

Marcos Eduardo Coelho Garcia

Método de Análise de Tráfegos VoIP Sobrepostos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações.
Aprovação em 01/06/2009

Banca Examinadora:

Prof. Dr. João Crisóstomo Weyl A. Costa - DEEC/UFPA

Prof. Dr. Leonardo de Souza Mendes - FEEC/UNICAMP

Prof. Dr. Max Henrique Machado Costa - FEEC/UNICAMP

Campinas, SP
2009

Marcos Eduardo Coelho Garcia

Método de Análise de Tráfegos VoIP Sobrepostos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações.

Orientador: Leonardo de Souza Mendes

Campinas, SP
2009

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

G165m Garcia, Marcos Eduardo Coelho
Método de análise de tráfegos VoIP sobrepostos /
Marcos Eduardo Coelho Garcia. – Campinas, SP: [s.n.],
2009.

Orientador: Leonardo de Souza Mendes.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Telecomunicações - Tráfego. 2. Markov, processos
de. 3. Telefonia pela Internet. 4. Sistemas de
comunicação em banda larga. 5. Interconexão em rede
(Telecomunicações). I. Mendes, Leonardo de Souza. II.
Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título

Título em Inglês: Method of analysis of VoIP traffic over
Palavras-chave em Inglês: Telecommunication traffic, Markov processes, Voice over,
IP networks, Broadband communication systems,
Internetworking (Telecommunication)
Área de concentração: Telecomunicações
Titulação: Mestre em Engenharia Elétrica
Banca Examinadora: João Crisóstomo Weyl A. Costa, Max Henrique Machado Costa
Data da defesa: 01/06/2009
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Marcos Eduardo Coelho Garcia

Data da Defesa: 1 de junho de 2009

Título da Tese: "Método de Análise de Tráfegos VoIP Sobrepostos"

Prof. Dr. Leonardo de Souza Mendes (Presidente): _____

Prof. Dr. João Crisóstomo Weyl Albuquerque Costa: _____

Prof. Dr. Max Henrique Machado Costa: _____

Resumo

O comportamento de tráfegos VoIP em enlaces de rede é abordado neste trabalho. As aplicações VoIP possuem requisitos rígidos de Qualidade de Serviço (QoS) que devem ser previstos no projeto da rede. Neste sentido, é apresentado o Processo Poissoniano Modulado por Markov (MMPP) como um processo de chegadas adequado para caracterizar o comportamento de tráfegos VoIP sobrepostos. É levado em conta que as aplicações VoIP usam o mecanismo Supressão de Silêncio para economizar largura de banda, o que torna o tráfego muito variável e com chegadas em rajadas. Para análise do comportamento do enlace, é utilizado o modelo de fila MMPP/M/1. Para avaliar a viabilidade do modelo, foram feitos estudos comparativos com resultados do simulador ns-2.

Palavras-chave: VoIP, Tráfegos Sobrepostos, MMPP.

Abstract

The behavior of VoIP traffic in network links is addressed in this work. VoIP applications have strict requirements for quality of service (QoS) that should be provided for in the network design. In this sense, the Markov Modulated Poisson Process (MMPP) is displayed as model of process of arrivals that is appropriate to characterize the behavior of superposition of VoIP traffic sources. It is taken into account that VoIP applications use the mechanism Suppression of Silence to save bandwidth, which makes the traffic very variable and with bursty arrivals. In order to analyze the behavior of the link, the MMPP/M/1 queue model is used. To evaluate the viability of the model, comparative studies were made with results of the ns-2 simulator.

Keywords: VoIP, Traffic Superposition, MMPP.

Agradecimentos

Aos meus pais e meu irmão que sempre me deram apoio, compreensão e carinho em todos os momentos da minha vida.

A minha querida mulher e meu maravilhoso filho pelo amor, paciência e compreensão nesses anos longe.

Ao meu orientado, Prof. Leonardo Mendes, que me convidou pro mestrado e pela grande ajuda nesses anos.

Aos amigos da kitnet, que foram irmãos para mim: Rafael Marinho, Rafael Pereira, Kim Souza, Danilo, Vinicius, Alex e Henrique.

Aos amigos paraenses, que sempre estavam por perto, especialmente: Marcus Pantoja, Néilton Maciel, Helen Paiva, Erick Rocha, Fábio e Bruno.

Aos amigos do laboratório, principalmente: Ricardo Alberti, Eder Ignatiwicz, Paulo Marchi, Alexandre Guimarães, Rodrigo Miani, Gean Breda e Fábio Pessoa.

E a Deus.

À minha família

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Trabalhos Publicados Pelo Autor	xii
1 Introdução	1
1.1 Objetivo	2
1.2 Organização	2
2 Voz sobre IP (VoIP)	3
2.1 Introdução	3
2.2 Protocolos de Transporte de Mídia	5
2.2.1 <i>Real-time Transport Protocol</i> (RTP)	5
2.2.2 <i>Real-Time Control Protocol</i> (RTCP)	6
2.3 Protocolos de Sinalização	7
2.3.1 H.323	7
2.3.2 <i>Session Initiation Protocol</i> (SIP)	8
2.3.3 IAX	10
2.3.4 Protocolos de Controle de <i>Gateways</i> de Mídia	10
2.4 Codificação do Sinal de Voz	11
3 Fatores que Influenciam a Qualidade de Voz	14
3.1 Introdução	14
3.2 Largura de Banda	15
3.2.1 Supressão de Silêncio	16
3.3 Atraso	17
3.4 <i>Jitter</i>	19
3.5 Perda de Pacotes	19
3.6 Eco	20
3.7 Arquiteturas de QoS	21
3.7.1 IntServ	21
3.7.2 DiffServ	23
3.8 Mecanismos de Controle de Congestionamento	26

3.8.1	<i>First-In First-Out (FIFO)</i>	26
3.8.2	<i>Priority Queuing (PQ)</i>	27
3.8.3	<i>Custom Queuing (CQ)</i>	28
3.8.4	<i>Flow-Based Weighted Fair Queuing (WFQ)</i>	29
3.8.5	<i>Class-Based Weighted Fair Queuing (CBWFQ)</i>	31
4	Processo Poissoniano Modulado por Markov - MMPP	32
4.1	Introdução	32
4.2	Modelagem MMPP	33
4.3	Caracterização do Tráfego VoIP	35
4.3.1	Períodos <i>ON/OFF</i>	36
4.4	Aproximação do Tráfego VoIP pelo modelo MMPP	36
4.4.1	Auto-similaridade	39
4.4.2	IDI	40
4.5	Modelo de Fila	42
4.5.1	MMPP/M/1	43
5	Resultados	45
5.1	Introdução	45
5.2	Topologia	45
5.3	Parâmetros	46
5.4	Implementação do Método Matriz-Analítico	47
5.5	Simulador NS-2	47
5.6	Atraso Médio	49
5.7	Probabilidade Escalar do Tamanho de Fila	53
6	Conclusão	56
	Referências bibliográficas	59
A	Script do MATLAB	64
B	Script TCL	67

Lista de Figuras

2.1	Componentes de um sistema H.323.	8
2.2	Componentes de um sistema SIP.	9
2.3	Componentes de um sistema SIP.	11
2.4	Etapas de codificação.	12
3.1	Pacote RTP.	16
3.2	Atividade de voz em uma conversao.	17
3.3	Processo de sinalizao.	23
3.4	Acordo de Nvel de Servio (SLA) entre dois domnios.	24
3.5	Arquitetura em termos funcionais.	25
3.6	Enfileiramento FIFO.	27
3.7	<i>Priority Queuing</i> (PQ).	28
3.8	<i>Custom Queuing</i> (CQ).	29
3.9	<i>Flow-Based Weighted Fair Queuing</i> (WFQ).	30
4.1	Diagrama de transio de um MMPP qualquer.	34
4.2	Fluxo constante de pacotes durante perodos ON.	35
4.3	Chegadas poissonianas durante perodos ON.	37
4.4	Diagrama do trfego unitrio de VoIP modelado por um MMPP de 2 estados.	37
4.5	Diagrama do trfego agregado de N fontes de VoIP modelado por um MMPP de $N + 1$ estados.	38
4.6	Processo auto-similar em diferentes escalas de tempo.	39
4.7	Curvas IDI para N	41
4.8	Trfego gerado por um MMPP injetado em um sistema de filas.	42
5.1	Topologia usada para o estudo da sobreposio de trfegos de VoIP.	46
5.2	Fragmento de cdigo Tcl definindo os parmetros da fonte VoIP.	48
5.3	Procedimento “attach-expoo-traffic”.	48
5.4	Animao do trfego sobreposto de 5 fontes de VoIP no Nam.	49
5.5	Atrasos mdios de $N = 1$ a $N = 30$	50
5.6	Atrasos mdios de $N = 40$ a $N = 500$	51
5.7	Probabilidade escalar do tamanho de fila para $N = 20$	54
5.8	Probabilidade escalar do tamanho de fila para $\rho = 0.86$	55

Lista de Tabelas

3.1	Larguras de banda resultante dos principais codecs da ITU.	16
3.2	Limite de perdas toleráveis de alguns dos principais codecs da ITU.	20
4.1	Modelos de atividade de voz.	36
4.2	Parâmetros H estimados.	40

Trabalho Publicado Pelo Autor

1. Marcos E. C. Garcia, Marcus P. da Silva e Leonardo de S. Mendes. “Método de Análise de Tráfegos VoIP Sobrepostos”. *XXVI Simpósio Brasileiro de Telecomunicações (SBrT’08)*, Rio de Janeiro, RJ, Brasil, Setembro 2008.

Capítulo 1

Introdução

Por muitos anos, a Rede de Telefonia Pública Comutada (*Public Switched Telephone Network - PSTN*), foi amplamente utilizada para a transmissão de voz. Ela tem uma grande importância para história e para o comportamento da sociedade por ser a primeira rede que possibilitou que várias cidades e nações pudessem se comunicar remotamente [1]. Com os avanços da rede IP (*Internet Protocol*), muito do tráfego de voz, que antes era transmitido pela PSTN, está migrando para um tipo de rede que foi primeiramente criada para transmissão de dados [2].

A PSTN é uma rede baseada em comutação de circuitos, ou seja, para que duas partes conversem é necessário que no caminho da chamada cada enlace aloque um canal que permanecerá dedicado até término da conversação. Como a conversação não é constante, existem períodos de silêncio, ocasionando desperdício de recursos. Por outro lado, a qualidade de transmissão da voz é garantida pelo fato de não haver qualquer disputa pelo recurso. [2]

Diferentemente da PSTN, a Rede IP é uma rede baseada em comutação de pacotes. As informações de voz são codificadas e divididas em pacotes. Não é necessário um caminho dedicado entre as partes e os enlaces podem ser compartilhados. Em um enlace são transmitidos, de forma intercalada, pacotes de tipos, origens e destinos variados. Desta forma, um pacote ocupa um enlace apenas durante o seu tempo de transmissão. O tempo restante pode ser usado para transmissão de outros pacotes. Pacotes de voz podem ser transmitidos somente em momentos em que realmente haja informações de voz, e, nos momentos de silêncio, a largura de banda não é desperdiçada enviando pacotes inúteis para conversação. [2]

Já que na PSTN os canais ficam dedicados até que a conversa seja encerrada, a qualidade da transmissão é a melhor possível. Por outro lado, a rede IP entregará uma qualidade de transmissão de acordo com a utilização dos enlaces de rede. Em casos de congestionamento, os enlaces começam não comportar o tráfego gerado, e atrasos e perdas começam a ocorrer. [2]

Para que seja possível a transmissão de voz pela rede, tem que se obedecer a certos requisitos de

qualidade (atraso e perdas de pacotes, por exemplo). O ideal é prever a qualidade no projeto da rede, estimando o comportamento dos tráfegos em cada enlace. [2]

Neste trabalho é usado o Processo Poissoniano Modulado por Markov (*Markov Modulated Poisson Process* - MMPP) para modelar o tráfego de VoIP. Ele é a abordagem mais promissora para descrever as características variáveis do tráfego de voz (devido aos momentos de silêncio, como já foi explicado) [32]. Para estimar e dimensionar a qualidade de transmissão de pacotes de voz nos enlaces da rede IP, é usado o modelo de filas MMPP/M/1.

1.1 Objetivo

Este trabalho tem o objetivo de estudar o MMPP como forma de prever o comportamento do tráfego de VoIP em um enlace. O MMPP tem sido muito utilizado para modelar processos de chegadas de voz. Ele trata efetivamente a variação de chegadas no tempo, captura a correlação das chegadas e é bastante simples de ser tratado matematicamente.

É apresentado como múltiplas fontes independentes de pacotes de voz podem ser modelados por um MMPP. A fila MMPP/M/1 é usado como modelo dos enlaces estudados que recebem um tráfego de VoIP, sendo os parâmetros de comportamento das filas calculados por métodos numéricos.

Levando-se em conta os requisitos de QoS de uma aplicação de voz de tempo-real, é proposto usar as filas e os métodos de cálculo de seus parâmetros como ferramentas práticas para dimensionar um enlace responsável por trafegar pacotes de voz.

1.2 Organização

O trabalho foi organizado como segue:

- No capítulo 2 faz-se um visão geral sobre como funciona atualmente a tecnologia de VoIP;
- No capítulo 3 discutem-se os requisitos de qualidade na transmissão de pacotes de voz e como é possível atendê-los em uma rede IP;
- No capítulo 4 é apresentado o processo MMPP e o modelo de fila MMPP/M/1, sendo mostrado como eles são aplicados para tráfegos de VoIP;
- No capítulo 5, são apresentados os resultados comparativos entre o método e as simulações;
- Finalmente, no capítulo 6 são feitas conclusões dos estudos e sugestões para trabalhos futuros.

Capítulo 2

Voz sobre IP (VoIP)

2.1 Introdução

Com o surgimento e evolução das telecomunicações, várias formas de comunicação à distância se tornaram possíveis. Sejam nas mais variadas formas de dados, sons, imagens ou vídeos, as informações conseguiam chegar a lugares distantes em velocidades nunca antes imaginados. É seguro afirmar que a transmissão de voz em tempo-real foi a forma de comunicação não presencial que causou maior impacto no comportamento e estilo de vida da sociedade mundial. [1]

Em 1876, Alexander Graham Bell submeteu sua patente do telefone. Desde então, a telefonia evoluiu e expandiu-se lentamente, aos poucos tornando-se uma poderosa ferramenta na vida moderna. O telefone pode ser visto como um instrumento de economia de tempo, que aproxima pessoas e fortalece contatos. O telefone foi certamente um fator importante na determinação da estrutura da sociedade moderna, possibilitando os diversos estilos de vida e de negócios que se têm desenvolvido. [1]

Pode-se afirmar que o telefone possibilitou a construção de grandes cidades e de seus arranha-céus. Para chegar a esta conclusão, basta tentar imaginar a inexistência de telefones e que cada mensagem precisasse que uma pessoa se deslocasse fisicamente para entregá-la. Haveria uma grande congestão de pessoas. [1]

A telefonia tradicional é baseada em comutação (ou chaveamento) de circuitos, por isso o uso do termo Rede Telefônica Pública Comutada, em inglês, *Public Switched Telephone Network* (PSTN), para se referir ao sistema de telefonia pública em geral. Para haver uma conversação telefônica é preciso estabelecer um caminho (circuito) entre os telefones que pretendem se comunicar. Este caminho deve permanecer ativo durante todo o tempo da conversação. [2]

Até 1958, as redes telefônicas eram totalmente analógicas. Foi quando surgiram, nos laboratórios Bell (empresa fundada por Alexander Graham Bell), as primeiras centrais digitais. Na década de 1980, o sistema telefônico passou a ser predominantemente digital, exceto pelas linhas de assinantes

(usuários finais). Esta evolução ofereceu uma série de vantagens em termos de operação manutenção e provisão de serviços de telefonia. [2]

Sendo que todos tipos de informação, incluindo texto, áudio, imagem, vídeo, etc. poderiam ser todos igualmente representados no formato digital. Sugere-se que podem existir sistemas de comunicações com a capacidade de transmitir todas estas informações de maneira integrada. O interesse na integração de serviços motivou a criação de diversas tecnologias, entre elas a *Distributed Queue Dual Bus* (DQDB), *Fiber Distributed Data Interface* (FDDI) e Redes Públicas Digitais de Serviços Integrados (RDSI). Mas a tecnologia que efetivamente se definiu como padrão para integração de serviços foi a rede IP (*Internet Protocol*). [2]

A rede IP surgiu na década de 1970. Na época, existiam no mundo várias rede de computadores de tecnologia específicas e que não se comunicavam entre si, sendo a ARPAnet (rede de computadores da Agência de Projetos de Pesquisas Avançadas nos Estados Unidos) a maior e mais famosa. Na intenção de interligar essas diversas redes em uma grande inter-rede e aproveitando a estrutura das redes existentes, foram desenvolvidos os protocolos TCP/IP. Em 1985 a ARPAnet foi rebatizada de Internet. [2] [3]

Na década de 1990, por meio de *modems* (moduladores/demoduladores), a rede IP ganhou grande popularidade, por intermédio da rede telefônica que possuía uma grande capilaridade [2]. Então, pode-se dizer que a PSTN foi uma das grandes impulsionadoras da popularidade da internet. Com isso, o crescimento da internet foi espetacular e criou toda uma nova linha de interesses em vários setores da sociedade e é considerada hoje a mais nova etapa dentro do processo de desenvolvimento da comunicação humana.

Com o considerável aumento das taxas de transmissão da internet cada vez mais serviços de tempo-real conquistam espaço na rede. Entre elas, a que incontestavelmente conquistou maior popularidade foi o *Voice over Internet Protocol* (VoIP), ou Voz sobre Protocolo de Internet. VoIP pode ser definido como transmissão de voz através da internet ou qualquer rede de dados baseada em IP.

Os termos “VoIP” e “Telefonia IP” costumam ser tratados de forma diferente tanto na indústria quanto na mídia. O termo VoIP é usado quando se refere ao empacotamento e transmissão de amostras de voz sobre a rede IP e aos mecanismos de sinalização necessários ao estabelecimento de chamadas telefônicas nessas redes. Já o termo telefonia IP é usado para se referir à aplicação de tecnologias de VoIP na transmissão e na sinalização, com o oferecimento de serviço de qualidade similar ao da telefonia convencional. [2]

O primeiro *software* comercial para transportar voz por internet, o Internet Phone da Vocaltec Communications, surgiu em 1995. A qualidade era precária, mas, com o tempo, a tecnologia VoIP evoluiu consideravelmente. No final desta mesma década, algumas pequenas companhias já ofereciam serviços de VoIP com certa qualidade e com possibilidade de interligação na PSTN. Iniciou-se a

produção de equipamentos específicos para VoIP (*gateways*, adaptadores, telefones IP). Nesta mesma época surgiam os primeiros padrões relacionados à VoIP, tanto por parte da International Telecommunication Union (ITU), quanto por parte da Internet Engineering Task Force (IETF). [2]

A grande procura do mercado por soluções VoIP iniciou quando se percebeu o seu potencial de redutor de custos. Com o uso do VoIP, as ligações de longas distâncias ficaram bem mais baratas. Ao invés das ligações serem processadas através de linhas comerciais de telecomunicação convencional, elas trafegam pela internet ou redes de dados privadas, que são muito mais baratas. [2]

Quando se fala em redução de custos com VoIP, vale comentar sobre Asterisk, o mais famoso *software* PABX em código-aberto. O Asterisk é uma poderosa ferramenta que incorpora os vários serviços dos extremamente caros PABXs comerciais de telefonia tradicional e possui flexibilidade de somar mais serviços, sendo que o custo pode ser reduzido a de um simples computador pessoal. [4]

2.2 Protocolos de Transporte de Mídia

A rede pode perder pacotes, inserir atrasos, ou tirar os pacotes de seqüência em um tráfego de pacotes de voz, da mesma maneira para qualquer outro pacote IP. A qualidade de aplicações de voz, assim como qualquer aplicação de tempo-real, é muito sensível a estas características de rede IP. Dado isto, é necessário protocolos que otimizem a transmissão de dados em tempo-real, podendo também obter informações a respeito da rede. Eis que surgem os protocolos RTP e RTPC. [2] [5]

2.2.1 *Real-time Transport Protocol (RTP)*

Real-time Transport Protocol (RTP) [6], ou Protocolo de Transporte em Tempo-Real, é um protocolo que fornece serviços de entrega fim a fim para dados com características de tempo-real, tais como, áudio e vídeo interativos, através de serviços de rede *unicast* e *multicast*.

Esses serviços incluem:

- Identificação do tipo de dado transportado;
- Número de seqüência;
- Indicação do tempo da amostragem (*timestamping*);
- Monitoração de entrega.

O RTP foi elaborado para ser independente de camada de transporte. Entretanto, em redes IP, ele é tipicamente usado sobre *User Datagram Protocol (UDP)* para fazer uso de sua multiplexação e serviços de soma de verificação; as funcionalidades de transporte são contribuídas em partes por ambos protocolos. O *Transmission Control Protocol (TCP)* não é usado por ser um protocolo de

transporte que garante a transmissão de dados e em seqüência, o que deixa o processo de transmissão mais demorado. Além do mais, o *overhead* do TCP é maior que o do UDP.

Vale comentar que o RTP sozinho não fornece mecanismo algum para garantir tempo máximo de entrega ou dá qualquer outra garantia de qualidade de serviço. Ele não garante a entrega ou evita entregas fora de ordem, nem considera que rede é confiável e entrega os pacotes em seqüência. Porém, o número de seqüência incluso no RTP permite que o destino reorganize a seqüência dos pacotes.

2.2.2 *Real-Time Control Protocol (RTCP)*

Real-Time Control Protocol (RTCP) [6], ou Protocolo de Controle em Tempo-Real, é um protocolo que complementa o RTP, serve para monitorar a qualidade de serviço e transmitir informações sobre os participantes de uma sessão em andamento.

O protocolo RTCP é baseado na transmissão periódica de pacotes de controle para os participantes de uma sessão. Assim como o RTP, o RTCP é tipicamente usado sobre UDP, que deve fornecer multiplexação dos pacotes de mídia e de controle, usando números de portas diferentes.

O RTCP tem quatro funções principais, descritas abaixo.

- Fornece informações sobre a qualidade da distribuição dos dados. Essa é parte essencial do papel do RTP como um protocolo de transporte, estando relacionada às funções de controle de fluxo e de congestão de outros protocolos de transporte.
- Transporta um identificador de nível de transporte persistente para um transmissor em uma sessão RTP, chamado de “nome canônico” (*canonical name*) ou “CNAME”. Isso evita conflitos de identificação em casos de anomalia.
- As duas funções anteriores exigem que todos os participantes de uma sessão RTP enviem pacotes RTCP periodicamente. Assim, as taxas devem ser controladas para que o RTP seja escalável para um grande número de participantes.
- Transporta informações mínimas de controle de sessão, como por exemplo, a identificação de participante que deve ser apresentada na interface de usuário.

A seguir são apresentadas as informações geradas pelo RTCP:

- SR (*Sender Report*) - para estatísticas de transmissão e recepção de participantes que são transmissores ativos em uma sessão;
- RR (*Receiver Report*) - para estatística de recepção de participantes que não são transmissores ativos em uma sessão;
- SDES (*Source Description*) - itens que descrevem um transmissor em uma sessão, como o CNAME;

- BYE - para indicar fim de participação de uma aplicação em uma sessão;
- APP - para funções específicas de aplicação.

2.3 Protocolos de Sinalização

Em VoIP, o protocolo de sinalização é responsável pela troca de informações necessárias para [2] [5]:

- Estabelecimento de chamadas;
- Controle e gerência de chamadas;
- Terminação de chamadas;
- Identificar e autenticar usuários;
- Gerar informações complementares (ex.: dados de tarifação).

Diferentemente do transporte de mídia, existem diversos padrões de protocolos de sinalização. Aqui serão apresentados os padrões que merecem maior destaque.

2.3.1 H.323

H.323 [7] é a recomendação da International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) para comunicações de multimídia em redes de comutação de pacotes. Essa recomendação descreve os componentes do sistema H.323, e descreve as mensagens de controle e procedimentos através dos quais os componentes se comunicam. A recomendação H.323 engloba várias outras recomendações da ITU-T, incluindo H.225, H.245.0, entre outras. O H.323 foi o primeiro protocolo que permitiu interoperabilidade dos produtos VoIP e afastou a indústria das soluções proprietárias. [5]

A Figura 2.1 ilustra a organização lógica típica de um sistema baseado em H.323. Abaixo, é destacado e apresentado as funcionalidades dos principais componentes de um sistema H.323:

- *Terminal* - é o elemento que provê comunicação em tempo-real. Pode ser um computador pessoal ou um dispositivo dedicado executando uma aplicação multimídia H.323, por exemplo.
- *Gateway* - é um elemento que faz uma ponte entre a rede H.323 e outras redes de mídia fora do padrão H.323 (ex.: rede de telefonia convencional).
- *Gatekeeper* - é um elemento opcional no sistema H.323 que é usado para as funcionalidade de controle de admissão e resolução endereço. Sua utilização permite um controle centralizado do sistema, já que todos os elementos devem estar registrados nele.

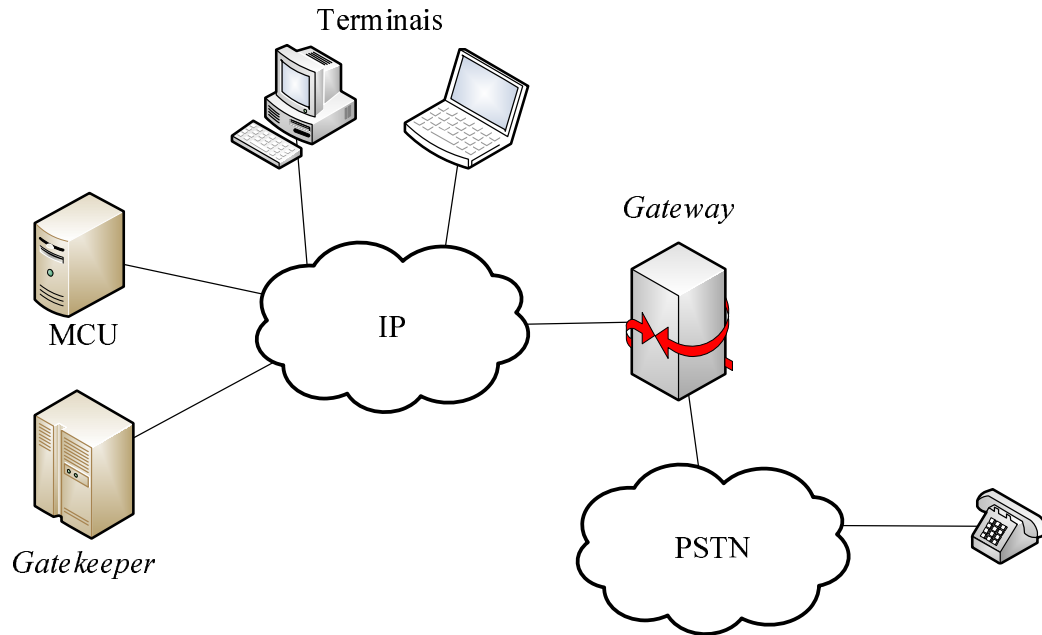


Fig. 2.1: Componentes de um sistema H.323.

- MCU - é um elemento opcional responsável por administrar conferências multipontos. [2]

O H.323 é um padrão que engloba os quatro protocolos a seguir:

- RAS (*Registration, Admission and Status*) - é um protocolo para troca de informações de registro, admissão e estado entre pontos finais (terminais ou *gateways*) e *gatekeepers*. Um ponto final pode usar o RAS para descobrir um *gatekeeper*, se (des)registrar em um *gatekeeper*, requisitar admissão de chamadas e alocação de banda, liberação de chamada. O *gatekeeper* pode fazer uso do RAS para obter informações do estado de um ponto final. O RAS é usado somente quando um *gatekeeper* está presente.
- Q.931 - é protocolo de sinalização para estabelecimento e encerramento de uma chamada entre dois pontos finais H.323.
- H.245 - é usado para controle de conexão, permitindo que dois pontos finais negociem capacidades de processamento em termos de tipos de mídia aceitos, codecs presentes e taxas de transmissão toleráveis.
- RTP/RTCP - já foram apresentados nas seções 2.2.1 e 2.2.2, respectivamente.

2.3.2 *Session Initiation Protocol (SIP)*

Session Initiation Protocol (SIP) [8], ou Protocolo de Iniciação de Sessão, é um protocolo definido pelo grupo de trabalho Multiparty Multimedia Session Control (MMUSIC) do Internet Engineering

Task Force (IETF) para estabelecer e gerenciar sessões de multimídia. O SIP é utilizado em conjunto com outros protocolos, tais como, RTP/RTCP para transportar a mídia, *Session Description Protocol* (SDP) para descrever sessões multimídia, entre outros.

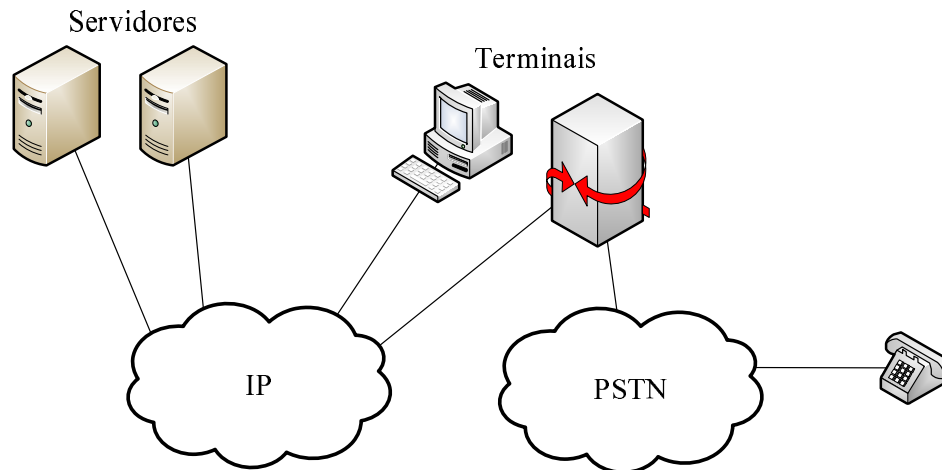


Fig. 2.2: Componentes de um sistema SIP.

Como pode ser visto na Figura 2.2, que ilustra a organização lógica de um sistema baseado em SIP, o padrão SIP é formado por dois componentes, os Terminais e Servidores, descritos a seguir.

- Terminal - é o agente usuário, capaz de iniciar, receber e responder requisições SIP. Pode ser um elemento de usuário final (ex.: telefone IP) ou um *gateway*.
- Servidor - ele fornece um ponto único de acesso para localização de usuários, mapeia nomes amigáveis em endereços roteáveis, roteia mensagens de sinalização e solicitação de redirecionamento entre terminais. Existem três tipos de servidores dentro de uma rede:
 - Servidor *Proxy* (*Proxy Server*) - faz o roteamento de mensagens de requisição de forma que elas sejam encaminhadas para um elemento mais próximo do destino desejado. É útil também para execução de políticas de segurança. Ele interpreta e, se necessário, reescreve partes específicas das mensagens de requisição antes de serem encaminhadas.
 - Servidor de Redirecionamento (*Redirect Server*) - recebe mensagens de requisição e a localização do outro Terminal ou Servidor onde o usuário poderá ser encontrado.
 - Servidor de Registro (*Registrar Server*) - ele aceita mensagens de registro, mantendo-se informado sobre a localização atual de um usuário.

O protocolo SIP é parecido com o *Hyper Text Transfer Protocol* (HTTP), também do IETF, pois

é baseado em mensagens de texto. As mensagens SIP ou são requisições ou são respostas. Algumas requisições são:

- INVITE - convida um usuário para participar de uma sessão;
- BYE - solicita o encerramento de uma sessão;
- REGISTER - enviada a um servidor de registro por um terminal para registrar sua localização.

As respostas são divididas em seis categorias e são identificadas por códigos de três dígitos:

- 1xx - pedido recebido, continuando a processar o pedido;
- 2xx - a ação foi recebida, entendida e aceita com sucesso;
- 3xx - uma ação adicional deve ser tomada para completar o pedido;
- 4xx - o pedido contém sintaxe inválida e não pode ser processada;
- 5xx - erro de servidor;
- 6xx - falha global.

O SIP tem ganhado cada vez mais espaço como alternativa ao H.323, na verdade, já pode ser considerado atualmente como o principal padrão de VoIP. Isso se deve à sua grande simplicidade e flexibilidade para introduzir novos serviços. A Figura 2.3 ilustra as mensagens necessárias para estabelecimento de uma chamada nos padrões H.323 e SIP, onde a maior simplicidade do SIP pode ser observada.

2.3.3 IAX

O *Inter-Asterisk eXchange* (IAX) [4] é um protocolo desenvolvido pela empresa Digium com o propósito de intercomunicar servidores Asterisk. É um protocolo aberto, mas ainda não é um padrão.

O IAX é muito parecido com o SIP, sendo uma das diferenças a que ele utiliza uma mesma porta UDP tanto para o fluxo de sinalização de canal como para o protocolo RTP. Isso facilita o trabalho do *firewall* e a passagem por NATs (*Network Address Translation*).

Ele permite também o agrupamento de sessões múltiplas, sendo os múltiplos fluxos de dados representados por meio de um único cabeçalho de datagrama. Isso permite reduzir a banda necessária quando existem muitos canais ativos.

2.3.4 Protocolos de Controle de Gateways de Mídia

Uma forma de simplificar o *gateway* consiste em separar as funções lógicas de tratamento de mídia e de sinalização em equipamentos diferentes, usando protocolos de controle específicos entre

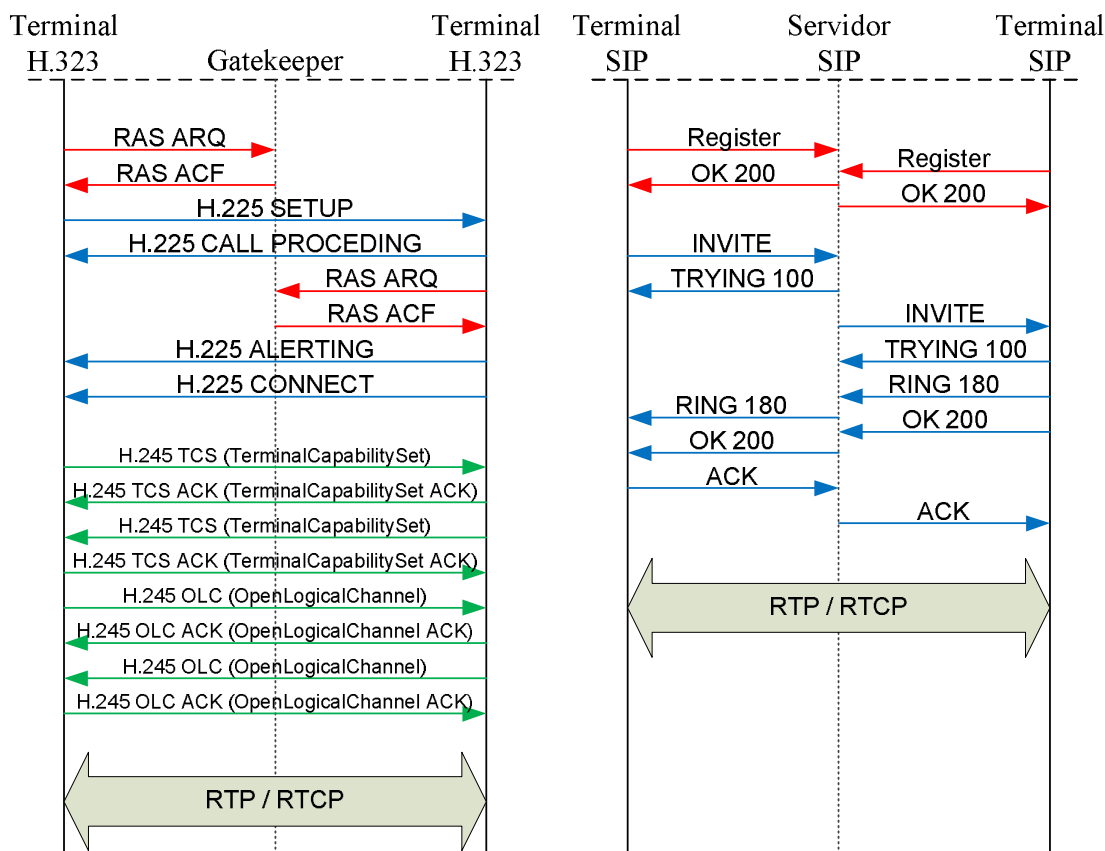


Fig. 2.3: Componentes de um sistema SIP.

eles. Assim, têm-se o Controlador de *Gateways* de Mídia (*Media Gateway Controller* - MGC) e o *Gateway* de Mídia (*Media Gateway* - MG). O MGC, também conhecido por Agente de Chamada (*Call Agent*) ou *Softswitch*, é responsável pela sinalização entre os MGs e outros elementos da rede VoIP (exemplos: Servidores SIP ou *Gatekeepers* H.323). O MG é responsável pela tradução dos sinais de áudio de redes tradicionais de telefonia e pacotes de mídia (RTP) de redes VoIP. Esses protocolos não tem pretensão de substituírem o SIP ou o H.323, e sim complementá-los. [2]

Sem entrar em detalhes, os principais protocolos de controle de *gateways* de mídia são o MGCP [9] e o Megaco/H.248 [10] [11]. O MGCP é um protocolo desenvolvido pelo IETF. O Megaco/H.248 é um protocolo desenvolvido pelo grupo de trabalho Megaco da IETF em conjunto com a ITU-T, é considerado uma extensão do MGCP.

2.4 Codificação do Sinal de Voz

O algoritmo básico de codificação de voz, usado na maioria dos equipamentos digitais de voz nas redes de telefonia, é dividido em três etapas [12]: amostragem, quantização e codificação. Como

pode ser visto na Figura 2.4.

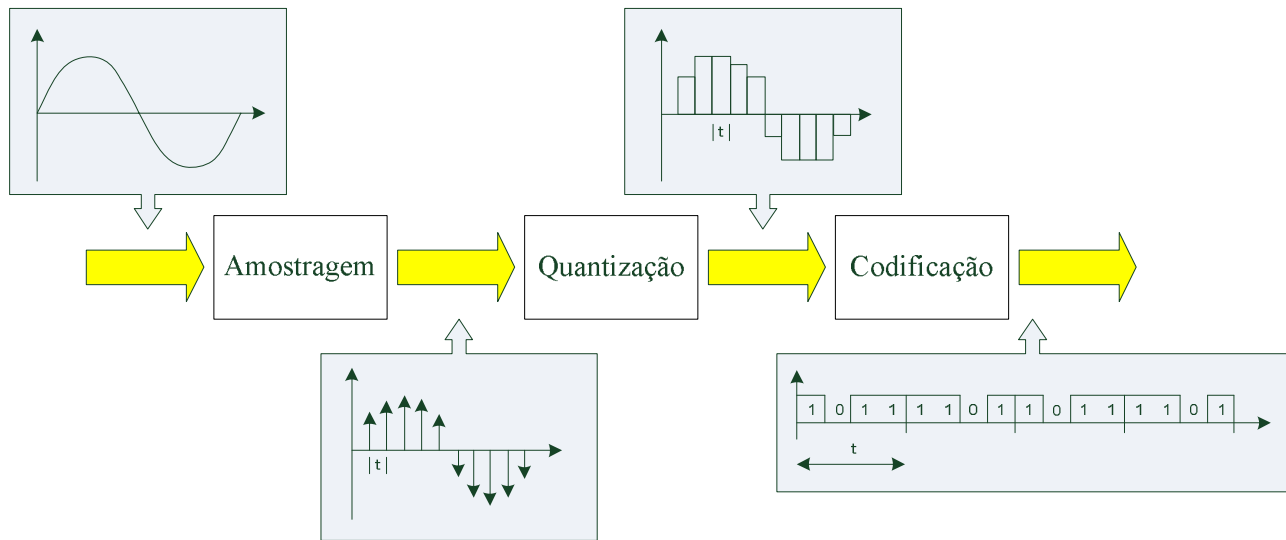


Fig. 2.4: Etapas de codificação.

- Amostragem - Nesta etapa a forma de onda é periodicamente amostrada. Segundo o teorema de Nyquist [13], todas as informações necessárias para reconstruir a forma de onda original da voz estão contidas em amostragens que ocorrem a uma taxa de 8 kHz.
- Quantização - Onde é identificado qual é a amplitude discreta que cada uma das amostras pertence, dentro de um número contável de valores.
- Codificação - Finalmente, na etapa da codificação, as amostras discretas de amplitude são convertidas para códigos binários. Assim, as informações de voz podem ser transmitidas por pulsos binários.

As técnicas de digitalização de voz podem ser largamente categorizadas em duas classes: as que codificam as formas de ondas analógicas da maneira mais fiel possível e as que processam a forma de onda para codificarem somente os aspectos que são perceptivelmente importantes na fala. [12]

A primeira categoria é representante do problema genérico de conversão analógico/digital, não se restringindo a digitalização da fala. As três técnicas mais comuns usadas para codificar formas de onda de voz são PCM (*Pulse Code Modulation*), DPCM (*Differential PCM*) e DM (*Delta Modulation*). Exceto em casos especiais, equipamentos de telefonia projetados para reproduzir uma forma de onda analógica usa umas destas técnicas. [12]

A segunda categoria de técnicas de digitalização de voz se preocupa principalmente em produzir codificadores e decodificadores de taxas de transmissão muito baixas (normalmente abaixo de 4 kbps). Os codificadores que fazem uso destas técnicas são comumente conhecidos como *vocoders*

(*voice coders*), por serem codificadores específicos para voz. Os *vocoders* produzem som de fala de baixa qualidade e soando de forma sintética, não sendo adequado para telefonia. [12]

Existem codificadores de voz de taxas médias (8 kbps, por exemplo) que são implementados como uma combinação de codificadores de forma de onda e as técnicas de taxas baixas. Logo, estas técnicas híbridas (como são conhecidas) representam uma terceira categoria de técnicas de digitalização de voz. Estes codificadores possuem níveis de qualidade mais próximos dos codificadores de forma de onda, podendo ser usados em sistemas de telefonia. [12]

Capítulo 3

Fatores que Influenciam a Qualidade de Voz

3.1 Introdução

Para um usuário, um serviço de conversação de voz em tempo-real de qualidade permite que ele e seu parceiro entendam perfeitamente as sentenças pronunciadas e ambos reconheçam a voz um do outro. [2]

O desenvolvimento da rede PSTN, baseada em comutação de circuitos, foi dirigida para oferecer tais características. Ela estabelece um caminho (circuito) dedicado durante todo o período de conversa entre os dois terminais. A comunicação via comutação de circuitos envolve três fases [2]:

- Estabelecimento do circuito;
- Fase de conversação;
- Desconexão do circuito.

É importante notar que, durante a fase de conversação, o circuito alocado entre os terminais permanece dedicado até que um deles (ou ambos) resolva desfazê-lo. É observado, em uma conversação, que o tráfego entre os terminais não é contínuo, ou seja, existem momentos de silêncio em ambos os lados. Sendo assim, a capacidade do canal é desperdiçada. Em compensação, existe a garantia que a capacidade de transmissão estará sempre disponível quando necessário, pois não há disputa alguma por recursos.

Em VoIP, baseado em comutação de pacotes, os dados a serem transmitidos são fragmentados, organizados e agrupados em pacotes. Estes pacotes possuem o endereço de destino e são transmitidos pela rede de nó a nó. Em cada nó, o pacote é recebido e o próximo caminho da rota é determinado baseado no endereço contido nele. Não é necessário o estabelecimento de um caminho dedicado entre os terminais, podendo os enlaces serem compartilhados, de forma intercalada, com pacotes de diferentes origens e destinos. Um pacote ocupa um enlace apenas durante o seu tempo de transmissão,

o tempo restante podendo ser utilizado para transmitir outros pacotes. O tempo de transferência aumentará sempre que a carga dos enlaces e tamanho das filas aumentarem. Em casos de grandes congestionamentos, as filas chegam em seus limites de armazenamento e perdas de pacotes começam a ocorrer. [2]

Segundo a definição da ITU-T [14], Qualidade de Serviço, ou QoS, é o efeito coletivo dos diversos fatores de desempenho num sistema, que determina o grau de satisfação de um usuário do serviço de telecomunicações.

A garantia de QoS pressupõe uma série de funções que devem ser fornecidas em uma rede de telecomunicações para que requisitos de qualidade sejam atendidos. Para garantir a qualidade de transmissão de voz em uma rede de pacotes, devem ser observados os seguintes parâmetros: [2]

- Largura de banda;
- Atraso;
- *Jitter*;
- Perdas de pacotes.

3.2 Largura de Banda

Para que conversações de longas distâncias em tempo-real possam acontecer com qualidade satisfatória entre os usuários, o meio de transmissão necessita ser capaz de transmitir um taxa mínima de dados entre os dois pontos. Por muito tempo, os antigos usuários da internet tinham conexões muito baixas, o que inviabilizava o uso da mesma para esta aplicação.

Para se saber a banda necessária para transmissão de voz, deve-se obter a taxa de saída do codificador e leva-se em conta o *overhead* dos cabeçalhos IP/UDP/RTP dos pacotes [5] [15]. Como pode ser visto na Figura 3.1, o tamanho do pacote é obtido pela seguinte soma:

Tamanho de pacote = 20 bytes (IP) + 8 bytes (UDP) + 12 bytes (RTP) + Tamanho de *payload* de voz

A banda é obtida nos cálculos a seguir:

$$\text{Taxa de Pacotes} = \frac{(\text{Taxa de bits do codec})}{(\text{Tamanho do } \textit{payload} \text{ de voz})}$$

$$\text{Largura de banda} = (\text{Tamanho de pacote}) \times (\text{Taxa de Pacotes})$$

As larguras de banda relacionadas aos principais codecs da ITU se encontram na Tabela 3.1.

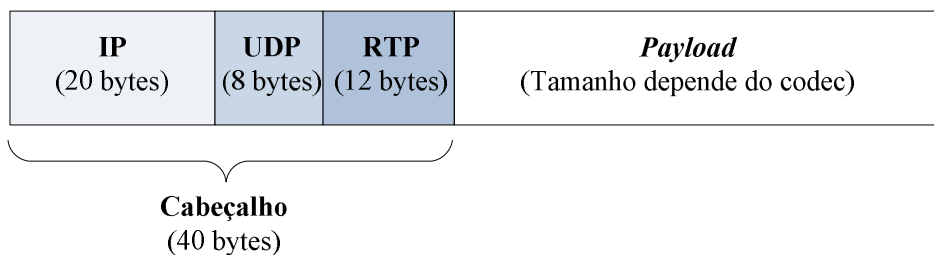


Fig. 3.1: Pacote RTP.

<i>Codec</i>	<i>Taxa do codec (kbps)</i>	<i>Tamanho do Payload (bit)</i>	<i>Largura de banda (kbps)</i>
G.711	64	1280	80
G.723.1	5,3	160	15,9
	6,3	192	16,8
G.726	24	480	40
	32	640	48
G.729	8	160	24

Tab. 3.1: Larguras de banda resultante dos principais codecs da ITU.

Para se obter a taxa de bits real na camada física, é preciso acrescentar em cada pacote o *overhead* da camada de enlace (+26 octetos de encapsulamento *Ethernet*, por exemplo).

Em enlaces que possuem disponibilidade de largura de banda limitada é preciso usar *codecs* com taxas de saída menores; em compensação a qualidade da voz transmitida é menor.

Uma forma de economizar largura de banda em conversas de voz é usar Supressão de Silêncio, mecanismo que aproveita os momentos de silêncio do locutor para evitar enviar pacotes inúteis para a conversa.

3.2.1 Supressão de Silêncio

Em um conversa, enquanto uma pessoa fala, a outra faz silêncio para escutar. Intuitivamente, pode-se concluir que, pelo menos na metade do tempo da conversa, cada pessoa está em silêncio. Além disso, no meio de um discurso a pessoa costuma fazer pausas (ver Figura 3.2). O estudo em [16] mostra, que em uma conversa normal, cada locutor está ativo somente 40% do tempo; segundo [17], são 35%. Na rede PSTN, existe um grande desperdício inevitável dos recursos para enviar informações de silêncio. Por outro lado, redes baseadas em comutação de pacotes podem fazer um uso mais inteligente da rede, usando o algoritmo de Supressão de Silêncio para não enviar pacotes de voz durante os períodos de silêncio. Assim, a banda que seria desperdiçada pode ser usada para outros propósitos.

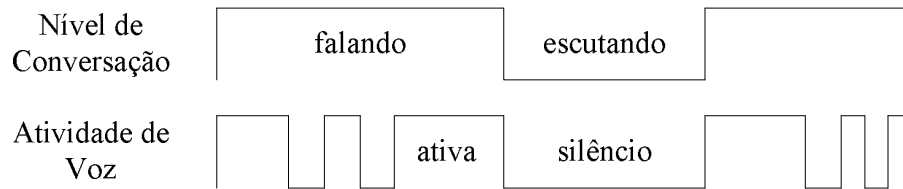


Fig. 3.2: Atividade de voz em uma conversação.

O processo de separar a fala do silêncio é chamado de VAD (*Voice Activity Detection*) e é usado para indicar a presença de voz de forma que possibilite a delimitação do começo e fim de um segmento de conversa. A função básica do algoritmo VAD é de extrair e medir o sinal de entrada, e comparar este valor com um limiar, que costuma ser extraído das características de sinal/ruído da fala. A decisão de voz ativa é feita quando os valores medidos ultrapassam o limiar. [18]

Qualquer codificador de fala de taxa fixa pode ser combinado com um VAD, o que fará dele um sistema simples de taxa que varia em dois estados: estados de taxa alta (voz ativa) e de taxa baixa (silêncio). A taxa baixa, ou pode ser zero, ou uma taxa baixa necessária para enviar informações de ruído para o gerador de ruído de conforto. O gerador de ruído de conforto, ou CNG (*Comfort Noise Generator*), é usado para simular o ruído de fundo nos momentos de silêncio para que o ouvinte não tenha uma sensação de desconexão pelo fato de a transmissão ficar completamente muda. [19] [20]

3.3 Atraso

O atraso ou latência é o retardo na transmissão ocasionado pelos elementos do sistema. O que é percebido pelo usuário é o atraso fim-a-fim, ou seja, o montante de todos os atrasos do sistema. Numa linguagem simples, o atraso fim-a-fim de voz é tempo que leva para o som sair da boca do locutor, passar pelo sistema e chegar ao ouvido do ouvinte [21]. Em uma conversa, cada pessoa normalmente espera o fim do discurso da outra para dar início à sua fala. Se o atraso fim-a-fim for muito grande, podem resultar em conversas frustrantes, se não completamente ininteligíveis [2].

Segundo a ITU [22], em uma conversa, atrasos fim-a-fim de voz menores do que 150 ms não são percebidos pelo ouvido humano, atrasos entre 150 e 400 ms podem ser aceitáveis e atrasos que excedem 400 ms são intoleráveis.

Os atrasos que compõem o atraso fim-a-fim são correspondentes aos seguintes fatores: *codec*, empacotamento, enfileiramento, serialização e propagação; eles variam de acordo com os equipamentos, materiais e tecnologia usados no sistema. [2] [23]

Atraso do *Codec*

O atraso do *codec*, podendo também ser chamado de atraso de processamento, corresponde ao tempo que o processador de digitalização de sinal (DSP) leva para comprimir um bloco de amostras PCM. Por exemplo, o algoritmo ACELP analisa um bloco de amostra PCM de 10 ms para depois comprimi-lo. [23]

Conforme a Cisco [23], o tempo de compressão do codificador CS-ACELP varia de 2,5 ms a 10 ms em função da carga do processador DSP. Se o DSP se encontrar totalmente carregado com 4 canais de voz, o atraso do codificador é de 10 ms. Se o DSP estiver carregado com somente um canal de voz o atraso é de 2,5 ms.

O tempo de descompressão é aproximadamente 10% do tempo de compressão para cada bloco. Entretanto, o tempo de descompressão de cada quadro é proporcional ao número de blocos amostras por quadro. [23]

Para processar corretamente as amostras de um bloco, vários codificadores procuram informações no bloco seguinte. Esta ação se chama *look-ahead* e adiciona mais atraso ao codificador. [23]

Atraso de Empacotamento

O atraso de empacotamento é o tempo necessário para se preencher o *payload* de um pacote com fala codificada e comprimida. Este atraso é uma função do tamanho do bloco de amostra determinado pelo codificador e do número de blocos em cada quadro. Como regra geral, este atraso não deve ultrapassar 30 ms. [23]

Atraso de Enfileiramento

Os pacotes de voz são colocados em filas para transmissão nas conexões da rede. O atraso de enfileiramento corresponde ao tempo variável que o pacote passa em uma fila, ele depende da velocidade do tronco e do estado da fila [23]. Ocorre quando são enviados mais pacotes que a capacidade de uma interface em um dado intervalo [21]. Os pacotes de voz têm maior urgência contra pacotes de dados, algoritmos de enfileiramento (fila com prioridade, por exemplo) devem ser usados nos roteadores de forma que o tráfego de VoIP receba maior vantagem na transmissão.

Atraso de Serialização

O atraso de serialização é o tempo necessário para que um quadro de voz ou dado seja transferido em uma interface de rede. É um tempo fixo e está relacionado à velocidade do *clock* do tronco. [23]

Atraso de Propagação

O atraso de propagação corresponde ao tempo fixo de propagação em um tronco [23]. Este tempo depende da distância percorrida e da velocidade de propagação do meio.

3.4 Jitter

Basicamente, *jitter* é a variação de atrasos entre pacotes adjacentes [21]. Em uma aplicação de voz, os pacotes são gerados de forma constante, mas na transmissão eles podem tomar caminhos diferentes na rede, ou receber níveis de congestão diferentes, tornando as chegadas no destino inconstantes.

O algoritmo do *codec* no destino precisa que os pacotes de voz sejam entregues constantemente; para isso, é necessário um mecanismo que compense o *jitter*. Este mecanismo é chamado de *Jitter Buffer*. [21]

O *jitter buffer* assegura uma reserva de pacotes antes de dar início ao processo de reprodução, introduzindo um atraso inicial à voz [2]. Quanto maior for o *jitter*, maior será o atraso inicial necessário.

3.5 Perda de Pacotes

As redes IP não garantem a entrega de pacotes, sendo esperada a perda de alguns pacotes. As causas das perdas são dadas a seguir:

- Erros na rede;
- Descarte pelos roteadores devido a congestionamentos;
- Pacotes de voz que se atrasam a ponto de chegar após o momento em que deveriam ser reproduzidos, também são considerados pacotes perdidos.

Devido a restrições rigorosas a atrasos para aplicações de VoIP, protocolos de transporte confiáveis, como TCP, não podem ser usados. A perda de pacotes de VoIP são inevitáveis, mas podem ser compensados por esquemas chamados de *Packet Loss Concealment* (PLC). Por exemplo, o *codec* G.723.1 interpola o quadro perdido simulando as características vocais do quadro anterior e vagarosamente atenua o sinal. [24]

O efeito da perda de pacotes depende da qualidade da codificação da voz e se o esquema PLC é usado pelo *codec*. Ao contrário do *codec* G.711, os *codecs* G.723.1 e G.729 já possuem o esquema PLC incorporados. Entretanto, o esquema pode ser implementado sobre o *codec* G.711. Na

Tabela 3.2, encontram-se os limites toleráveis de perdas de pacotes para uma conversação de qualidade para alguns *codecs* da ITU, considerando que as perdas ocorrem aleatoriamente e que o atraso fim-a-fim <150 ms é respeitado. [25]

<i>Codec</i>	<i>Limite de perdas de pacotes (%)</i>
G.711 sem PLC	1
G.711 com PLC	10
G.729A + VAD	3,4
G.723.1 (6.3 kbps) + VAD	2,1

Tab. 3.2: Limite de perdas toleráveis de alguns dos principais codecs da ITU.

3.6 Eco

O eco é a percepção que o locutor tem de sua própria voz, porém com atraso [5]. Para cada locutor de uma conversação, é comum e esperado ouvir a própria voz enquanto fala. Por outro lado, ouvir a própria voz no fone após um atraso de 25 ms pode causar interrupções e quebrar a cadência em uma conversa. O eco tem duas características: ele pode ser alto e longo. Quanto mais alto e longo for o eco, é claro, mais irritante o mesmo se torna. [21]

Em uma rede PSTN, o eco é normalmente causado pelo descasamento de impedância no comutador que faz a conversão da rede de quatro fios para a rede local de dois fios [21]. Este tipo de eco pode ser chamado de Eco Híbrido [5]. Outro tipo de eco é o Eco Acústico. Este consiste do sinal acústico realimentado do alto-falante de um dispositivo para o microfone desse mesmo dispositivo [5].

Em lugares do mundo onde se usa principalmente a voz analógica, para limitar o eco, empregam-se Supressores de Eco [21]. Eles introduzem uma grande perda no caminho de envio quando a parte distante está falando. O problema é que esta técnica tende a atropelar a fala do locutor quando a parte distante fala ao mesmo tempo [5].

Outra forma de limitar o eco é usando Canceladores de Eco. Eles constroem uma estimativa de eco para removê-la do sinal que chega. O eco é modelado como uma soma de sinais, semelhante ao sinal de entrada, porém atrasado e com menor amplitude [5]. Nas redes baseadas em pacotes, os canceladores de eco podem ser montados em codecs de baixas taxas de bits e operá-los em cada DSP. Alguns fabricantes implementam o cancelador de eco em software (esta prática reduz drasticamente os benefícios do cancelador) [21].

3.7 Arquiteturas de QoS

A partir de meados dos anos 90, houve extraordinário desenvolvimento e ampla disseminação das aplicações em rede que transmitem e recebem conteúdo de áudio e vídeo pela internet. Novas aplicações IP (rádio por internet, *sites* multimídia, teleconferência, jogos interativos, mundos virtuais, aprendizado à distância, etc) surgem continuamente. As exigências de serviços dessas diferem de modo significativo daquelas das aplicações tradicionais orientados a dados, como texto/imagem, *web*, *e-mail*, FTP, etc. [3]

O protocolo de camada de rede da internet de hoje fornece um serviço de melhor esforço (*best effort*) para todos os datagramas que transporta. Faz-se melhor esforço para transportar cada datagrama do remetente ao receptor o mais rápido possível. Contudo, o serviço de melhor esforço não dá nenhuma garantia sobre o atraso fim-a-fim para um pacote individual. Tampouco garante alguma diferenciação de prioridade de cada aplicação. Em outras palavras, a Internet não promete nada quanto à QoS que uma aplicação receberá. Uma aplicação receberá o nível de desempenho que a rede estiver capacitada a dar no determinado momento. [3]

Há muitos esforços para o desenvolvimento de arquiteturas que atendem as necessidades de QoS das diversas aplicações, entre as quais merecem maior destaque as de Serviço Integrado (IntServ) e a de Serviço Diferenciado (DiffServ). [26]

3.7.1 IntServ

A arquitetura IntServ define duas grandes classes de serviços: serviço garantido e serviço de carga controlada. [3] [26]

Serviço Garantido - Introduz mecanismo de reserva de recurso e mecanismo de controle de admissão. Tem o objetivo de atender os requisitos de atraso máximo associado aos fluxos de dados de tempo-real intolerantes. Uma aplicação obtém uma banda mínima que é indisponibilizada para outros tráfegos. [3]

Serviço de Carga Controlada - Procura espelhar o desempenho de uma rede de pouca carga ou sem congestionamento. Seus principais objetivos de serviço são que uma percentagem elevada de pacotes seja entregue ao destino com sucesso e que o atraso experimentado pela maioria dos pacotes não exceda muito o atraso mínimo experimentado na média por todos os pacotes do fluxo. Esse serviço é adequado a aplicações de tempo-real tolerantes, pois essas aplicações funcionam muito bem quando a rede não está carregada, mas o seu desempenho decai rapidamente à medida que a rede fica mais carregada. [3] [26]

Framework de Implementação

O roteador deve disponibilizar recursos para cada fluxo de acordo com o modelo de serviço definido. Ele recebe a função de controle de tráfego, que é implementado através de algumas componentes, tais como: escalonador de pacotes, classificador e controle de admissão. [26]

O escalonador de pacotes gerencia o encaminhamento dos diferentes fluxos de pacotes através de um conjunto de filas. Um estimador é utilizado para medir o tráfego de saída e permitir o estabelecimento de estatísticas que permitam avaliar o comportamento do escalonador de pacotes e do controle de admissão.

O classificador é encarregado da tarefa de mapeamento e atua baseado no conteúdo do cabeçalho do pacote ou, eventualmente, em alguma informação adicional de classificação incluída no pacote. Isso viabiliza o controle de tráfego e funções de bilhetagem. O controle de admissão é utilizado nos roteadores e nos terminais para determinar se um nível de QoS requisitado pode ser concedido a um novo fluxo sem comprometer os níveis de QoS já garantidos para outros fluxos.

O Resource Reservation Protocol (RSVP)

Na arquitetura de IntServ, é necessário um protocolo para a criação e manutenção do estado de cada fluxo nos roteadores e nos terminais ao longo do caminho percorrido pelo fluxo. RSVP é o protocolo responsável por isso, estabelecendo reservas de recursos para atender os requisitos de qualidade de serviço de um determinado fluxo. [26]

As principais informações transportadas pelo RSVP incluem: [26]

- informação sobre a rota (*last hop*);
- caracterização do tráfego;
- especificação da QoS para um determinado fluxo.

O receptor é o elemento encarregado de realizar as reservas de recursos. Essas reservas são estabelecidas no sentido inverso ao tráfego de pacotes de dados. [26]

A interação do RSVP ocorre para a determinação das rotas necessárias à transmissão dos pacotes de dados e controle, ficando claro que o RSVP não é um protocolo de roteamento. De modo a manter a compatibilidade com o caráter não orientado à conexão do protocolo IP, RSVP introduz o conceito de *soft state*. Isso significa que cada reserva de largura de banda armazenada em um roteador tem um temporizador associado. Se o temporizador de reserva expirar, a reserva será removida; se um receptor deseja manter uma reserva, ele deve renová-la periodicamente, enviando mensagens de reserva. [26]

Processo de Sinalização

O processo é ilustrado na Figura 3.3. O transmissor envia uma mensagem PATH para o receptor especificando características do tráfego. Cada roteador intermediário ao longo do caminho, passa a mensagem PATH para o próximo *hop*, determinado pelo protocolo de roteamento. Ao receber uma mensagem PATH, o receptor responde com uma mensagem RESV para requisitar recursos para o fluxo. Cada roteador intermediário ao longo do caminho, pode rejeitar ou aceitar as requisições da mensagem RESV. Se a mensagem for rejeitada, o roteador enviará uma mensagem de erro para o receptor e, o processo de sinalização terminará. Se a sinalização for aceita, largura de banda no enlace e espaço nos *buffers* são alocados para cada fluxo, e as informações de estado do fluxo relatado, serão instaladas no roteador.

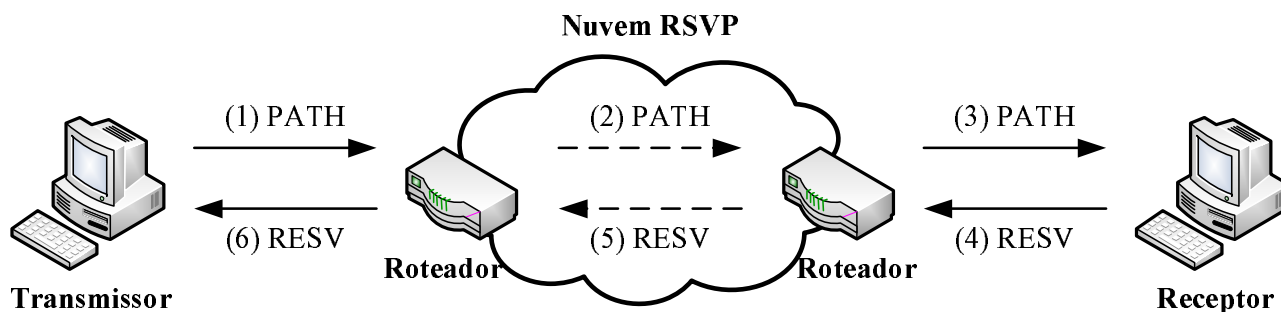


Fig. 3.3: Processo de sinalização.

Considerações Importantes

A capacidade da arquitetura IntServ de requisitar reservas de recursos por fluxo faz com que seja possível fornecer qualidade de serviço para fluxos individuais. No entanto, a reserva de recursos por fluxo usando o RSVP, implica na necessidade de um roteador para processar as reservas e para manter o estado por fluxo para cada fluxo que passa pelo roteador. Dessa maneira, o processamento de reserva por fluxo representa considerável sobrecarga para redes de grande porte. [3]

A estrutura IntServ fornece um pequeno número de classes de serviços pré-especificadas. Esse conjunto específico de classes de serviços não comporta definições mais qualitativas ou relativas para as diferenças entre as classes [3]

3.7.2 DiffServ

Devido às limitações da arquitetura IntServ, surge uma solução mais simples, a arquitetura de Serviços Diferenciados (DiffServ). A DiffServ possui alta escalabilidade a custo de uma forte agregação

de fluxos em umas poucas classes de serviços. DiffServ emprega classificação como mecanismo de QoS e utiliza o campo TOS do datagrama IPv4. A DiffServ é um refinamento do esquema de prioridade relativas cuja principal desvantagem é assegurar desempenho às aplicações apenas em termos relativos, ou seja, só garante que aplicações de classes diferentes obterão tratamento diferenciado. Todavia, ela sozinha não especifica a quantidade de recurso a ser disponibilizada para cada classe. A DiffServ emprega um mecanismo de priorização denominado de comportamento por *hop* (*per hop behavior* - PHB). PHBs definem comportamentos padrões para o encaminhamento de datagramas. [26]

Um PHB agrega fluxos gerados por diferentes aplicações e define uma classe de serviço. Nós DS alocam recursos para PHB suportado e empregam uma política de escalonamento de forma que o comportamento associado ao PHB seja satisfeito. A definição de PHBs é flexível, sendo função de grandezas, tais como: [26]

- recursos (banda, *buffer*);
- parâmetros de QoS (atrasos, *jitter*, taxa de perda);
- presença de outros PHBs.

DiffServ emprega *codepoints* para a identificação de PHBs (DiffServ *codepoints* - DSCP) que ocupam 6 bits do campo TOS (IPv4). A atribuição de PHBs à datagramas IP requer um contrato entre domínios (Figura 3.4). Esse contrato é chamado de SLA (*Service Level Agreement*). Ele estabelece alguns parâmetros de QoS que devem ser respeitados pelo domínio upstream (domínio que está injetando pacotes em outro domínio). O SLA é traduzido para parâmetros de rede como banda, taxa máxima de pacotes, tamanho de rajada, *jitter*, atraso máximo, etc., em um documento auxiliar conhecido por SLS (*Service Level Specification*). Esses contratos também existem entre um cliente final e a rede ao qual pertence. [26]

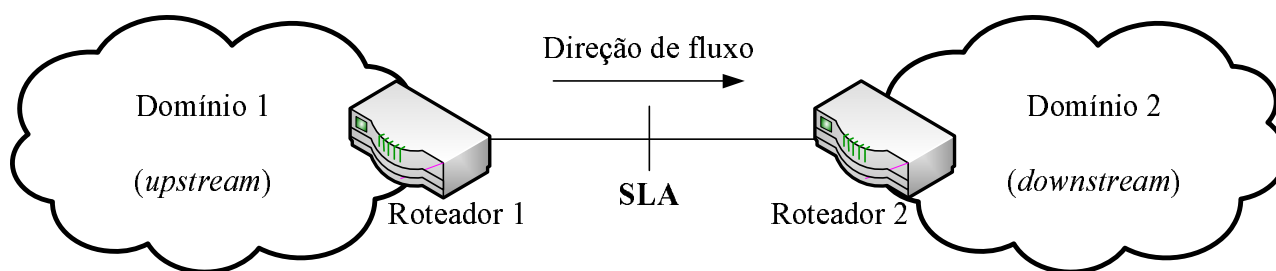


Fig. 3.4: Acordo de Nível de Serviço (SLA) entre dois domínios.

Arquitetura Funcional

A Figura 3.5 mostra a arquitetura DiffServ em termos funcionais. O classificador seleciona os pacotes recebidos nas interfaces de entrada, baseado no conteúdo de alguma parte do seu cabeçalho. Foram definidos dois tipos principais de classificadores. O classificador BA (*Behavior Aggregate*) classifica os pacotes, baseado somente no conteúdo do campo DSCP. Esse caso ocorre quando o domínio anterior é compatível com DiffServ e os pacotes já vêm marcados. Quando o domínio anterior não é habilitado para enviar os pacotes com o campo DSCP previamente marcado, o classificador pode avaliar vários campos dos pacotes. Nesse caso, ele é chamado de classificador MF (*Multi-Field*). [26]

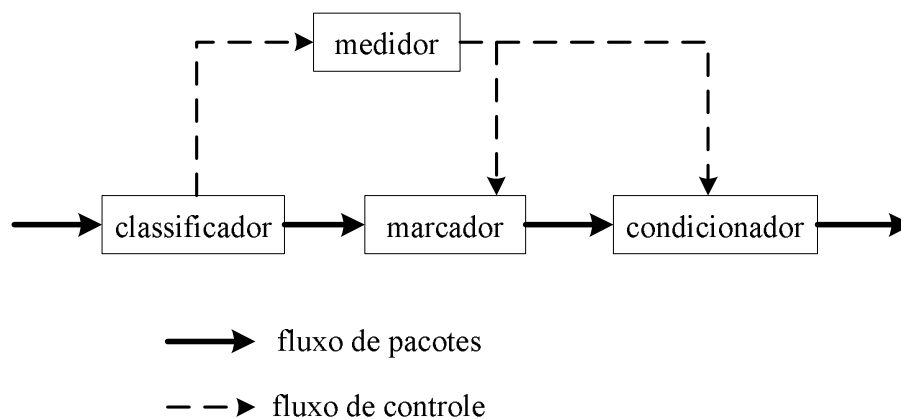


Fig. 3.5: Arquitetura em termos funcionais.

O medidor de tráfego mede as propriedades temporais de um fluxo de pacotes selecionados pelo classificador de acordo com o perfil de tráfego especificado. [26]

O marcador atribui um *codepoint* ao datagrama, ou altera um valor previamente atribuído. O *codepoint* atribuído ao datagrama é função do estado informado pelo medidor. [26]

O condicionador molda o tráfego para que fique forçosamente dentro do perfil contratado. Descarta pacotes que foram considerados fora do perfil pelo medidor, para que o fluxo seja considerado dentro do perfil. [26]

Per Hop Behavior (PHB)

A implementação de um BA requer que todos os pacotes recebam o mesmo tratamento de encaminhamento nos roteadores por onde passam. Um PHB (*Per-Hop Behavior*) é uma descrição de um comportamento de encaminhamento observável externamente de um roteador DiffServ, aplicado a um determinado BA. O PHB é a maneira como um roteador aloca recursos para os BAs, e é em cima desse mecanismo local que serviços são construídos. A maneira mais simples de implementar

um PHB é destinar a ele um determinado percentual de utilização da largura de banda de um enlace de saída. [26]

Foram padronizados dois tipos de PHBs: Encaminhamento Expresso (EF) e Encaminhamento Assegurado (AF). Além desses, DiffServ deve ser compatível com as implementações e usos já existentes, assim definidos: BE (*best effort*), para o comportamento de encaminhamento de tráfego de melhor esforço e CSC (*class selector compliant*) para compatibilidade com esquemas de roteamento e encaminhamento que levam em conta o subcampo Precedência do protocolo IPv4. [26]

3.8 Mecanismos de Controle de Congestionamento

Para controle de congestionamento, as interfaces de rede necessitam de técnicas de enfileiramento de pacotes, para que possam separar os pacotes de tráfegos diferentes em filas e servi-los de forma distinta. Assim, tráfegos de maior prioridade podem receber vantagens maiores na transmissão e chegar à frente de tráfegos de menor prioridade.

Nos equipamentos da Cisco [27], por exemplo, incluem-se as seguintes técnicas de enfileiramento:

- FIFO *Queuing*;
- *Priority Queuing* (PQ);
- *Custom Queuing* (CQ);
- *Flow-Based Weighted Fair Queuing* (WFQ);
- *Class-Based Weighted Fair Queuing* (CBWFQ).

Cada técnica de enfileiramento foi desenvolvida para resolver um problema específico de tráfego de rede e tem um efeito particular no desempenho da rede. [27]

3.8.1 *First-In First-Out* (FIFO)

Entre todas as técnicas de enfileiramento, o enfileiramento FIFO é o mais simples. Ele consiste em armazenar os pacotes, em uma única fila, quando a rede está congestionada e enviá-los quando descongestionar. Como diz o nome, “primeiro a entrar, primeiro a sair”, somente a ordem de chegada determina a ordem de saída. O enfileiramento FIFO é ilustrado na Figura 3.6. [27]

O enfileiramento FIFO foi o primeiro passo necessário no controle de tráfego, mas ele acumula várias deficiências como são mostradas a seguir. [27]

- Não toma qualquer decisão em relação à prioridade de pacotes. Independente da prioridade, todos os pacotes serão tratados da mesma forma.

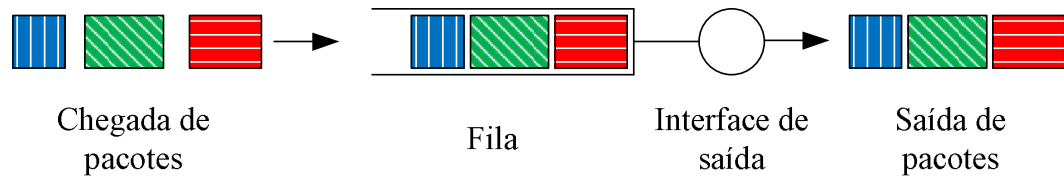


Fig. 3.6: Enfileiramento FIFO.

- Não oferece proteção contra aplicações com tráfegos mal comportados. Tráfegos com rajadas podem causar grandes demoras na entrega de tráfegos sensíveis a atrasos.
- Quando a fila está completamente cheia, os pacotes que chegam são descartados. Isto é indesejado, pois o pacote descartado pode ser um pacote de alta prioridade.

FIFO é uma técnica inadequada para suporte à qualidade de serviço por ser incapaz de diferenciar pacotes e tratá-los de acordo com seus requisitos de QoS. Isto tem motivado o desenvolvimento de novas técnicas de enfileiramento. [26]

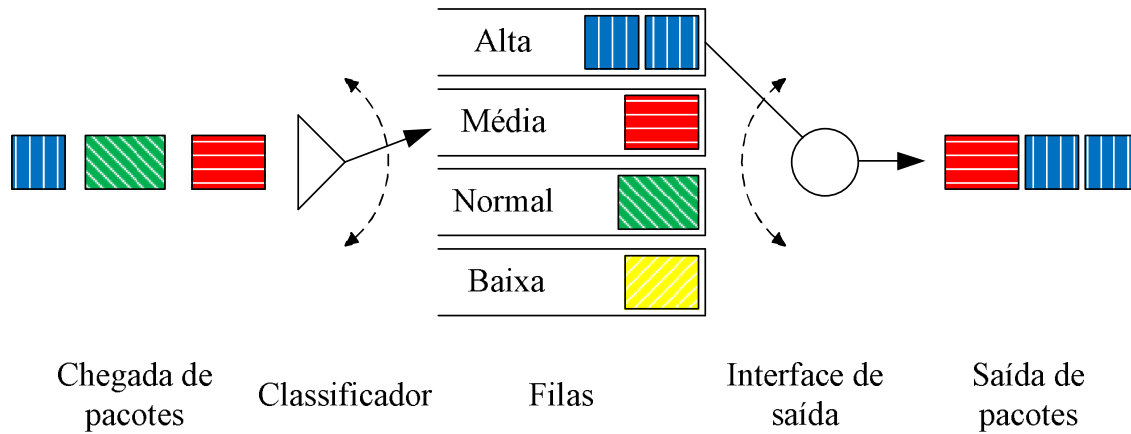
3.8.2 Priority Queuing (PQ)

A técnica de enfileiramento PQ foi desenvolvida para dar prioridade rígida para tráfegos importantes fazendo uso de múltiplas filas. Cada pacote é classificado e colocado em uma fila com prioridade correspondente (para a Cisco, são quatro filas: alta, média, normal e baixa) [27]. Na classificação de prioridades dos pacotes podem-se usar critérios como protocolos de rede (IP, IPX, etc.), de transporte (TCP, UDP) ou tipo de aplicação (VoIP, HTTP, etc.) [26].

Durante a transmissão, tratamento preferencial absoluto é entregue às filas de maiores prioridades diante das filas de menores prioridades. Somente quando todas as filas de maiores prioridades estão vazias é que os pacotes das filas de prioridades menores serão servidos, como pode ser visto na Figura 3.7. Estes pacotes são servidos até a fila ficar vazia, ou outro pacote chegar em uma fila de maior prioridade. [27]

PQ, ao contrário do enfileiramento FIFO, consegue diferenciar pacotes e entregá-los com tratamento diferenciado, mas ainda tem algumas desvantagens que são apresentadas a seguir. [26]

- Filas de baixa prioridade podem permanecer muito tempo sem serem servidas caso um fluxo constante de pacotes continuem a chegar em uma fila de prioridade maior. Pacotes esperando em filas de menor prioridade podem nunca serem servidos, fenômeno chamado de “inanição”;
- Nenhuma garantia de atraso e *jitter* pode ser dada, empregando-se o esquema de prioridades;
- As prioridades são configuradas de forma estática, logo, não se adaptam automaticamente a mudanças nas condições de rede;

Fig. 3.7: *Priority Queuing* (PQ).

- Filas individuais são processadas como enfileiramento FIFO, portanto, para cada fila individual, as desvantagens do FIFO se manifestam.

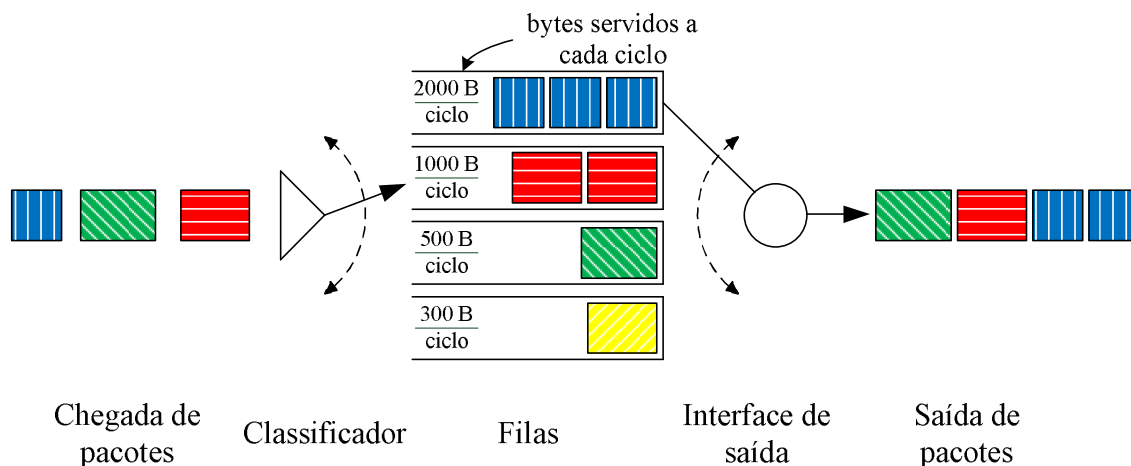
3.8.3 *Custom Queuing* (CQ)

O *custom queing* (CQ), ou enfileiramento customizado, é uma técnica de enfileiramento que foi desenvolvida para que várias aplicações compartilhem a rede entre elas com requisito mínimo de largura de banda [27]. Esta técnica é também conhecida como *Weighted Round Robin* (WRR) ou ainda *Class Based Queuing* (CBQ) [26]. Pode-se usar o CQ para compartilhar a largura de banda proporcionalmente com as necessidades de cada aplicação e usuário. Assim, assegura que um tráfego específico receba porção fixa de largura de banda e deixando a largura restante para outros tráfegos [27].

A técnica de enfileiramento CQ coloca cada classe de pacotes em filas diferentes que serão servidas de forma cíclica (*round robin*). As filas são configuradas com prioridades que são dadas em pesos. O peso da fila determina o número de bytes que serão servidos em cada ciclo. Sendo assim, nenhuma aplicação (ou classe de tráfego especificado) obtém mais que uma proporção predeterminada do total da capacidade quando rede estiver congestionada. [27]

A Figura 3.8 ilustra o enfileiramento CQ, seja uma interface onde são definidas 4 filas. Para cada fila é estabelecido um número de bytes que o CQ utiliza para decidir a troca de fila. Quanto maior o peso da fila maior é este número. Por exemplo, 2000, 1000, 500, 300 bytes para as 4 filas. A cada ciclo de serviço um contador associado a cada fila é iniciado com o número de bytes da fila. O CQ serve a fila de maior prioridade subtraindo do contador o tamanho do pacote transmitido. Quando o contador atingir um valor menor ou igual a zero, ou a fila se esvaziar, a próxima fila é servida. No exemplo, 2000 bytes da primeira fila são transmitidos, 1000 da segunda, e assim por diante. [26]

Ao contrário do PQ, que somente começa a servir a próxima fila quando a fila de prioridade maior

Fig. 3.8: *Custom Queuing (CQ)*.

estiver totalmente servida, o CQ também passa a servir a próxima fila quando o percentual de largura de banda atribuído à fila for atingido. Essa característica resolve o problema de “inanição” nas filas de baixa prioridade, mas as demais desvantagens do enfileiramento PQ ainda persistem. [26]

3.8.4 *Flow-Based Weighted Fair Queuing (WFQ)*

Em redes nos quais é desejável garantir tempo de resposta consistente (como nas arquiteturas IntServ e DiffServ), tanto para tráfegos pesados quanto a leves, de maneira que não se comprometa excessiva largura de banda, a solução é usar *flow-based* WFQ (comumente chamado de somente WFQ). É uma técnica de enfileiramento baseada em fluxos, permitindo que cada fila seja servida de forma justa em termos de contagem de bytes. Por exemplo, caso uma fila tenha pacotes de 100 bytes e outra fila pacotes de 50 bytes, o WFQ serve 2 pacotes da primeira fila para cada pacote da segunda fila. Isso faz o serviço justo para cada fila: 100 bytes para cada vez que a fila é servida. Esse exemplo é ilustrado na Figura 3.9. [27]

O WFQ é inspirado na justiça do algoritmo *Bit-by-bit Round Robin (BRR)*. No BRR cada fluxo é colocado em uma fila exclusiva e um bit é enviado a cada ciclo de serviço. Portanto, o BRR divide a largura de banda igualmente para N fluxos. A implantação de algoritmo é inviável devido ao *overhead* gerado por se transmitir um bit por vez através do enlace. [26]

Uma técnica de enfileiramento chamado de *Fair Queuing (FQ)* simula o BRR sem a restrição de servir as filas bit-a-bit, enviando os pacotes integralmente através do enlace. Ele procura servir cada pacote no mesmo tempo que seria servido caso o BRR estivesse em uso. O WFQ passa a ser uma variação do FQ que permite atribuir pesos aos fluxos. Ao simular BRR, ao invés de transmitir um único bit, o WFQ supõe a transmissão de uma quantidade de bits proporcional ao peso do fluxo. [26]

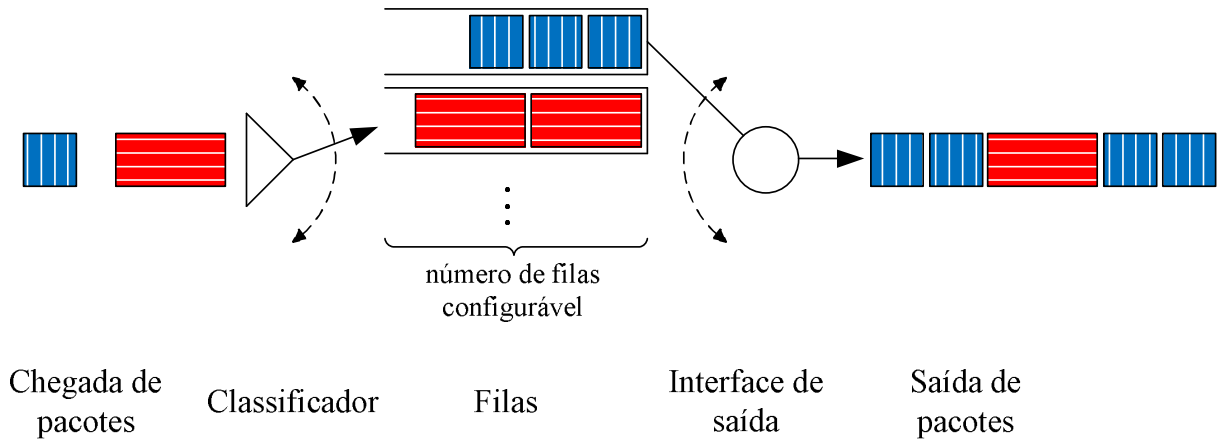


Fig. 3.9: *Flow-Based Weighted Fair Queuing (WFQ)*.

O WFQ utiliza o campo precedência (são os 3 primeiros bits do campo TOS) do cabeçalho IP para determinar os pesos para cada fluxo. Ele possui valores entre 0 (valor padrão) e 7. Na medida em que o valor de precedência aumenta, o WFQ aloca mais largura de banda para o fluxo para garantir que seus pacotes sejam servidos mais rapidamente em momento de congestionamento. [27]

Por exemplo, caso se tenha um fluxo para cada nível de precedência em uma interface, cada fluxo terá o peso precedência+1, e o total será:

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$$

Assim, os fluxos obterão 1/36, 2/36, 3/36 da largura de banda e assim segue. Entretanto, caso haja 18 fluxos com precedência 1 e um com precedência 7, a soma fica:

$$2 \times 18 + 8 = 44$$

Assim, 18 fluxos obterão 2/44 da largura de banda enquanto um obterá 8/44.

Como o WFQ diferencia cada fluxo e lhe garante largura de banda, ele consegue proteger a rede de tráfegos em rajadas. Assegura que filas não sofram “inanição” de largura de banda e que o tráfego receba um serviço previsível. O WFQ é projetado para minimizar necessidades de configurações, e se adapta automaticamente a mudanças das condições de tráfego da rede [27]. Porém, possui algumas desvantagens, com segue: [26]

- O algoritmo para implementar o WFQ é mais complexo que as técnicas apresentadas anteriormente;
- O WFQ precisa manter o estado de cada fluxo, pois o cálculo da parcela de largura de banda do enlace é em função de quais fluxos estão ativos.

Como WFQ permite tratar um fluxo específico ou um conjunto de fluxos agregados numa classe de serviço, esta política é adequada tanto para IntServ que opera por fluxo quanto para DiffServ que opera por classe de serviço. A atribuição dos pesos se dá em função dos parâmetros da reserva para o caso de IntServ ou do campo DS presente no datagrama IP (ou precedência com já foi visto) para o caso de DiffServ. [26]

3.8.5 *Class-Based Weighted Fair Queuing (CBWFQ)*

Existem situações em que se precisa garantir uma porção mínima da largura de banda. Nessas situações, deve-se usar a técnica de enfileiramento *Class-Based WFQ (CBWFQ)*. O CBWFQ permite ao gerente de rede criar classes. Ao invés de prover uma fila individual para cada fluxo, uma classe consiste de um ou mais fluxos. Para cada classe pode ser garantido uma porção mínima de largura de banda. [27]

Um exemplo no qual o CBWFQ pode ser usado é em prevenir que múltiplos fluxos de baixas prioridades prejudiquem um fluxo de alta prioridade. Por exemplo, um fluxo de vídeo que precisa da metade da largura de banda de um E1, será fornecido se houver dois fluxos. Quanto mais fluxos forem somados, o fluxo de vídeo obtém menos largura de banda, pois o WFQ cria justiça entre os fluxos. Se houver 10 fluxos, o fluxo de vídeo receberá somente 1/10 do total da largura de banda, o que é insuficiente. Mesmo configurando o valor de precedência igual a 5, não resolve o problema. [27]

$$1 \times 9 + 6 = 15$$

O vídeo obtém 6/15 da largura de banda, o que é ainda insuficiente. Um mecanismo deve ser invocado para garantir a metade da largura de banda necessária para o vídeo. O CBWFQ possibilita isso. O gerente de rede define uma classe, coloca o fluxo de vídeo na classe e informa ao roteador para fornecer 1024 kbps (metade de um E1) de serviço a classe. Então o vídeo obtém a largura de banda que necessária. Uma classe padrão é usada para o resto dos fluxos. Essa classe é servida pelo esquema WFQ, alocando o restante da largura de banda (no caso, a outra metade do E1). [27]

Capítulo 4

Processo Poissoniano Modulado por Markov - MMPP

4.1 Introdução

A implantação de VoIP é uma tarefa desafiadora para engenheiros e projetistas de rede de IP. O sucesso de tal implantação depende da avaliação da capacidade da rede. Todavia, isto requer uma avaliação acerca da capacidade de uma rede IP ser usada para fazer ligações VoIP ainda em fase de pré-projeto. Tudo isto depende de uma boa metodologia. [28]

Quando se implanta o serviço de VoIP em uma rede, muitos gerentes de rede se fazem os seguintes questionamentos:

- Quais são os requisitos de QoS de uma conversação?
- Quantos tráfegos de VoIP a minha rede consegue suportar e satisfazer os requisitos padronizados de QoS?
- Quanto de recurso devo alocar para VoIP para atender adequadamente uma demanda de tráfego?

Essas e outras perguntas têm levado o desenvolvimento de algumas ferramentas comerciais para testar o desempenho de VoIP em redes IP. A maioria dessas ferramentas usa duas abordagens para avaliar a implantação: (1) através de medições, avaliar a saúde da rede existente e depois prever a capacidade de suporte a VoIP; (2) por meio de injeção de tráfego de VoIP na rede medir parâmetros resultantes, tais como latência, *jitter* e perda de pacotes. [28]

Segundo Salah [28], nenhuma dessas ferramentas disponibilizam uma abordagem compreensiva para implantar VoIP com sucesso. Ele desenvolveu uma metodologia para analisar o desempenho do tráfego de VoIP em uma rede. Essa análise é baseada em teoria de filas usando os modelos M/M/1 e

M/D/1. Estes modelos supõem que as chegadas são poissonianas, aproximação aceitável quando se tem um tráfego constante de pacotes de voz. Quando as fontes de tráfego de VoIP usam o mecanismo de supressão de silêncio, as chegadas ocorrem em rajadas, logo, chegadas poissonianas se tornam um processo não representativo. É necessário um outro processo de chegadas que consegue descrever o comportamento variável do tráfego.

Na literatura, surgiram vários modelos de tráfegos auto-similares que conseguem modelar melhor chegadas em rajadas, entre eles, pode-se destacar *Chaotic map*, FARIMA e *Fractional Brownian Motion* (FBM). Este último possui como extensão o modelo *Multi-fractal*. Esses modelos são muito eficientes computacionalmente, mas a maioria é considerada complexa e é difícil entender a relação entre os coeficientes do modelo e os parâmetros do tráfego real. Em contraparte a esses modelos citados, uma das abordagens mais simples e mais frequentemente usadas para modelar um processo estocástico com característica de rajadas é o *Markov Modulated Poisson Process* (MMPPs). Este modelo tem o apelo de ser simples, fácil de entender, intuitivo, e acima de tudo, pode ser tratado matematicamente usando um pequeno número de parâmetros. [29]

Neste trabalho faz-se uso do processo MMPP. Vários autores ([29]-[36]) têm mostrado que o MMPP é adequado para tráfegos de VoIP com supressão de silêncio.

4.2 Modelagem MMPP

As redes IP têm sido usadas para transportar tráfegos das mais variadas aplicações. Tráfegos de voz, vídeo e dados possuem propriedades muito diferentes. Um modelo de tráfego deve ser flexível e capaz as propriedades dessas aplicações. Em geral, o tráfego nas redes IP não é suave, ocorre em rajadas e o tempo entre chegadas de pacotes é altamente correlacionado. Em muitos casos, isso pode resultar efeitos devastadores no desempenho em alguns cenários. Os modelos de tráfego devem ter a habilidade de reproduzir este efeito, podendo ser previsto em que circunstâncias esse efeito devastador ocorre. [37]

O MMPP tem sido largamente usado para modelar esses processos, pois ele modela qualitativamente a variação no tempo da taxa de chegadas e captura algumas das importantes correlações entre o tempo das chegadas, enquanto permanece analiticamente simples. O MMPP, como já diz o nome, é um processo de Poisson duplamente estocástico, em que a transição entre os estados segue uma cadeia de Markov. Cada estado possui uma taxa de chegadas que obedece a uma distribuição de Poisson [38]. A Figura 4.1 ilustra o diagrama de transição de um MMPP qualquer.

O MMPP é parametrizado por uma matriz geradora \mathcal{Q} e uma matriz diagonal \mathcal{L} , contendo as taxas de chegadas para cada estado do processo markoviano. Estas duas matrizes são dadas a seguir:

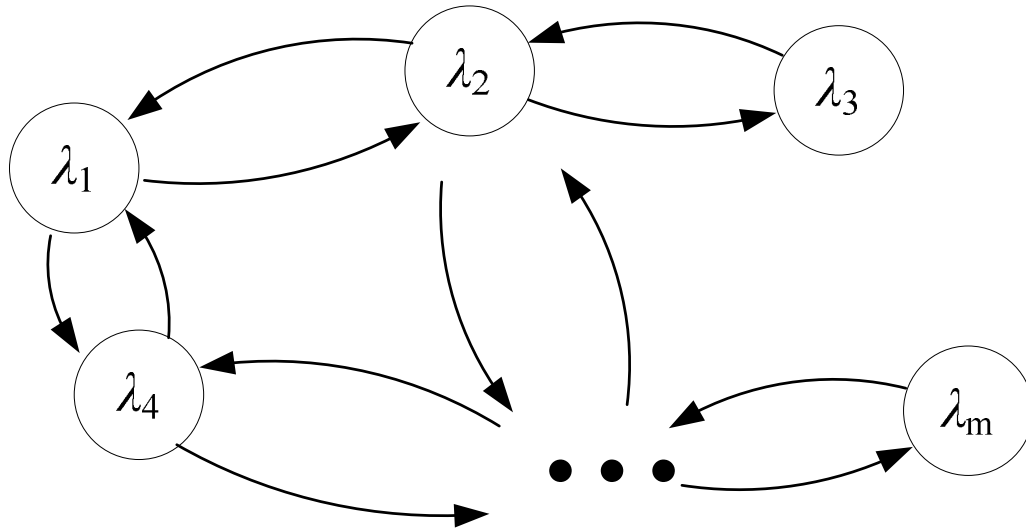


Fig. 4.1: Diagrama de transição de um MMPP qualquer.

$$Q = \begin{bmatrix} -\sigma_{11} & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1m} \\ \sigma_{21} & -\sigma_{22} & \sigma_{23} & \cdots & \sigma_{2m} \\ \sigma_{31} & \sigma_{32} & -\sigma_{33} & \cdots & \sigma_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \sigma_{m3} & \cdots & -\sigma_{mm} \end{bmatrix} \quad (4.1)$$

e

$$\mathcal{L} = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_m \end{bmatrix}. \quad (4.2)$$

O vetor de probabilidade dos estados estacionários da cadeia de Markov é π , de forma que:

$$\pi Q = 0, \quad \pi \varepsilon' = 1, \quad (4.3)$$

onde $\varepsilon' = (1, 1, \dots, 1)^T$ é um vetor coluna de largura m . [38]

Vários estudos têm mostrado que MMPP tem flexibilidade suficiente para descrever uma larga variedade de tráfegos com processos de chegada em rajada e com correlação [30] [39] [40]. Trabalhos usaram o MMPP para estudar tráfegos específicos como: voz [30]-[36], video [30] [41] [42] e

dados [37] [40] [43] [44] [45].

4.3 Caracterização do Tráfego VoIP

Os algoritmos de codificação de voz têm uma taxa de transmissão de bits fixa e um tempo fixo de empacotamento. A fonte é caracterizada por um fluxo constante de pacotes durante o período ativo (período *ON*) e nenhuma chegada durante o silêncio (período *OFF*). Durante o período *ON* a fonte envia pacotes em intervalos regulares de tamanho T (tempo de empacotamento), como pode ser visto na Figura 4.2. A duração dos períodos ativos e inativos é geralmente estimada por distribuições exponenciais independentes com parâmetros α e β . A média de duração do período *ON* é dada por $T_{on} = 1/\alpha$ e a média de duração do período *OFF* por $T_{off} = 1/\beta$. A média de taxa de bits do modelo *ON/OFF* é igual à:[35]

$$\bar{\lambda} = \frac{T_{on}}{(T_{on} + T_{off}) \cdot T} = \frac{\beta}{(\beta + \alpha) \cdot T} = \frac{P_{on}}{T}. \quad (4.4)$$

Sendo P_{on} a probabilidade de uma fonte estar no estado *ON*, calculada pela seguinte equação:

$$P_{on} = \frac{T_{on}}{(T_{on} + T_{off})} = \frac{\beta}{(\beta + \alpha)}. \quad (4.5)$$

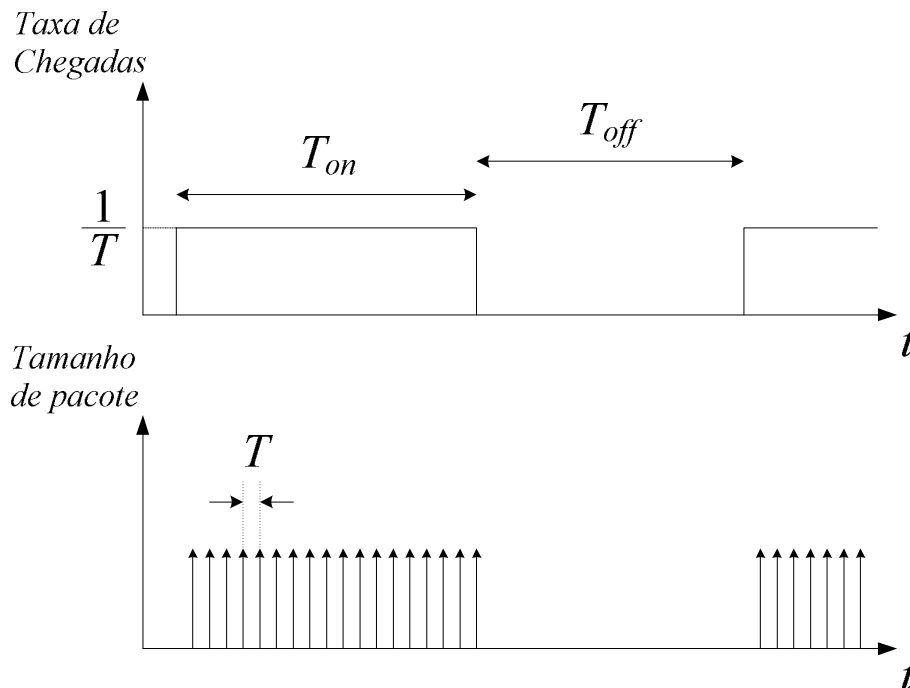


Fig. 4.2: Fluxo constante de pacotes durante períodos *ON*.

4.3.1 Períodos *ON/OFF*

Os valores de T_{on} e T_{off} usados para modelar a atividade da voz não são únicos na literatura. Os valores mais usados são os do modelo de May e Zebo [17], e do modelo de Brady [16]. Hassan [35] trabalha com três modelos: o modelo de May e Zebo, que ele denomina *Clássico*; os dois outros modelos são tirados das medições de tráfego telefônico e áudio em redes de comutação de pacotes feitas por Deng [46], que distinguem duas categorias de aplicações de áudio, *Leitura* e *Conversação*. A Tabela 4.1 mostra os valores dos modelos de atividade de voz.

<i>Modelo</i>	T_{on} (s)	T_{off} (s)	P_{on}
May e Zebo (<i>Clássico</i>)	0.352	0.65	0.351
Brady	1.004	1.587	0.387
Deng (<i>Conversação</i>)	7.24	5.69	0.560
Deng (<i>Leitura</i>)	3.23	0.41	0.887

Tab. 4.1: Modelos de atividade de voz.

A diferença entre esses modelos pode ser explicada por sensibilidade de silêncio dos codecs usados. Além disso, a atividade do modelo *Leitura* é mais fluida que o *Conversação*, o que resulta períodos de silêncio curtos. [35]

4.4 Aproximação do Tráfego VoIP pelo modelo MMPP

Quando se modela um tráfego unitário de VoIP, durante o período *ON*, a chegada constante de pacotes é aproximada por chegadas poissonianas, fazendo com que o intervalo entre as chegadas de pacotes τ obedeça a uma distribuição exponencial, como pode ser visto na Figura 4.3. Na Figura 4.4 é mostrado o tráfego unitário modelado por um MMPP de 2 estados na forma de um diagrama de transição, onde no estado *ON* a intensidade de chegadas poissonianas λ_{on} e no estado *OFF* não há chegadas, $\lambda_{off} = 0$. Este MMPP específico, em que o tráfego é interrompido em alguns instantes, também é conhecido como *Interrupted Poisson Process* (IPP) [38].

Sabendo que a soma de processos poissonianos independentes resulta em um novo processo poissoniano [47], é possível somar as intensidades das fontes que estão em estado *ON* resultando em uma nova intensidade, essa propriedade determina a simplicidade do MMPP para descrever tráfegos sobrepostos. No entanto, não é tão simples quando se tem intervalos determinísticos entre chegadas. Isso é porque as diferentes fontes estarão dessincronizadas, assim os tráfegos sobrepostos não serão determinísticos, sendo difícil de descrever matematicamente [48].

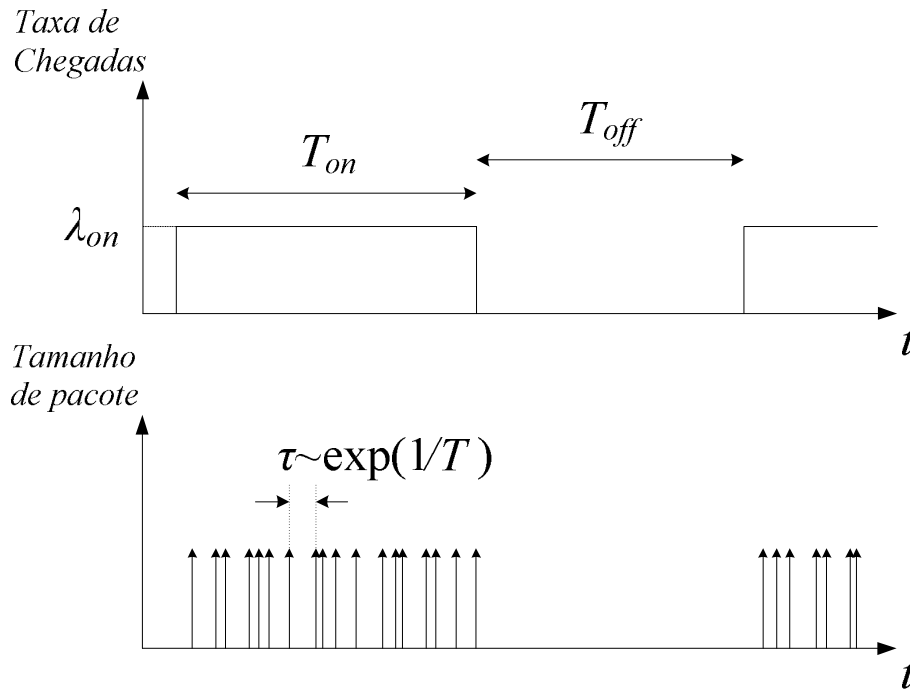


Fig. 4.3: Chegadas poissonianas durante períodos ON.

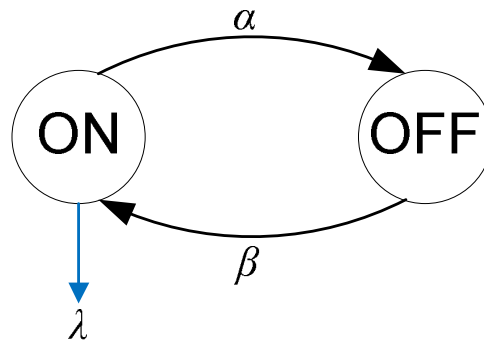


Fig. 4.4: Diagrama do tráfego unitário de VoIP modelado por um MMPP de 2 estados.

Quando há N idênticos MMPPs de 2 estados (N MMPPs modelando N tráfegos unitários de VoIP, usando o mesmo modelo de atividade de voz), a sobreposição dos mesmos podem ser modelados por um MMPP de $N + 1$ estados em que a transição dos estados forma um processo de nascimento e morte, como pode ser visto na Figura 4.5. Cada estado representa um número de vozes ativas. Quando o processo markoviano está em um estado n , as chegadas de pacotes ocorrem de acordo com um processo poissoniano de taxa λ_n . [38]

Os campos da matriz Q podem ser calculados pelas seguintes equações: [38]

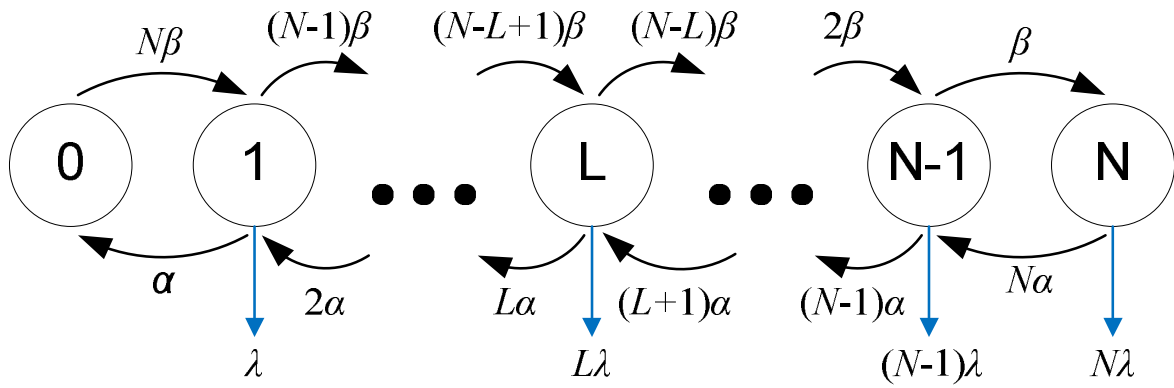


Fig. 4.5: Diagrama do tráfego agregado de N fontes de VoIP modelado por um MMPP de $N + 1$ estados.

$$\begin{aligned}
 \sigma_{i,i} &= -[(i-1)\alpha + (N+1-i)\beta], & \text{para } 1 \leq i \leq N+1 \\
 \sigma_{i,i-1} &= (i-1)\alpha, & \text{para } 2 \leq i \leq N+1 \\
 \sigma_{i,i+1} &= (N+1-i)\beta, & \text{para } 1 \leq i \leq N \\
 \sigma_{i,j} &= 0, & \text{para } 1 \leq i \leq N
 \end{aligned} \tag{4.6}$$

e

$$\lambda_i = (i-1)\lambda, \quad \text{para } 1 \leq i \leq N+1 \tag{4.7}$$

A ordem das matrizes \mathcal{Q} e \mathcal{L} é então:

$$\dim(\mathcal{Q}) = \dim(\mathcal{L}) = N+1. \tag{4.8}$$

Assim, as matrizes de parametrização do MMPP de chegadas de VoIP sobrepostas têm os seguintes aspectos:

$$\mathcal{Q} = \begin{bmatrix} -N\beta & N\beta & 0 & \cdots & 0 \\ \alpha & -[\alpha + (N-1)\beta] & (N-1)\beta & \cdots & 0 \\ 0 & 2\alpha & -[2\alpha + (N-2)\beta] & \ddots & \\ \vdots & \vdots & \ddots & \ddots & \beta \\ 0 & 0 & & N\alpha & -N\alpha \end{bmatrix} \tag{4.9}$$

e

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_{on} & 0 & \cdots & 0 \\ 0 & 0 & 2\lambda_{on} & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n\lambda_{on} \end{bmatrix}. \quad (4.10)$$

4.4.1 Auto-similaridade

Tráfegos com chegadas em rajadas são comumente relacionados a comportamentos de auto-similaridade. A auto-similaridade é uma propriedade referente à invariância da distribuição para diferentes escalas de tempo. Na Figura 4.6, um processo auto-similar é observado em diferentes escalas de tempo. Para um inteiro positivo m , um novo processo $X_k^{(m)}$ é construído, calculando-se médias de blocos sequenciais m do processo original, como mostra a equação a seguir: [49]

$$X_k^{(m)} = \frac{(X_{km} + \dots + X_{k(m+1)m-1})}{m}. \quad (4.11)$$

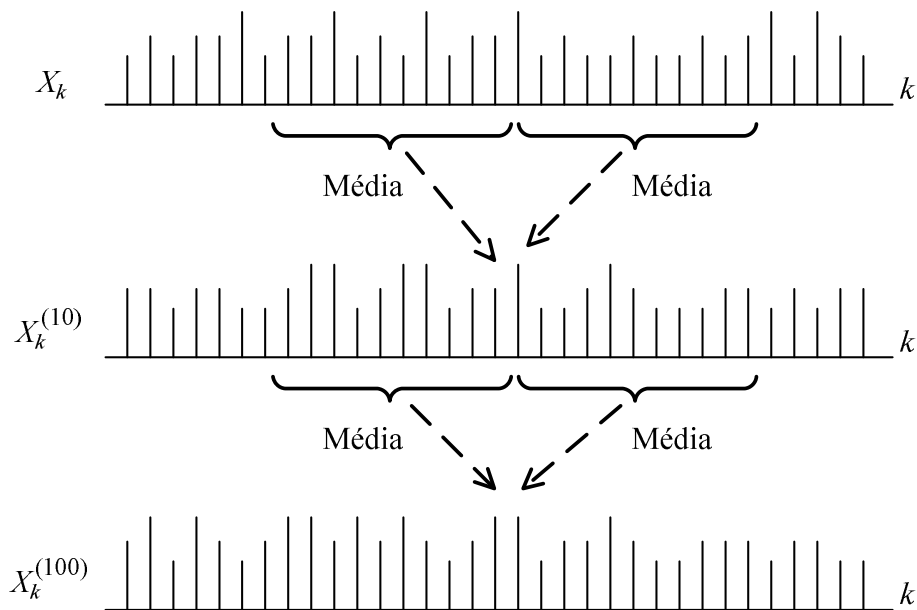


Fig. 4.6: Processo auto-similar em diferentes escalas de tempo.

Essencialmente, auto-similaridade significa que um novo processo $a^{-H} X_{ak}$ possui as mesmas

propriedades estatísticas que o processo original X_k em uma grande faixa de escalas $a > 0$ e algum parâmetro $0 < H < 1$. Isso não é verdadeiro para o processo poissoniano, que tende a ficar mais suave para m grandes. [49]

O parâmetro Hurst H [50] mede o grau de auto-similaridade. A faixa de interesse de H é $[0.5, 1]$, sendo que $H = 0.5$ significa ausência de auto-similaridade e $H = 1$ processo exatamente auto-similar.

A Tabela 4.2 apresenta parâmetros H estimados dos tráfegos de VoIP de acordo com números de fontes sobrepostas. O tráfego de N fontes de VoIP foram tiradas das simulações no NS-2 detalhadas no Capítulo 5, e os resultados de seus modelos MMPP correspondentes foram obtidas usando uma simulação do modelo no Matlab. É observado que o tráfego unitário de VoIP possui mínima auto-similaridade, semelhante a chegadas poissonianas, e o tráfego fica cada vez mais auto-similar com o aumento do número de fontes. É feita uma comparação com os tráfegos aproximados pelo modelo MMPP, e o modelo se mostrou coerente com o tráfego original. O método usado para estimar o parâmetro H é o *Rescaled Range Analysis* [51].

N	N Fontes Sobrepostas	MMPP de $N + 1$ Estados
1	0.5018	0.5033
5	0.5656	0.5627
10	0.6388	0.6309
50	0.6501	0.6461
100	0.6733	0.6707
500	0.7148	0.6936

Tab. 4.2: Parâmetros H estimados.

4.4.2 IDI

Para descrever a dependência entre chegadas sucessivas de um processo de chegadas, o *Index of Dispersion of Intervals* (IDI) é usado. Na literatura, é usado com uma medida de intensidade de rajadas de um sinal. Seja $\{X_k, k \geq 1\}$ a duração de um processo estocástico; assume-se que $\{X_k, k \geq 1\}$ é estacionário, o que significa que a distribuição conjunta de $(X_k, X_{k+1}, \dots, X_{k+m})$ é independente de k para todo m . A soma de k tempos entre-chegadas consecutivas é dada por $S_k = (X_1 + X_2 + \dots + X_k)$. O IDI é definido como: [48] [52]

$$c_k^2 = \frac{k \text{Var}(S_k)}{[\text{E}(S_k)]^2} = \frac{\text{Var}(S_k)}{k[\text{E}(X_1)]^2}. \quad (4.12)$$

O IDI mede a covariância acumulativa normalizada pelo quadrado da média ao longo de k interchegadas consecutivas. [35]

A Figuras 4.7 mostra as medidas de IDI para chegadas de VoIP à esquerda (calculadas apartir de resultados obtidos no NS-2) e a aproximação MMPP à direita (calculadas apartir de resultados obtidos da simulação do modelo MMPP correspondente no Matlab), para k de 1 a 1000 e N igual a 1, 5, 10, 50 e 100. Para referência, foram adicionados os valores de IDI de um processo poissoniano. O MMPP, mais uma vez, se mostra muito adequado para modelar chegadas de VoIP uma vez que seus resultados são muito semelhantes.

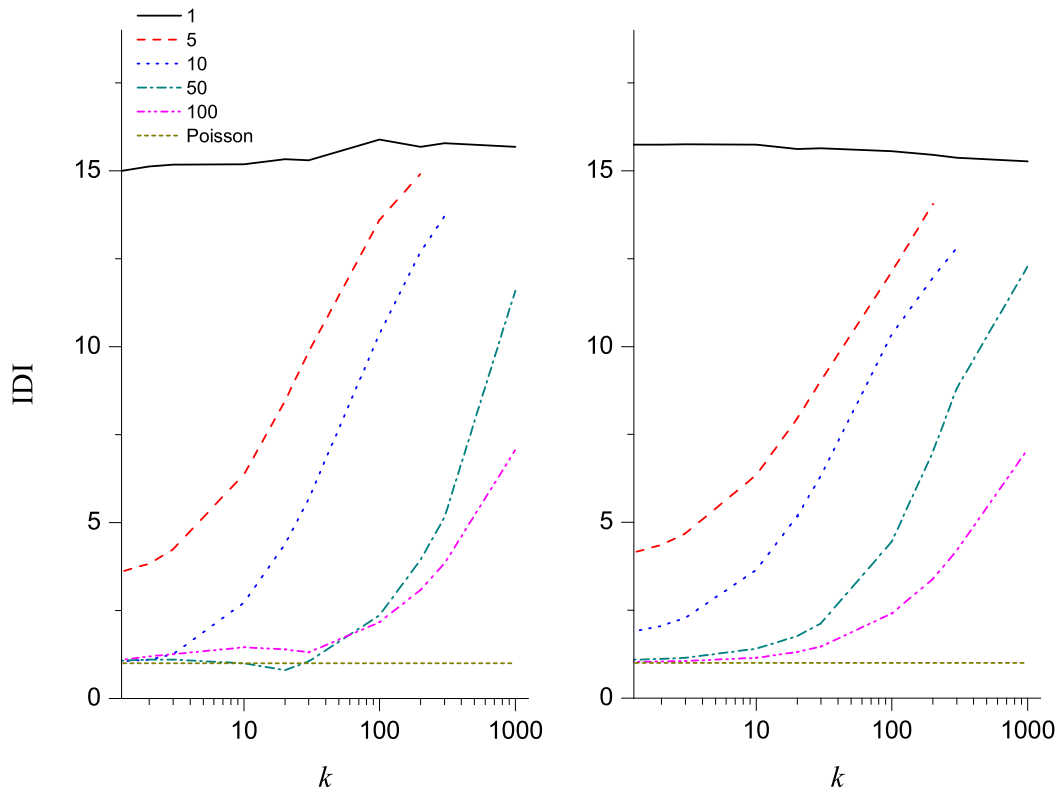


Fig. 4.7: Curvas IDI para N .

Observando os valores IDI do tráfego VoIP, características interessantes podem ser notadas: [52]

- Os valores de IDI de um tráfego unitários são altos e aproximadamente constantes;
- Em curtos intervalos, o processo sobreposto se parece com um processo poissoniano; mas em intervalos mais longos, ele se afasta e é altamente variável;
- O IDI de um tráfego sobreposto tende ao valor de um tráfego unitário para intervalos muito longos;

- Para um número muito grande de tráfegos VoIP sobrepostos, o IDI se assemelha ao IDI de um processo poissoniano.

4.5 Modelo de Fila

Para analisar o comportamento do tráfego gerado por um MMPP em um enlace, ele precisa ser injetado em um sistema de filas, como mostra a Figura 4.8. Como os pacotes de VoIP possuem tamanhos constantes, a melhor representação de um enlace com *buffer* infinito seria um uma fila do tipo MMPP/D/1. Entretanto, por ser matematicamente mais simples, usa-se uma fila do tipo MMPP/M/1. Como este tipo de fila considera que o tamanho dos pacotes obedece a uma distribuição exponencial negativa, os resultados tenderão a serem mais conservadores.

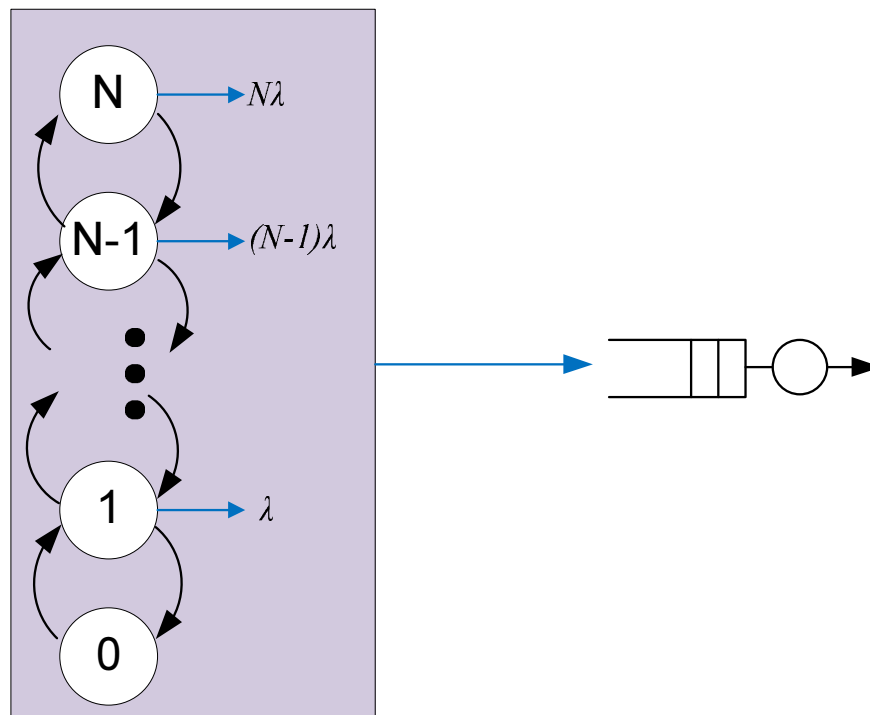


Fig. 4.8: Tráfego gerado por um MMPP injetado em um sistema de filas.

Neuts [53] usa o método matriz-analítico para calcular a distribuição probabilidade de largura e o atraso das filas SM/M/1, onde a fila MMPP/M/1 é tida como um caso especial. Este método foi muito bem documentado por Schwefel [37] e é usado neste trabalho.

Harrison e Zhang [54] apresentam um algoritmo alternativo para analisar filas com chegadas MMPP. Esta abordagem pode ser facilmente estendida para filas com prioridades. Este algoritmo exige muitos cálculos complexos e não foi testado em ambientes com sobrecargas.

Para evitar cálculos complexos, Ciciani et al. [45] elaboraram uma aproximação analítica para obter a largura média de uma fila MMPP/M/1. Esta aproximação se resume na soma ponderada de diferentes filas M/M/1, podendo ser facilmente adaptada para uma fila MMPP/D/1. Este método só funciona quando todos os estados possuem taxas de chegada menores que a taxa de serviço, logo, não pode ser usadas para situações onde existem momentos de sobrecarga.

4.5.1 MMPP/M/1

O espaço de estados de uma fila MMPP/M/1 é o produto do espaço de estados do processo de chegadas com os possíveis tamanhos de fila [37]. A matriz de transição de taxas bloco diagonal \hat{Q} para fila MMPP/M/1 é a matriz infinita

$$\hat{Q} = \begin{bmatrix} \bar{\mathcal{A}}_1 & \mathcal{A}_0 & 0 & 0 & \cdots \\ \mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_0 & 0 & \cdots \\ 0 & \mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_0 & \\ 0 & 0 & \mathcal{A}_2 & \mathcal{A}_1 & \ddots \\ \vdots & \vdots & & \ddots & \ddots \end{bmatrix}, \quad (4.13)$$

sendo

$$\mathcal{A}_0 = \mathcal{L}, \quad (4.14)$$

$$\bar{\mathcal{A}}_1 = \mathcal{Q} - \mathcal{L}, \quad (4.15)$$

$$\mathcal{A}_1 = \mathcal{Q} - \mathcal{L} - v\mathcal{I}, \quad (4.16)$$

e

$$\mathcal{A}_2 = v\mathcal{I}, \quad (4.17)$$

onde v é a taxa de utilização.

O vetor de distribuição estacionária do tamanho de fila $\hat{\pi}$ pode ser particionado em estados com o mesmo tamanho de fila:

$$\hat{\pi} = [\hat{\pi}(0), \hat{\pi}(1), \hat{\pi}(2), \dots]. \quad (4.18)$$

Então, a equação $\hat{\pi}\hat{Q} = 0$ possui soluções do tipo:

$$\hat{\pi}(k) = \hat{\pi}(0)\mathcal{R}^k, \quad (4.19)$$

onde a matriz de taxas \mathcal{R} é a solução da seguinte equação quadrática matricial:

$$\mathcal{A}_0 + \mathcal{R}\mathcal{A}_1 + \mathcal{R}^2\mathcal{A}_2 = 0 \quad (4.20)$$

onde a condição de contorno dada por:

$$\hat{\pi}(0) = \pi(\mathcal{I} - \mathcal{R}). \quad (4.21)$$

Schwefel [37] explica várias formas numéricas de resolver a equação quadrática matricial (4.20), a forma adotada neste trabalho foi o *Cyclic Reduction*.

Finalmente, a probabilidade escalar do tamanho da fila e o tamanho médio da fila são dados, respectivamente, pelas equações a seguir:

$$r(k) = \hat{\pi}(k)\varepsilon' = \pi(\mathcal{I} - \mathcal{R})\mathcal{R}^k\varepsilon', \quad k = 0, 1, \dots \quad (4.22)$$

e

$$E\{Q\} = \sum_{k=0}^{\infty} k\hat{\pi}(k)\varepsilon' = \pi\mathcal{R}(\mathcal{I} - \mathcal{R})^{-1}\varepsilon'. \quad (4.23)$$

O atraso médio é obtido usando o teorema de Little, que se resume na seguinte equação: [47]

$$E\{W\} = \frac{E\{Q\}}{\lambda}. \quad (4.24)$$

Capítulo 5

Resultados

5.1 Introdução

No capítulo anterior, foi proposto o uso do modelo de filas MMPP/M/1 para estimar parâmetros de desempenho de um enlace que recebe tráfego sobrepostos de pacotes VoIP com o mecanismo de supressão de silêncio. Neste capítulo, será estudado o comportamento da fila de um enlace submetido a este tipo de tráfego e comparado com o comportamento da fila MMPP/M/1.

Devido à dificuldade de se montar vários cenários de teste no mundo real, a melhor maneira de verificar a validade do modelo é comparar seus resultados com os resultados de simulações do sistema real. Simulações toleram modelos complexos, sendo possível então construir modelos muito próximos à realidade. O simulador de redes usado neste trabalho foi o NS-2, muito conhecido no meio acadêmico.

Para fazer um estudo completo do comportamento do sistema de filas com tráfegos de VoIP sobrepostos, foram estudados vários cenários diferentes. Estes vários cenários constam da variação do número de usuários e da carga do enlace.

5.2 Topologia

A Figura 5.1 apresenta a topologia usada para o estudo da sobreposição de tráfegos de VoIP. Todas as fontes de tráfego estão diretamente conectadas ao roteador. Este roteador está diretamente conectado ao destino de todos os pacotes por um enlace de capacidade C . Este enlace será compartilhado por todos os tráfegos e será estudado o comportamento da fila do roteador.

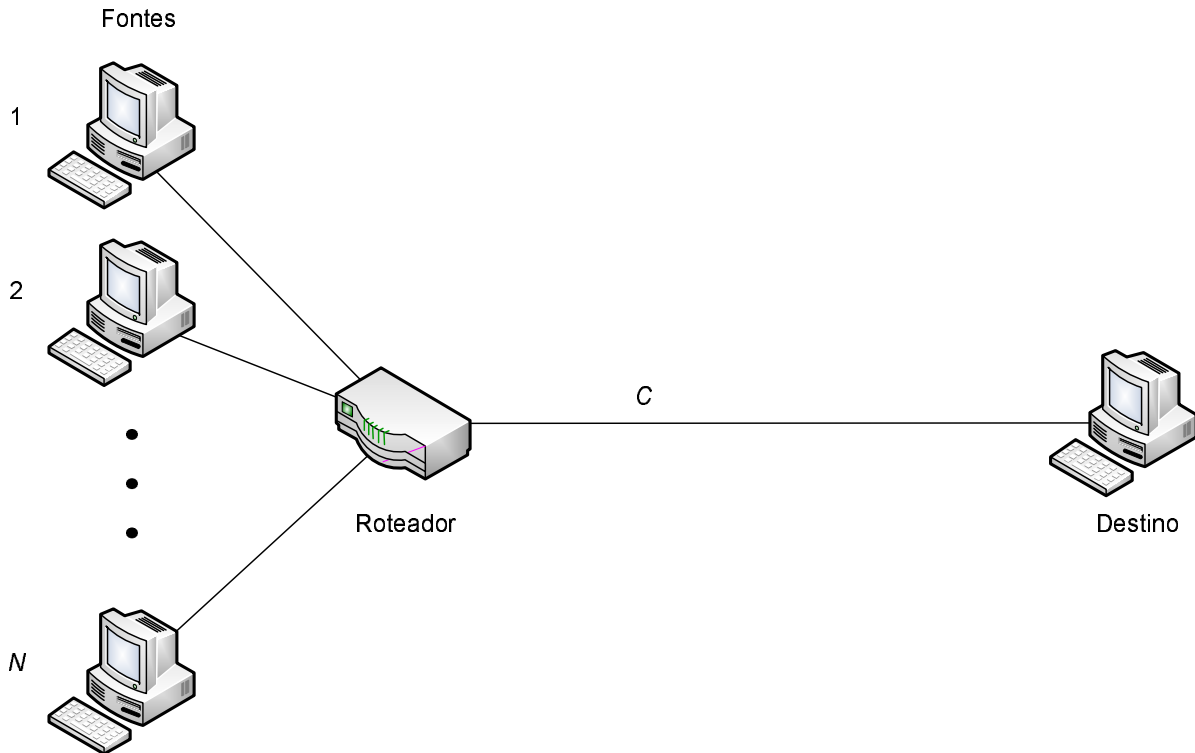


Fig. 5.1: Topologia usada para o estudo da sobreposição de tráfegos de VoIP.

5.3 Parâmetros

Os parâmetros usados para a simulação e o MMPP/M/1 analítico são descritos abaixo.

- A duração dos períodos *ON* e *OFF*, obedecem uma distribuição exponencial. De acordo com o modelo de May e Zebo, os valores de T_{on} e T_{off} são de 0,352 e 0,65 segundos, respectivamente.
- O codec usado é o G.711 que possui uma taxa de bits de 64 kbps e payload de pacote é de 160 Bytes, logo a taxa de pacotes no estado *ON* é $\lambda_{on} = 50$ pps.
- Não sendo considerado nenhum cabeçalho de camada de enlace, o tamanho de pacote (cabeçalhos IP/UDP/RTP + *payload*) é de 200 kBytes. Logo, a taxa de bits para fontes no estado *ON* é de 80 kbps.

A capacidade do enlace C onde os tráfegos serão sobrepostos vai depender do número de usuários N e da carga ρ estabelecidos em cada cenário. A capacidade é calculada pela equação que segue:

$$C = \frac{NP_{on}\lambda_{on}}{\rho}, \quad (5.1)$$

onde P_{on} é a probabilidade de uma fonte estar no estado *ON*, pode ser calculado pela equação (4.5).

5.4 Implementação do Método Matriz-Analítico

Para implementar o método matriz-analítico, optou-se em usar o MATLAB [55]. Ele possibilita rápida implementação devido ao grande suporte a operações de vetores e matrizes, fundamental para o problema em estudo. O MATLAB é o principal produto da MathWorks e é muito usado pela comunidade científica em estudos das mais diversas áreas. Este trabalho fez uso da versão 7.1, e foi usado sob licença acadêmica da UNICAMP.

No Apêndice A está integralmente disponível o script em linguagem do MATLAB usado neste trabalho.

5.5 Simulador NS-2

Para chegar a conclusões da utilidade do método, os resultados são comparados com resultados gerados de simulações. O simulador de redes usado foi o Network Simulator (versão 2.33) [56] [57], muito conhecido como NS-2. O NS-2 é a segunda versão de uma ferramenta de simulação de redes desenvolvida pelo projeto Virtual InterNetwork Testbed (VINT).

O simulador NS-2 possui uma rica biblioteca de objetos de redes e protocolos. Ele é baseado na linguagem orientada a objetos C++, em conjunto com interpretador OTcl/Tcl, usado para executar scripts de comandos do usuário. O C++ compilado permite que se consiga eficiência em uma simulação e maior velocidade de execução. Então, no script pode se definir uma topologia de rede particular, os protocolos específicos e aplicações que se desejam simular (cujo comportamento já está definido no C++ compilado) e o formato da saída que se deseja obter do simulador [57]. Devido ao NS-2 ser uma ferramenta de código-aberto e dar total liberdade de se criarem novos protocolos e aplicações, ele é um dos simuladores de redes mais populares na comunidade científica.

O script Tcl usado para gerar as simulações no NS-2 é baseado no trabalho de Andersson [48]. A Figura 5.2 mostra os códigos Tcl que iniciam os “agente”, chamando o procedimento “attach-expoo-traffic” (Figura 5.3). Eles são fontes de taxas constantes, denotados por “CBR/UDP”. O “Traffic/Expo” gera tráfegos baseados na distribuição *ON/OFF*, com parâmetros especificados nas quatro linhas seguintes, que estão de acordo com os parâmetros dados na seção 5.3. Cada fonte “\$i” usa sementes diferentes. Em outras palavras, para cada fonte os números geradores de seqüências aleatórias são distintos. Logo, as fontes se comportam de forma independente entre si. [48]

Para que a simulação chegue a valores exatos, é necessária a simulação de um tempo infinito. No entanto, isso não é prático devido a limitações de tempo e recurso computacional [48]. O tempo de simulação adotado é de 2000 segundos. Para evitar perdas, foi configurado um *buffer* bastante grande.

No Apêndice B está integralmente disponível o script Tcl usado neste trabalho.

```

for {set i 0} {$i < $N} {incr i} {
  set sinkudp($i) [new Agent/Null]
  $ns attach-agent $sink $sinkudp($i)
  set voip($i) [attach-expoo-traffic $n($i) $sinkudp($i) 200 0.352s 0.65s 80k]
}

```

Fig. 5.2: Fragmento de código Tcl definindo os parâmetros da fonte VoIP.

```

proc attach-expoo-traffic { node sink size burst idle rate } {
  # Criar uma instancia do simulador
  set ns [Simulator instance]

  # Criar um UDP agent e atrela-lo a um node
  set src [new Agent/CBR/UDP]
  $ns attach-agent $node $src

  # Criar um Expoo traffic agent e inserir os parâmetros
  set traffic [new Traffic/Expoo]
  $traffic set packet-size $size
  $traffic set burst-time $burst
  $traffic set idle-time $idle
  $traffic set rate $rate

  # Atrelar o source ao gerador de trafego
  $traffic attach-agent $src

  # Conectar o source e o sink
  $ns connect $src $sink
  return $traffic
}

```

Fig. 5.3: Procedimento “attach-expoo-traffic”.

O programa Nam [58] permite visualização da simulação a partir de um arquivo de saída específico. Ele mostra a topologia da rede, animação em nível de pacote e possui ferramentas de inspeção de dados. A Figura 5.4 mostra a animação do tráfego sobreposto de 5 fontes de VoIP no Nam.

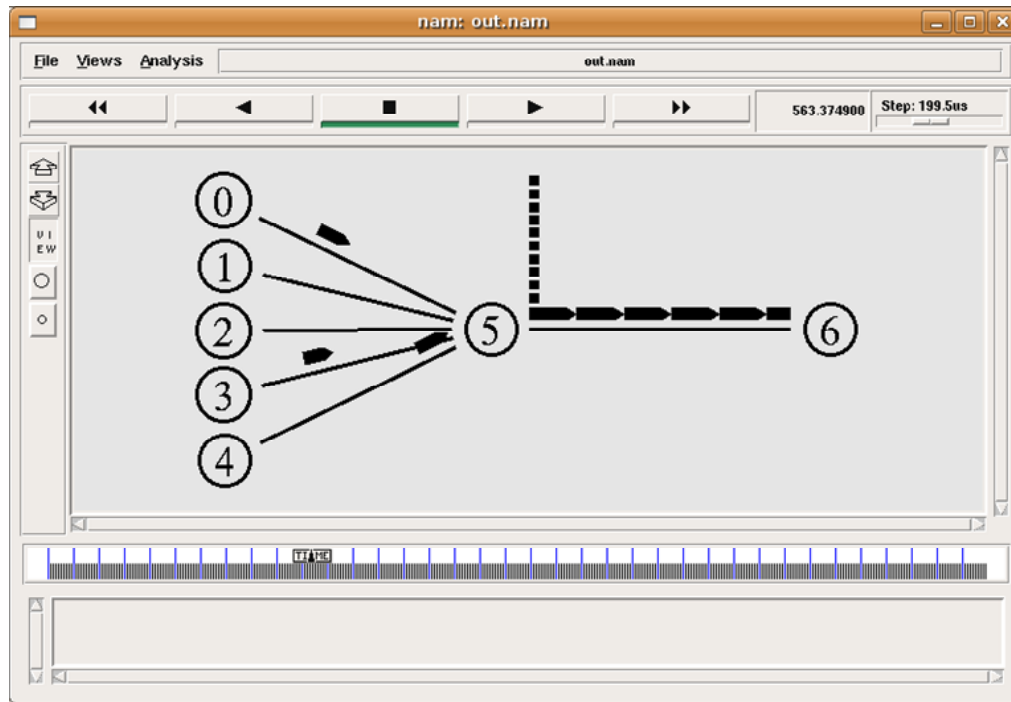


Fig. 5.4: Animação do tráfego sobreposto de 5 fontes de VoIP no Nam.

5.6 Atraso Médio

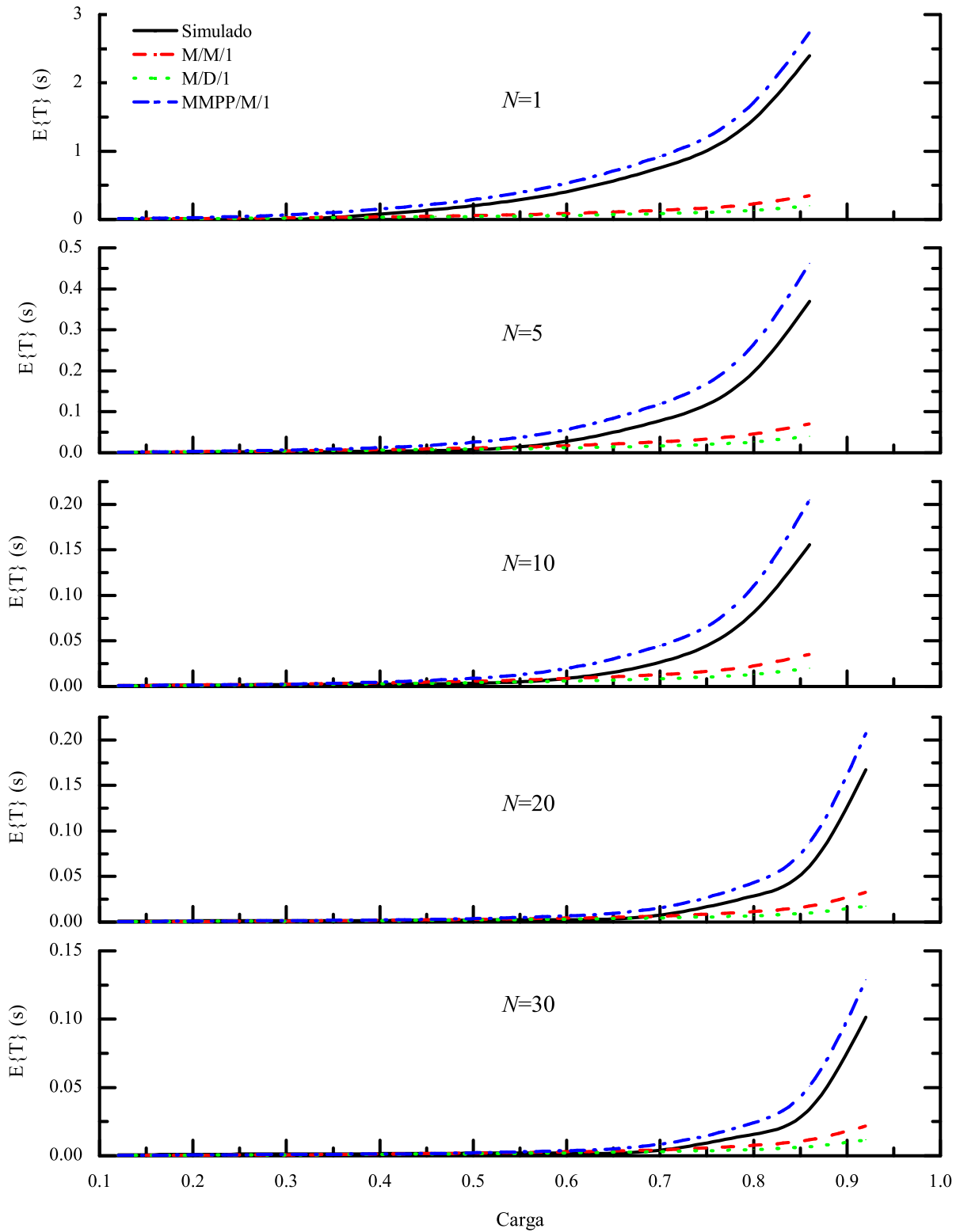
Nesta seção são apresentados e comparados os resultados de simulações e do modelo MMPP/M/1. Nas Figuras 5.5 e 5.6 encontra-se um gráfico de curvas para cada quantidade N de tráfegos sobrepostos, mostrando a sensibilidade ao valor de carga.

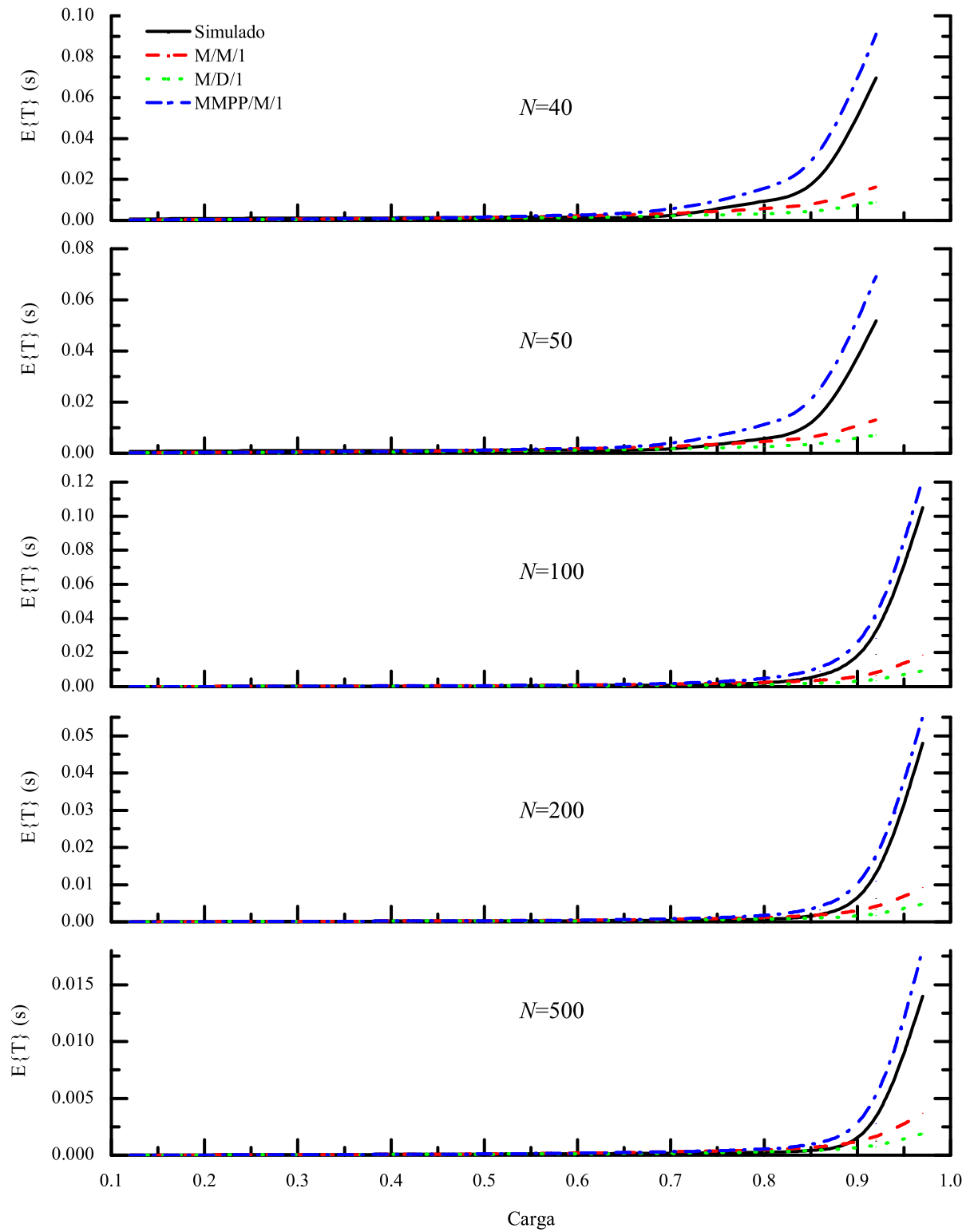
Houve 10 intensidades de tráfego em estudo: um equivale ao tráfego gerado por uma fonte somente e o resto, à sobreposição de 5, 10, 20, 30, 40, 50, 100, 200 e 500 fontes. Para cada uma das intensidades se variou a carga; os valores de carga considerados foram 0.12, 0.3, 0.48, 0.56, 0.7, 0.8, 0.86, 0.92 e 0.97. Nas cargas mais altas, alguns resultados foram descartados por não terem convergido. Assim, foram apresentados resultados para o total de 80 situações diferentes.

Os tempos médios de espera dos modelos M/M/1 e M/D/1 foram acrescentados aos gráficos para referência, eles possuem equações fechadas, dadas por: [47]

$$E\{T\} = \frac{1}{\mu(1 - \rho)} \quad (5.2)$$

e

Fig. 5.5: Atrasos médios de $N = 1$ a $N = 30$.

Fig. 5.6: Atrasos médios de $N = 40$ a $N = 500$.

$$E\{T\} = \frac{1}{\mu(1 - \rho)} \left(1 - \frac{\rho}{2}\right), \quad (5.3)$$

respectivamente.

Nota-se que os resultados do modelo MMPP/M/1 têm um comportamento parecido com os simulados; eles acompanham os resultados simulados, mantendo-se sempre mais conservadores. Esse conservadorismo se deve ao fato de o modelo supor que o tamanho de pacote obedece a uma distribuição exponencial negativa, enquanto que, na realidade, os pacotes possuem tamanhos fixos.

Os modelos M/M/1 e M/D/1 mostram resultados melhores nas cargas baixas do que o modelo MMPP/M/1. Na medida em que se aumenta o número de fontes, maiores são os valores de carga em que os resultados continuam próximos dos resultados simulados, mas continuam a se afastarem nas cargas mais altas.

É interessante notar que o tempo médio de espera na fila se torna cada vez menos sensível a variação dos valores de carga na medida em que se aumenta o número de fontes. Isso pode ser notado ao se observar que quando se aumenta o número de fontes há uma diminuição da escala de tempo nos resultados e o deslocamento para a direita do ponto de aumento intenso da espera em relação a carga.

5.7 Probabilidade Escalar do Tamanho de Fila

As três curvas encontradas nos gráficos das Figuras 5.7 e 5.8 representam os resultados das probabilidades escalares do tamanho de fila obtidos por simulação, pelo modelo MMPP/M/1 e pelo modelo M/M/1; sendo que a probabilidade escalar do tamanho de fila para o modelo M/M/1 é dada pela equação [47]:

$$r(k) = \rho^k(1 - \rho). \quad (5.4)$$

As curvas simuladas apresentam caudas ruidosas abaixo de 10^{-5} devido ao número limitado de amostras.

A Figura 5.7 apresenta a probabilidade escalar do tamanho de fila para a sobreposição de 20 tráfegos. Os gráficos apresentam quatro cargas diferentes: 0.56, 0.7, 0.8 e 0.86. Quanto maior a carga, maior é a precisão com que a curva de modelo MMPP/M/1 se encaixa com a curva da simulação. Na medida em que se diminui a carga, a curva simulada se afasta do modelo MMPP/M/1 e se aproxima do modelo M/M/1.

A Figura 5.8 apresenta a influência da quantidade de tráfegos sobrepostos. Para isso, tendo a carga fixada em $\rho = 0.86$, os gráficos apresentam quatro quantidades de fontes diferentes: 5, 30, 100 e 500. Nesse caso, a maior precisão do modelo MMPP/M/1 ocorre quando há menores números de fontes. Para uma grande quantidade de fontes, não se nota mais as grandes vantagens do modelo MMPP/M/1 em comparação ao modelo M/M/1.

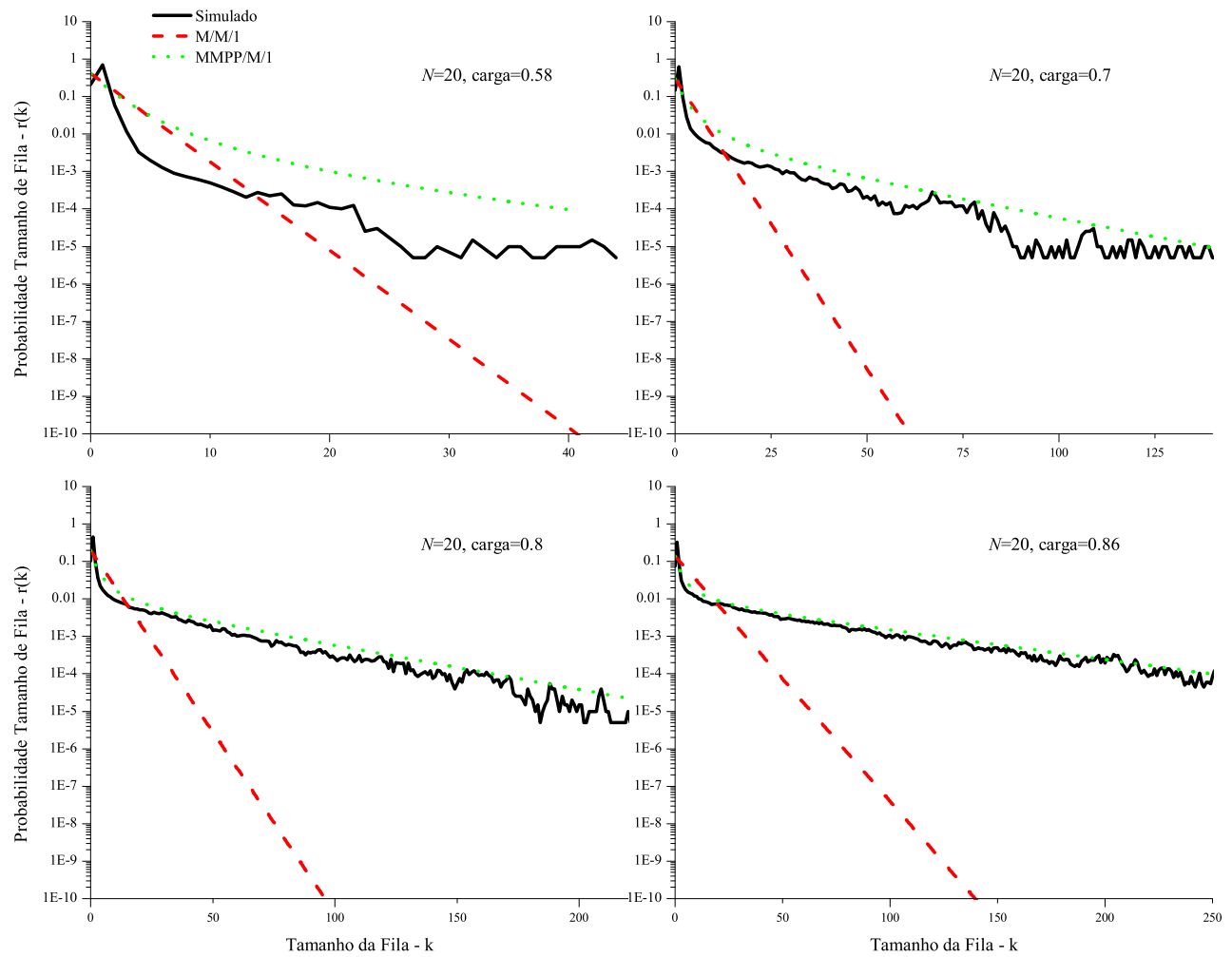


Fig. 5.7: Probabilidade escalar do tamanho de fila para $N = 20$.

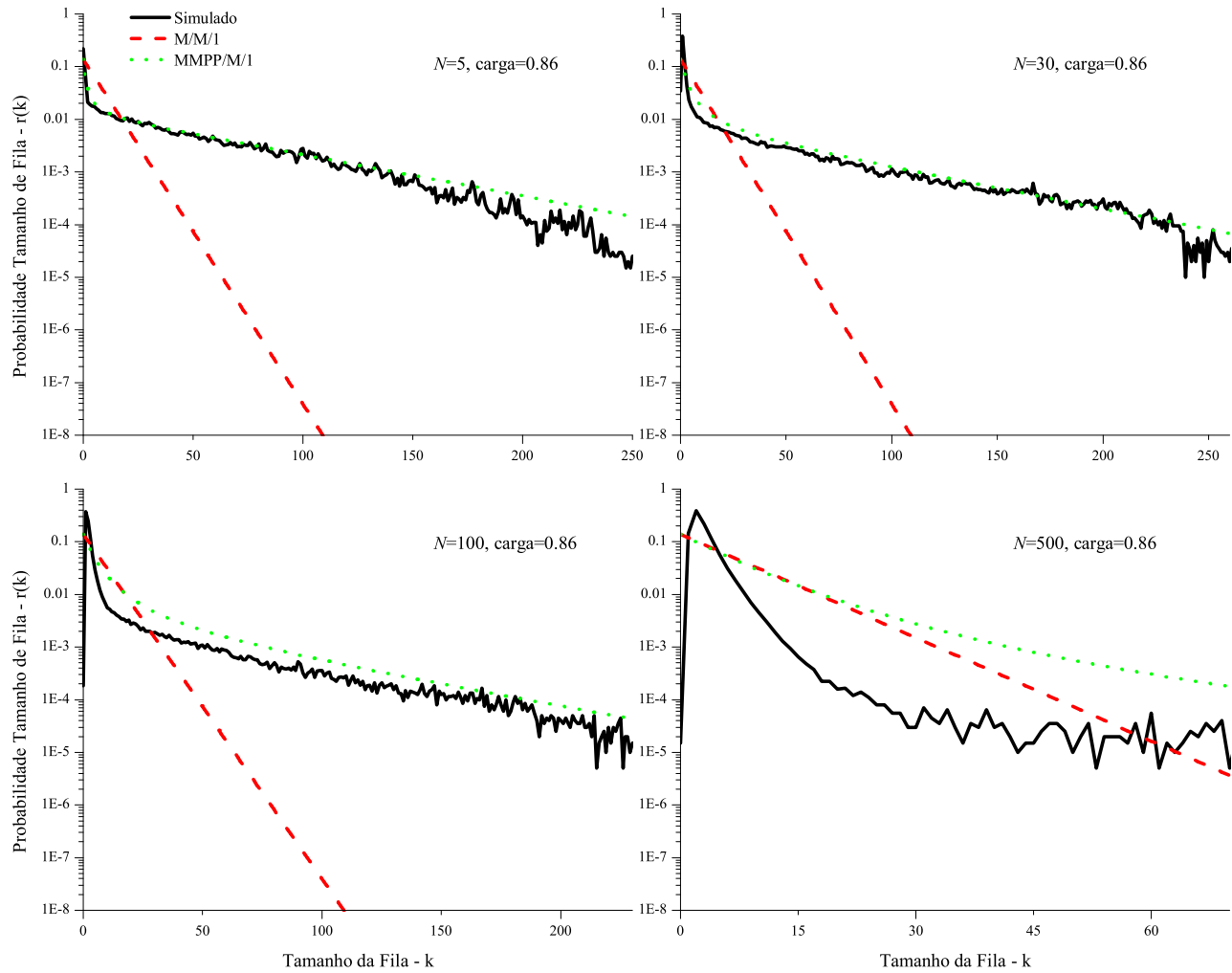


Fig. 5.8: Probabilidade escalar do tamanho de fila para $\rho = 0.86$.

Capítulo 6

Conclusão

Neste trabalho, foi estudado o comportamento de tráfegos VoIP sobrepostos em um enlace de rede IP e como se pode estimar este comportamento de forma analítica. É levado em conta que as aplicações de VoIP usam um mecanismo para economizar largura de banda, chamado de supressão de silêncio. Este mecanismo torna o tráfego mais variável e com chegadas em rajadas. Primeiramente, foi discutida a grande importância, para a sociedade, da transmissão da voz humana através de grandes distâncias e tempo-real, apresentando a evolução da rede PSTN. Foi apresentado que a rede IP está ganhando cada vez mais espaço como rede responsável pela transmissão de voz devido a diversas vantagens em comparação à rede PSTN. Este tipo de tráfego de voz ficou conhecido como VoIP.

Foram apresentados alguns padrões VoIP usados no mercado, entre eles, os que merecem maiores destaque são H.323 e SIP. O H.323 foi o primeiro padrão com grande aceitação, mesmo sendo criticado por ser muito complexo. Devido à sua simplicidade e grande flexibilidade, o SIP ganhou preferência do mercado e é considerado o principal padrão de VoIP. Ambos usam os protocolos RTP/RTCP para o transporte das informações de voz, tornando a qualidade de conversação independente do padrão de VoIP escolhido. Ou seja, os estudos em torno da qualidade de conversação são válidos para ambos os padrões. Foi introduzido o conceito de qualidade de serviço (QoS) em sistemas de telecomunicações. Foram apresentados quais são os requisitos de qualidade de uma aplicação de voz interativa (atraso, *jitter*, eco, etc.) e como eles afetam a qualidade de uma conversação.

A internet, como foi criada, não previa a necessidade de garantir os requisitos de QoS de uma aplicação VoIP. Para garantir melhores serviços para aplicações VoIP, ou outras aplicações que exigem garantias de qualidade, o IETF iniciou dois projetos (IntServ e DiffServ) que propõem novas arquiteturas para que redes IP tornem-se capazes de garantir tais requisitos, mesmo em momentos de congestionamento.

Foram apresentadas as principais técnicas enfileiramento, usadas nos elementos de uma rede para tratar congestionamento de tráfego. Cada técnica foi desenvolvida para resolver um problema espe-

cífico. Algumas técnicas de enfileiramento possibilitam alocar uma largura de banda para uma classe de tráfego específica, podendo, então, alocar uma largura de banda para tráfegos de VoIP.

São estudadas as características dos tráfegos VoIP sobrepostos. O processo MMPP é usado para modelar os tráfegos VoIP. Foi mostrado que é muito simples e intuitivo modelar VoIP com o MMPP. Na verdade, esta modelagem ajuda até a compreender melhor o comportamento do tráfego. Foram apresentados estudos de auto-similaridade e de comportamento de rajadas, que comprovam a eficiência do MMPP para descrever o comportamento do tráfego de VoIP.

Para estudar o comportamento de uma fila quando várias fontes são multiplexadas em um enlace, foi apresentada uma forma matemática de calcular a probabilidade escalar do tamanho da fila, seu tamanho médio e seu atraso médio. O modelo de fila usado é o MMPP/M/1. O serviço é exponencial negativo por ser matematicamente mais simples, o que torna esperados os resultados mais conservadores em relação a um serviço determinístico.

Para avaliar a utilidade do modelo proposto, foi usado o simulador de redes NS-2. Os primeiros resultados a serem comparados foram os atrasos médios. Foram estabelecidos 10 cenários, todos representando uma quantidade de fontes VoIP diferentes (de 1 a 500 fontes). A capacidade de serviço dos enlaces variou de forma que os cenários fossem submetidos a diversos níveis de carga (de 0,12 a 0,97).

Os resultados do modelo acompanham o comportamento dos resultados simulados, mas sempre se mantendo mais conservadores nas cargas pesadas. Isso já era esperado pelo fato de o método usar serviço exponencial negativo. Os resultados simulados são comparados, também, com resultados de filas M/M/1 e M/D/1 (filas muito mais simples que o MMPP/M/1 e cujas equações são largamente conhecidas). Observa-se que todos os modelos de fila concordam com os resultados simulados em cargas baixas, o que faz com que se prefira escolher os modelos M/M/1 ou M/D/1 por simplicidade. Na medida em que se aumenta a carga e diminui o número de fontes, os resultados simulados se afastam dos modelos M/M/1 e M/D/1, passando a ser acompanhados somente pelo modelo MMPP/M/1.

Os outros resultados comparados foram os de probabilidade escalar do tamanho da fila. Observa-se que as curvas do modelo MMPP/M/1 tendem a concordar cada vez melhor com as curvas dos resultados simulados na medida em que se diminui o número de fontes e se aumenta o valor da carga. As curvas do modelo M/M/1 foram apresentadas para referência, mas em nenhum momento conseguem acompanhar o comportamento dos resultados simulados.

O método proposto por esse trabalho é de grande utilidade para o projeto de redes. Pode ser usado como ferramenta para se prever quando os requisitos de atraso são obedecidos com bastante segurança, já que os resultados do modelo são conservadores.

Por fim, ficam algumas sugestões para trabalhos futuros. Uma primeira sugestão seria sobre o estudo do modelo MMPP/M/1/m, considerando perdas no enlace devido ao tamanho de *buffer*

limitado. Outra sugestão seria o desenvolvimento de algoritmos iterativos para obter número máximo de fontes VoIP obedecendo a um limite de atraso, dada uma taxa de serviço; ou obter a carga ideal para um dado número de fontes. Uma última sugestão seria o desenvolvimento de um aplicativo que determine rápido e facilmente a qualidade de transmissão de voz em uma rede, baseando-se no método deste trabalho e nos algoritmos iterativos aqui sugeridos.

Referências Bibliográficas

- [1] Arthur C. Clarke, Michael L. Dertouzos, et al. *O Telefone: Ontem, Hoje e Amanhã*. Telebras, 1977.
- [2] Sérgio Colcher, Antônio Tadeu A. Gomes, Anderson Oliveira da Silva, Guido L. de Souza Filho, Luiz Fernando G. Soares. *VoIP: Voz Sobre IP*. Elsevier, 2005.
- [3] James F. Kurose, Keith W. Ross. *Redes de Computadores e a Internet: Uma Nova Abordagem*. Addison Wesley, 2003.
- [4] Jim Van Meggelen, Jared Smith, Leif Madsen. *Asterisk: O Futuro da Telefonia*. Alta Books, 2005.
- [5] Oliver Hersent, David Guide, Jean-Pierre Petit. *Telefonia IP: Comunicação multimídia baseada em pacotes*. Addison Wesley, 2002.
- [6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RFC 3550: RTP, A Transport Protocol for Real-Time Applications. Rfc, Internet Engineering Task Force, July 2003.
- [7] ITU-T. Recommendation H.323: Packet-based multimedia communications systems. Rec., International Telecommunication Union, June 2006.
- [8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. RFC 3261: SIP, Session Initiation Protocol. Rfc, Internet Engineering Task Force, June 2002.
- [9] F. Andreassen, B. Foster. RFC 3435: Media Gateway Control Protocol (MGCP) Version 1.0. Rfc, Internet Engineering Task Force, January 2003.
- [10] C. Groves, M. Pantaleo, T. Anderson, T. Taylor. RFC 3525: Gateway Control Protocol Version 1. Rfc, Internet Engineering Task Force, June 2003.

- [11] ITU-T. Recommendation H.248.1 : Gateway control protocol Version 3. Rec., International Telecommunication Union, September 2005.
- [12] John C. Bellamy. *Digital Telephony*. Wiley-Interscience, 3rd edition, 2000.
- [13] Harry Nyquist. Certain Factors Affecting telegraph Speed. *Bell System Technical Journal*, 3(2):324–346, April 1924.
- [14] ITU-T. Recommendation E.800 : Terms and definitions related to quality of service and network performance including dependability. Rec., International Telecommunication Union, September 2008.
- [15] Cisco. Voice Over IP - Per Call Bandwidth Consumption. Document ID: 7934. Design techniques, Cisco Systems, Inc., August 2003. http://www.cisco.com/application/pdf/paws/7934/bwidth_consume.pdf.
- [16] Paul T. Brady. A Model for Generating On-Off Speech Patterns in Two-way Conversation. *Bell System Technical Journal*, 48(7):2245–2272, September 1969.
- [17] C.E. May, T.J. Zebo. A Summary of Speech Statistics Measured During the TASI-E Rego Park-Ojus Field Trial. *submitted for publication*.
- [18] Project Group 841. ROBUST VOICE ACTIVITY DETECTION and Noise Reduction-Mechanism USING HIGHER-ORDER STATISTICS. Technical report, Department of Communication Technology, Institute of Electronic Systems, Aalborg University, 2005.
- [19] Richard V. Cox. Three new speech coders from ITU cover a range of applications. *IEEE communication Magazine*, 35(9):40–47, September 1997.
- [20] ITU-T. Recommendation G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP). Rec., International Telecommunication Union, January 2007.
- [21] Jonathan Davidson. *Voice over IP Fundamentals*. Cisco Press, 2000.
- [22] ITU-T. Recommendation G.114 : One-way transmission time. Rec., International Telecommunication Union, May 2003.
- [23] Cisco. Understanding Delay in Packet Voice Networks. Document ID: 5125. White paper, Cisco Systems, Inc., August 2008. <http://www.cisco.com/application/pdf/paws/5125/delay-details.pdf>.

- [24] Thomas J. Kostas, Michael S. Borella, Ikhtlaq Sidhu, Guido M. Schuster, Jacek Grabiec, Jerry Mahler. Three new speech coders from ITU cover a range of applications. *IEEE Network*, February 1998.
- [25] Jan Janssen, Danny De Vleeschauwer, Guido H. Petit. Delay and Distortion Bounds for Packetized Voice Calls of Traditional PSTN Quality. *Proceeding of IPTel'2000*, April 2000.
- [26] Mauricio F. Magalhães, Eleri Cardozo. Qualidade de Serviço na Internet. Draft, Unicamp, Outubro 1999. <http://www.dca.fee.unicamp.br/~mauricio/>.
- [27] Cisco Systems, et al. *Internetworking Technology Handbook*, chapter Quality of Service Networking. Cisco Press, 3rd edition, 2000.
- [28] Khaled Salah. On the deployment of VoIP in Ethernet networks: methodology and case study. *Computer Communications*, 29:1039–1054, 2006.
- [29] L. Muscariello, M. Mellia, M. Meo, R. Lo Cigno, M. Ajmone Marsan. A Simple Markovian Approach to Model Internet Traffic at Edge Routers. Technical report, Dipartimento di Elettronica, Politecnico di Torino, May 2003.
- [30] San-qi Li, Chia-lin Hwang. On the Convergence of Traffic Measurement and Queueing Analysis: A Statistical-Matching And Queueing (SMAQ) Tool. *IEEE/ACM Transactions on Networking (TON)*, 5(1):95–110, 1997.
- [31] K. Khalil, Y. Sun. The effect of bursty traffic on the performance of local area networks. In *Proc. IEEE Globecom*, pages 597–603, Orlando, FL, USA, December 1992.
- [32] Shahram Shahi-Heydari, Tho Le-Ngoc. MMPP models for multimedia traffic. *Telecommunication Systems*, 15:273–293, 2000.
- [33] H. Heffes, D. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications*, 4:855–868, September 1986.
- [34] Ramesh Nagarajan, James F. Kurose, Don Towsley. Approximation Techniques for Computing Packet Loss in Finite-Buffered Voice Multiplexers. *IEEE Journal on Selected Areas in Communications*, 9:368–377, 1991.
- [35] H. Hassan, J. M. Garcia, C. Bockstal. Aggregate Traffic Models for VoIP Applications. In *ICDT'06: Proceedings of the International Conference on Digital Telecommunications*, page 70, Washington, DC, USA, August 2006. IEEE Computer Society.

- [36] Bengt Ahlgren, Anders Andersson, Olof Hags, Ian Marsh. Dimensioning Links for IP Telephony. In *Proceedings of 2nd IP Telephony Workshop (IPTel 2001)*, number 2, New York, NY, USA, April 2001.
- [37] Hans-Peter Schwefel. Performance Analysis of Intermediate Systems Serving Aggregated ON/OFF Traffic with Long-Range Dependent Properties. Dissertation, Technischen Universität München, September 2000.
- [38] K. Meier-Hellstern, W. Fischer. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18(2):149–171, September 1993.
- [39] Daniel P. Heyman, David Lucantoni. Modeling and multiple IP traffic streams with rate limits. *IEEE/ACM Transactions on Networking (TON)*, 11(6):948–958, December 2003.
- [40] L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, R. Lo Cigno. Markov models of internet traffic and a new hierarchical MMPP model. *Computer Communications*, 28:1835–1851, October 2005.
- [41] György Dán, Viktória Fodor. Analysis of the Packet Loss Process in an MMPP+M/M/1/K queue. Technical report, Department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm, Sweden, April 2003.
- [42] Surapich Phuvoravan. Fast Timescale Traffic Engineering in MPLS Networks. Ph.d dissertation, University of Maryland, 2004.
- [43] M. Mellia, A. Carpani, R. Lo Cigno. Measuring TCP and IP Behaviour on a Edge Node. In *IEEE Globecom 2002*, Taipei, November 2002.
- [44] K. Salamatian, B. Bruno, T. Bugnazet. Interpretation of losses observed on the Internet by inferring traffic characteristics. In *Proceedings of the DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling*, Piscataway, NJ, USA, February 2002. Rutgers University.
- [45] Bruno Ciciani, Andrea Santoro, Paolo Romano. Approximate Analytical Models for Networked Servers Subject to MMPP Arrival Processes. In *Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007)*, Cambridge, MA, USA, July 2007. IEEE Computer Society.
- [46] Shuang Deng. Traffic Characteristics of Packet Voices. In *IEEE International Conference on Communications, 1995, ICC '95 Seattle, 'Gateway to Globalization'*, volume 3, pages 1369–1374, Seattle, WA, USA, June 1995.

- [47] Leonard Kleinrock. *Queueing Systems, Volume I: Theory*. Wiley-Interscience, 1975.
- [48] Anders Andersson. Analysis of the Packet Loss Process in an MMPP+M/M/1/K queue. Technical report, Swedish Institute of Computer Science, Kista, Sweden, January 2000.
- [49] Thomas M. Chen. *The Handbook of Computer Networks*, volume 3, chapter Network Traffic Modeling. Wiley, 2007.
- [50] H. E. Hurst. Long-term Storage Capacity of Reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770–799, 1951.
- [51] Ian Kaplan. Estimation of the Hurst Exponent. Página na internet, Bear Products International, Setembro 2008. http://www.bearcave.com/misl/misl_tech/wavelets/hurst/.
- [52] K. Sriram, W. Whitt. Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data. *IEEE Journal on Selected Areas in Communications*, 4:833–846, September 1986.
- [53] Marcel F. Neuts. *Matrix-Geometric Solution in Stochastic Models, An Algorithmic Approach*. The Johns Hopkins University Press, 1981.
- [54] P.G. Harrison, Yu Zhang. Delay Analysis of Priority Queues with Modulated Traffic. In *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 280–287, Washington, DC, USA, September 2005. IEEE Computer Society.
- [55] MATLAB® The Language of Technical Computing. Página na internet, The MathWorks, Inc, Setembro 2008. <http://www.mathworks.com/products/matlab/>.
- [56] The Network Simulator - ns-2. Página na internet, Information Sciences Institute (ISI), University of Southern California, Setembro 2008. <http://www.isi.edu/nsnam/ns/>.
- [57] Eitan Altman, Tania Jiménez. NS Simulator for beginners. Lecture notes, Univ. de Los Andes, December 2003. <http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>.
- [58] Nam: Network Animator. Página na internet, Information Sciences Institute (ISI), University of Southern California, Setembro 2008. <http://www.isi.edu/nsnam/nam/>.

Apêndice A

Script do MATLAB

Script de Matlab usado para implementar a solução da fila MMPP/M/1.

*% Calcula Tamanho médio da fila, Atraso Médio e Probabilidade escalar de tamanho de fila de uma
% fila MMPP/M/1.*

function [tamM,atrasoM,r]=solucaoMMPP(N,lambda,mi,T_on,T_off);

clear;

L=**diag**([0:N]*lambda);

I=**eye**(N+1);

Q=**find_Q**(N,T_on,T_off);

A0=L;

A1=Q-L-mi*I;

A2=mi*I;

[R,iter,t0]=**solve_R**(A0,A1,A2);

eps=**ones**(1,N+1);

pi=[**zeros**(1,N+1), 1] / [Q, **eps**'] ;

LambdaM=**pi***L***eps**' ;

% Tamanho médio da fila

tamM=((**pi***R)/(I-R))***eps**' ;

% Atraso médio

atrasoM=tamM/LambdaM

nmax=300;

r=**zeros**(1,nmax+1); *% Probabilidade escalar de tamanho de fila*

10

20

```

n=0:nmax;
pi_k=pi*(I-R);
r(1)=sum(pi_k);
30

for i=1:nmax,
    pi_k=pi_k*R;
    r(i+1)=sum(pi_k);
end;

qbar=(pi*R)*((I-R)\eps');
disp(qbar);

figure;
semilogy(n,r,'b-');
xlabel('Tamanho da fila - k');
ylabel('Probabilidade de tamanho de fila - r(k)');
title('Distribuição probabilidade de tamanho de fila.');
```

```

%
%      Sub-Funções
%
% Monta a Matriz Q
50
function [Q]=find_Q(N,T_on,T_off);

Q=zeros(N+1,N+1);
for i=1:N,
    Q(i,i+1)=(N+1-i)/T_off;
end;

for i=2:N+1,
    Q(i,i-1)=(i-1)/T_on;
end;
60

for i=1:N+1,
    Q(i,i) = - sum(Q(i,:));
end;

%
% Calcula a solução da equação da matriz quadrada usando o método "Cyclic Reduction"
% Fonte:
% Hans-Peter Schwefel. Performance Analysis of Intermediate Systems Serving Aggregated
% ON/OFF Traffic with Long-Range Dependent Properties. Dissertation, Technischen Universität
70
```

% München, September 2000. (Appendix F)

function [R,iter,t0]=solve_R(A0,A1,A2);

t0=cputime; iter=1;

A0k=A0; A1k=A1; A2k=A2;

A1d=A1;

k=0;

delta=1; dold=2;

while (delta>1e-12)&(abs(delta-dold)>1e-12),

 iter=iter+1;

 dold=delta;

 H0=A0k*inv(-A1k);

 H2=A2k*inv(-A1k);

 A1d=A1d+H0*A2k;

 A1k=A1k+H2*A0k+H0*A2k;

 A2k=H2*A2k;

 A0k=H0*A0k;

 k=k+1;

 delta=max(max(A0k));

end;

80

90

Apêndice B

Script TCL

Script TCL usado para simular Sobreposição de tráfegos ON-OFF no ns-2.

```
scan $argv "%d %s " N rho
puts "$N $rho"
```

```
set ns [new Simulator]
```

```
#Parâmetros de simulação
```

```
set Ton 0.352
```

```
set Toff 0.65
```

10

```
set Pon [expr $Ton/($Ton+$Toff)]
```

```
set C [expr ($N*$Pon*80000)/$rho]
```

```
# Hora de início da simulação
```

```
exec date >> hora$N-$rho.txt &
```

```
# Abrir arquivo NAM
```

```
set namfile [open out.nam w]
```

```
$ns namtrace-all $namfile
```

20

```
# Definir o procedimento "finish"
```

```
proc finish {} {
```

```
    global ns N rho tracefile namfile
```

```
    exec date >> hora$N-$rho.txt &
```

```
    $ns flush-trace
```

```
    close $tracefile
```

```

    close $namfile
    exec /home/marcos/ns-allinone-2.33/nam-1.13/nam out.nam &
    exit 0
}

proc attach-expoo-traffic { node sink size burst idle rate } {
    # Criar uma instancia do simulador
    set ns [Simulator instance]

    # Criar um UDP agent e atrela-lo a um node
    set src [new Agent/CBR/UDP]
    $ns attach-agent $node $src

    # Criar um Expoo traffic agent e inserir os parâmetros
    set traffic [new Traffic/Expoo]
    $traffic set packet-size $size
    $traffic set burst-time $burst
    $traffic set idle-time $idle
    $traffic set rate $rate

    # Atrelar o source ao gerador de trafego
    $traffic attach-agent $src

    # Conectar o source e o sink
    $ns connect $src $sink
    return $traffic
}

# Criar nós
for {set i 0} {$i < $N} {incr i} {
    set n($i) [$ns node]
}

set router [$ns node]
set sink [$ns node]

# Criar enlaces entre os nodes
for {set i 0} {$i < $N} {incr i} {
    $ns simplex-link $n($i) $router 1Mb 0ms DropTail
}

$ns simplex-link $router $sink $C 0ms DropTail

```

30

40

50

60

70

Abrir arquivo Trace

set tracefile [**open** N\$Nrho\$rhoout.tr w]

\$ns trace-queue \$router \$sink \$tracefile

Monitorar fila

set qmon [\$ns monitor-queue \$router \$sink [**open** qm\$N-\$rho.out w] 0.01]

[\$ns link \$router \$sink] queue-sample-timeout; # [\$ns link \$router \$sink] start-tracing

Posição dos nodes para o NAM

for {**set** i 0} {\$i < \$N} {**incr** i} {

 \$ns simplex-link-op \$n(\$i) \$router orient right

}

80

\$ns simplex-link-op \$router \$sink orient right

\$ns simplex-link-op \$router \$sink queuePos 0.5

Configurar o tamanho da fila para o enlace router-sink para 1000 pacotes

\$ns queue-limit \$router \$sink 10000

for {**set** i 0} {\$i < \$N} {**incr** i} {

set sinkudp(\$i) [new Agent/Null]

 \$ns attach-agent \$sink \$sinkudp(\$i)

set voip(\$i) [attach-expoo-traffic \$n(\$i) \$sinkudp(\$i) 200 0.352s 0.65s 80k]

}

90

Agenda de eventos

for {**set** i 0} {\$i < \$N} {**incr** i} {

 \$ns at 0.001 "\$voip(\$i) start"

}

\$ns at 2000.0 "finish"

\$ns run

100