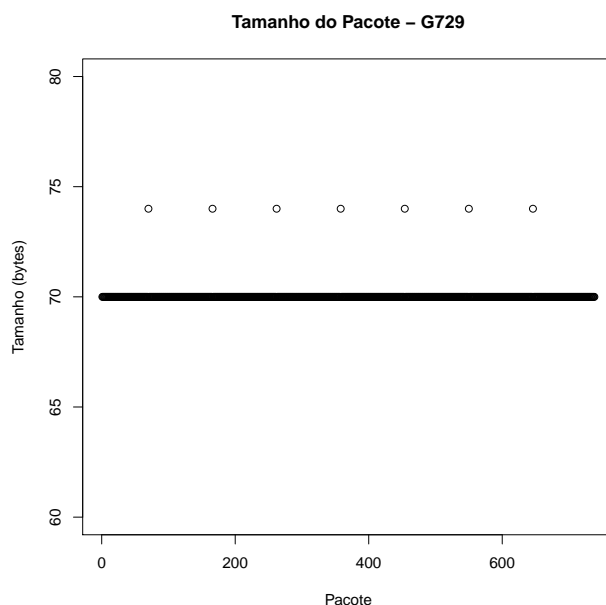


Figura 3.6: Comportamento do *Tamanho do Pacote* para o codec G729.

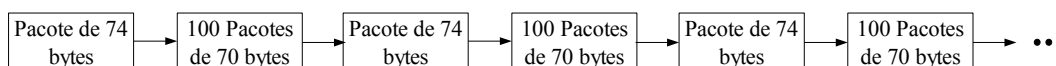


Figura 3.7: Periodicidade dos Pacotes para o codec G729.

3.1.3.3 iSAC

O *Tamanho do Pacote* para o codec iSAC varia ao longo de uma chamada, fato que o caracteriza como um codec VBR. Essa variação está ilustrada na figura 3.8.

A primeira abordagem para modelar o comportamento da figura 3.8 foi utilizar o método AR. Esse método pode ser utilizado, pois os dados são estacionários. O modelo AR não proveu um bom resultado, assim como o MA. Todavia, o modelo ARMA proveu um ótimo resultado. Foram testadas diversas combinações de coeficientes e a

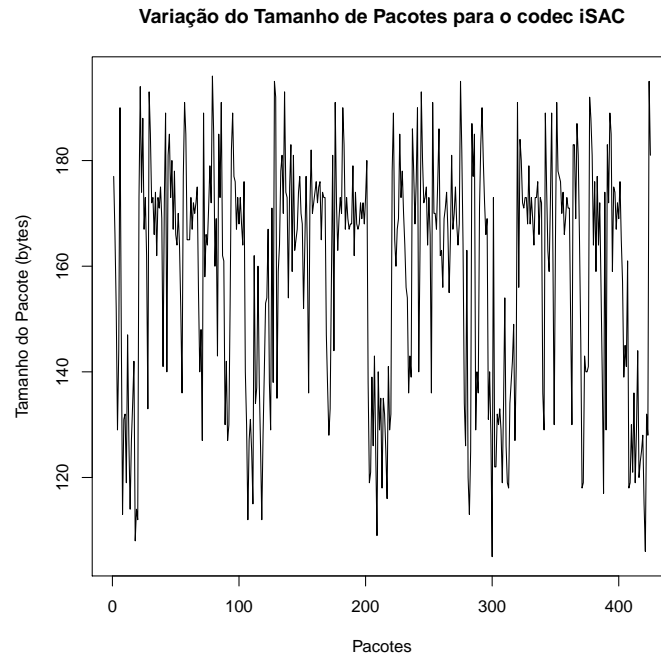


Figura 3.8: Varição do *Tamanho do Pacote* para o codec iSAC ao longo de uma Chamada VoIP.

que apresentou resultado mais confiável, e estável, foi o ARMA(2,1). A tabela 3.3 mostra os valores médios obtidos para os coeficientes do modelo ARMA(2,1).

Tabela 3.3: Coeficientes para o modelo ARMA(2,1) do codec iSAC.

Parâmetro AR 1	Parâmetro AR 2	Parâmetro MA 1
1,117	-0,190	-0,631

A figura 3.9 contém uma análise dos resíduos do modelo ARMA(2,1). O gráfico superior esquerdo mostra o comportamento dos resíduos, os quais não apresentam tendências. Já o gráfico do canto superior direito mostra a correlação entre os pontos. Quanto mais aleatórios e descorrelacionados forem os resíduos, maior é a qualidade do modelo. O teste de Ljung-Box avalia o grau de correlação e a aleatoriedade dos resíduos. O resultado desse teste está no gráfico inferior direito. Quanto mais próximo de 1 os pontos estiverem, melhor o modelo. A aleatoriedade dos resíduos deve ser descrita pela distribuição Normal³, com média 0 e desvio padrão s . O gráfico inferior esquerdo mostra o QQ-Plot dos resíduos com a distribuição Normal. Avaliando essa análise, concluí-se que o modelo adequou-se bem aos dados originais.

³ $f(x, \mu, \sigma) = (1/\sqrt{2\pi\sigma^2})e^{-(x-\mu)^2/(2\sigma^2)}$, com $\mu > 0, \sigma > 0$

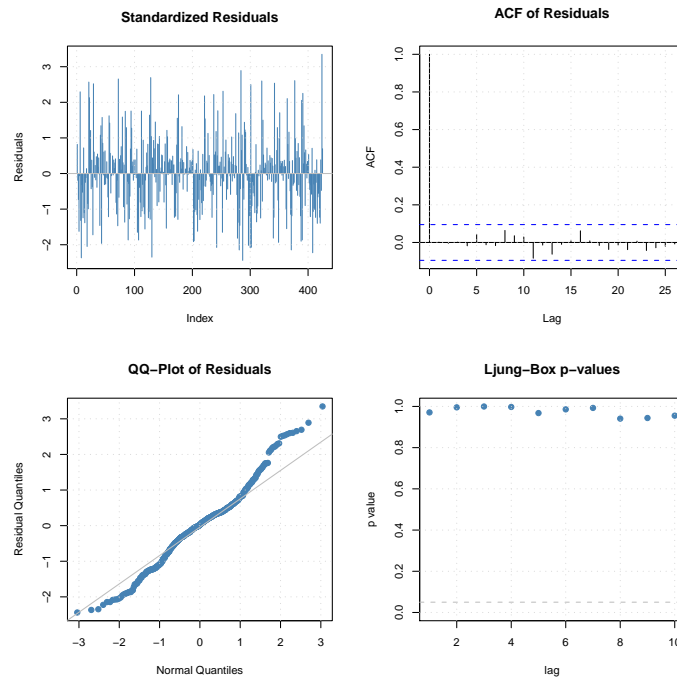


Figura 3.9: Análise do Resíduo do modelo ARMA(2,1) para o codec iSAC.

Uma outra forma de comprovar a eficiência do modelo é pela comparação das estruturas de auto-correlação dos dados reais e previstos. A figura 3.10 mostra essa comparação. Note que a ACF do modelo ARMA(2,1) teve um comportamento semelhante à ACF dos dados reais. Dessa forma, o modelo proposto mantém as características dos dados reais. Para os dados reais a média foi de 158,85 bytes e o desvio padrão de 22,02, enquanto que para os dados gerados pelo modelo ARMA(2,1) a média foi de 158,88 bytes e o desvio padrão de 26,20.

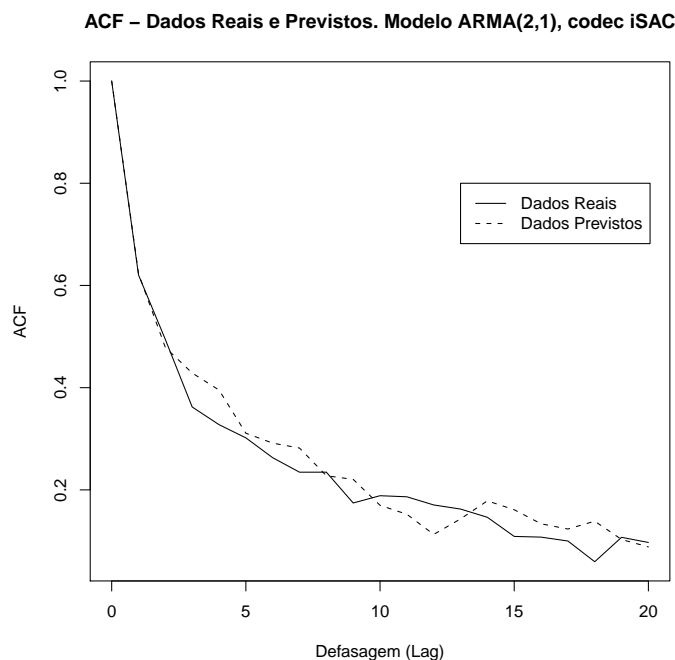


Figura 3.10: Comparação das Estruturas de Auto-Correlação para os Dados Reais e os Dados Gerados pelo modelo ARMA(2,1).

3.1.3.4 SILK v.3

O codec SILK possui um comportamento VBR. O modelo ARMA(2,1) também se adequou bem à esse codec. Porém, para o SILK, foram detectados dois conjuntos de parâmetros ARMA(2,1) diferentes. A tabela 3.4 contém um sumário dos parâmetros. O conjunto 1 possui maior ocorrência que o conjunto 2. Essa análise foi feita a partir de 64 conversas telefônicas pelo Skype 4.2. Nessas conversas o áudio utilizado foi proveniente dos arquivos da ITU-T [ITU-T].

Tabela 3.4: Coeficientes para os modelos ARMA(2,1) do codec SILK.

	Parâmetro AR 1	Parâmetro AR 2	Parâmetro MA 1	Ocorrência
Conjunto 1	1,281	-0,332	-0,600	76,56%
Conjunto 2	0,240	0,462	0,548	23,44%

A análise dos resíduos para o conjunto 1 pode ser vista na figura 3.11. Nessa figura é possível verificar nos gráficos superiores esquerdo e direito, nessa ordem, a estacionariedade dos resíduos e decorrelacionamento entre os mesmos. O teste de Ljung-Box foi aprovado, conforme pode ser visto no gráfico inferior direito. O QQ-Plot, gráfico inferior esquerdo, mostra aderência com a distribuição Normal.

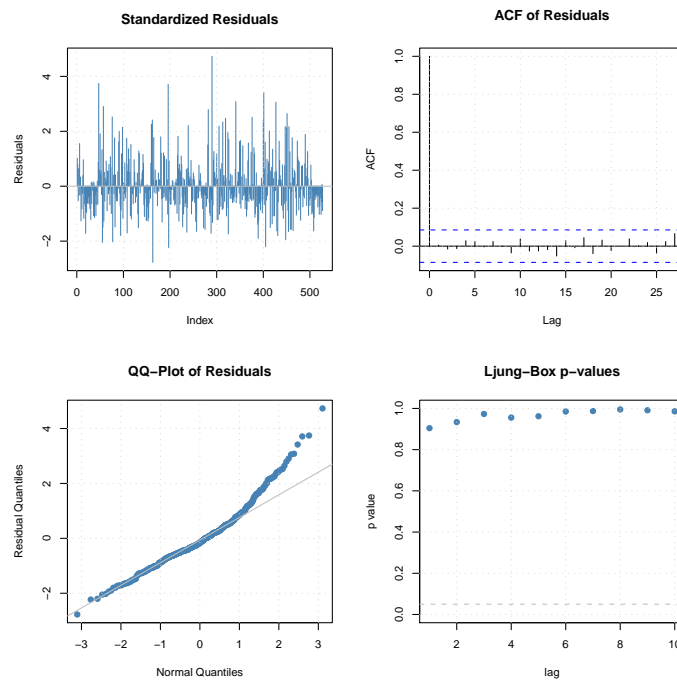


Figura 3.11: Análise do Resíduo do Conjunto 1 para o codec SILK.

O conjunto 2 também aderiu satisfatoriamente aos dados. A figura 3.12 possui as análises que comprovam essa aderência. Os resíduos foram estacionários, gráfico superior esquerdo, com baixa correlação, gráfico superior direito, aderentes à distribuição Normal, gráfico inferior esquerdo, e aprovados no teste de Ljung-Box, gráfico inferior direito.

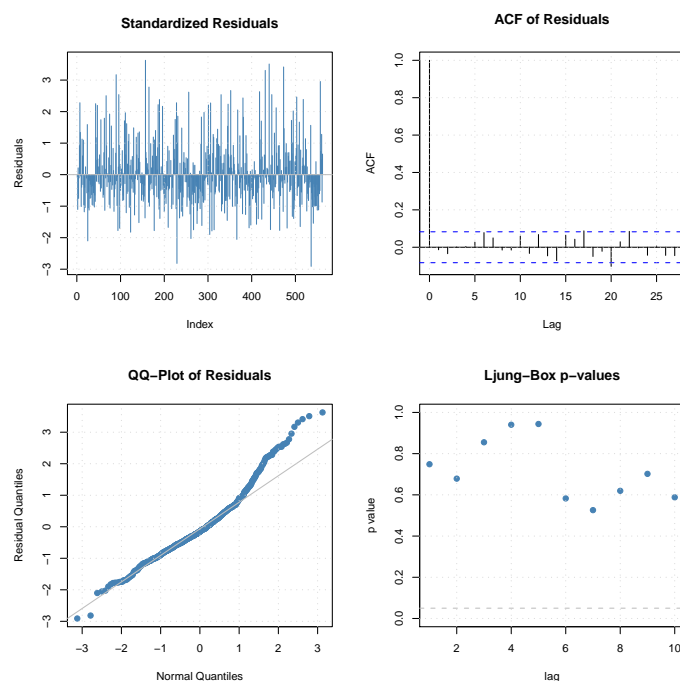


Figura 3.12: Análise do Resíduo do Conjunto 2 para o codec SILK.

As estruturas de auto-correlação dos dados gerados pelos conjuntos 1 e 2 possuem comportamento semelhante aos dados reais, como mostram as figuras 3.13 e 3.14, respectivamente. Para o conjunto 1, os dados reais apresentaram média em torno de 139,50 e desvio padrão de 18,19, enquanto que os dados gerados pelo modelo tiveram média de 139,8324 e desvio padrão de 26,54. No conjunto 2 as médias foram 138,91 e 138,75 para os dados reais e simulados, respectivamente. Já os desvios padrões foram 15,80 e 22,11 para a série capturada e gerada, nessa ordem.

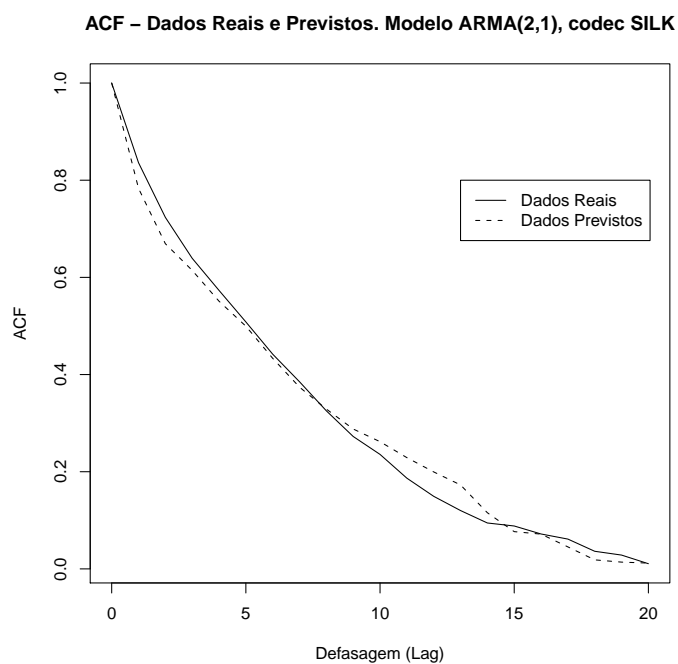


Figura 3.13: Comparação das Estruturas de Auto-Correlação para os Dados Reais e os Dados Gerados pelo modelo ARMA(2,1) com Parâmetros do Conjunto 1.

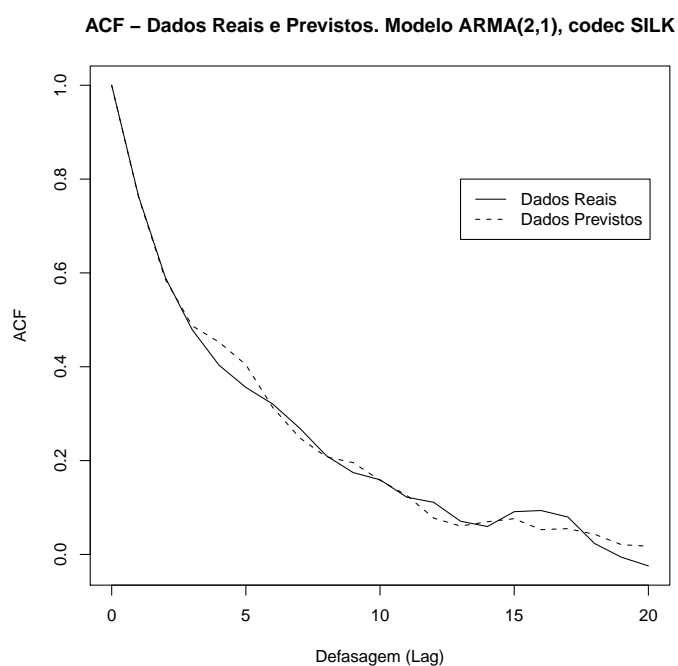


Figura 3.14: Comparação das Estruturas de Auto-Correlação para os Dados Reais e os Dados Gerados pelo modelo ARMA(2,1) com Parâmetros do Conjunto 2.

3.1.4 Intervalo de Tempo entre Pacotes (ITP)

Pelo fato da variável ITP ser dependente do codec, os resultados serão apresentados separadamente.

3.1.4.1 G.711

O tráfego para análise do codec G.711 lei-A foi gerado no Ekiga. Este *softphone* utiliza o protocolo SIP para sinalização, o qual informa no INVITE, campo do SDP, o codec a ser utilizado na comunicação. Dessa forma, o Wireshark decodifica e permite ter acesso ao conteúdo do protocolo RTP.

Para obter o valor do *Intervalo de Tempo entre Pacotes* foi analisando o campo *timestamp* do pacote RTP. Observou-se um valor fixo nas 64 chamadas realizadas, cujo valor foi de 20ms. Não foi realizado um número maior de chamadas porque o G.711 opera de forma CBR.

3.1.4.2 G.729

O tráfego com o codec G.729 foi gerado no Skype, o qual utiliza um protocolo proprietário e não o SIP como protocolo de sinalização para ligações PC para PC. Dessa forma, o Wireshark não consegue decodificar o protocolo de sinalização que está acima do UDP.

O *Intervalo de Tempo entre Pacotes* foi obtido pela análise da média do tempo de captura do pacote pelo Wireshark. Esse tempo sofre variações, como será visto no tópico 3.1.5, mas na média segue o ITP padrão do codec. Para o codec G.729, a variável ITP assume um valor fixo de 20ms.

3.1.4.3 iSAC

O tráfego para o codec iSAC foi gerado no Skype, por essa razão o raciocínio para a obtenção do valor do *Intervalo de Tempo entre Pacotes* foi o mesmo daquele realizado para o codec G.729. O valor de ITP para o codec iSAC é de 30ms.

3.1.4.4 SILK v.3

Todas as 64 chamadas realizadas com o codec SILK v.3 tiveram ITP de 20ms. O tráfego foi gerado no Skype e a obtenção do valor de ITP teve o mesmo processo que para os codecs G.729 e iSAC.

3.1.5 Erro do Tempo entre Pacotes (ETP)

A variável ETP serve para modelar as variações em torno do *Intervalo de Tempo entre Pacotes*. Processamento de máquina e tempos de fila enfrentados pelo pacote alteram o valor nominal de ITP. Por causa das filas, alguns pacotes chegam depois e outros antes no destino, quando comparados ao *timestamp*. Todavia, um estudo aprofundado sobre as reais causas dessa variação e o grau de influência de cada uma delas se faz necessário. Ressalta-se que as filas comentadas são aquelas enfrentadas dentro da máquina do transmissor, antes de o pacote sair para a rede. As filas que o pacote pode enfrentar em roteadores e *switches* não foram abordadas no estudo por ser um caso particular de cada rede. Nesse trabalho será apresentado um modelo simplificado da variável ETP. Estudos adicionais também são necessários para verificar o grau de influência dessa variável no modelo proposto.

O comportamento da variável ETP foi semelhante entre os codecs, pois todos tenderam a aderir a distribuição Normal. Na figura 3.15, o gráfico superior esquerdo, refere-se ao codec G.711 Lei-A. O codec G.729, gráfico superior direito, foi o que melhor aderiu a distribuição Normal em relação aos codecs estudados. Os codecs iSAC e SILK v.3, gráficos inferiores esquerdo e direito, respectivamente, tiveram comportamentos semelhantes.

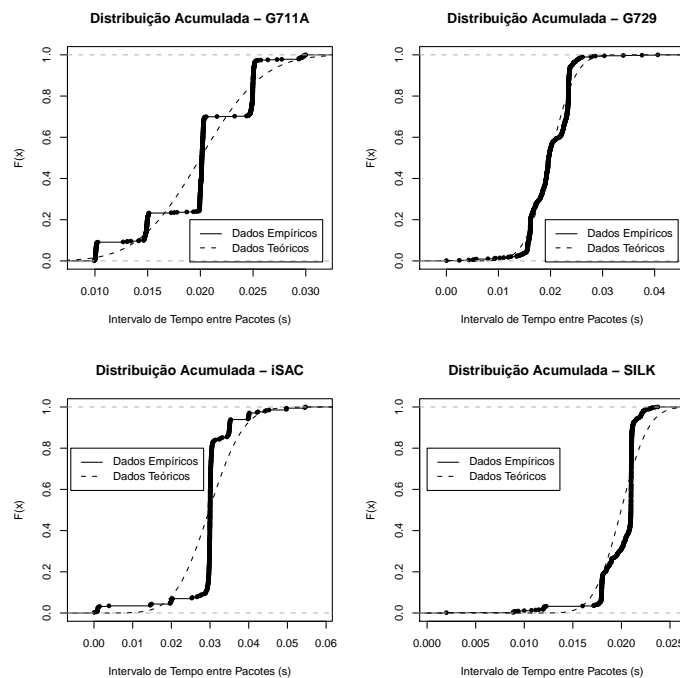


Figura 3.15: Comparação da Distribuição Empírica com a Distribuição Normal por codec.

A tabela 3.5 contém os parâmetros da distribuição Normal para cada codec. Note que na média, o valor do *timestamp* foi mantido e que o desvio padrão foi baixo.

Tabela 3.5: Parâmetros Obtidos da Distribuição Normal para Modelar o ETP dos Codecs Estudados.

	Média (s)	Desvio Padrão
G711	0,02	0,0047
G729	0,02	0,0038
iSAC	0,03	0,0070
SILK v.3	0,02	0,0022

Baseado no modelo apresentado no capítulo 3, foi implementado um gerador de tráfego VoIP sintético. Baseado nesse gerador pode-se comprovar a integridade e eficácia do modelo proposto, assim como planejar redes VoIP. O capítulo 4 aborda detalhes do funcionamento do gerador e apresenta um aplicativo para auxílio da análise de desempenho. Esse aplicativo auxilia na interpretação dos dados de saída do simulador e facilita o processo de planejamento de redes de telecomunicações.

Capítulo 4

Simulação

4.1 Gerador de Tráfego Sintético

De forma a validar o modelo proposto, descrito no capítulo 3, foi implementado um simulador para geração de eventos discretos, os quais consistem no tráfego VoIP sintético. A necessidade da implementação de um simulador adveio do princípio de funcionamento do nosso modelo. Primeiramente, modela-se o comportamento do cliente, o qual é baseado em distribuições de probabilidade. Dessa forma, faz-se necessário a utilização de geradores de variáveis aleatórias. Segundo, o tráfego agregado é gerado a partir da superposição de fontes (clientes). A combinação desses fatores consiste no tráfego agregado gerado, o qual possui solução numérica. Para tornar o simulador independente da plataforma no qual será executado, utilizou-se a linguagem de programação Java.

O princípio de funcionamento do simulador consiste em gerar vários fluxos VoIP paralelos e na simulação de uma fila a qual os pacotes podem vir a enfrentar. Os clientes, que geram o tráfego VoIP, iniciam na condição inativa, sem chamada, e passam para o estado de TRC, em chamada, de acordo com a distribuição exponencial. A permanência no estado de TRC é descrita pela distribuição de Pareto tipo 2. Enquanto o cliente está em chamada, o simulador gera pacotes de acordo com as variáveis TDP, ITP e ETP. A formação de fila ocorre quando há rajadas de pacotes ou a taxa de geração de tráfego é maior que a capacidade do canal. A fila segue a disciplina FIFO (*First In - First Out*) com tamanho infinito. Baseado nesse princípio de funcionamento, as variáveis fornecidas como saída pelo simulador são:

- (i) Tempo de Início da Transmissão do Pacote;
- (ii) Tempo de Término da Transmissão do Pacote;

- (iii) Identificação do Pacote;
- (iv) Identificação do Cliente;
- (v) Identificação da Sessão;
- (vi) Tamanho do Pacote;
- (vii) Tempo de Fila.

A topologia implementada no simulador pode ser visualizada na figura 4.1. Cada fonte de tráfego representa um cliente ativo no sistema. Todos os tráfegos gerados são roteados para um único canal, o qual opera como um *backbone*. A fila implementada está presente no primeiro roteador (esquerda).

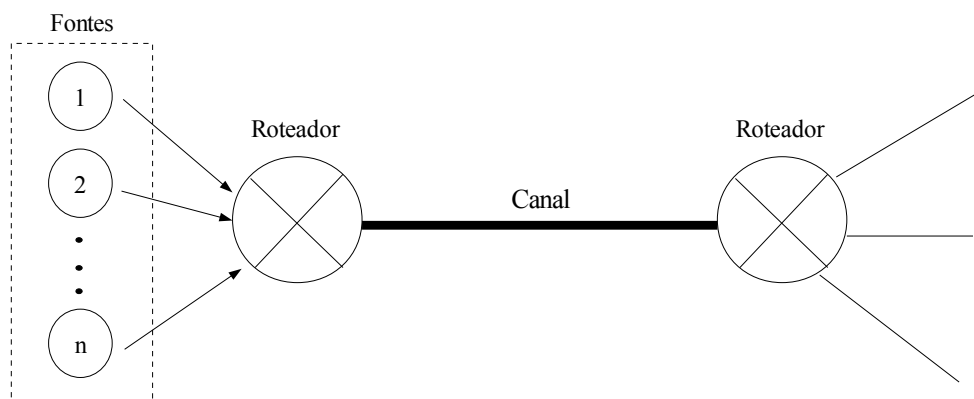


Figura 4.1: Topologia Implementada no Simulador.

Para demonstrar o correto funcionamento do simulador, foram realizados testes em dois cenários previamente conhecidos e avaliado o comportamento das variáveis de saída do simulador.

4.1.1 Cenários de Validação

O primeiro cenário testado verificou o comportamento de quatro variáveis do modelo, TRC, ITC, TDP e ITP. Para facilitar a interpretação das variáveis de saída, as variáveis de entrada do simulador foram configuradas com valores constantes. Esses valores estão dispostos nos tópicos abaixo.

- (i) $TRC = 2s$;
- (ii) $ITC = 1s$;

- (iii) Quantidade de Clientes = 1;
- (iv) TDP = 100 bits;
- (v) ITP = 100 μ s;
- (vi) ETP = 0;
- (vii) Taxa de Transmissão = 1Mbps.

Com uma taxa de transmissão do canal de 1Mbps, cada bit leva 1μ s para ser transmitido. O tamanho do pacote é de 100 bits e o intervalo de tempo entre pacotes é de 100μ s, então não deve haver formação de fila no cenário 1. Como os clientes começam na condição inativo, o único cliente configurado deve iniciar a transmissão de pacotes após 1s de simulação. A sessão desse cliente deve durar 2s e uma nova sessão deve ser iniciada após 1s de inatividade, com um novo identificador. Nessa nova sessão, o identificador do cliente deve ser o mesmo, pois há somente um cliente no sistema. Os resultados de saída do simulador para o cenário 1 estão na tabela 4.1, a qual confirma o correto funcionamento das variáveis.

Tabela 4.1: Resultado da Validação do Modelo com a Configuração do Cenário 1.

ID Pacote	Início do Pacote (s)	Término do Pacote (s)	ID Cliente	ID Sessão	Tamanho do Pacote (bits)	Tempo de Fila (s)
1	1,000100	1,000200	0	0	100	0,000000
2	1,000200	1,000300	0	0	100	0,000000
3	1,000300	1,000400	0	0	100	0,000000
...
19998	2,999800	2,999900	0	0	100	0,000000
19999	2,999900	3,000000	0	0	100	0,000000
20000	3,000000	3,000100	0	0	100	0,000000
20001	4,000100	4,000200	0	1	100	0,000000
...

A segunda configuração possui o objetivo de avaliar o comportamento da fila. Os parâmetros foram configurados de forma a gerar uma fila de 2μ s a cada pacote. A tabela 4.2 mostra os resultados obtidos. A análise destes resultados confirma o correto funcionamento do simulador.

- (i) TRC = 2s;
- (ii) ITC = 1s;

- (iii) Quantidade de Clientes = 1;
- (iv) TDP = 100 bits;
- (v) ITP = 198 μ s;
- (vi) Taxa de Transmissão = 500Kbps.

Tabela 4.2: Resultado da Validação do Modelo com a Configuração 2.

ID Pacote	Início do Pacote (s)	Término do Pacote (s)	ID Cliente	ID Sessão	Tamanho do Pacote (bits)	Tempo de Fila (s)
1	1,000198	1,000398	0	0	100	0,000000
2	1,000396	1,000598	0	0	100	0,000002
3	1,000594	1,000798	0	0	100	0,000004
4	1,000792	1,000998	0	0	100	0,000006
5	1,000990	1,001188	0	0	100	0,000008
...
4996	1,989208	1,999398	0	0	100	0,009990
4997	1,989406	1,999598	0	0	100	0,009992
4998	1,989604	1,999798	0	0	100	0,009994
...

4.1.2 Geradores - Variáveis Aleatórias e Modelo ARMA(2,1)

Para gerar dados que seguem o comportamento das distribuições de probabilidade utilizadas foram implementados geradores de variáveis aleatórias. Os geradores para a distribuição exponencial e a distribuição de Pareto tipo 2 utilizaram o método de Monte Carlo [Banks et al. 2001]. Esse método consiste em isolar a variável aleatória e gerar números aleatórios para $F(x)$. A equação 4.1 gera variáveis aleatórias segundo a distribuição exponencial e a equação 4.2 segundo a distribuição de Pareto tipo 2. Em ambas as equações, a variável R_i representa números aleatórios e x_i a saída do gerador. Na equação 4.1 a variável μ consiste na média.

$$x_i = -\mu \ln(R_i), \mu > 0. \quad (4.1)$$

$$x_i = [R_i^{\left(\frac{-1}{\alpha}\right)} - 1]\beta \quad (4.2)$$

O gerador de variáveis aleatórias que segue a distribuição Normal foi gerado pelo método de Box-Müller [Box and Muller 1958], definido na equação 4.3. Nessa equação

μ é a média, σ é o desvio padrão, R_i é o número aleatório e x_i é a saída do gerado Box-Müller.

$$x_i = \mu + \sigma(\cos(2\pi R_i)\sqrt{-2\log(R_i)}) \quad (4.3)$$

Foi implementado um gerador da série ARMA(2,1) no simulador. Para comprovar o correto funcionamento do mesmo, foram gerados dados com os parâmetros do codec iSAC e comparados com os dados reais. A tabela 4.3 possui um resumo dessa comparação. A série gerada pelo simulador foi modelada com o ARMA(2,1). Os parâmetros obtidos foram semelhantes ao parâmetros reais. Esses parâmetros tendem a ser cada vez mais parecidos conforme a quantidade de dados gerados seja maior. Na figura 4.2 há uma análise dos resíduos do modelo ARMA(2,1) aplicado aos dados gerados. Os gráficos superiores esquerdo e direito, nessa ordem, mostram que não houve tendência e nem correlação entre os resíduos. Esses seguem a distribuição normal, vide gráfico inferior esquerdo, e foram aprovados no teste de Ljung-Box, gráfico inferior direito.

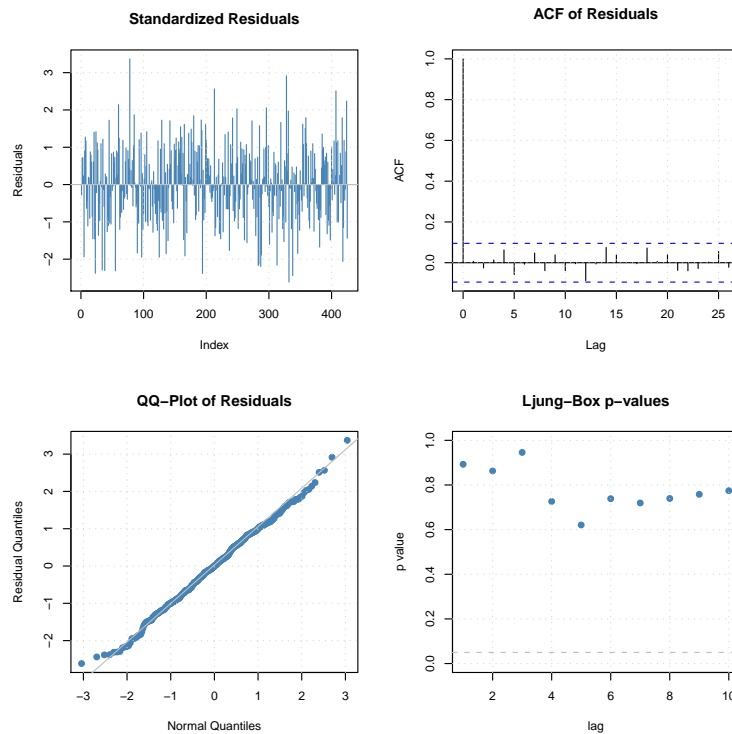


Figura 4.2: Análise dos Resíduos para os Dados Gerados no Simulador.

Na figura 4.3 há uma comparação das funções de auto-correlação dos dados reais e gerados. O comportamento de ambos os dados foram semelhantes.

Tabela 4.3: Comparação de Parâmetros - Dados Reais e Modelo Implementado.

	Parâm. AR 1	Parâm. AR 2	Parâm. MA 1	Média	Desv. Pad.
Dados Reais	1,117	-0,19	-0,631	158,85	22,02
Dados Gerados	1,154	-0,231	-0,666	167,83	27,54

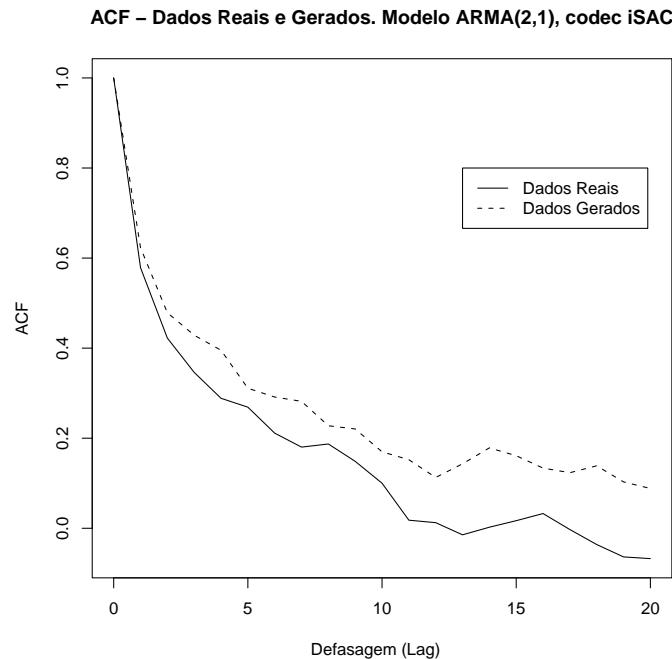


Figura 4.3: Comparação da Função de Auto-Correlação dos Dados Reais com os Dados Gerados.

4.2 Um Aplicativo para Auxílio à Análise de Desempenho

Os dados provenientes do simulador permitem verificar o tráfego de pacotes e o desempenho de fila. Todavia, relacionar esses dados com os parâmetros de modelamento de redes, como o MOS e o *jitter*, não é uma tarefa trivial. Por essa razão foi desenvolvido um aplicativo de análise de tráfego e qualidade de serviço, o qual interage com a saída do gerador de tráfego VoIP. Este aplicativo tem como objetivo verificar a qualidade das ligações, a avaliação do *jitter* e obter a ocupação e capacidade do canal. Em posse dessas métricas, o planejamento de redes fica mais rápido e eficiente. Esse trabalho foi desenvolvido por Rafael Alessi Muntsch e Willian Daniel de Mattos [Muntsch and de Mattos 2010], com orientação do professor Carlos Marcelo Pedroso e co-orientação de Carlos Ignácio de Mattos.

O aplicativo sobre análise de desempenho aborda dois cenários: (1) obter os dados

de tráfego e (2) estimar a capacidade do canal. O cenário 1 serve para avaliar a qualidade de voz da rede, baseada no eMOS [ITU-T 2000], o *jitter* e a ocupação do canal. O cenário 2 serve para estimar a capacidade do canal baseada numa qualidade de serviço desejada. A simulação do tráfego VoIP em conjunto com este aplicativo permite um planejamento otimizado de uma rede de telecomunicações. Além do planejamento, o uso desse aplicativo permite avaliar eventuais problemas na rede VoIP.

4.2.1 Interação Simulador - Aplicativo

Na figura 4.4 está ilustrado o diagrama de fluxo de dados entre o aplicativo e o simulador. Observando o diagrama verifica-se que os parâmetros da rede e da geração de tráfego, fornecidos pelo usuário, serão passados pelo aplicativo como entrada para o simulador. Baseado nesses parâmetros, o simulador irá atuar e fornecer os dados de saída, os quais serão analisados pelo aplicativo. Os possíveis parâmetros de entrada são:

- (i) Parâmetros da Distribuição de Pareto - Forma e Escala;
- (ii) Média do Tempo de Retenção da Chamada;
- (iii) Média do Intervalo de Tempo entre Chamadas;
- (iv) Quantidade de Clientes;
- (v) Qualidade da Voz - MOS;
- (vi) Intervalo de Confiança para o Tratamento dos Resultados;
- (vii) Taxa de Transmissão;
- (viii) Codec da Rede.



Figura 4.4: Diagrama de Fluxo de Dados da Interação entre o Aplicativo e o Simulador.

Na figura 4.5, uma visão mais aprofundada da interação entre o aplicativo e o simulador é mostrada. A primeira etapa do funcionamento consiste na geração do

tráfego baseado nos parâmetros fornecidos. O tráfego gerado é roteado para o canal, o qual é configurado de acordo com dados fornecidos pelo usuário. Caso o tráfego gerado seja maior que a capacidade do canal, haverá formação de fila. Após essa etapa o simulador realiza o cálculo do atraso de cada pacote. O analisador automático captura e interpreta esses dados de acordo com os parâmetros fornecidos pelo usuário.

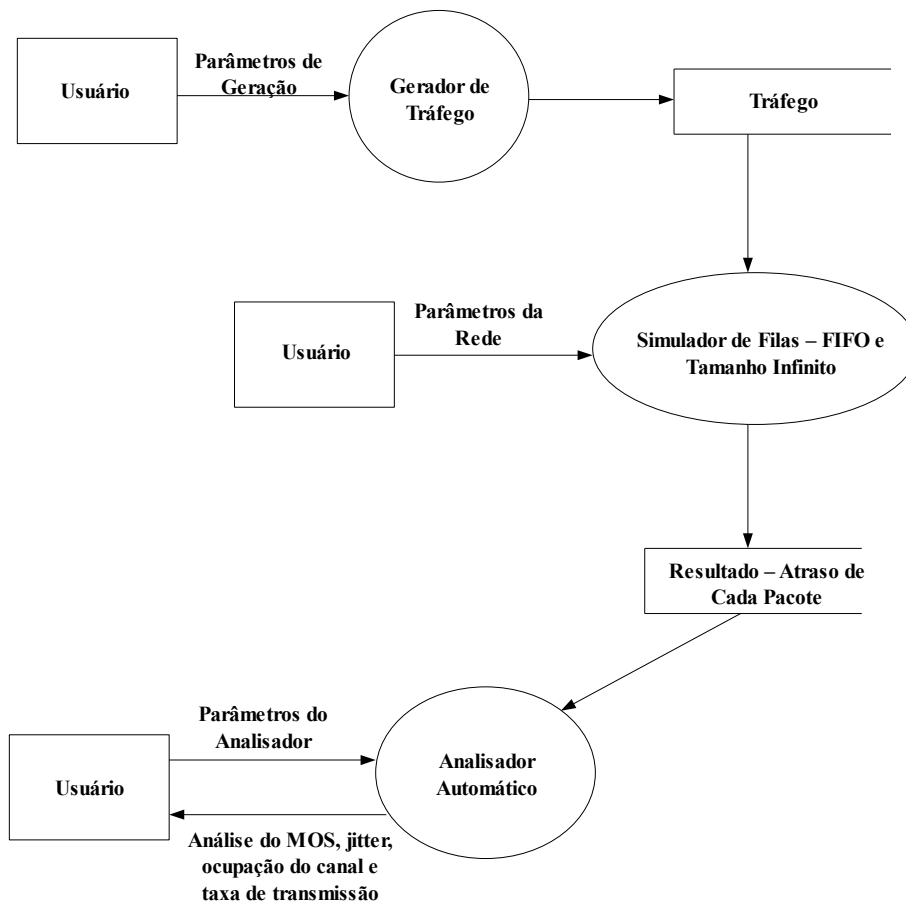


Figura 4.5: Funcionamento Interno da Interação entre o Aplicativo e o Simulador.

4.2.2 Cenários Propostos

O cenário 1, ilustrado na figura 4.6, permite avaliar a qualidade do serviço que está sendo fornecida aos clientes. O usuário fornece os parâmetros da distribuição de Pareto, o tempo médio entre chamadas e a taxa de transmissão. Caso os parâmetros da distribuição de Pareto não sejam conhecidos, o usuário deve fornecer o tempo médio de retenção de chamada. O aplicativo irá calcular o valor do parâmetro de escala

utilizando um valor do parâmetro de forma fixo de $2,16^1$. Baseado nesses parâmetros de entrada, o simulador irá fornecer os resultados da simulação de tráfego. Estes serão analisados de forma a fornecer a ocupação do canal, o *jitter* e o MOS.

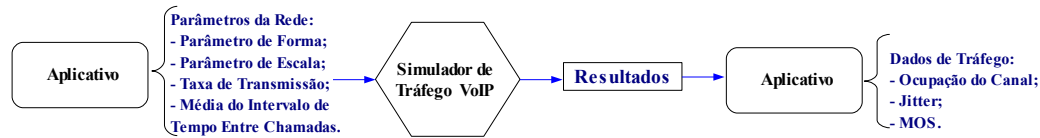


Figura 4.6: Diagrama de Fluxo de Dados do Cenário 1.

No cenário 2 o usuário será capaz de descobrir a taxa de transmissão necessária de forma a atingir o MOS desejado, como mostra a figura 4.7. Analogamente ao cenário 1, os parâmetros de entrada são fornecidos ao simulador o qual provê os resultados da geração de tráfego. Com base nesses resultados o aplicativo realizará o cálculo do eMOS. O resultado desse será comparado com o MOS de entrada. O aplicativo realizará quantas iterações forem necessárias até atingir o MOS desejado.

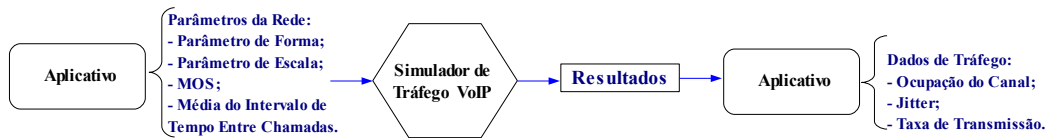


Figura 4.7: Diagrama de Fluxo de Dados do Cenário 2.

4.2.3 Resultados do Aplicativo

Um dos fatores que degradam bastante a qualidade de uma comunicação VoIP é o *jitter*. Por essa razão, essa variável é disponibilizada como saída nos dois cenários de avaliação. O cálculo do *jitter* no aplicativo é realizado de acordo com a equação 4.4 [Schulzrinne et al. 2003]. As variáveis $J(i)$ e $J(i - 1)$ representam o *jitter* no instante atual e no instante anterior, respectivamente. A variável $D(i - 1, i)$ é igual a $(S_i - C_i) - (S_{i-1} - C_{i-1})$, sendo S o tempo de saída do pacote e C o tempo de chegada.

$$J(i) = J(i - 1) + \frac{|D(i - 1, i)| - J(i - 1)}{16} \quad (4.4)$$

¹ $\mu = \frac{\alpha\beta}{\alpha-1}$, sendo μ a média

A ocupação do canal é obtida pela razão entre a taxa de chegada e a taxa de atendimento. A taxa de chegada é expressa pelo tamanho médio dos pacotes dividido pela taxa de transmissão do enlace, enquanto que a taxa de atendimento é o intervalo de tempo médio entre pacotes.

Para o cenário 1, o aplicativo fornece um gráfico que mostra o tempo de fila por pacote, vide figura 4.8. Com base nesse gráfico, o usuário pode realizar diversas verificações, como, por exemplo, se há alguma tendência de a rede congestionar.

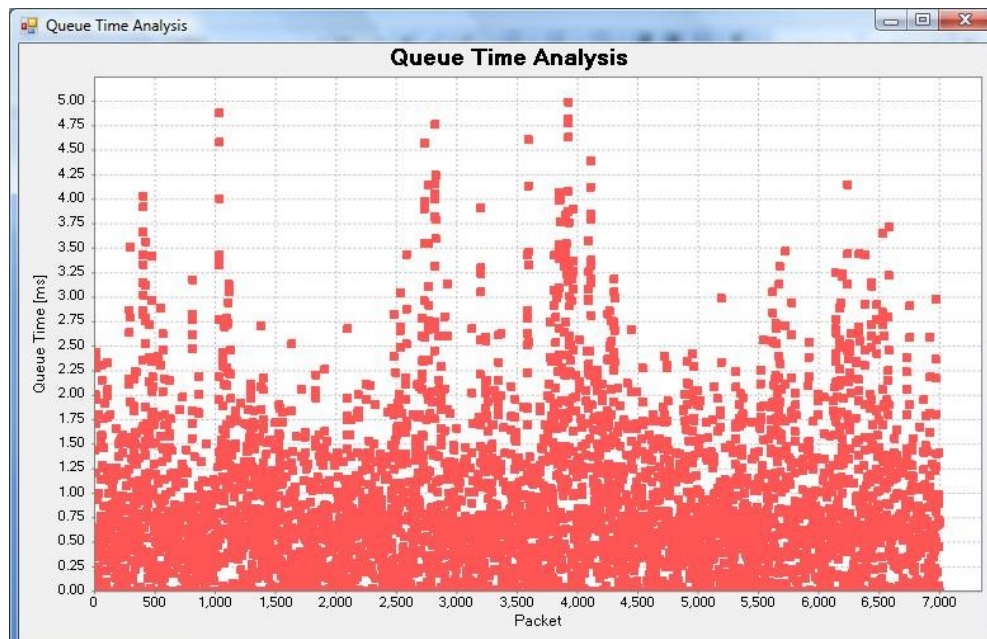


Figura 4.8: Gráfico do Tempo de Fila por Pacote.

Os valores de *jitter*, ocupação do canal e tempo de fila são fornecidos com um intervalo de confiança, o qual é calculado segundo a equação 4.5. Nessa equação, σ é o desvio padrão e n o número de amostras. O valor crítico t_α é calculado pela distribuição t de Student e em função do grau de confiança desejado [de Freitas Filho 2001].

$$h = \frac{t_\alpha \sigma}{\sqrt{n}} \quad (4.5)$$

O cenário 2, a resposta gráfica consiste no eMOS calculado em função da taxa de transmissão. A figura 4.9 ilustra um exemplo do gráfico do cenário 2. Para definir o eMOS, varia-se a taxa de transmissão segundo uma pesquisa binária. Essa pesquisa consiste em definir uma amplitude possível de resultados e encontrar o valor médio desta. Com este valor médio, calcula-se a saída. Se o valor obtido for insatisfatório, a nova amplitude de resultados é entre a média e o valor superior ou inferior, dependendo do caso. Esse método iterativo é realizado até obter um resultado dentro da margem

de erro desejada.

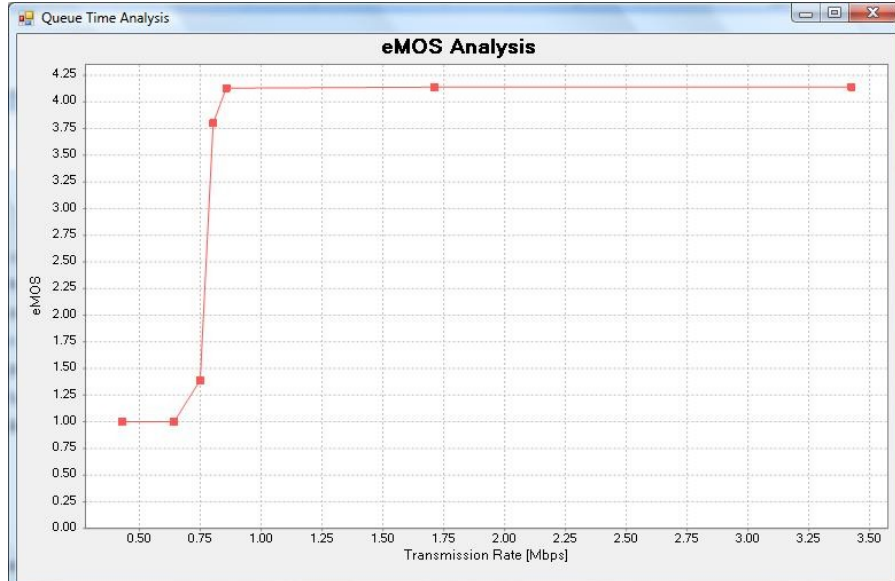


Figura 4.9: Gráfico do eMOS em função da Taxa de Transmissão.

De acordo com [ITU-T 2000], o cálculo do eMOS (eModel) é baseado no fator R . Para $R < 0$, tem-se o $eMOS = 1$ e para $R > 100$, o valor de eMOS é 4,5. Para $0 < R < 100$, o eMOS é obtido pela equação 4.6. A equação 4.7 serve para calcular o fator R , cujas variáveis significam: a) R_0 é o fator base da relação sinal-ruído, b) I_s é o fator de prejuízo que afeta o sinal de voz, c) I_d representa os prejuízos causados pelo atraso, d) I_e representa os prejuízos pelas baixas taxas de transmissão dos codecs e e) A é um fator de vantagem que depende da tecnologia de transmissão de voz utilizada e da situação de uso. A variável A representa basicamente a expectativa do usuário em relação a qualidade da ligação dada uma condição. Por exemplo, uma ligação de celular tem um A superior a de uma ligação pela telefonia tradicional. Isso porque o usuário aceita que a ligação de celular tenha uma pior qualidade que uma ligação da telefonia fixa.

$$eMOS = 1 + 0,035R + 7 * 10^6 R(R - 60)(100 - R) \quad (4.6)$$

$$R = R_0 - I_s - I_d - I_e + A \quad (4.7)$$

Utilizando os valores padrões para telefonia VoIP [ITU-T 2000] das variáveis da equação 4.7, chega-se a equação 4.8. A variável I_e depende do codec em uso. O valor de I_e é 0 para o codec G.711 e 10 para o codec G.729. O fator I_d está relacionado aos atrasos eco na transmissão da voz. O cálculo detalhado dessa variável pode ser

encontrado em [ITU-T 2000].

$$R = 93,4 - I_d - I_e \quad (4.8)$$

Com base no gerador de tráfego apresentado neste capítulo será realizada uma análise de resultados. Essa análise consiste em comparar o tráfego agregado real com o tráfego sintético. Adicionalmente, o capítulo 5 apresenta uma análise do número de elementos na fila. Essa análise permite verificar que o uso dos modelos de tráfego tradicionais para planejar redes VoIP subestima a capacidade da rede.

Capítulo 5

Análise de Resultados

5.1 Avaliação da Qualidade da Geração do Tráfego

A validação do modelo, apresentada neste capítulo, utilizou os dados apresentados na tabela 5.1. Esses dados foram coletados na mesma operadora de telecomunicações, porém dois meses após a coleta de dados utilizada na parametrização do modelo. Os dados utilizados para validação foram coletados na HMM e apresentam estacionariedade. No capítulo 3 foi apresentado o modelo proposto que foi caracterizado e parametrizado com o conjunto de dados apresentado na tabela 3.1.

Tabela 5.1: Dados Utilizados para Validar o Modelo/Simulador Desenvolvido.

Coleta de Validação 1	21/11/2007
Coleta de Validação 2	22/11/2007

Para avaliar a qualidade do modelo proposto, o tráfego simulado foi comparado com o tráfego real. Nesta comparação, a quantidade de informação enviada no tráfego agregado foi quantizada em escalas de 100ms. A escala de 100ms foi escolhida de forma a não gerar arquivos de dados muito extensos. Essa análise foi realizada num tráfego com um pouco mais de 3 horas.

Sabe-se que séries temporais que apresentam auto-similaridade possuem um decaimento lento da função de auto-correlação, como visto no capítulo 2. A figura 5.1 ilustra a função de auto-correlação do tráfego real e do tráfego sintético. Analisando as duas curvas nota-se uma grande semelhança entre as mesmas e um decaimento lento da ACF. Com base nesses resultados observa-se que o tráfego sintético gerado imitou as características de dependência temporal contidas no tráfego real.

Uma outra forma de avaliar a presença da dependência de longa duração é pelo

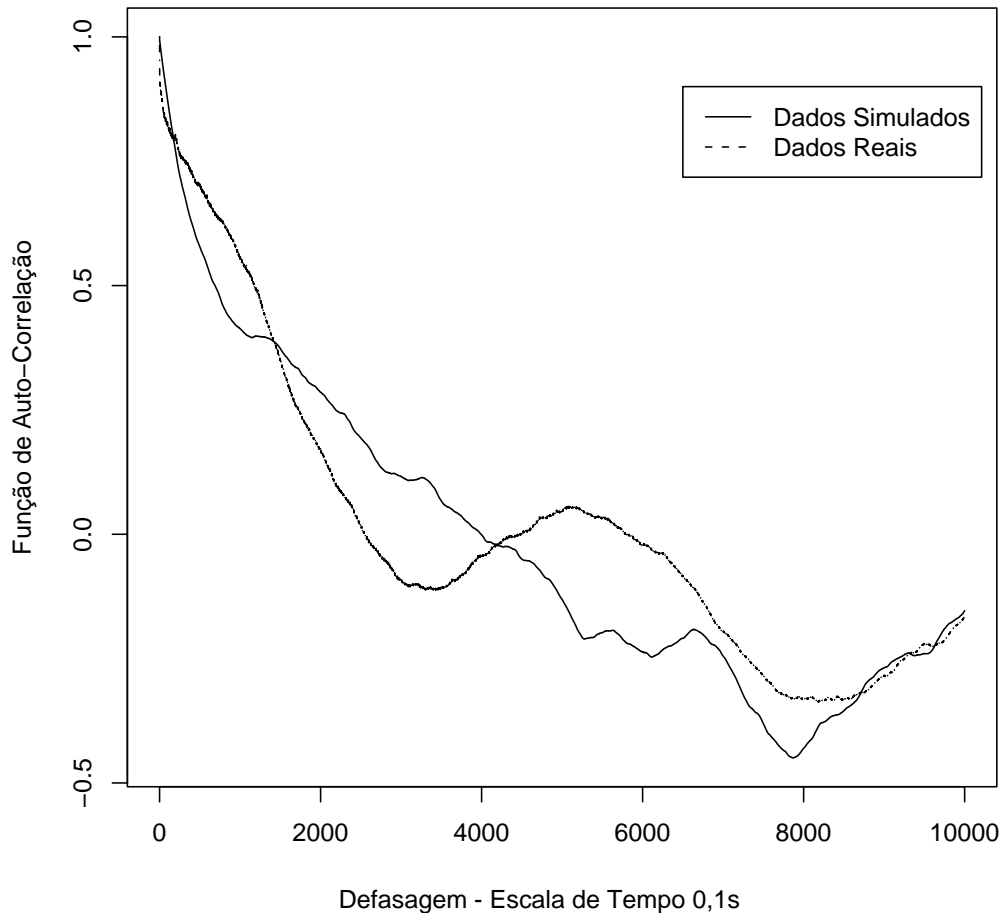


Figura 5.1: Função de Auto Correlação do Tráfego Agregado. Decaimento Lento da ACF para o tráfego sintético e para o tráfego real.

parâmetro de Hurst (H). Séries temporais que apresentam LRD possuem o valor do parâmetro de Hurst entre $1/2 < H < 1$. Quanto mais próximo do valor unitário, maior é o grau de auto-similaridade e da presença de LRD. Para estimar o parâmetro de Hurst foi utilizado o método de Wavelet [Abry and Veitch 1998]. O tráfego agregado real, quantizado numa escala de 10ms, forneceu um valor de 0,6671 para H , enquanto que o tráfego sintético, quantizado na mesma escala, teve um valor de H de 0,6555. Esse resultado confirma numericamente o que foi constatado na figura 5.1. Ambos os tráfegos apresentam características auto-similares com dependência de longo alcance com intensidades semelhante.

Os resultados apresentados, tanto gráfico como numérico, mostram que o modelo imitou as características observadas no tráfego VoIP real. O próximo tópico 5.2 irá

explicar, com base em simulações, o comportamento de uma fila quando alimentada com o tráfego gerado modelo proposto em comparação com o tráfego gerado pelo modelo tradicional.

5.2 Análise do Comportamento de Fila

Uma forma bastante comum de avaliar um sistema é pelo comportamento do tempo médio de espera ou pelo número médio de elementos na fila. Dado a taxa de chegada (λ) e a taxa de atendimento (μ) do sistema avaliado, pode ser obtido o fator de ocupação ($\rho = \frac{\lambda}{\mu}$). A telefonia tradicional possui o intervalo entre chamadas e a duração das mesmas modeladas por uma distribuição de decaimento exponencial. Esse comportamento é descrito pelo modelo M/M/1¹. Para esse modelo, o tempo médio na fila (TF) é dado por: $TF = \frac{\lambda}{\mu(\mu-\lambda)}$. Pela lei de Little, a relação entre o número médio de elementos na fila (NF) e TF é: $NF = \lambda TF$.

Para verificar o comportamento de um sistema de filas, foi utilizado o simulador de filas, apresentado em detalhes no capítulo 4, que foi alimentado por um tráfego gerado pelo modelo proposto. A taxa de atendimento foi considerada constante, de modo a reproduzir as condições de um enlace de transmissão de dados e o algoritmo de gerência de filas foi o FIFO com *buffer* infinito. A intensidade de geração de tráfego foi variada de forma a produzir diversas situações de ocupação do canal e foi registrado o número de elementos na fila para cada situação. A geração de tráfego foi realizada de duas formas: (a) considerando o TRC modelado com distribuição de Pareto e (b) considerando o TRC modelado com a distribuição Exponencial. Em ambos os casos, para cada sessão ativa (em chamada), o fluxo de pacotes foi gerado com o modelo ARMA parametrizado de acordo com os dados provenientes do codec SILK v.3, conforme identificado no capítulo 3. A figura 5.2 ilustra uma comparação do número de elementos na fila observados na simulação para os casos (a) e (b). Foram gerados 10.000.000 de pacotes para obter cada ponto no gráfico. Escolheu-se o codec SILK v.3 por esse ser um codec moderno e bastante utilizado atualmente. O gráfico mostra as tolerâncias, para cada ponto simulado, com um nível de confiança de 95%. Utilizou-se na comparação a variável NF porque esta facilita a verificação de quão grande é a fila. Caso for desejado descobrir o TF, basta multiplicar NF pela taxa de chegadas (λ) da rede em estudo.

¹Notação de Kendall (A/B/N): Comportamento das Chegadas/Comportamento do Atendimento/Número de Servidores.

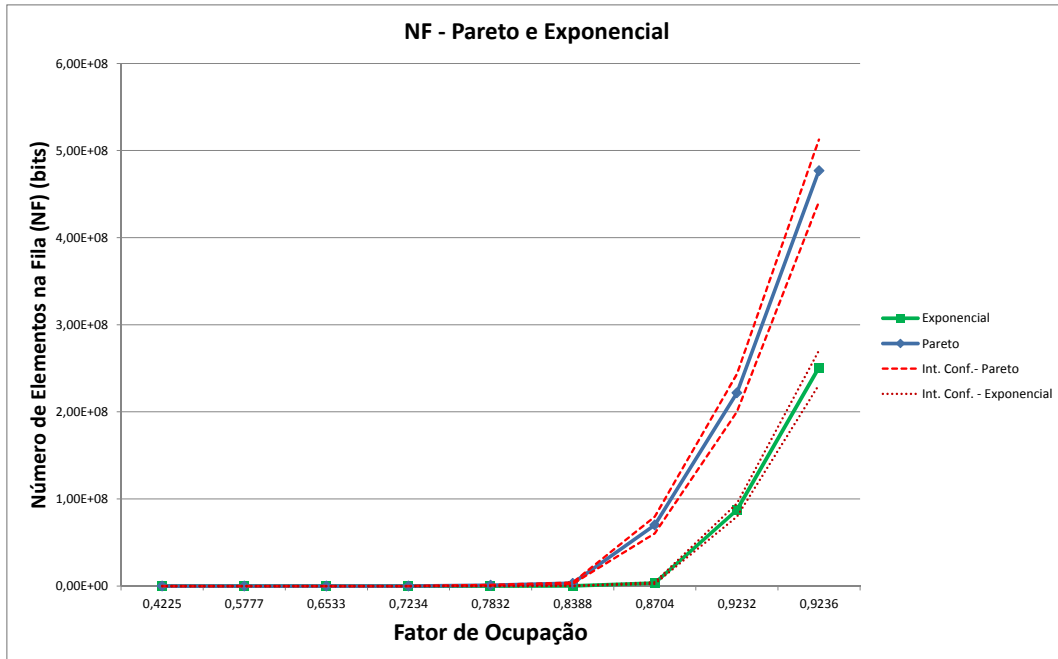


Figura 5.2: Número de Elementos na Fila - Codec SILK v.3.

A análise da figura 5.2 permite concluir que, se a telefonia VoIP for modelada de acordo com o modelo habitualmente utilizado na telefonia tradicional, no qual o TRC é modelado com a distribuições de decaimento exponencial, haverá uma subestimação da capacidade do sistema. Isso porque há uma grande diferença na quantidade de elementos na fila, NF, previstos por cada modelo. Caso um sistema seja dimensionado utilizando modelos de telefonia tradicionais, a qualidade do serviço oferecida pode ficar comprometida. Note que o número de chamadas simultâneas não é necessariamente o mesmo entre as duas curvas para o mesmo valor de ocupação do canal. Na figura 5.3 são apresentados os mesmos resultados obtidos na figura 5.2, porém esses estão em escala logarítmica. Nessa escala fica mais fácil visualizar a diferença entre os resultados das curvas com a distribuição de Pareto e a de decaimento exponencial.

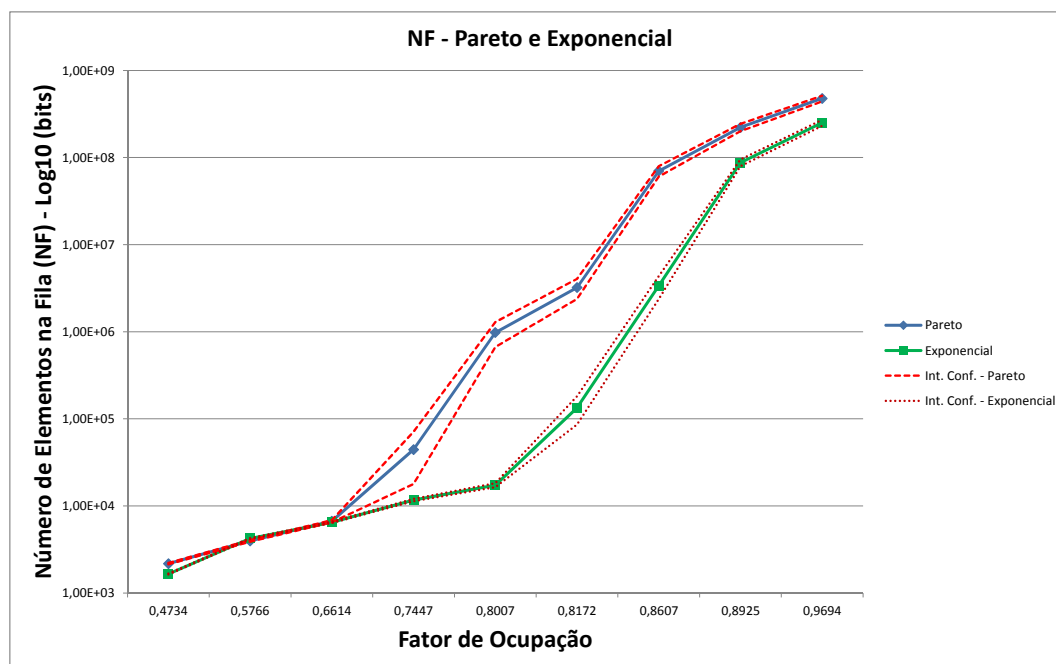


Figura 5.3: Número de Elementos na Fila sem Intervalo de Confiança - Codec SILK v.3.

O presente capítulo apresentou que o modelo proposto e sua implementação geram resultados que caracterizam propriamente o tráfego VoIP. A característica de auto-similaridade foi observada tanto graficamente quanto numericamente. Adicionalmente, os resultados do tópico 5.2 mostram o desempenho de uma fila FIFO quando submetida ao tráfego gerado pelo modelo proposto.

Capítulo 6

Conclusão e Trabalhos Futuros

Os sistemas telefônicos estão gradualmente migrando para VoIP devido a maior eficiência na utilização de recursos, quando comparada à telefonia comutada. A mudança da tecnologia para a transmissão da voz parece estar provocando uma mudança do comportamento do usuário de telefonia. A principal mudança está na duração das chamadas. Chamadas de longa duração não são mais raras, como eram na telefonia comutada, e os modelos tradicionais falham em prever o tráfego VoIP.

Em [Leland et al. 1994] foi comprovado que o tráfego agregado de uma rede Ethernet possui características auto-similares. O trabalho de [Crovella and Bestavros 1997] complementou o trabalho de Leland explicando a razão do tráfego de rede ser auto-similar. Vários modelos foram propostos para caracterizar o tráfego VoIP e o comportamento dos codecs. O trabalho de [Pedroso et al. 2009] mostrou que o tempo de retenção da chamada pode ser modelado por uma distribuição de cauda pesada e em [Menth et al. 2009] foram propostos modelos para caracterizar os principais codecs de áudio utilizados no VoIP. O modelo apresentado neste trabalho propõe modelar o tráfego VoIP através da caracterização do comportamento do usuário e do comportamento do fluxo de pacotes, produzindo tráfego pela agregação de fontes. O modelo proposto foi elaborado de modo a ser bastante genérico e de fácil parametrização, pois a mesma pode ser realizada somente com a média de ITC e TRC, uma vez que o parâmetro de forma da distribuição de Pareto tem sido reportado com pouca variação entre diversos autores.

A parametrização das distribuições de probabilidade que modelam o comportamento do usuário foi realizada com a análise do tráfego real coletado no *backbone* de uma grande operadora de telecomunicações do Brasil. Para modelar o comportamento do fluxo de pacotes, os quais são dependentes do codec utilizado, os dados foram gerados em ambiente laboratorial. Esses dados foram obtidos de conversas telefônicas

cujo áudio consiste em arquivos utilizados para o desenvolvimento de aplicações de processamento de voz [ITU-T].

Foram apresentados conceitos fundamentais sobre a telefonia VoIP: o protocolo SIP que inicializa, controla e finaliza as sessões VoIP e o protocolo RTP que transporta a voz codificada. Uma explicação sobre o princípio de funcionamento dos principais codecs utilizados na telefonia VoIP foi fornecida. Para completar o embasamento teórico necessário para entendimento do modelo proposto, foram apresentados os conceitos de estatística e processos estocásticos utilizados no trabalho e os principais modelos de tráfego disponíveis atualmente.

O modelo proposto foi desenvolvido de forma a gerar um tráfego VoIP através da agregação de fontes. O comportamento do usuário foi descrito por duas variáveis, sendo essas o *Tempo de Retenção da Chamada* e o *Intervalo de Tempo entre Chamadas*. Já o fluxo de pacotes foi modelado pelas variáveis *Tamanho do Pacote*, *Intervalo* e *Tempo entre Pacotes* e *Erro do Tempo entre Pacotes*.

O *Tempo de Retenção de Chamada* foi modelado com uma distribuição de cauda pesada, Pareto tipo 2. Essa distribuição prevê, por definição, a ocorrência de eventos extremos. O *Intervalo de Tempo entre Chamadas* seguiu o modelo tradicional, caracterizado por uma distribuição exponencial.

As variáveis TDP, ITP e ETP foram modeladas separadamente por codec. Para os codecs G.711 e G.729, o modelo da variável TDP é bastante simples, dado que ambos os codecs possuem comportamento CBR. Os comportamentos do Tamanho dos Pacotes para codecs iSAC e SILK foram modelados com o modelo ARMA(2,1), sendo que para o codec SILK foram obtidos dois conjuntos de parâmetros. A variável ITP possui um valor constante para cada codec. O ETP foi modelado com o uso da distribuição Normal.

Baseado no modelo proposto, foi implementado um simulador de tráfego VoIP. Este utiliza internamente geradores de variáveis aleatórias para gerar valores que seguem as distribuições de probabilidade utilizadas. O tráfego agregado é produzido pela superposição de fontes, sendo que cada fonte reproduz o comportamento de um usuário. Todavia, os dados de saída do simulador precisam ser analisados para serem utilizados no planejamento de redes. Para suprir essa necessidade foi desenvolvido um aplicativo que automatiza a análise dos dados de saída do simulador.

A eficácia do simulador foi comprovada pela comparação do comportamento do tráfego real com o tráfego sintético gerado. Ambos os tráfegos apresentaram lento decaimento da função de auto-correlação e um valor do parâmetro de Hurst semelhante. Adicionalmente foi realizado um estudo sobre o desempenho de uma fila FIFO com *buffer* infinito quando alimentada com o tráfego gerado pelo modelo proposto. Os

resultados obtidos comprovam que, se o modelo da telefonia tradicional for utilizado para modelar o tráfego VoIP, haverá uma subestimação dos recursos necessários. Para esse estudo foi utilizado o codec SILK v.3.

Os resultados obtidos confirmam que o modelo proposto caracteriza propriamente o tráfego VoIP. Dessa forma, o mesmo pode ser utilizado para estudos de tráfego VoIP, geração de carga, planejamento de redes, entre outros. O simulador implementado pode ser configurado para se adequar às características da rede em estudo.

Como trabalho futuro, pode ser realizada uma análise mais profunda da variável ETP, de forma a quantificar a influência do tempo de processamento e da formação de fila interna. Outro ponto importante a ser trabalhado é a mudança no padrão de geração de pacotes em função da perda dos mesmos na rede. O trabalho de [Huang et al. 2010] já realizou uma análise prévia do assunto. Por último, deseja-se adaptar o modelo proposto nesse trabalho para o tráfego de IPTV (*IP Television*).

Referências Bibliográficas

- [MMC 1999] (1999). *A memory markov chain model for VBR traffic with strong positive correlations*. 16th International Teletraffic Congress. (ITC). pp. 827-836.
- [3rd Generation Partnership Project (3GPP)] 3rd Generation Partnership Project (3GPP). GSM AMR Speech Codec. *http : //www.3gpp.org/ftp/Specs/archive/26_series/26.073/*, acesso em janeiro de 2011.
- [Abry and Veitch 1998] Abry, P. and Veitch, D. (1998). Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2 –15.
- [Adas 1997] Adas, A. (1997). Traffic models in broadband networks. *IEEE Communications Magazine*.
- [Aldrich 1997] Aldrich, J. (1997). R. A. Fisher and the Making of Maximum Likelihood 1912 - 1922. *Statistical Science*, 12:162–176.
- [Andersen et al. 2004] Andersen, S., A.Duric, Astrom, H., Kleijn, W., and Linden, J. (2004). RFC 3951 - Internet Low Bit Rate Codec (iLBC). IETF Network Working Group.
- [Banks et al. 2001] Banks, J., John S, C. I., Nelson, B. L., and Nicol, D. M. (2001). *Discrete-Event System Simulation*. Prentice Hall International Series in Industrial and Systems Engineering, fourth edition edition.
- [Barford and Crovella 1998] Barford, P. and Crovella, M. (1998). Generating representative Web workloads for network and server performance evaluation. *SIGMETRICS Performance Evaluation Review*, 26(1):151–160.
- [Barford 2001] Barford, P. R. (2001). *Modeling, Measurement and Performance of the World Wide Web Transactions*. PhD thesis, University of Illinois.

- [Box et al. 1994] Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time Series Analysis - Forecasting and Control*. John Wiley and Sons, fourth edition.
- [Box and Muller 1958] Box, G. E. P. and Muller, M. E. (1958). A note on the generation of random normal deviates. *Ann. Math. Stat.* pp. 610-611.
- [Chen et al. 2007] Chen, W.-E., Hung, H.-N., and Lin, Y.-B. (December 2007). Modeling VoIP Call Holding Times for Telecommunications. *IEEE Network*, 21:pp. 22-28.
- [CISCO] CISCO. Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation. Technical report. [http : //www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00800b6710.shtml](http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00800b6710.shtml), acesso em agosto de 2010.
- [Crovella et al. 1998] Crovella, M., Taqqu, M., and Bestavros, A. (1998). *A Practical Guide to Heavy Tail*. Chapman & Hall.
- [Crovella and Bestavros 1997] Crovella, M. E. and Bestavros, A. (1997). Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835-846.
- [de Freitas Filho 2001] de Freitas Filho, P. J. (2001). *Introdução a Modelagem e Simulação de Sistemas com Aplicações em Arena*. Visual Books.
- [de Mattos et al. 2010] de Mattos, C. I., Ribeiro, E. P., and Pedroso, C. M. (2010). A New Model for VoIP Traffic Generation. *International Telecommunications Symposium*.
- [E.Casilari et al. 2002] E.Casilari, H.Montes, and F.Sandoval (2002). Modelling of voice traffic over IP networks. in *Proceedings of Communication Systems, Networks and Digital Signal Processing (CSNDSP)*.
- [F.-K.Chen et al. 2002] F.-K.Chen, J.-F.Yang, and Y.-L.Yan (2002). Candidate scheme for fast ACELP search. *IEE Proceedings: Vision, Image and Signal Processing*, 149:10-16.
- [Flood 1997] Flood, J. (1997). *Telecommunications Networks*. The Institution of Electrical Engineers, second edition edition.
- [GIPS] GIPS. iSAC. Available: <http://www.globalipsound.com/datasheets/iSAC.pdf>, acesso em janeiro de 2011.

- [GIPS 2011] GIPS (2011). Global IP Solutions. <http://www.gipscorp.com/>, acesso em janeiro de 2011.
- [Global-IP-Solutions 2010] Global-IP-Solutions (2010). How to Select the Best Codec for HD Voice. Disponível em: [www.eetrend.com/files – eetrend/GIPS_HD_Voice_WP.pdf](http://www.eetrend.com/files-eetrend/GIPS_HD_Voice_WP.pdf), acesso em janeiro de 2011.
- [Gong et al. 2005] Gong, W.-B., Liu, Y., Misra, V., and Towsley, D. (2005). Self-similarity and long range dependence on the internet: a second look at the evidence, origins and implications. *Computer Networks*, 48(3):377–399.
- [Ha 1999] Ha, N. K. (1999). A fast search method of algebraic codebook by reordering search sequence. *Proceedings of IEEE international conference on Acoustics, speech, and signal processing*, 1:21–24.
- [Handley et al. 2006] Handley, M., Jacobson, V., and C.Perkins (2006). RFC 4566 - SDP: Session Description Protocol. IETF Network Working Group.
- [Heffes and Lucsmtoni 1986] Heffes, H. and Lucsmtoni, D. M. (1986). Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Selected Areas in Communications*, SAC-4:856–868.
- [Huang et al. 2010] Huang, T.-Y., Huang, P., Chen, K., and Wang, P.-J. (2010). Could Skype be more Satisfying? A Qo-E Centric Study of the FEC Mechanism in an Internet-Scale VoIP System. *IEEE Network*, 24:pp. 42–48.
- [ITU-R 2003] ITU-R (2003). ITU-R Recommendation BS.1534-1: Methods for the subjective determination of intermediate quality levels of coding systems.
- [ITU-T] ITU-T. ITU-T Test Signals for Telecommunication Systems. Test Vectors Associated to Rec. ITU-T P.50 Appendix I.
- [ITU-T 1988] ITU-T (1988). Pulse Code Modulation (PCM) of Voice Frequencies.
- [ITU-T 1996a] ITU-T (1996a). Coding of Speech at 8kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP).
- [ITU-T 1996b] ITU-T (1996b). Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 And 6.3 kbit/s.
- [ITU-T 1996c] ITU-T (1996c). Methods for subjective determination of transmission quality.

- [ITU-T 2000] ITU-T (2000). G.107 - The E-model, a computational model for use in transmission planning.
- [ITU-T 2008] ITU-T (2008). Wideband embedded extension for G.711 pulse code modulation.
- [Jain 1991] Jain, R. (1991). *The Art of Computer System Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley and Sons.
- [Jain and Routhier 1986] Jain, R. and Routhier, S. A. (1986). Packet trains: Measurements and a new model for computer network traffic. *IEEE Journal on Selected Areas in Communications*, 4:pp. 986–995.
- [Jiang and Schulzrinne 2000] Jiang, W. and Schulzrinne, H. (2000). Analysis of On-Off Patterns in VoIP and Their Effect on Voice Traffic Aggregation. *in Proceedings of 9th IEEE International Conference on Computer Communication Networks*.
- [Johnston 2004] Johnston, A. B. (2004). *SIP - Understanding the Session Initiation Protocol*. Second edition.
- [Kaplan and Glass 1998] Kaplan, D. and Glass, L. (1998). *Understanding Non Linear Dynamics*. Springer, New York, first edition.
- [Kleiber and Kotz] Kleiber, C. and Kotz, S. *Statistical Size Distributions in Economics and Actuarial Sciences*.
- [Leland et al. 1994] Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. (1994). On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15.
- [Limited 2010] Limited, S. (2010). SILK - Datasheet. <http://developer.skype.com/resources/SILKDataSheet.pdf>, acesso em janeiro de 2011.
- [Limited 2011] Limited, S. (2011). Skype. <http://www.skype.com/intl/en-us/home>, acesso em janeiro de 2011.
- [Lindblom 2004] Lindblom, J. (2004). A Sinusoidal Voice Over Packet Coder Tailored for the Frame-Erasure Channel.
- [Ljung and Box 1998] Ljung, G. M. and Box, G. E. P. (1998). On a Measure of a Lack of Fit in Time Series Models. *Biometrika*, 65:pp. 297–303.

- [Menascé and Almeida 1998] Menascé, D. A. and Almeida, V. A. (1998). *Capacity Planning for Web Performance*. Prentice Hall.
- [Menth et al. 2009] Menth, M., Binzenhöfer, A., and Mühleck, S. (2009). Source models for speech traffic revisited. *IEEE/ACM Transactions on Networking*, 17(4):1042–1051.
- [Muntsch and de Mattos 2010] Muntsch, R. A. and de Mattos, W. D. (2010). Planejamento e Otimização de Redes VoIP através de Simulação de Tráfego. Trabalho de Conclusão de Curso na UFPR.
- [Orebaugh et al. 2006] Orebaugh, A., Ramirez, G., Burke, J., and Pesce, L. (2006). Wireshark Ethereal Network Protocol Analyzer Toolkit (Jay Beale’s Open Source Security).
- [Park and Willinger 2000] Park, K. and Willinger, W. (2000). *Self-Similar Network Traffic and Performance Evaluation*. John Wiley and Sons, New York, first edition.
- [Pedroso et al. 2009] Pedroso, C. M., Ribeiro, E. P., and Munetiko, E. M. (Outubro 2009). Modelagem de Sistemas VoIP: Análise das Características da Sessão. *XXVI SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES SBrT09*.
- [Postel 1980] Postel, J. (1980). RFC 768 - User Datagram Protocol.
- [Ramirez and Gerken 1998] Ramirez, M. A. and Gerken, M. (1998). Efficient algebraic multipulse search. *Proceedings of SBT/IEEE International Telecommunications Symposium*, 1:231–236.
- [Ramirez and Gerken 1999] Ramirez, M. A. and Gerken, M. (1999). A multistage search of algebraic CELP codebooks. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1:17–20.
- [Rosenberg et al. 2002] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., R.Sparks, M.Handley, and Schooler, E. (2002). RFC 3261: Session Initiation Protocol. IETF Network Working Group.
- [Sandras 2010] Sandras, D. (2010). Ekiga. <http://ekiga.org/>.
- [Schiel 1996] Schiel, F. (1996). Bavarian Archive for Speech Signals (BAS) Verbmobil 6.1. <http://www.phonetik.uni-muenchen.de/Bas/BasHomedeu.html>.

- [Schroeder and Atal 1985] Schroeder, M. R. and Atal, B. S. (1985). Code Excited Linear Prediction (CELP): High-Quality Speech at very low bit rates. *Proceedings ICASSP*, pages 937–940.
- [Schulzrinne et al. 2003] Schulzrinne, H., S.Casner, Frederick, R., and Jacobson, V. (2003). RFC 3550 - RTP: A Transport Protocol for Real-Time Applications. IETF Network Working Group.
- [Spittka et al. 2009] Spittka, J., Astrom, H., and Vos, K. (2009). RTP Payload Format and File Storage Format for SILK Speech and Audio Codec. IETF Network Working Group.
- [Tanenbaum 2002] Tanenbaum, A. S. (2002). *Computer Networks*. Prentice Hall Professional Technical Reference.
- [Team 2009] Team, R. D. C. (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
- [Teleco 2010] Teleco (2010). Seção VoIP - Estatísticas. Acesso em Julho de 2010.
- [Vitez 2009] Vitez, M. (2009). Picophone. Disponível em: <http://www.vitez.it/picophone/>, acesso em Setembro de 2010.
- [Vos et al. 2010] Vos, K., Jensen, S., and Soerensen, K. (2010). SILK Speech Codec draft-vos-silk-02. IETF Network Working Group.