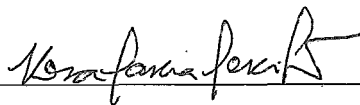


MEDIÇÃO DE TRÁFEGO BASEADA NA
DECOMPOSIÇÃO TEMPORAL DE FLUXOS

José Carlos Bellora Junior

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E
COMPUTAÇÃO.

Aprovada por:



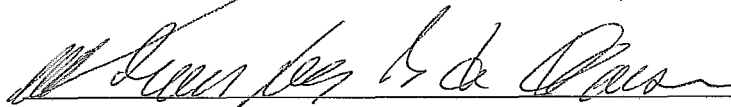
Prof. Rosa Maria Meri Leão, Dr.



Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.



Prof. José Ferreira de Rezende, Dr.



Prof. Nelson Luis Saldanha da Fonseca, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2005

BELLORA JUNIOR, JOSÉ CARLOS

Medição de Tráfego Baseada na Decomposição Temporal de Fluxos [Rio de Janeiro] 2005

XVI, 109 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Tese - Universidade Federal do Rio de Janeiro, Pesc/COPPE

1. Medição Ativa
2. Modelos de Tráfego
3. Monitoração de Tráfego
4. Fluxos de pacotes

I. COPPE/UFRJ II. Título (Série)

À Mariza de Queiroz Pessoa, minha companheira inseparável, pelo seu apoio, compreensão e incentivo que me conduziram à realização deste grande sonho.

Agradecimentos

Durante os anos de dedicação a este Mestrado, foram muitos os momentos que abdiquei de estar com meus familiares para ficar isolado em longos períodos de estudo que pareciam não terminar. Agradeço a eles pela paciência e espera e, principalmente, pelo orgulho das minhas conquistas.

Agradeço aos meus pais pela educação, pelo amor e carinho que sempre me propiciaram. À minha irmã pelo exemplo de dedicação e, aos meus filhos, por me motivarem a vencer. Agradeço, sobretudo, à minha esposa Mariza, por ser a catalizadora da busca de meus objetivos.

Gostaria também de agradecer a todos aqueles que, de alguma forma, contribuíram para a conclusão deste trabalho. Ao colega Fernando Hagemann por sua ajuda em adquirir a literatura que utilizei durante as disciplinas. Aos colegas do LAND Allyson, Ana, Bernardo, Bruno, Flávio, Guto, Kleber e Sidney por estarem sempre dispostos a colaborar.

À minha orientadora profa. Rosa pelo aprendizado, atenção e direcionamento dado a esta pesquisa.

Por fim, agradeço as contribuições enviadas por András Horváth e Jörg Micheel, que possibilitaram ampliar os experimentos aqui desenvolvidos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MEDIÇÃO DE TRÁFEGO BASEADA NA
DECOMPOSIÇÃO TEMPORAL DE FLUXOS

José Carlos Bellora Junior

Abril/2005

Orientador: Rosa Maria Meri Leão

Programa: Engenharia de Sistemas e Computação

Medidas de tráfego são essenciais para o planejamento e o gerenciamento de redes de alta velocidade. Entretanto, a obtenção dos dados necessários para a caracterização precisa do tráfego requer uma infra-estrutura de medição bastante complexa. A coleta de estatísticas de pacotes, como o tamanho e o tempo de chegada de cada um deles, exige uma infra-estrutura complexa de medição e não é escalável com o aumento da velocidade dos enlaces. Já as estatísticas de fluxos são mais fáceis de serem obtidas, pois os roteadores possuem ferramentas de medição passiva que coletam e exportam essas estatísticas. O objetivo deste trabalho é propor um método para gerar uma seqüência de pacotes a partir de informações coletadas para fluxos. O método é baseado em modelos analíticos do protocolo TCP. Foram obtidas estatísticas de primeira e de segunda ordem para as seqüências de pacotes geradas pelo método e seqüências reais da Internet para validar o método. Os resultados mostraram que o método é capaz de capturar com bastante precisão estas estatísticas, assim como a distribuição da ocupação de uma fila. São apresentados modelos de tráfego capazes de serem parametrizados com o método e analisado o desempenho na ocupação da fila para ser aplicado em projetos de dimensionamento. Como exemplo de aplicação, apresentamos um estudo de dimensionamento de rede baseado exclusivamente em medições de fluxos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

TRAFFIC MEASUREMENTS BASED ON
TEMPORAL FLOW DECOMPOSITION

José Carlos Bellora Junior

April/2005

Advisors: Rosa Maria Meri Leão

Department: Computer and System Engineering

Traffic measurements are essential for capacity planning and managing high speed data networks. Nevertheless, the generation of packet level statistics, such as sizes and timestamps of individual packets, require a complex measurement infrastructure and do not scale well with link speed. Flow level measurements can be more easily obtained than packet level statistics, since routers offer tools that give information about the flow of packets that traverse them. We propose a method to generate a packet trace from flow level measurements. The method is based on analytical TCP models. We compute first and second order statistics from the packet traces and several real traces of the Internet to analyze the performance of the method. The results show that the packet traces obtained from the method reproduce the first and the second order statistics of the real traces very accurately as well as the queue length distribution. Traffic models that can go under parametrization by using the proposed method are presented and their performance analysed to be applied in dimensioning projects. We present a network capacity planning study based on exclusively in flow measurements, as an example of application.

Palavras-chave

1. Medição Ativa
2. Modelos de Tráfego
3. Monitoração de Tráfego
4. Dimensionamento de Redes

Glossário

ACK:	<i>Acknowledgments;</i>
ATM:	<i>Asynchronous Transfer Mode;</i>
BGP:	<i>Border Gateway Protocol;</i>
CDF:	<i>Cumulative Distribution Function;</i>
FIN:	Sinalização do protocolo TCP para fim de conexão;
FTP:	<i>File Transfer Protocol;</i>
HSSI:	<i>High-Speed Serial Interface;</i>
ICMP:	<i>Internet Control Message Protocol;</i>
IP:	<i>Internet Protocol;</i>
LAN:	<i>Local-area network;</i>
MSS:	<i>Maximum Segment Size;</i>
OSPF:	<i>Open Shortest Path First;</i>
PDF:	<i>Probability Density Function;</i>
PMF:	<i>Probability Mass Function;</i>
POS:	<i>Packet Over SONET ;</i>
QoS:	<i>Quality of Service;</i>
RFC:	<i>Request for comments;</i>
RTT:	<i>Round Trip Time;</i>
SYN:	Sinalização do protocolo TCP para início de conexão;
SNMP:	<i>Simple Network Management Protocol;</i>
SONET:	<i>Synchronous Optical Network;</i>
TCP:	<i>Transmission Control Protocol;</i>
UDP:	<i>User Datagram Protocol;</i>
WAN:	<i>Wide-area network;</i>

Sumário

Resumo	v
Abstract	vi
Glossário	viii
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Objetivos	3
1.3 Estrutura da Tese	4
2 Fundamentação Teórica, Fluxos e Medição Passiva do Tráfego	6
2.1 Métricas de Desempenho	6
2.2 Descritores de Tráfego	8
2.2.1 Auto-Similaridade	9
2.2.2 Dependência de Longo Alcance	11
2.3 Fluxos e Medição Passiva do Tráfego	13
2.3.1 Monitoração de Fluxos	15
2.3.2 Monitoração SNMP	17

2.3.3	Monitoração de Pacotes	18
2.3.4	Resumo	20
3	Método para Decomposição Temporal de Fluxos em Pacotes	21
3.1	Introdução	21
3.2	Método para a geração de pacotes	23
3.2.1	Estrutura Geral	25
3.2.2	Fase de Inicialização	26
3.2.3	Fase de <i>Slow Start</i>	29
3.2.4	Fase de <i>Congestion Avoidance</i>	30
3.2.5	Distribuição do tamanho dos pacotes	33
3.2.6	Determinação do tempo entre chegadas de pacotes	34
3.3	Influência dos Fluxos UDP	35
3.4	Validação, Experimentos e Resultados	37
3.4.1	Resultados para o número de pacotes por intervalo de tempo .	39
3.4.2	Resultados para o tempo entre chegadas de pacotes	45
3.5	Análise do Erro Médio Produzido pelo Método	46
3.6	Aplicação em Modelos de Tráfego	48
3.7	Resumo e Conclusões	50
4	Modelos de Tráfego	52
4.1	Introdução	52
4.2	Modelo de Sobrecarga e Folga	53
4.2.1	Cadeia de Markov do modelo	54

4.2.2	Parametrização	56
4.2.3	Ocupação da fila	57
4.3	Modelo MAP Multifractal	58
4.3.1	Cadeia de Markov do modelo	59
4.3.2	Parametrização	61
4.3.3	Ocupação da fila	62
4.4	Modelo MMPP Hierárquico	63
4.4.1	Cadeia de Markov do modelo	64
4.4.2	Parametrização	66
4.4.3	Ocupação da fila	69
4.5	Modelo de Markov Oculto	70
4.5.1	Cadeia de Markov do modelo	72
4.5.2	Parametrização	73
4.5.3	Ocupação da fila	74
4.6	Resumo e Conclusões	75
5	Dimensionamento de Capacidade Baseado em Medição de Fluxos:	
	Um Estudo de Caso	78
5.1	Introdução	78
5.2	Dimensionamento de Capacidade	79
5.2.1	O processo de Dimensionamento	80
5.3	Estudo de Caso	82
5.3.1	Arquitetura da rede	82
5.3.2	Monitoração do Tráfego	83

<i>SUMÁRIO</i>	xii
5.3.3 Caracterização e Modelagem do Tráfego	85
5.3.4 Avaliação de Desempenho	87
5.4 Resumo e Conclusões	94
6 Contribuições e Trabalhos Futuros	96
6.1 Contribuições	97
6.1.1 Método de geração de pacotes	97
6.1.2 Análise de Modelos de tráfego	98
6.1.3 Estudo de Caso	99
6.2 Trabalhos Futuros	100
Referências Bibliográficas	102

Lista de Figuras

2.1	Comparativo do tráfego observado para diferentes níveis de agregação.	10
2.2	Acesso e criação dos fluxos e datagrama UDP de exportação.	15
2.3	Estrutura do registro do pacote após a coleta.	19
3.1	Estrutura de distribuição e contagem dos pacotes	26
3.2	Módulo principal do algoritmo com a inicialização dos parâmetros. . .	28
3.3	Cálculo de RTT para as fases de <i>slow start</i> e <i>congestion avoidance</i> . .	30
3.4	Algoritmo para a fase de <i>slow start</i>	30
3.5	Algoritmo para a fase de <i>congestion avoidance</i>	32
3.6	Distribuição de pacotes executada pelo algoritmo para dois fluxos diferentes.	33
3.7	CDF do tamanho dos pacotes para a seqüência real e estimada. . . .	34
3.8	Tempo de chegada dos pacotes e multiplexação estatística	35
3.9	Composição TCP e UDP do tráfego total.	36
3.10	Comportamento real e estimado do tráfego UDP.	37
3.11	Número de pacotes por intervalo de tempo para as seqüências real e estimada para intervalo de tempo de 1s.	41

3.12	Função de autocorrelação para as seqüências real e estimada para intervalo de tempo de 1s.	42
3.13	Distribuição do tamanho da fila para as seqüências real e estimada.	43
3.14	Distribuição do tamanho da fila para as seqüências real e estimada.	44
3.15	Função de autocorrelação para as seqüências real e estimada do tempo entre chegadas dos pacotes.	45
3.16	Erro médio para as seqüências estimadas de 1s e 200ms.	47
3.17	Erro médio das seqüências para fluxos individuais e simultâneos.	48
3.18	a) Segundo momento dos coeficientes wavelet de Haar para diferentes níveis de agregação k. b) Tamanho médio da fila para a seqüência real de pacotes e os modelos HMM e MAP.	49
4.1	Modelo com 6 estados de sobrecarga e 3 de folga.	55
4.2	Desempenho do modelo de sobrecarga e folga na ocupação da fila.	57
4.3	Cadeia do modelo Multifractal de 3 níveis e 8 estados com suas respectivas taxas.	60
4.4	Cadeia do modelo Multifractal de 4 níveis e 16 estados.	60
4.5	Desempenho do modelo MAP de 16 e 32 estados na ocupação da fila.	63
4.6	Cadeia do modelo MMPP Hierárquico.	65
4.7	<i>trace</i> de fluxos obtidos com o <i>Netflow</i>	67
4.8	Algoritmo de ajuste do modelo HMM	68
4.9	Distribuição do número de sessões e fluxos.	69
4.10	Estados observados e ocultos em uma HMM.	70
4.11	Estrutura utilizada na cadeia HMM.	72
4.12	Desempenho do modelo HMM na ocupação da fila.	74

5.1	Processo de dimensionamento de capacidade	81
5.2	Rede de análise.	82
5.3	Monitoração SNMP para o <i>link</i> de 34Mb/s.	84
5.4	Tráfego com granularidade de 200ms medido no <i>link</i> de 34Mb/s.	86
5.5	Ocupação média da fila para os quatro <i>links</i> em estudo.	87
5.6	Curva ajustada para as funções de atraso versus capacidade dos enlaces.	90
5.7	Funções candidatas. A) Potência, B) Exponencial.	91

Lista de Tabelas

3.1	Proporção de tráfego dos <i>traces</i> da Internet usados nas análises . . .	36
3.2	<i>Traces</i> da Internet usados nas análises	38
3.3	Estatísticas obtidas para as seqüências real e estimada para intervalo de tempo de 1s.	39
3.4	Estatísticas obtidas para as seqüências real e estimada para intervalo de tempo de 200ms.	40
3.5	Estatísticas obtidas para as seqüências real e estimada para o tempo entre chegadas dos pacotes.	45
4.1	Características dos modelos analisados	76
5.1	Proporção de tráfego IP nos <i>links</i> da rede em análise.	84
5.2	Proporção de tráfego <i>Web</i> nos <i>links</i> da rede em análise.	85
5.3	Especificação dos <i>traces</i> de pacotes usados na modelagem do tráfego.	85
5.4	Capacidade atual e dimensionada para a rede em estudo.	94

Capítulo 1

Introdução

1.1 Considerações Iniciais

ESTA pesquisa é motivada pelo interesse em obter medidas precisas de tráfego, que possam ser facilmente coletadas por administradores de rede, e que sirvam para o dimensionamento e modelagem do tráfego Internet.

A medição e a modelagem do tráfego têm sido realizadas desde que houve a necessidade de que computadores remotos trocassem informações entre si. Os circuitos dedicados das redes de telefonia existentes na época demonstraram-se totalmente inadequados para esta tarefa, uma vez que a natureza da comunicação de dados difere consideravelmente da comunicação de voz. O tráfego de dados possui períodos de rajadas seguidos por longos períodos de silêncio na proporção de 1:1000 ou até mesmo de 1:10000 em comparação com a de 1:3 do tráfego de voz. Esta ineficiência na utilização dos canais de comunicação abriu a oportunidade para uma nova tecnologia baseada no princípio do compartilhamento dinâmico dos recursos da rede (roteadores, switches, enlaces). Os dados da comunicação entre os computadores seriam multiplexados em um único canal, não de forma determinística com reserva de tempo por exemplo, mas de forma aleatória, de modo que o acesso à rede fosse imediato a qualquer instante e com qualquer duração. Esta aleatoriedade no envio de dados pelas diversas fontes em um único canal levou ao que é conhecido como

multiplexação estatística. Desta forma, os computadores poderiam comunicar-se através da troca de mensagens sem a necessidade de circuitos dedicados. Estudos posteriores demonstraram que o tempo de resposta da rede era diretamente influenciado pelo tamanho da mensagem trafegada, sendo necessário tamanhos menores para otimizar o tempo de transmissão. Este conceito fez com que a comunicação fosse executada através da troca de pequenos segmentos de informação conhecidos como *pacotes*, dando origem às redes de pacotes atuais [47].

De acordo com o paradigma de serviço datagrama da rede de pacotes, cada segmento contém o endereço de origem e destino da comunicação, sendo direcionado de forma independente pela rede até seu destino sem requerer reservas de recursos através de seu caminho. Como resultado, a qualidade de serviço oferecida não traz garantias nem limites de desempenho, apenas fornece o que é possível no momento da comunicação na base do *best-effort*. A multiplexação estatística e o empacotamento da informação, aliados a uma arquitetura simples e aberta de roteadores e protocolos de comunicação, viabilizaram a criação de uma infra-estrutura na qual computadores de várias redes fossem cada vez mais interligados, desenhando assim as origens da Internet [48].

Com a evolução da tecnologia e das aplicações, cada vez mais a Internet é utilizada para transportar tráfego com características bem distintas a velocidades da ordem de gigabits. Desta forma a medição, caracterização e modelagem do tráfego, tarefas essenciais para o planejamento e dimensionamento da rede, vem recebendo especial atenção nos últimos anos.

O ponto de partida para a análise e modelagem do tráfego está na medição. A obtenção dos dados necessários para uma caracterização precisa do tráfego em backbones e redes de alta velocidade requer uma infra-estrutura de monitoração bastante complexa. No entanto, esta caracterização é essencial para (i) o desenvolvimento de novos mecanismos para a rede, (ii) o planejamento de capacidade da rede, (iii) o gerenciamento da rede e (iv) a engenharia de tráfego. A maioria destas atividades necessita de um modelo para fazer uma previsão de curto ou longo prazo do tráfego. As ferramentas SNMP [15], embora largamente utilizadas pelos admi-

nistradores de rede, não fornecem as informações com o nível de detalhe adequado para a parametrização dos modelos de tráfego. Já a monitoração de pacotes exige uma infra-estrutura complexa de equipamentos, com memórias e barramentos de alta velocidade para atender às elevadas taxas de transferência de dados das placas de captura, além da grande capacidade de armazenamento externo necessária. Dez minutos de captura em um canal 10Gbs (OC192) na rede Abilene, por exemplo, consomem mais de 5GB de espaço em disco.

Recentemente, uma nova abordagem no estudo e modelagem do tráfego tem sido evidenciada na literatura. Trata-se da análise em nível de fluxos que fornece informações agregadas do tráfego entre pontos de origem e destino da rede, possibilitando completa abstração da rede física. As informações de fluxo têm sido usadas para, por exemplo, inferir certas propriedades da rede como os fluxos que estão compartilhando um determinado caminho ou um enlace que está congestionado [4], ou para estudar a autocorrelação do tráfego em um backbone da Internet [56], além de grande aplicação na engenharia de tráfego. O que facilita a utilização das informações em nível de fluxo para estudo do tráfego é o fato de que os fabricantes implementam em seus roteadores mecanismos de coleta [16, 43], permitindo que os fluxos gerados sejam analisados por ferramentas estatísticas específicas sem a necessidade de uma infra-estrutura complexa.

Apesar de as informações de fluxos terem sido recentemente usadas para estudar certas características da rede, elas não fornecem o comportamento do tráfego em escalas pequenas de tempo e não são suficientes para parametrizar grande parte dos modelos de tráfego encontrados na literatura. Desta forma, novos mecanismos que ampliem as vantagens na utilização de fluxos precisam ser desenvolvidos, com o objetivo de fornecer o mesmo nível de informação que a monitoração de pacotes.

1.2 Objetivos

O objetivo principal desta tese é apresentar um método para caracterizar o tráfego no nível de pacotes valendo-se da monitoração exclusiva de fluxos. Este

método reúne a facilidade da coleta dos fluxos com a precisão das informações obtidas em nível de pacotes. Ou seja, a partir das informações do número de pacotes, bytes e duração do fluxo, é gerada uma seqüência de pacotes com uma determinada granularidade definida pelo usuário. Mostraremos que é possível utilizar este método para parametrizar diversos modelos da literatura sem a necessidade da coleta direta de pacotes, possibilitando inúmeras aplicações, entre elas a análise, o gerenciamento e o dimensionamento de redes.

1.3 Estrutura da Tese

Esta tese é organizada da seguinte forma: No Capítulo 2 faremos as definições das estatísticas utilizadas e a conceituação das propriedades mais relevantes encontradas no tráfego Internet atual, especificamente a auto-similaridade e a dependência de longo alcance. Apresentaremos o estado-da-arte em medição de tráfego, evidenciando a monitoração dos fluxos como alternativa de grande potencial para a medição do tráfego, fato que vem recebendo cada vez mais atenção do meio científico.

No Capítulo 3 é apresentado o método para decomposição dos fluxos em pacotes. Mostraremos que é possível obter seqüências de pacotes próximas às reais, tanto para o número de pacotes por intervalo de tempo, como para o tempo entre chegadas. Foram usados, para análise e comparações, *traces* reais de domínio público de canais de comunicação de acesso à Internet, bem como canais de alta velocidade em uso no backbone da Internet [57]. Os resultados mostram que o método é capaz de capturar com precisão estatísticas de primeira e segunda ordem da seqüência de pacotes, assim como a distribuição da ocupação de uma fila alimentada por esta seqüência.

No Capítulo 4 são estudados modelos Markovianos de tráfego que aproximam o comportamento dos processos auto-similares para uma determinada escala de tempo, com o objetivo de reproduzir corretamente a ocupação em uma fila. A parametrização destes modelos foi feita com o auxílio do método proposto, demonstrando flexibilidade de capturar os diferentes descritores necessários. Os modelos foram resolvidos

anlítica e comparamos tanto seu desempenho quanto sua eficiência e precisão em determinar o comportamento real do tráfego, o que viabiliza sua aplicação no projeto e dimensionamento de capacidade das redes.

No Capítulo 5 apresentamos um procedimento para o dimensionamento de capacidade baseado no método proposto e nos modelos de tráfego estudados. Com este procedimento procuramos otimizar a utilização dos recursos compartilhados, minimizando custos e atendendo os requisitos de desempenho percebidos pelos usuários. Através de um estudo de caso, conduziremos o dimensionamento de uma rede real, onde verificaremos a necessidade de novos investimentos para atender a demanda das aplicações.

Finalmente, no Capítulo 6, faremos um resumo das contribuições apresentadas e discutiremos possíveis direcionamentos para trabalhos futuros.

Capítulo 2

Fundamentação Teórica, Fluxos e Medição Passiva do Tráfego

ESTE capítulo apresenta os conceitos e definições usados ao longo do trabalho, e aborda técnicas passivas de coleta de tráfego para análise de desempenho em *links* de acesso ou backbones. Inicialmente são apresentadas as métricas e os descritores utilizados, bem como a influência das características de auto-similaridade e dependência de longo alcance no dimensionamento dos recursos de rede. Em seguida na seção 2.3 são apresentados o conceito e aplicações de fluxos e os métodos de medição passiva do tráfego através da monitoração de fluxos, monitoração SNMP e monitoração de pacotes.

2.1 Métricas de Desempenho

A análise de desempenho determina o quanto um sistema atende satisfatoriamente ou não os requisitos especificados. Em redes LANs e WANs estes requisitos são expressos pela qualidade de serviço (QoS) necessária ao bom desempenho das aplicações. Entretanto, o desempenho desejado nem sempre é garantido na maior parte das redes atuais, onde os serviços são entregues na forma do melhor esforço (*best-effort*).

A definição de QoS é bastante ampla, abrangendo uma série de exigências como desempenho, disponibilidade, confiabilidade, segurança, entre outros [28]. Para atender aos objetivos específicos do dimensionamento de redes, as métricas tipicamente utilizadas como indicadores da qualidade de serviço são:

- **Latência:** Indica o número de *rounds* ou RTT's envolvidos na troca de mensagens entre o cliente e o servidor, desde o início da comunicação até o recebimento da resposta. A latência é função da natureza dos protocolos envolvidos na comunicação.
- **Atraso (*delay*):** É o retardo experimentado pelos pacotes entre dois pontos da rede. O atraso é aleatório para cada pacote do fluxo de dados, ocasionando a variação do retardo ou *jitter*, elemento crítico para aplicações multimídia.
- **Probabilidade de perda:** indica a quantidade de pacotes perdidos em uma transmissão ou fluxo de dados.
- **Vazão (*throughput*):** indica a taxa média na qual os pacotes podem ser servidos pela rede. A máxima vazão possível de ser alcançada é chamada de **capacidade** ou **banda máxima da rede**, sendo normalmente expressa em bits por segundo (bps ou b/s).
- **Utilização:** é a razão entre a taxa de chegada dos pacotes e a taxa de serviço da rede.
- **Distribuição da ocupação da fila:** indica o número médio de pacotes na fila para diferentes taxas de serviço.

É importante salientarmos a relação existente entre desempenho, planejamento e gerenciamento de capacidade no contexto desta tese. Tanto o planejamento quanto o gerenciamento de capacidade têm a finalidade de atender os requisitos de desempenho a custos justificáveis. A diferença entre os dois processos é que o planejamento dimensiona os recursos de rede para atender futuras demandas de carga, enquanto que o gerenciamento faz a otimização dos recursos disponíveis para atingir o máximo

desempenho (*tuning*). Ou seja, o planejamento está focado no futuro e o gerenciamento no presente.

Utilizaremos os termos *dimensionamento* e *planejamento de capacidade* como sendo equivalentes, tendo como propósito comum determinar a capacidade necessária de um enlace para atender às demandas de carga com desempenho e custos adequados.

2.2 Descritores de Tráfego

Os descritores são usados para definir o comportamento estatístico do tráfego, apresentando de forma concisa suas propriedades mais relevantes.

Em nossa abordagem utilizamos um conjunto de descritores para caracterizar os *traces* de pacotes. Para isto, modelamos os *traces* por um processo estocástico $X(i), i = 1, \dots, N$ onde $X(i)$ é o número de pacotes no intervalo de tempo i , ou o tempo entre a chegada do $(i - 1)$ -ésimo e o i -ésimo pacote (*inter-arrival times*). Consideramos os seguintes descritores neste trabalho:

- Taxa média ou Tempo médio $E[X]$: Representa o valor médio das amostras da seqüência. É calculado através da expressão:

$$\mu = E[X] = \frac{1}{N} \sum_{i=1}^N X(i)$$

Onde N é o número de amostras do *trace*.

- Variância $Var[X]$: É definida como sendo o valor médio do quadrado da distância entre $X(i)$ e a média μ . É calculado pela expressão:

$$Var[X] = \frac{1}{N-1} \sum_{i=1}^N (X(i) - \mu)^2$$

- Função de Autocorrelação $autocor(\tau)$: Esta função mostra a relação que um elemento da série temporal tem com outro. Digamos, $X(i)$ com o próximo

elemento $X(i + \tau)$, onde τ indica o deslocamento no tempo.

$$autocor(\tau) = \frac{\sum_{i=1}^{N-\tau} (X(i) - E[X])(X(i + \tau) - E[X])}{\sum_{i=1}^N (X(i) - E[X])^2}$$

A função de autocorrelação é normalmente avaliada para vários valores de τ e seu valor está compreendido no intervalo $[-1, 1]$. Uma função negativa implica em uma relação inversa, isto é, quando $X(i)$ cresce é esperado que $X(i + \tau)$ diminua. Uma função de autocorrelação positiva implica em uma relação direta, isto é, quando $X(i)$ aumenta é esperado que $X(i + \tau)$ também aumente. Quando $autocor(\tau) = 0$ temos uma relação nula ou independência estatística.

- **Parâmetro de Hurst H :** O parâmetro de Hurst indica o grau de dependência temporal em processos estocásticos. É usado como métrica de auto-similaridade e varia no intervalo $(0, 1)$. Para $(1/2 < H < 1)$, o processo tem dependência de longo alcance (*long-range dependence - LRD*). Para $H = 1/2$ as amostras não são correlacionadas e a função autocorrelação tem decaimento exponencial. Para $(0 < H < 1/2)$ o processo apresenta dependência de curto alcance ou dependência negativa. O parâmetro Hurst é estimado através de vários métodos que, em alguns casos, produzem resultados conflitantes [45]. Devido a isto, nas análises realizadas, os resultados foram validados por três métodos diferentes para garantir a real existência de LRD $(1/2 < H < 1)$, e como certificação de que a medida estava correta. Foram utilizados o métodos R/S (*rescaled adjusted range statistic*), o estimador *Whittle* e o método A-V (*Abry-Veitch*). O valor do método R/S foi escolhido para ser representado nos resultados.

2.2.1 Auto-Similaridade

Uma figura geométrica é chamada de auto-similar se a estrutura geométrica que a caracteriza é sempre vista independente da distância de observação. Ou seja, a mesma forma do objeto se repete a medida em que se observa partes mais detalhadas do mesmo. Em séries temporais que representam o tráfego medido em pacotes, bytes

ou bits, a auto-similaridade é uma propriedade referente à invariância da distribuição para qualquer incremento do processo [10], ou quando a série temporal apresenta a mesma estrutura de autocorrelação para diferentes níveis de agregação [49].

Considerando esta última definição, seja a série temporal X_k , $k = 1, 2, \dots$ representando o número de pacotes que passa no enlace no intervalo de tempo δ , e X_k^m a série agregada de nível m definida por $X_k^m = 1/m(X_{km-m+1} + \dots + X_{km})$, $k \geq 1$. Por exemplo, $X_1^3 = (X_1 + X_2 + X_3)/3$ representa a média do tráfego observado durante o intervalo 3δ .

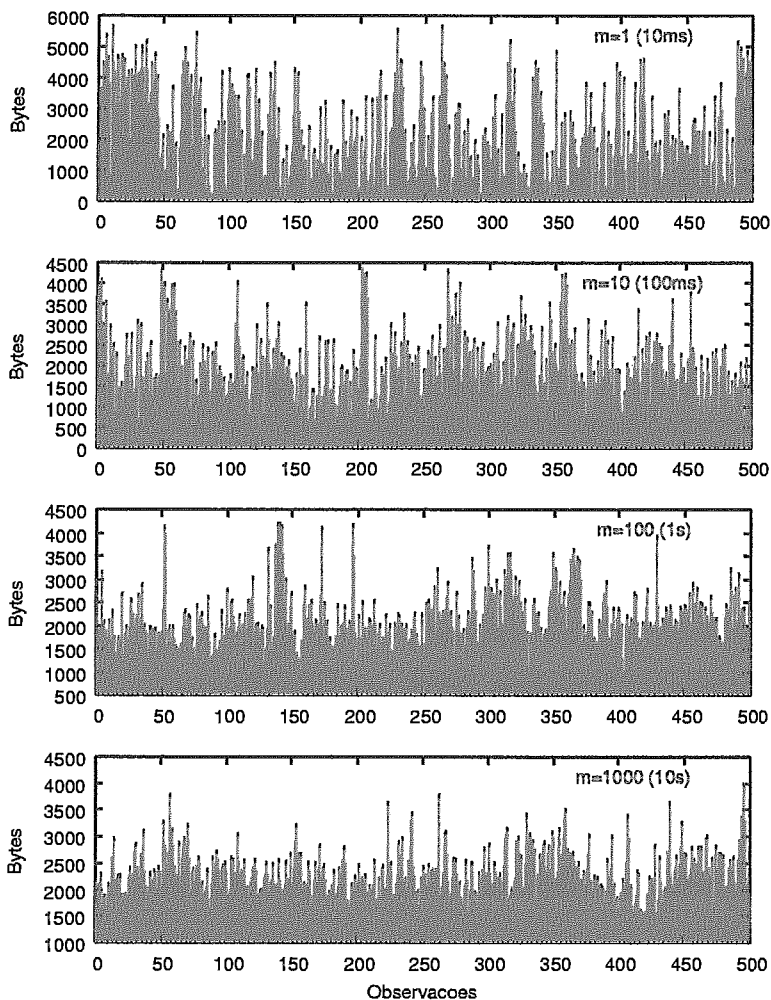


Figura 2.1: Comparativo do tráfego observado para diferentes níveis de agregação.

Sendo a série estacionária, o processo é auto-similar se X_k tem a mesma função de autocorrelação que X_k^m para diferentes valores de agregação m . Portanto a série possui as mesmas propriedades estatísticas independente da escala em que é analisada. A Figura 2.1 mostra quatro séries com diferentes níveis de agregação m , para

$X_k^1, X_k^{10}, X_k^{100}$ e X_k^{1000} , com δ representando os intervalos de tempo de 10ms, 100ms, 1s e 10s. São mostradas as primeiras 500 observações de cada série. Estas séries foram retiradas do *trace* Auckland-6 de 06/12/2001, disponível em [57].

Em contraste com a auto-similaridade determinística de uma figura geométrica, o tráfego observado para diferentes níveis de agregação não repete exatamente a mesma forma para escalas menores de tempo. O que se nota é uma constância de rajadas presente nas várias escalas. Neste caso, a auto-similaridade é expressa pela invariância das estatísticas de segunda ordem do processo estacionário que capturam as rajadas ou variabilidade, e da função de autocorrelação que permanece a mesma para os diferentes níveis de agregação da série.

$$autocor^m(k) \rightarrow autocor(k), \text{ para } m \rightarrow \infty$$

Pode ser observado na figura que o decréscimo de variabilidade é muito lento sendo consistente com

$$Var[X_k^m] = O(m^{-\beta}), \beta \approx 0.4 \in (0, 1)$$

sendo que o parâmetro de Hurst é dado por $H = 1 - \beta/2 \approx 0.80$ [1].

A importância do comportamento auto-similar está no fato de ser difícil determinar uma escala natural de tempo para o dimensionamento, pois o tráfego não se torna suavizado em torno de um valor médio para escalas maiores. Esta característica tem influência direta na distribuição da ocupação e do tempo de espera dos pacotes em uma fila. Esta característica invariante das rajadas observadas no tráfego resulta em uma baixa utilização dos recursos da rede para uma determinada qualidade de serviço. Ou seja, é necessário deixar uma banda ociosa para acomodar o tráfego em eventuais períodos de rajadas.

2.2.2 Dependência de Longo Alcance

Na seção 2.2.1 apresentamos o comportamento auto-similar do tráfego através da invariância da função de autocorrelação para diferentes níveis de agregação de um

processo X_k estacionário. A dependência de longo alcance, por outro lado, refere-se à correlação existente entre as amostras distantes entre si, sendo definida para $1/2 < H < 1$.

Seja a função de autocorrelação em função do parâmetro de Hurst dada por [10]:

$$r(k) = \frac{1}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}]$$

Para $0 < H < 1$ e $H \neq 1/2$, temos que $r(k)$ é equivalente a:

$$r(k) \approx H(2H-1)k^{2H-2}, \quad k \rightarrow \infty$$

Quando $1/2 < H < 1$, $r(k)$ tende assintoticamente para $r(k) \approx ck^\beta$, com $0 < \beta < 1$, $\beta = 2 - 2H$ e a constante $c > 0$. Neste caso, a função de autocorrelação apresenta um decaimento lento, isto é, hiperbólico, tendo soma infinita.

$$\sum_{k=-\infty}^{\infty} r(k) = \infty$$

Isto indica que o processo apresenta dependência positiva, mesmo para amostras distantes entre si no tempo. Estes processos são definidos como tendo dependência de longo alcance ou de longa duração (*long-range dependence - LRD*). A restrição $1/2 < H < 1$ imposta na definição de LRD implica que processos com auto-similaridade assintótica de segunda ordem também apresentam LRD e vice-versa. Por esta razão, os termos auto-similaridade e dependência de longo alcance são usados indistintamente dentro deste contexto.

O reflexo auto-similar ou LRD na ocupação da fila está no decaimento assintótico mais lento do que o exponencial, podendo ocasionar o aumento do tamanho médio da fila. Aumentar os buffers dos roteadores para reduzir as perdas não traz resultados satisfatórios, uma vez que as perdas não diminuem na mesma proporção que o tempo de espera na fila aumenta. Isto implica em políticas de provisionamento que utilizam

buffers pequenos (entre 50 a 100 pacotes) para canais de alta velocidade. Além disto, a utilização de buffers menores tende a limitar os efeitos da dependência de longo alcance. Ou seja, se a capacidade do buffer for menor, então a memória LRD é diminuída, sendo mais relevante para o dimensionamento considerar a correlação de curto alcance [33, 36, 71].

Como o aumento do buffer traz pouco impacto na melhoria da taxa de perdas, é necessário dimensionar corretamente a capacidade do canal para atender aos picos e rajadas mais prolongados, oferecendo o nível de desempenho (QoS) adequado, principalmente para os fluxos de tempo real que são os mais sensíveis aos atrasos e perdas de pacotes.

2.3 Fluxos e Medição Passiva do Tráfego

Esta seção apresenta o conceito de Fluxo e sua utilização na análise do tráfego, bem como os métodos usados em medições passivas.

Um fluxo IP é um conjunto de pacotes observados em um intervalo de tempo e que possuem propriedades em comum. Os fluxos em redes de dados foram inicialmente observados por Jain [41] que classificou as chegadas dos pacotes trafegando em rajadas entre os vários nós da rede, como *packet train model*. O modelo de Jain consistia em caracterizar o conjunto de pacotes que fluía nas duas direções entre dois nós da rede. Era definido um tempo máximo *maximum allowed intercar gap - MAIG* que determinava o fim de um fluxo *packet train* e o início de outro.

Em [18] é apresentada uma caracterização dos fluxos quanto a (i) sua direcionalidade (unidirecionais ou bidirecionais), (ii) seu nível de agregação (domínio, rede, subrede, IP, porta, interface) e (iii) protocolo de transporte usado pela aplicação (TCP ou UDP). Os fluxos podem ser definidos usando diferentes níveis de informações, tornando-os mais agregados a medida que sua definição envolve parâmetros de rede mais abrangentes. A definição mais apropriada depende do objetivo do estudo. Para análises envolvendo o comportamento do protocolo de transporte em uma conexão, por exemplo, o fluxo deve conter informações mais detalhadas de sua

fonte e destino, como o endereço e as portas de origem e destino. Já para o planejamento de redes e problemas envolvendo engenharia de tráfego, a análise pode ser feita em um nível mais agregado de informação (domínio, rede ou subrede). Os fluxos são formados contabilizando-se os pacotes que coincidem com sua definição. Eles são finalizados quando é sinalizado o fim da conexão (flag FIN para o protocolo TCP), ou quando não é mais observado tráfego para o fluxo durante um intervalo de tempo (*timeout*). As informações armazenadas para cada fluxo, além da sua identificação, são o tempo de início, a duração, o número de bytes e de pacotes transmitidos. Entretanto, os fluxos, pela sua característica de agregar dados de pacotes, resumizam a informação de forma que as propriedades do tráfego em escalas pequenas de tempo são perdidas.

A definição de fluxo aplicada no método do Capítulo 3 segue a proposta de [18], sendo a mesma utilizada pela ferramenta *Netflow* incorporada nos roteadores Cisco [16]. Esta definição consiste na seqüência de pacotes que flui entre dois pontos da rede, de forma unidirecional e caracterizada por um conjunto de cinco parâmetros: Endereço IP de origem, Endereço IP de destino, Porta de origem, Porta de destino e Protocolo de transporte. Fluxos com esta definição representam o comportamento de geração de tráfego das aplicações, pois as conexões TCP são bem caracterizadas por fluxos porta a porta [26]. Requisições HTTP, por exemplo, caracterizam-se pelo grande número de pequenos fluxos, que raramente entram para a fase de *congestion avoidance* do TCP, desenvolvendo apenas a fase de *slow-start* [50, 78]. Já os fluxos provenientes de conexões FTP, como o *download* de arquivos, são longos, permanecendo a maior parte de sua duração na fase de *congestion avoidance*.

A classificação dos fluxos conforme sua duração, taxa e quantidade de pacotes, tem sido alvo de recentes pesquisas [75, 62, 12, 79]. Os resultados destes estudos promovem um melhor entendimento da dinâmica do tráfego agregado, evidenciando a necessidade de um tratamento diferenciado para as diferentes classes de fluxos, fator essencial para a melhoria de desempenho das aplicações.

2.3.1 Monitoração de Fluxos

Medidas em nível de fluxos estão disponíveis em roteadores comerciais Cisco e Juniper [16, 43], sem a necessidade de adicionar à rede equipamentos externos de coleta. Outras implementações como o *Netramet* [11] fazem a monitoração passiva de fluxos, mas necessitam de equipamento adicional para a captura.

O *Netflow* é uma ferramenta de medição passiva que armazena as informações dos fluxos que passam por uma determinada interface do roteador. É amplamente utilizado pelos administradores para análise e gerência do tráfego [64], [9]. Resumidamente, o *Netflow* funciona da forma descrita a seguir.

Um fluxo *Netflow* é identificado por cinco parâmetros: Endereço IP de origem e destino, porta IP de origem e destino e protocolo. O roteador mantém uma tabela de registros dos fluxos ativos que passam por ele. Os registros possuem contadores que contabilizam os bytes, os pacotes e a duração do fluxo, e são acessados através de uma chave. Para cada pacote que chega ao roteador é verificado se existe um fluxo ativo para aquele pacote. Caso positivo, os contadores do registro são atualizados. Caso contrário, um novo registro é criado para este fluxo iniciado. O início do fluxo é determinado pelo *time-stamp* do primeiro pacote para o fluxo. Quando o fluxo é terminado, seu registro é exportado para um coletor e a memória do roteador liberada para novos fluxos. O processo de formação dos fluxos pelo *Netflow* e o datagrama UDP com os fluxos exportados são ilustrados na Figura 2.2.

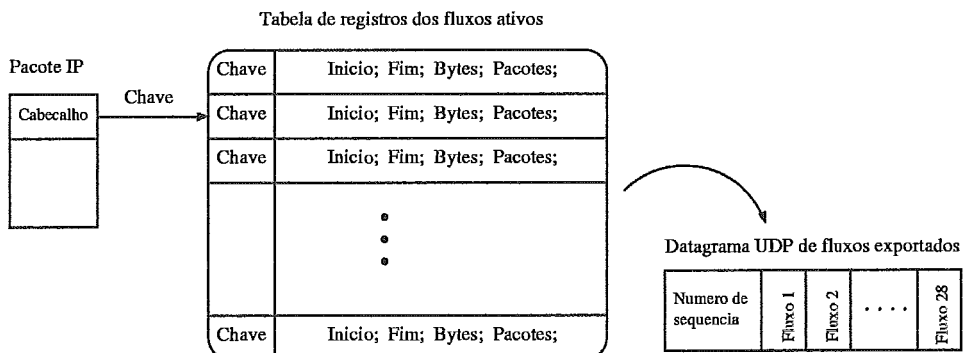


Figura 2.2: Acesso e criação dos fluxos e datagrama UDP de exportação.

Os fluxos se mantêm na memória do roteador até serem terminados, conforme

as seguintes regras de expiração:

- Após o recebimento de um pacote RST ou FIN para fluxos TCP;
- Nenhum pacote pertencente ao fluxo é recebido durante mais de 15 segundos;
- Quando a duração do fluxo atinge 30 minutos;
- Quando tabela de fluxos do roteador está cheia.

Os fluxos expirados são agrupados em datagramas UDP e exportados para um equipamento coletor com software específico para tratamento dos fluxos [32, 17]. Os datagramas exportados para o coletor podem ser perdidos na transmissão, dado que o protocolo UDP não oferece serviço confiável. Estes datagramas contêm um número de seqüência para permitir a identificação de possíveis perdas no processo de transmissão de fluxos. Um total de 28 fluxos formam o datagrama UDP de exportação na versão 7 do *Netflow*.

A precisão das informações de fluxos fornecidas pelo *Netflow* foi avaliada em [73] e [74], onde foi implementada uma estrutura de coleta de fluxos e pacotes simultânea para comparação entre os dois. Os resultados mostraram que as medidas de bytes, pacotes e duração da conexão produzidas pelo *Netflow* fornecem valores com muito boa precisão.

Os fluxos ativos são mantidos em memória pelo *Netflow* consumindo importantes recursos do roteador que deve manter atualizada a tabela de fluxos na chegada de cada pacote. Em *links* de grande velocidade este consumo deve ser limitado para não comprometer o desempenho do roteador, pois o número de fluxos concorrentes cresce com o aumento da velocidade do *link*. O *Netflow* pode ser configurado para capturar um a cada "N" pacotes (1:N) no modo *sampled Netflow*, reduzindo consideravelmente o processamento. Os fluxos formados pelas amostras possuem uma granularidade menor do que os fluxos coletados pelo processo normal. A frequência de amostragem é configurável e requer um compromisso entre precisão e consumo de memória. Uma baixa frequência de amostragem produz baixo consumo de memória, porém diminui a precisão da informação nos fluxos coletados.

Embora a utilização de fluxos seja de grande aplicação, atualmente não existe um consenso no padrão de formato utilizado pelos dispositivos de coleta. Esforços para padronizar medições em nível de fluxos tem sido feitos pelo IETF (*The Internet Engineering Task Force*) [39], pelo grupo RTFM (*Real-Time Traffic Flow Meter*) [13] e pelo grupo de trabalho IPFIX (*IP Flow Information Export*) [34] que estuda a definição de um formato e protocolo para envio dos dados do equipamento de coleta para outros sistemas para análise e armazenamento das informações.

2.3.2 Monitoração SNMP

O SNMP (*Simple Network Management Protocol*) [15] é um protocolo em nível de aplicação que facilita a troca de informações entre as estações de gerenciamento e os agentes residentes nos elementos gerenciados, como roteadores, switches, servidores entre outros. Uma rede gerenciada por SNMP possui três elementos chaves: os dispositivos gerenciados, os agentes e as estações de gerenciamento. Através desta infra-estrutura, o SNMP disponibiliza aos administradores de rede informações sobre a utilização dos recursos e alarmes, o que permite identificar e solucionar problemas e planejar o crescimento da rede.

Os dispositivos gerenciados coletam e armazenam as informações na sua base de dados (*Management Information Base - MIB*) e a tornam disponível, via SNMP, para as estações de gerenciamento. O agente tem função específica de coletar os dados armazenadas na base MIB e transformá-los em informações compatíveis com o SNMP.

O padrão MIB que contempla a base de dados relacionado com a coleta de tráfego é a MIB-II [52]. Sua organização contém um conjunto de grupos que monitoram a execução de vários protocolos no elemento de rede, como o IP, TCP/UDP, BGP, OSPF, fornecendo o estado (ativo/inativo) das interfaces do roteador ou contabilizando as estatísticas dos contadores de bytes e pacotes enviados, recebidos ou perdidos para cada interface. Informam também dados relevantes ao desempenho do roteador como utilização de CPU e memória.

As estações de gerenciamento coletam as informações dos dispositivos normalmente a cada 5 minutos, ou seja, no caso do número de bytes enviados por uma interface, a informação recebida é a média neste intervalo de tempo. Este intervalo é demasiado grande para a parametrização de modelos de tráfego ou determinação de rajadas que ocasionam perdas sucessivas de pacotes. Por outro lado, todas as informações recebidas ficam armazenados na estação de gerenciamento, formando um completo histórico de atividade dos dispositivos gerenciados, podendo ser usadas para análise de tendências de longo prazo [60]. Softwares de domínio público são usados para armazenar e organizar os dados em forma de relatórios e gráficos que são configurados pelos administradores conforme suas necessidades [58, 70]

2.3.3 Monitoração de Pacotes

Em uma rede de pacotes, monitorar a unidade básica de informação, o pacote, representa obter o maior nível possível de detalhamento do tráfego. Podem ser analisadas as características predominantes nas escalas menores de tempo, bem como o agregado de pacotes ou bytes em escalas maiores.

Tipicamente, esta monitoração é realizada por um computador equipado com placa de captura compatível com o meio físico, ou seja, em mídia compartilhada como a Ethernet ou enlaces sem fio, um monitor de pacotes pode ser um computador conectado à rede com sua interface de rede operando em modo *promíscuo* para criar uma cópia local de cada pacote. Já em enlaces ponto a ponto é necessário uma derivação do meio físico para a conexão do monitor, através de uma conexão em "Y", sempre observando a compatibilidade da placa de captura com o tipo de mídia (ATM, HSSI, POS, etc..). No entanto, a divisão de um *link* pode interferir na qualidade elétrica ou ótica do sinal entre dois roteadores. Nestes casos, pode ser necessário o uso de amplificadores ou adaptadores de impedância para evitar a degradação do sinal recebido pelos roteadores e monitor de pacote.

O processo de captura envolve a análise e a cópia do pacote. A análise compreende a validação do pacote mediante alguma regra ou filtro de captura, onde o

monitor pode ser configurado para selecionar uma faixa específica de endereços ou portas de rede, caso contrário o pacote é descartado. Para pacotes válidos é realizada a cópia apenas do cabeçalho IP (20 bytes), ou dos cabeçalhos IP e TCP (40 bytes) e adicionado o tempo de chegada do pacote (*time-stamp*) (8 bytes). Cada pacote, após a coleta, terá um tamanho que depende da quantidade de informação capturada. Alguns monitores capturam inclusive dados ao nível de enlace, elevando o tamanho do pacote para 72 ou 88 bytes. A Figura 2.3 mostra a formação de um pacote típico com as informações normalmente capturadas. Quando o protocolo de transporte é o UDP, apenas as informações de porta de origem e destino são adicionadas ao pacote.

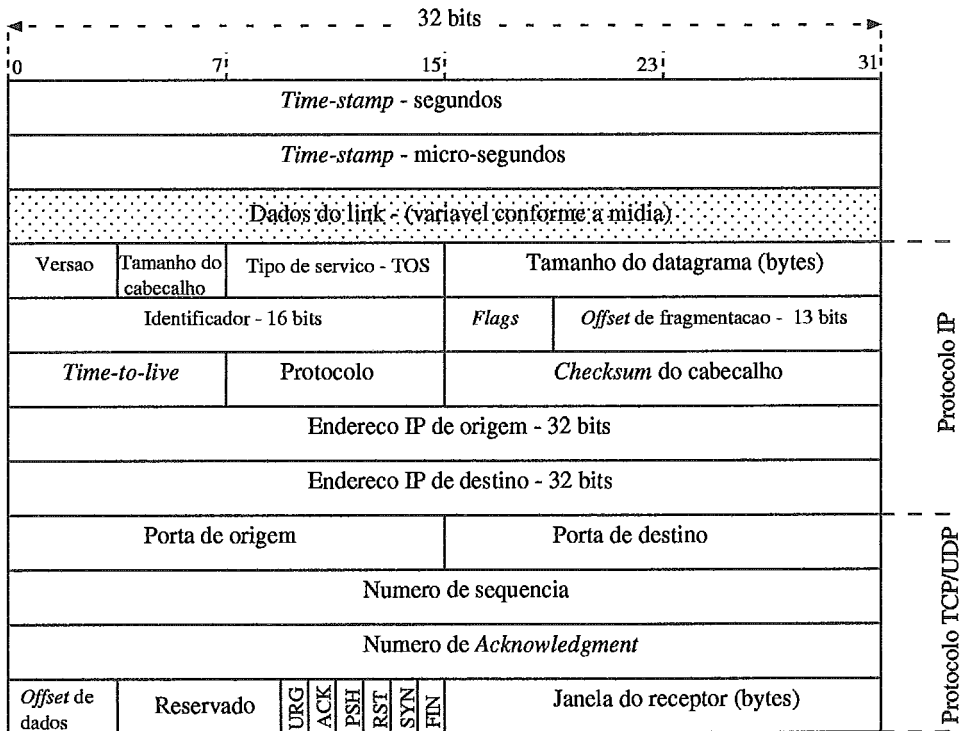


Figura 2.3: Estrutura do registro do pacote após a coleta.

A partir dos pacotes coletados é possível derivar um nível mais agregado de informação, como a reconstrução de fluxos, com base no *time-stamp* e dos campos IP e TCP, ou na contagem de pacotes ou bytes por intervalo de tempo, onde é necessário apenas o *time-stamp* e o tamanho do datagrama. No entanto, a monitoração de pacotes exige uma infra-estrutura específica de coleta, que torna-se mais complexa com o aumento da velocidade do *link* [29, 30]. O monitor deve ser capaz de capturar,

processar e armazenar os pacotes na medida em que chegam. Isto exige placas de captura co-processadas com barramentos de grande velocidade de transferência de dados. Muitas das funções de captura, como a geração do tempo de chegada do pacote, é realizada por hardware. Além disto, é necessária uma capacidade grande de armazenamento secundário dos pacotes coletados [38]. Um *link* com tráfego de 20000 pacotes por segundo (equivalente a 128 Mbps, considerando-se um pacote médio de 800 bytes) produz, em 60 minutos de coleta, com pacotes de 72 bytes, um volume de 5.2 Gbytes ($3600s \times 20000 \times 72$).

Recentemente, vários *traces* de pacotes vem sendo disponibilizados publicamente por grupos de pesquisas em medições passivas, com objetivo de fornecer à comunidade científica dados para o estudo do comportamento do tráfego Internet em *links* de alta velocidade [57].

2.3.4 Resumo

Neste capítulo apresentamos os conceitos básicos de representação do tráfego, desde sua coleta até a sua caracterização estatística através de descritores. A média, a variância, a autocorrelação e o parâmetro de Hurst serão utilizados ao longo desta tese. Estes descritores foram escolhidos pois são usados na parametrização de diversos modelos de tráfego da literatura conforme será mostrado no Capítulo 4.

Procuramos mostrar o estado-da-arte em medições passivas de tráfego, descrevendo as vantagens e desvantagens de cada método, com ênfase na monitoração de fluxos que pode ser obtida diretamente dos roteadores sem a necessidade de equipamentos de coleta específicos. Mesmo assim, a monitoração em nível de pacotes é indispensável para se capturar as propriedades do tráfego em pequenas escalas de tempo.

No próximo capítulo é proposto um método capaz de recuperar as informações em pequenas escalas de tempo a partir dos fluxos.

Capítulo 3

Método para Decomposição

Temporal de Fluxos em Pacotes

NESTE capítulo é proposto um método para determinar o tráfego agregado em pequenos intervalos de tempo a partir de informações de fluxos normalmente coletadas pelos roteadores. O método possibilita estudar o tráfego com uma granularidade mais fina, oferecendo maior precisão no planejamento de capacidade e no gerenciamento das redes de dados.

3.1 Introdução

Determinar o comportamento do tráfego através de medidas detalhadas é requisito fundamental para o dimensionamento e o gerenciamento dos recursos em uma rede de dados. Atualmente, os administradores se valem ou de medições SNMP, (mas estas são ineficientes para o detalhamento do tráfego), ou da monitoração de pacotes que traz como inconveniente a necessidade de uma infraestrutura específica para a coleta e armazenamento, tarefa que se torna mais complexa a medida que a velocidade e a disponibilidade dos canais de comunicação aumentam.

O comportamento do tráfego Internet atual é resultado de uma estrutura hierárquica de protocolos, aplicações e do próprio comportamento do usuário [66].

Cada nível desta organização apresenta sua própria dinâmica que ocorre em escalas de tempo diferenciadas. No contexto da Internet ou de redes WAN, um usuário ao acessar um servidor Web estabelece uma sessão que irá produzir uma seqüência de pacotes a serem transmitidos pelos protocolos inferiores da hierarquia de rede. Ou seja, uma sessão Web, por exemplo, dará origem a várias requisições HTTP que serão encapsuladas pelo protocolo TCP e roteadas até seu destino através de pacotes IP.

Como parte desta estrutura hierárquica estão os fluxos que ocupam posição intermediária entre as sessões e os pacotes. No caso do acesso Web estes são representados pelas solicitações e respostas HTTP. Assim sendo, os fluxos são diretamente influenciados pela dinâmica das sessões e fornecem informações sobre o tráfego para uma determinada escala de tempo. Porém, para caracterizar completamente o tráfego, considerando-se todos os níveis hierárquicos, é necessário também a análise em escalas menores de tempo, normalmente abaixo de 1s [1]. Nestas escalas existe uma alta correlação no padrão do tráfego influenciada pelo mecanismo de congestionamento do TCP [27].

Dado que as informações em nível de fluxos são relativamente fáceis de serem obtidas, mas não capturam as propriedades do tráfego em pequenas escalas de tempo, a pergunta que podemos fazer é: *É possível gerar a seqüência de pacotes a partir dos dados coletados para um fluxo ?* Ou, de uma outra forma, *como obter dados para uma escala de tempo menor partindo-se de informações coletadas para uma escala de tempo maior ?*

A obtenção de uma escala menor de tempo para a análise detalhada do tráfego, sem necessitar de uma infraestrutura complexa de captura de pacotes, é o objetivo do método proposto. A base para isto são os fluxos normalmente capturados pelos roteadores e sua posterior decomposição temporal em pacotes obedecendo a dinâmica do protocolo TCP, reproduzindo desta forma as propriedades do tráfego em várias escalas de tempo. Estudos recentes mostram que a maior parte do tráfego da Internet é gerado por aplicações que usam o protocolo TCP [62]. Este tráfego é formado por fluxos elásticos, pois variam sua taxa para se adaptar à banda disponível

do canal em função de possíveis congestionamentos. Portanto, o algoritmo para a geração da seqüência de pacotes é baseado em modelos da literatura do protocolo TCP em suas fases de *slow start* e *congestion avoidance*.

A organização deste capítulo é descrita a seguir. A seção 3.2 descreve o método e apresenta o algoritmo proposto para a geração de pacotes. A seção 3.3 apresenta a influência dos fluxos não TCP no desempenho do algoritmo. O procedimento utilizado na validação do método e um estudo comparativo entre as estatísticas obtidas dos *traces* real e estimado são apresentados na seção 3.4. O erro produzido pelo método é avaliado na seção 3.5. Na seção 3.6 utilizamos o método na parametrização de dois modelos de tráfego como exemplo de aplicação. Por fim, na seção 3.7 são apresentadas as conclusões e possíveis aplicações para o método.

3.2 Método para a geração de pacotes

Determinar a distribuição dos pacotes ao longo da duração do fluxo não é tarefa trivial. Um fluxo de 10s e 500 pacotes, por exemplo, pode ter sido formado por uma rajada de 200 pacotes no primeiro segundo e os demais pacotes distribuídos uniformemente no restante do tempo ou por diversas rajadas de 100 pacotes distribuídas aleatoriamente no tempo. Esta distribuição dos pacotes ao longo da duração do fluxo tem forte influência no cálculo de estatísticas de primeira e de segunda ordem da seqüência de pacotes, assim como na distribuição do tamanho de uma fila alimentada por esta seqüência.

O método proposto tem como objetivo estimar esta distribuição conforme a dinâmica do protocolo TCP. O comportamento do TCP Reno é reproduzido com base nos modelos propostos para as fases de *slow start* [14, 80], e *congestion avoidance* [51, 59]. Os modelos de [14] e [80] são muito semelhantes, no entanto foram usadas as equações de [80], pois este modelo apresenta uma aproximação melhor para o comportamento do TCP na fase de *slow start*. Com relação a fase de *congestion avoidance*, foi usado o modelo de [59] que é baseado em [51].

Nos trabalhos [14, 51, 59, 80] os autores consideraram as seguintes premissas: (i)

É utilizado o envio de ACK cumulativo no recebimento de dois pacotes consecutivos (*delayed ACK*); (ii) A cada RTT (*round trip time*) é transmitido um conjunto de W pacotes, definidos pelo tamanho da janela. Estes W pacotes são enviados em forma de rajada, sem intervalo de tempo entre eles; (iii) Foi considerado que não existe limitação no lado do transmissor, ou seja, a janela de congestionamento máxima é definida pelas perdas na rede ou pelo buffer do receptor; (iv) As perdas são modeladas de duas formas: na ocorrência de três ACKs duplicados (*triple-duplicated ACKs*) e por expiração do temporizador (*time-outs*). O primeiro caso será referenciado como TD e o segundo como TO.

Além das premissas descritas acima, consideramos também:

1. Durante a fase de *slow start* é assumido que o limite (*threshlod*) para início da fase de *congestion avoidance* é sempre maior que a janela máxima de transmissão, ou seja, não tem efeito inicial.
2. As perdas (p) na rede podem ser ajustadas, sendo que o valor adotado para a geração de todas as seqüências deste trabalho foi de 0,005 [51].
3. O buffer do receptor (*RecWin*) foi ajustado em 16 Kbytes, o que proporciona uma janela máxima de transmissão que varia entre 10 e 30 pacotes. Estes também são valores encontrados na literatura [7] e em simuladores como o NS [40].
4. A dinâmica do TCP é implementada para todos os fluxos coletados, independentemente do protocolo de transporte. Ou seja, é considerado que o tráfego agregado é fortemente influenciado pelo comportamento do TCP. Esta condição será analisada com maiores detalhes na seção 3.3. A exceção desta regra são os fluxos de ACKs identificados quando o MSS (*MSS - Maximum segment size*) é inferior a 80 bytes. Neste caso, os pacotes são distribuídos uniformemente pela duração do fluxo.
5. Um fluxo sempre inicia pela fase de *slow start*, independente se ele é continuação de algum outro que foi interrompido pelas regras de formação.

6. O algoritmo analisa individualmente cada fluxo do *trace* de entrada que representa um tráfego agregado.
7. Caso seja necessário gerar o *time-stamp* de cada pacote, é feita uma distribuição uniforme dos W pacotes no tempo RTT. Os *time-stamps* são úteis nos modelos de tráfego que utilizam os tempos entre chegadas em sua parametrização. O procedimento de geração, bem como as estatísticas comparativas entre o tempo real e o estimado será apresentado ao longo deste capítulo.
8. A definição de fluxo usada no método segue a proposta de [18] apresentada no capítulo anterior. Fluxos com esta definição são substitutos adequados para as conexões TCP [26].

3.2.1 Estrutura Geral

O método de distribuição de pacotes está dividido em duas etapas principais que são executadas para cada fluxo individualmente: (1) Leitura dos dados do fluxo e determinação do seu RTT, e (2) Distribuição dos pacotes em um vetor de contagem de pacotes, conforme a evolução da janela de congestionamento.

A entrada do algoritmo é um *trace* contendo o total de fluxos ordenados pelo início de cada fluxo (*time-stamp*). O custo é linear ao número de fluxos a serem analisados. Após a leitura dos dados do fluxo são calculados os parâmetros que permitirão determinar o RTT. O detalhamento é apresentado no algoritmo da Figura 3.2.

A principal estrutura de dados do algoritmo é um vetor $A[i], i = 0, 1, \dots, m$, de $m + 1$ posições onde cada posição representa o número de pacotes transmitidos em um determinado intervalo de tempo. O índice do vetor equivale a um período de tempo constante. Desta forma, se o intervalo de tempo for igual a $1s$, temos que $A[i], i = 0, 1, \dots, m$ é igual ao número de pacotes que foram enviados no intervalo $[i, i + 1s)$. A posição $A[0]$ do vetor corresponde aos pacotes transmitidos no primeiro intervalo de tempo do fluxo com o menor *time-stamp*. Desta forma, se o primeiro fluxo do *trace* tiver duração de $7.2s$, ele terá seus pacotes distribuídos nas 8 primeiras

posições do vetor ($A[0] \dots A[7]$).

A distribuição de pacotes segue a evolução da janela do TCP para cada RTT, até que o tempo decorrido seja igual ao tempo de duração do fluxo. A Figura 3.1 ilustra o mecanismo de distribuição dos pacotes para dois fluxos hipotéticos com os seguintes dados:

Fluxo 1: 2018984000;7244;29000;58.

Fluxo 2: 2018988000;5438;43985;31.

Neste exemplo, os fluxos 1 e 2 são representados pelo *time-stamp* que está em formato *UNIX* pela duração do fluxos expressa em milisegundos e pelo número de bytes e pacotes.

O Fluxo 1 tem 58 pacotes, duração de 7.244s, inicia no tempo 0 do vetor e tem $RTT = 0.9s$. A evolução da janela, que será detalhada na fase de *slow start*, ocorre da seguinte forma: 1, 2, 3, 5, 8, 13, 21, 5. O Fluxo 2 tem 31 pacotes, duração de 5.438s, inicia 4s após o primeiro e tem $RTT = 0.9s$. Sua janela evolui da seguinte forma: 1, 2, 3, 5, 8, 12. O resultado final produzido pelo algoritmo é uma seqüência S contendo o número de pacotes por intervalo de tempo (armazenado em cada posição do vetor $A[i]$) que é igual a $S = 1, 2, 3, 5, 9, 15, 24, 10, 8, 12$.

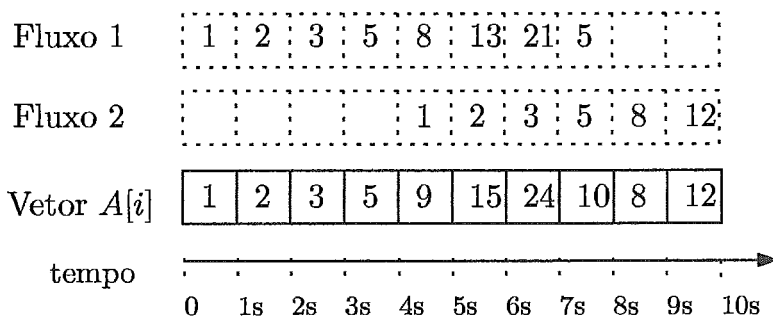


Figura 3.1: Estrutura de distribuição e contagem dos pacotes

3.2.2 Fase de Inicialização

O algoritmo tem como entrada um *trace* que representa um conjunto de fluxos contendo as informações de início (s), duração (l), quantidade total de bytes (b) e

número total de pacotes (d) de cada um dos fluxos. Os fluxos devem estar ordenados pelo seu início para a leitura do algoritmo. Estas informações são usadas para calcular os seguintes parâmetros:

O tamanho do segmento máximo MSS:

$$MSS = \frac{b}{d}$$

A janela do receptor W_m :

$$W_m = \lceil \frac{RecWin}{MSS} \rceil$$

A taxa média do fluxo B_w :

$$B_w = \frac{d}{l}$$

Os pacotes enviados até a ocorrência da primeira perda na fase de *slow start* [14]:

$$Edss = \frac{(1 - (1 - p)^d)(1 - p)}{p} + 1$$

Os pacotes enviados nas fase de *congestion avoidance* $Edca$:

$$Edca = d - Edss$$

A duração da fase de *slow start* Tss :

$$Tss = l - Tca$$

A duração da fase de *congestion avoidance* Tca :

$$Tca = \frac{Edca}{B_w}$$

A Figura 3.2 ilustra a sequência do cálculo destes parâmetros.

Nesta fase todas as posições do vetor $A[i], i = 0, 1, \dots, m$, são inicializadas com zero. O número de posições do vetor depende da duração do *trace*. Caso tenhamos um *trace* de fluxos de uma hora e intervalos de tempo de 1s, teremos um vetor de 3600 posições.

```

Entrada: Arquivo com os fluxos ordenados pelo tempo de início
Saída: Vetor com a contagem dos pacotes por intervalo de tempo

1. Inicialização:  $A[i] := 0, i = 0, 1, \dots, m, p = 0.005, RecWin = 16000$ 
2. foreach (fluxo do arquivo de entrada)
3.   Lê o início ( $s$ ), duração ( $l$ ), bytes ( $b$ ) e número de pacotes ( $d$ )
4.    $MSS = \frac{b}{d}$  /* MSS do fluxo */
5.    $W_m = \lceil \frac{RecWin}{MSS} \rceil$  /* Janela máxima do receptor (em pacotes) */
6.    $B_w = \frac{d}{l}$  /* taxa média do fluxo */
7.    $Edss = \frac{(1-(1-p)^d)(1-p)}{p} + 1$  /* pacotes enviados no slow start antes da primeira
perda */
8.    $Edca = d - Edss$  /* pacotes do congestion avoidance */
9.    $Tca = \frac{Edca}{B_w}$  /* Duração do congestion avoidance */
10.   $Tss = l - Tca$  /* Duração do slow start */
11.  if ( $Edca \leq 0$ )
12.    /* Fluxo tem apenas slow start. */
13.     $RTT = RTT\_SS()$  /* O RTT do fluxo é o  $RTT_{ss}$  */
14.    CALL slow start(); /* vai para a fase de slow start */
15.  else
16.    /* Fluxo tem slow start e congestion avoidance. */
17.    CALL  $RTT\_SS()$  /* Calcula o  $RTT_{ss}$  */
18.    CALL  $RTT\_CA()$  /* Calcula o  $RTT_{ca}$  */
19.     $RTT = \min(RTT_{ss}, RTT_{ca})$ . /* O RTT do fluxo será o menor dos dois */
20.    if ( $RTT < RTT_{ss}$ )
21.       $Tss = (Tss/RTT_{ss})RTT$  /* Ajusta  $Tss$  e  $Tca$  para o novo RTT */
22.       $Tca = l - Tss$ 
23.    end
24.    CALL slow start(); /* vai para a fase de slow start */
25.    CALL cong-avoid(); /* vai para a fase de congestion avoidance */
26.  end
27. end

```

Figura 3.2: Módulo principal do algoritmo com a inicialização dos parâmetros.

O RTT do fluxo (RTT) é definido escolhendo-se o menor valor entre o RTT calculado para a fase de *slow start* (RTT_{ss}) e o calculado para a fase de *congestion avoidance* (RTT_{ca}). A escolha do menor valor garante que o somatório dos RTT's de cada fase não seja maior do que a duração destas fases, ou seja, o tempo total de ciclos RTT deve estar contido na duração do fluxo. Abordagem semelhante foi adotada em [59]. O cálculo do RTT das fases de *slow start* e *congestion avoidance* é realizado com a chamada das funções RTT_SS e RTT_CA durante a fase de inicialização. O algoritmo para determinar o RTT_{ss} e o RTT_{ca} é apresentado na Figura 3.3.

As durações das fases de *slow start* (Tss) e *congestion avoidance* (Tca) são inicialmente calculadas através da taxa média B_w do fluxo. Caso o RTT_{ca} do fluxo seja

menor que o RTT_{ss} , estas durações são reajustadas para um novo valor de forma que a soma do número total de ciclos RTT (*rounds*) seja igual a duração do fluxo. Este ajuste é mostrado nas linhas 21 e 22 da Figura 3.2.

Após a inicialização dos parâmetros e determinação do RTT, são iniciadas as fases de *slow start* e *congestion avoidance*. O detalhamento destas fases será apresentado a seguir.

3.2.3 Fase de *Slow Start*

O algoritmo usado para distribuir os pacotes nas posições correspondentes do vetor $A[i]$, conforme evolução da janela de congestionamento do *slow start*, está detalhado na Figura 3.4. A evolução desta janela segue a seguinte relação [80]:

$$cwnd = 0.72(1.62)^j + 0.28(-0.62)^j, \quad j = 1, 2, 3...$$

Desta forma, para 8 *rounds*, a evolução da janela será 1, 2, 3, 5, 8, 13, 21, 34.

A determinação do RTT_{ss} na fase de *slow start* pode assumir duas formas, conforme o modelo de [80]:

(i) Quando não há limitação da janela do receptor. Neste caso, a janela máxima de transmissão será dada em função da primeira perda. Ou seja, a evolução máxima da janela de congestionamento para o *slow start* é calculada para o total de pacotes que são enviados sem perda, conforme a equação abaixo:

$$RTT_{ss} = \frac{T_{ss}}{\lceil \log_g \left(\frac{Ed_{ss}+2}{C_1} \right) \rceil - 2}$$

onde $g = 1.618033989$ e $C_1 = 0.723606798$ são constantes de Fibonacci.

(ii) Quando há limitação da janela do receptor. Neste caso o crescimento exponencial da janela de transmissão ocorre até o valor máximo da janela do receptor

W_m , permanecendo constante neste valor.

$$RTT_{ss} = \frac{T_{ss}}{\lceil \log_g \left(\frac{W_m}{C_1} \right) \rceil + \frac{1}{W_m} (Edss - g^2 W_m - 2)}$$

```

sub RTT_SS /* Cálculo do RTT do slow start */
/* Constantes:  $g = 1.618033989$ ,  $C_1 = 0.723606798$  */
1.  $W_{m_{ss}} = (Edss + 2)/g^2$  /* Determina a janela máxima para enviar  $Edss$  */
2. if ( $W_{m_{ss}} > W_m$ ) /* Ocorre limitação da janela do receptor */
3.  $RTT_{ss} = \frac{T_{ss}}{\lceil \log_g \left( \frac{W_m}{C_1} \right) \rceil + \frac{1}{W_m} (Edss - g^2 W_m - 2)}$ 
4. else
5.  $RTT_{ss} = \frac{T_{ss}}{\lceil \log_g \left( \frac{Edss+2}{C_1} \right) \rceil - 2}$ 
6. end
:
sub RTT_CA /* Cálculo do RTT do congestion avoidance */
/* Constantes:  $p = 0.005$ ,  $b = 2$  (ACK cumulativo) */
1.  $RTT_{ca} = \min \left( \frac{W_m}{B_w}, \frac{1}{B_w \sqrt{\frac{2bp}{8} + 3.3(\sqrt{\frac{3bp}{8}})p(1+32p^2)}} \right)$ 
2. end

```

Figura 3.3: Cálculo de RTT para as fases de *slow start* e *congestion avoidance*

```

Fase de Slow Start /* Distribui os pacotes nas posições de  $A[i]$  */
1. while ( $Edss$ )
2.  $cwnd = 0.72(1.62)^j + 0.28(-0.62)^j$  /* Calcula a janela para envio de pacotes */
3.  $T_j = T_j + RTT * j$  /* Calcula o tempo decorrido durante a evolução da janela */
4.  $A[T_j] += cwnd$  /* Adiciona o valor de  $cwnd$  na posição  $A[i]$  correspondente */
5.  $Edss = Edss - cwnd$ ;  $j = j + 1$ 
6. end

```

Figura 3.4: Algoritmo para a fase de *slow start*.

3.2.4 Fase de *Congestion Avoidance*

Para os fluxos que não tiveram os pacotes totalmente enviados na fase de *slow start*, entra em ação a fase de *congestion avoidance* para o envio dos pacotes restantes. Da mesma forma que para a fase anterior, é realizada a distribuição dos pacotes nas posições correspondentes do vetor $A[i]$, conforme evolução da janela de congestionamento. O modelo utilizado baseia-se no trabalho desenvolvido para o estado estacionário das conexões TCP de longa duração [59]. Este modelo leva em consideração as limitações impostas pela janela do receptor, as perdas envolvendo três ACKs duplicados e perdas por expiração do temporizador. Com estes parâmetros é

possível determinar o RTT_{ca} do fluxo TCP através da seguinte expressão:

$$RTT_{ca} = \min \left(\frac{W_m}{B_w}, \frac{1}{B_w \sqrt{\frac{2bp}{3}} + 3.3(\sqrt{\frac{3bp}{8}})p(1 + 32p^2)} \right)$$

Onde $p = 0.005$ representa a probabilidade de perda, e $b = 2$ o ACK cumulativo.

A evolução da janela pode ocorrer de duas formas: (i) Fluxos longos e com grande volume de pacotes, conhecidos como elefantes ou *heavy hitters*. Estes são a minoria e representam a maior parte do volume de tráfego [62, 35]. A evolução da janela é caracterizada pelo comportamento dente-de-serra, como pode ser observado no Fluxo A da Figura 3.6. A janela evolui até a ocorrência de perda por TD, atingindo o valor máximo dado por:

$$W_{m_{ca}} = \sqrt{\frac{8}{3bp}}$$

(ii) Fluxos que apresentam limitação de taxa imposta pelo receptor. A evolução da janela inicialmente apresenta um crescimento linear, até ser limitada pelo tamanho da janela do receptor, permanecendo constante neste valor como ilustrado pelo Fluxo B da Figura 3.6.

Inicialmente, conforme pode ser visto no algoritmo da Figura 3.5, é verificado se a janela do receptor impõe limitação na taxa do transmissor ou não. Em seguida é calculado o número de ciclos (EX) de duração igual a um RTT até a ocorrência da primeira perda. Este cálculo é feito tanto para o caso de ocorrer ou não limitação da janela do receptor. Quando o número de ciclos atinge o valor de EX (ocorrência de uma perda), é verificado se a perda ocorrida foi por recebimento de três $ACKs$ duplicados (TD) ou porque o temporizador expirou (TO). A perda por TO ocorre uma a cada $W_m/3$ perdas por TD. No caso de perda por TD, a janela de transmissão é reduzida à metade e o fluxo se mantém na fase de *congestion avoidance*, com o crescimento linear da janela até ocorrer nova perda. Quando as perdas são por expiração do temporizador, a janela de transmissão é reduzida a um pacote e a partir daí inicia-se um novo ciclo de *slow start*.

Um exemplo de distribuição dos pacotes executada pelo algoritmo pode ser visto

```

Fase de Congestion Avoidance /* Distribui os pacotes nas posições de A[i] */
1.  $Wm_{ca} = \sqrt{\frac{8}{3bp}}$  /* Máxima janela do congestion avoidance até ocorrer perda */
2. if ( $W_m < Wm_{ca}$ ) /* Janela do receptor impõe limitação */
3.    $EX = \frac{b}{8}W_m + \frac{1-p}{pW_m}$  /* Ciclos RTT com a limitação no crescimento da janela */
4. else
5.    $W_m = Wm_{ca}$ ;  $EX = (b/2)W_m$ ; /* Ciclos RTT de crescimento linear da janela */
6. end
7.  $cwnd = W_m/2$  /* Valor inicial da janela do congestion avoidance */
8.  $TO = W_m/3$  /* Indicador ocorrência de perdas por TO. */
9. while (Edca)
10.  if  $cwnd < W_m$  /* Evolução da janela */
11.     $cwnd = cwnd + 1$ 
12.  end
13.   $T_j = T_j + RTT * j$  /* Calcula o tempo decorrido durante a evolução da janela */
14.   $A[T_j] += cwnd$  /* Adiciona o valor da cwnd na posição A[i] correspondente */
15.   $Edca = Edca - cwnd$ ;  $j = j + 1$ ;
16.  if ( $ciclos = EX$ ) /* Ocorre perda */
17.     $ciclos = 0$ ;  $q = q + 1$ ;
18.    if ( $q = TO$ ) /* A perda ocorrida é por TO */
19.       $cwnd = 1$ ;  $q = 0$ ;
20.      CALL slow start() /* inicia o slow start após  $2.2RTT$  */
21.    else  $cwnd = W_m/2$ ; /* A perda ocorrida é por TD */
22.    end
23.  end
24.   $ciclos = ciclos + 1$ 
25. end

```

Figura 3.5: Algoritmo para a fase de *congestion avoidance*.

na Figura 3.6. Para os dois exemplos da figura $Wm_{ca} = 17$. O Fluxo A não apresenta limitação da janela do receptor na fase de *congestion avoidance*, resultando em uma evolução da janela de congestionamento na forma de dente de serra. Neste fluxo, o tamanho médio do pacote é de 500 bytes, logo $W_m = 32$ pacotes pois $RecWin = 16K$. O Fluxo B é um exemplo onde a evolução da janela é limitada pelo receptor. O tamanho médio do pacote é de 1400 bytes, o que resulta em uma janela máxima de transmissão de 12 pacotes, conforme a linha 5 do algoritmo da Figura 3.2: $W_m = \lceil \frac{RecWin}{MSS} \rceil = \lceil \frac{16000}{1400} \rceil = 12$.

Observa-se também, para o Fluxo B, a ocorrência de perda por expiração do temporizador (TO) por volta dos 110 segundos. Neste ponto, a janela de transmissão é reduzida a um pacote e após decorridos $2.2RTT$ é iniciado um novo ciclo de *slow-start*. O algoritmo considera perdas consecutivas por TO, ou seja, dado que ocorreu uma perda por TO a perda seguinte também será por TO [59]. Desta forma, o valor de $2.2RTT$ corresponde a o tempo médio que o temporizador do TCP aguarda

devido a uma perda consecutiva por TO.

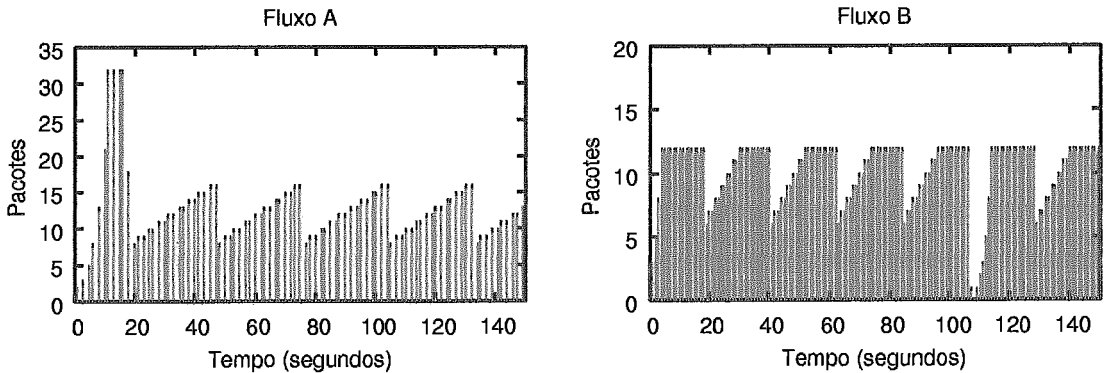


Figura 3.6: Distribuição de pacotes executada pelo algoritmo para dois fluxos diferentes.

3.2.5 Distribuição do tamanho dos pacotes

Com o objetivo de calcular a taxa em bits por segundo a partir do número de pacotes estimados pelo algoritmo da seção anterior, precisamos determinar corretamente o tamanho de cada pacote do fluxo. A taxa em bits por segundo normalmente é utilizada para determinar a capacidade dos enlaces, podendo ser expressa em Mb/s ou Gb/s.

A informação do número de bytes e pacotes do fluxo agrega tanto pacotes de sinalização (SYN, ACK, FIN) na ordem de 80 bytes, quanto pacotes de dados com tamanho até 1500 bytes. Desta forma, o valor médio $MSS = \frac{b}{a}$ é uma aproximação do tamanho real dos pacotes que são transmitidos em uma conexão. Entretanto esta aproximação não afeta a precisão do método, pois em um fluxo o número de pacotes de dados é muito maior que o número de pacotes de sinalização. Desta forma, a diferença entre o tamanho real do pacote de dados e o valor médio estimado pelo método é pequena. Para fazer esta verificação, comparamos a distribuição real do tamanho dos pacotes com a calculada pelo algoritmo. O resultado, apresentado na Figura 3.7, mostra uma boa aproximação entre as distribuições real e estimada.

A distribuição dos pacotes também é utilizada pelo algoritmo para determinar o total de Kbits registrados por intervalo de tempo. Este cálculo é feito substituindo-se as linhas 4 e 14 dos algoritmos das Figuras 3.4 e 3.5 respectivamente, pela seguinte

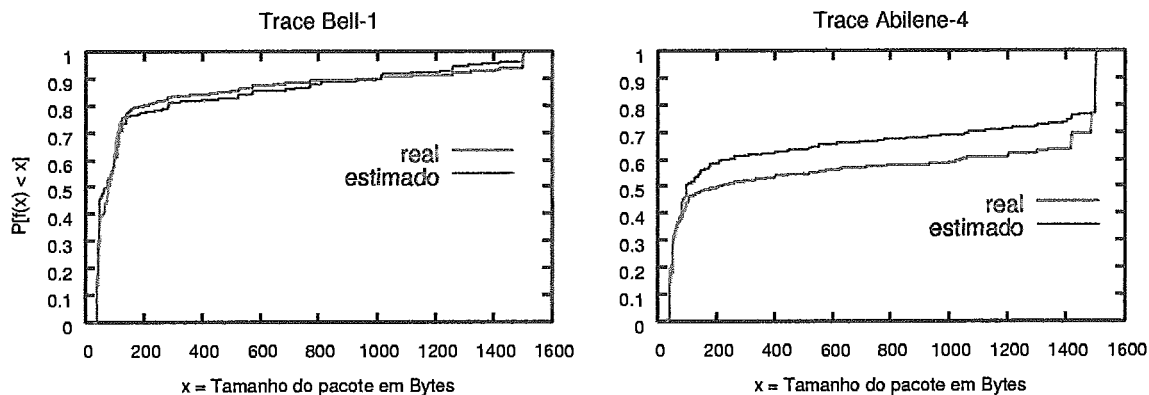


Figura 3.7: CDF do tamanho dos pacotes para a seqüência real e estimada.

relação:

$$A[T_j]_+ = \frac{cwnd \times MSS \times 8}{1000}$$

No Capítulo 5 usaremos este cálculo para determinar a taxa do tráfego em Mb/s e observar períodos de congestionamento existentes no enlace.

3.2.6 Determinação do tempo entre chegadas de pacotes

Conforme apresentado nas seções anteriores, o método proposto destina-se preferencialmente a determinar uma série temporal com o número de pacotes ou bytes por intervalo de tempo, representado pelo vetor $A[i], i = 0, 1, \dots, m$. Entretanto, através do mesmo mecanismo de distribuição de pacotes, é possível estimar o tempo entre chegadas dos pacotes e obter estatísticas relacionadas a esta variável aleatória. Alguns modelos de tráfego utilizam-se do tempo entre chegadas para sua parametrização, como o modelo MAP multifractal [37] que é apresentado em detalhes no Capítulo 4.

Para o cálculo do tempo de chegada dos pacotes utilizamos os mesmos algoritmos apresentados nas seções anteriores. Entretanto, ao invés de somar os pacotes da janela de congestionamento $cwnd$ na posição $A[i]$ do vetor, adotamos o seguinte procedimento:

- Os pacotes da janela de congestionamento $cwnd$ são divididos uniformemente

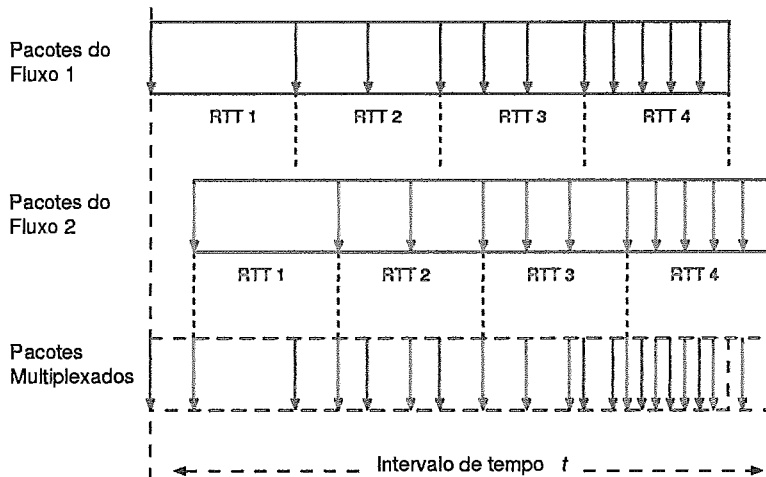


Figura 3.8: Tempo de chegada dos pacotes e multiplexação estatística

ao longo do RTT do fluxo. O intervalo uniforme entre os pacotes reproduz o comportamento do TCP ao longo do caminho e não na fonte onde os pacotes da janela são enviados um após o outro.

- Para cada pacote distribuído ao longo do RTT é atribuído um tempo de chegada relativo ao início do fluxo. Um exemplo desta distribuição é apresentado na Figura 3.8. Nesta figura são mostrados dois fluxos com o mesmo RTT. Estes fluxos possuem 11 pacotes e a janela *cwnd* evolui a cada RTT da seguinte forma: 1, 2, 3 e 5 pacotes. Pode-se notar que Fluxo 2 inicia após o Fluxo 1. Os pacotes da *cwnd* são distribuídos uniformemente no intervalo do RTT e, para cada pacote, é atribuído um tempo de chegada.
- A ordenação dos tempos de chegada dos pacotes obtidos no passo anterior faz com que ocorra uma intercalação dos pacotes conforme sua precedência no tempo, originando a multiplexação estatística dos pacotes. No final temos uma seqüência com os tempos de chegadas dos pacotes. Calcula-se então, o tempo entre chegadas.

3.3 Influência dos Fluxos UDP

Uma questão importante de ser abordada é a influência dos fluxos UDP no mecanismo de distribuição temporal de pacotes. Conforme apresentado, o método

implementa a dinâmica do TCP para todos os fluxos do *trace*, independente do protocolo. Esta implementação mostrou-se eficiente mesmo para proporções de 50% de tráfego TCP.

A Tabela 3.1 mostra a proporção dos protocolos para os diferentes *traces* analisados no período de observação. A relação de outros protocolos incluem o RIP, OSPF, BGP e o ICMP, entre outros.

<i>Trace</i>	% TCP	% UDP	% Outros	<i>Período</i>
Hpwren-1	43.7	38.8	17.5	4h
Bell-1	66.7	32.6	0.7	1h
Bell-2	92.5	6.3	1.2	1h
Auck6	95.2	3.9	0.9	3h
Abil-1	79.8	12.7	7.5	10min
NCAR-1	50.9	48.3	0.8	5min
Abil-4	75.4	22.5	2.1	90s
Abil-4a	73.1	25.3	1.6	90s
Hpwren-2	76.8	15.8	7.4	1h

Tabela 3.1: Proporção de tráfego dos *traces* da Internet usados nas análises

Uma análise mais detalhada destes *traces* mostrou que a variabilidade na taxa do tráfego é predominantemente determinada pelo comportamento dos fluxos TCP, independente de sua proporção no tráfego total, sendo que os fluxos UDP contribuem com uma parcela constante com pequena variação de taxa.

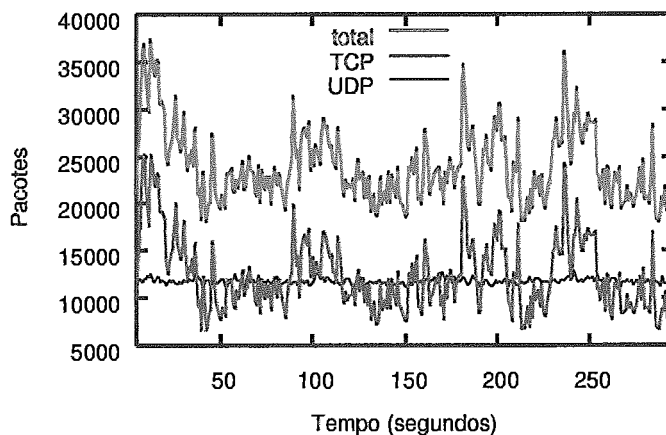


Figura 3.9: Composição TCP e UDP do tráfego total.

Para ilustrar esta constatação, a Figura 3.9 mostra a decomposição do tráfego NCAR-1, onde é possível observar separadamente as variações das taxas TCP e

UDP. Na Figura nota-se a influência da variação do tráfego TCP na composição total do tráfego e a contribuição de taxa constante do tráfego UDP para a escala de tempo de 1s. A Figura 3.10 mostra a comparação do tráfego UDP real e o estimado pelo método, onde é reproduzida a pouca variabilidade da taxa nesta escala de tempo.

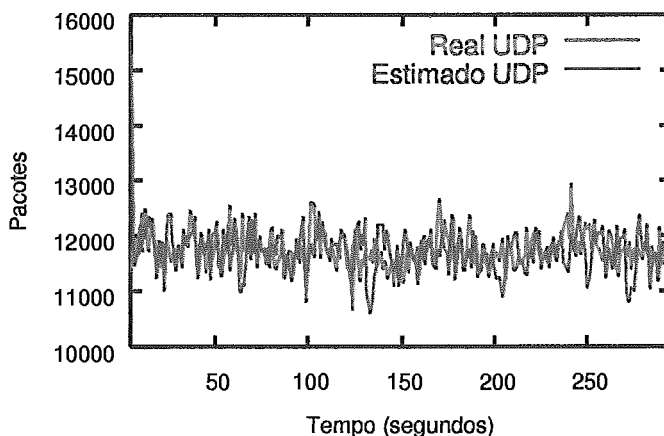


Figura 3.10: Comportamento real e estimado do tráfego UDP.

3.4 Validação, Experimentos e Resultados

Nesta seção é analisado o desempenho do método de geração de pacotes. Usamos em nossas análises *traces* de pacotes coletados da Internet com diferentes perfis de tráfego que estão publicamente disponíveis em [57]. A avaliação do método foi baseada nos seguintes critérios: (i) Comparamos estatísticas de primeira e de segunda ordem da seqüência de pacotes estimada com as estatísticas da seqüência real; (ii) obtivemos a distribuição do tamanho da fila quando esta é alimentada pela seqüência real e pela seqüência estimada de pacotes, considerando vários tamanhos de *buffer* e diferentes taxas de serviço.

As estatísticas usadas na validação foram a média, a variância, o parâmetro de Hurst e a função de autocorrelação do número de pacotes por intervalo de tempo. Considerando que a estimativa do parâmetro de Hurst é sensível ao método de coleta e tratamento das amostras, julgamos que este parâmetro é um bom indicador para a comparação das seqüências real e estimada. A função de autocorrelação foi escolhida,

pois é usada na parametrização de diversos modelos de tráfego da literatura.

Os *traces* estudados representam o tráfego medido em um sentido nos roteadores de acesso e do backbone da Internet. Cada um desses *traces* exibe características diferentes quanto ao percentual de tráfego TCP, a duração, a velocidade do enlace, e a data da coleta. A Tabela 3.2 resume as informações dos *traces*. O *trace* Hpwren foi coletado no enlace sem fio do centro de supercomputação da Universidade de San Diego. O *trace* Bell foi coletado no centro de pesquisa da Bell Labs, o Auck6 na Universidade de Auckland e o NCAR-1 no Centro Nacional de Pesquisas Atmosféricas em Boulder. Por fim, os *traces* Abil-1 e Abil-4/4a foram obtidos do backbone da rede Abilene (Internet 2), sendo que o Abil-1 foi medido na interface do roteador no sentido de Cleveland para Kansas City, e Abil-4/4a na interface do roteador no sentido de Atlanta para Indianápolis.

<i>Trace</i>	<i>Data</i>	<i>Início</i>	<i>Duração</i>	<i>Pacotes</i>	<i>% TCP</i>	<i>Link</i>
Hpwren-1	20/09/2004	21:06h	4h	4534892	43.7	45Mbps
Bell-1	23/05/2002	04:00h	1h	887685	66.7	9Mbps
Bell-2	21/05/2002	13:00h	1h	2806982	92.5	9Mbps
Auck6	06/12/2001	12:00h	3h	9164540	95.2	155Mbps
Abil-1	14/08/2002	10:30h	10min	45647327	79.8	2.5Gbps
NCAR-1	29/12/2003	22:05h	5min	7189640	50.9	1Gbps
Abil-4	17/06/2004	22:28h	90s	659851	75.4	2.5Gbps
Abil-4a	30/07/2004	10:52h	90s	1108435	73.1	2.5Gbps
Hpwren-2	11/01/2005	01:02h	1h	2961694	76.8	45Mbps

Tabela 3.2: *Traces* da Internet usados nas análises

Para cada *trace* de pacotes da Tabela 3.2 foram preparadas duas seqüências com o total de pacotes contabilizados em intervalos de tempo de 200ms e 1s, formando as seqüências reais. Os intervalos de tempo foram baseados em trabalhos da literatura que usam estatísticas de *traces* para parametrização de modelos de tráfego [3, 8, 54, 68]. Também foram criados *traces* de fluxos de acordo com a classificação utilizada pelo Netflow, ou seja, para cada *trace* de pacotes foi gerado um *trace* equivalente de fluxos obedecendo a lei de formação padrão do Netflow. Os *traces* de fluxos foram usados como entrada para o algoritmo proposto.

O algoritmo empregado na formação dos *traces* de fluxos, utiliza uma tabela *Hash* que faz a agregação dos pacotes em fluxos. Esta tabela é indexada por uma

chave de 5 elementos (IP de origem e destino, portas de origem e destino e protocolo) que representam os fluxos ativos. Para cada pacote do trace é verificado se existe uma chave *Hash* correspondente. Caso exista, o pacote pertence a um fluxo ativo e os contadores deste fluxo são atualizados. Caso contrário, um novo fluxo é criado (chave na tabela *Hash*).

Os fluxos são retirados da tabela *Hash* e exportados para um arquivo que forma o trace de fluxos mediante duas condições: Quando o pacote lido possui o tempo de chegada (*time-stamp*) superior a 15 segundos ao pacote anterior contabilizado para o fluxo, ou quando os bits de sinalização FYN ou RST estão ativos. Após a leitura de todos os pacotes do trace, os fluxos restantes na tabela são exportados para o trace de fluxos. A partir daí, utilizamos o método proposto para gerar as seqüências estimadas de pacotes para os intervalos de 200ms e 1s.

3.4.1 Resultados para o número de pacotes por intervalo de tempo

A média, a variância e a função de autocorrelação do número de pacotes foram calculadas usando a ferramenta Octave [25]. A estimativa do parâmetro de Hurst foi feita utilizando-se o método *R/S* do Octave e conferido com o programa SELFIS [44] através dos métodos *Abry-Veitch* e *Whittle*.

Trace	Média		Variância		Hurst	
	<i>real</i>	<i>estim.</i>	<i>real</i>	<i>estim.</i>	<i>real</i>	<i>estim.</i>
Hpwren-1	318.61	318.26	2.6268e+04	2.4018e+04	0.66989	0.67767
Bell-1	246.58	246.50	5.9287e+04	5.8936e+04	0.74062	0.74249
Bell-2	779.72	779.68	1.5485e+05	1.5011e+05	0.76609	0.76804
Auck6	848.57	848.28	1.7972e+04	1.6631e+04	0.75025	0.75398
Abil-1	76079	76020	7.5488e+06	8.3733e+06	0.81382	0.80104
NCAR-1	23965	23947	1.5186e+07	1.5219e+07	0.66410	0.67270
Abil-4	7331.7	7325.8	7.4647e+05	7.1284e+05	0.75214	0.72673
Abil-4a	12393	12412	1.7591e+07	2.0231e+07	0.74135	0.74060
Hpwren-2	822.69	822.12	2.6451e+06	2.6313e+06	0.78936	0.78959

Tabela 3.3: Estatísticas obtidas para as seqüências real e estimada para intervalo de tempo de 1s.

Trace	Média		Variância		Hurst	
	<i>real</i>	<i>estim.</i>	<i>real</i>	<i>estim.</i>	<i>real</i>	<i>estim.</i>
Hpwren-1	63.816	63.730	2161.7	1638.3	0.68901	0.70324
Bell-1	49.316	49.301	3168.4	2837.6	0.76845	0.77537
Bell-2	155.94	155.92	8437.6	7931.2	0.78874	0.79194
Auck6	169.71	169.66	1201.3	1333.4	0.76361	0.75842
Abil-1	15199	15207	3.7289e+05	3.5858e+05	0.84096	0.84336
NCAR-1	4793.1	4789.5	8.1080e+05	7.9169e+05	0.71828	0.72714
Abil-4	1467.9	1467.3	3.5822e+04	3.7154e+04	0.80742	0.78390
Abil-4a	2474.2	2476.2	7.3900e+05	8.4599e+05	0.80771	0.80680
Hpwren-2	164.54	164.42	1.0953e+05	1.0830e+05	0.82223	0.82274

Tabela 3.4: Estatísticas obtidas para as seqüências real e estimada para intervalo de tempo de 200ms.

A distribuição do tamanho da fila foi obtida através de simulação usando a ferramenta *Tangram-II* [21]. Foram consideradas as seqüências de pacotes geradas para intervalos de 1s. A taxa de serviço foi calculada para valores de utilização do enlace de 40% e 70%. A exceção foi o *trace* do backbone da Abilene *Abil-1*, onde foi considerada uma utilização de 95% e o intervalo de 200ms. Este *trace* possui um tráfego mais suavizado e, portanto, para valores menores de utilização e intervalos de tempo maiores, a fila permanece vazia ou com poucos pacotes durante grande parte do tempo. As Tabelas 3.3 e 3.4 sumarizam as estatísticas para intervalos de 1s e 200ms, respectivamente.

A Figura 3.11 mostra o número de pacotes transmitidos ao longo do tempo para as seqüências real e estimada para intervalos de 1s. Por questões de visibilidade e compreensão das figuras, são mostradas as duas primeiras horas das seqüências geradas a partir dos *traces* *Hpwren-1* e *Auck6*. As demais seqüências estão representadas na sua íntegra. Podemos observar, a partir da figura, que a seqüência estimada acompanha as variações na taxa de pacotes com bastante precisão. A semelhança entre as seqüências pode ser melhor observada para os *traces* de curta duração *Abil-1*, *NCAR-1*, *Abil-4* e *Abil-4a*.

Na Figura 3.12 é apresentada a autocorrelação do número de pacotes para intervalos de 1s. Podemos notar que as estatísticas de primeira e de segunda ordem obtidas para a seqüência estimada são muito próximas dos seus valores reais.

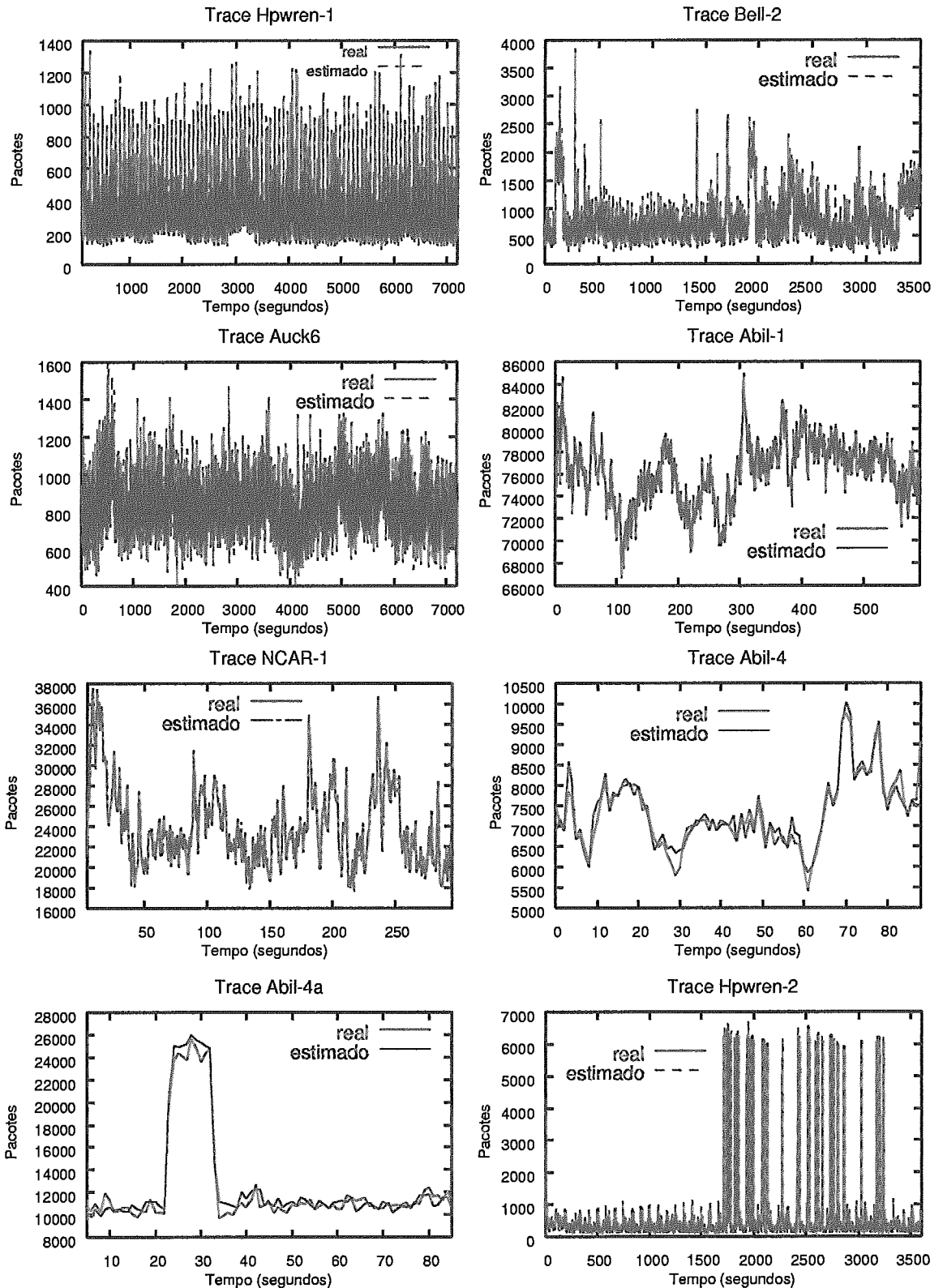


Figura 3.11: Número de pacotes por intervalo de tempo para as seqüências real e estimada para intervalo de tempo de 1s.

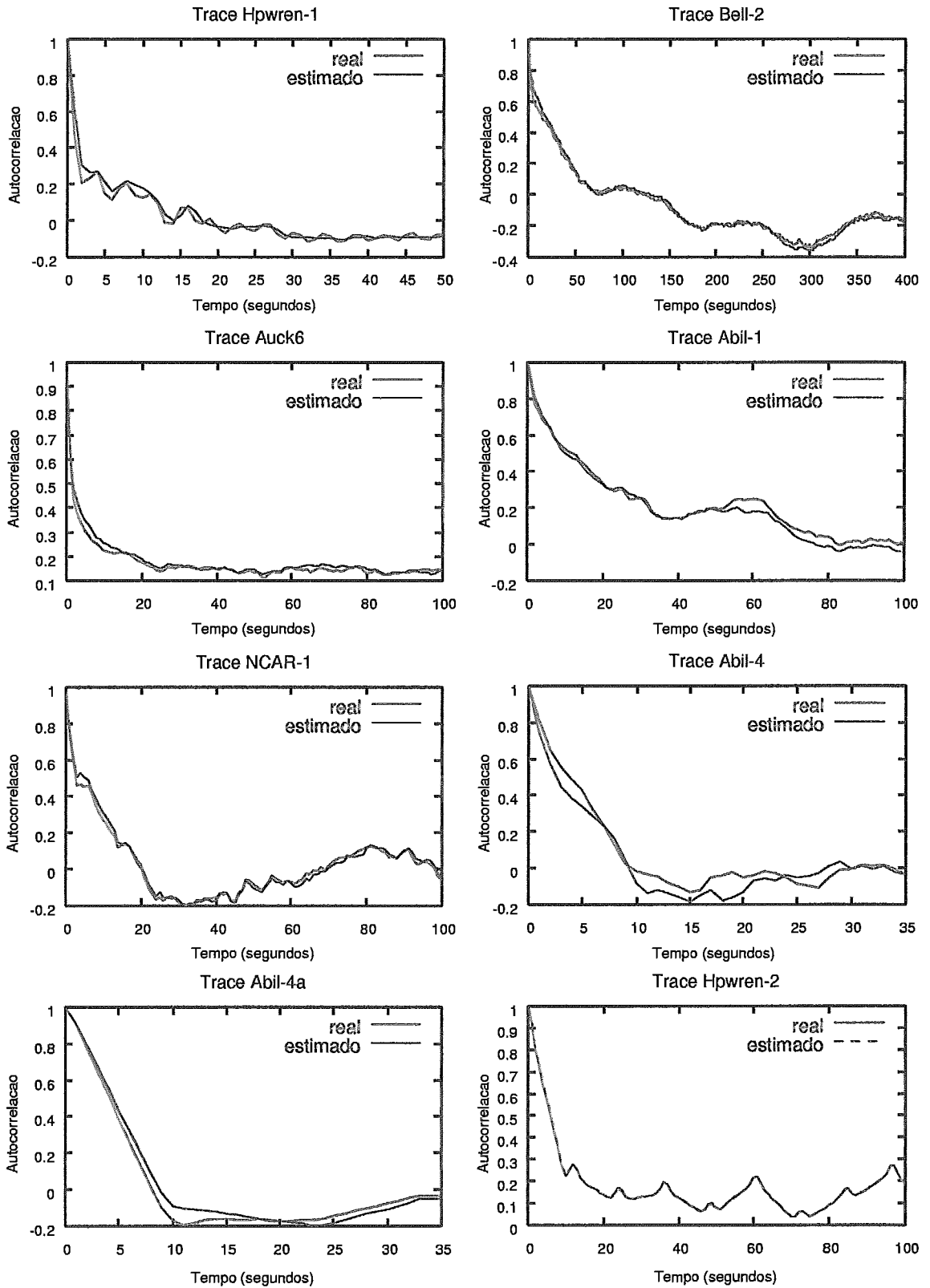


Figura 3.12: Função de autocorrelação para as seqüências real e estimada para intervalo de tempo de 1s.

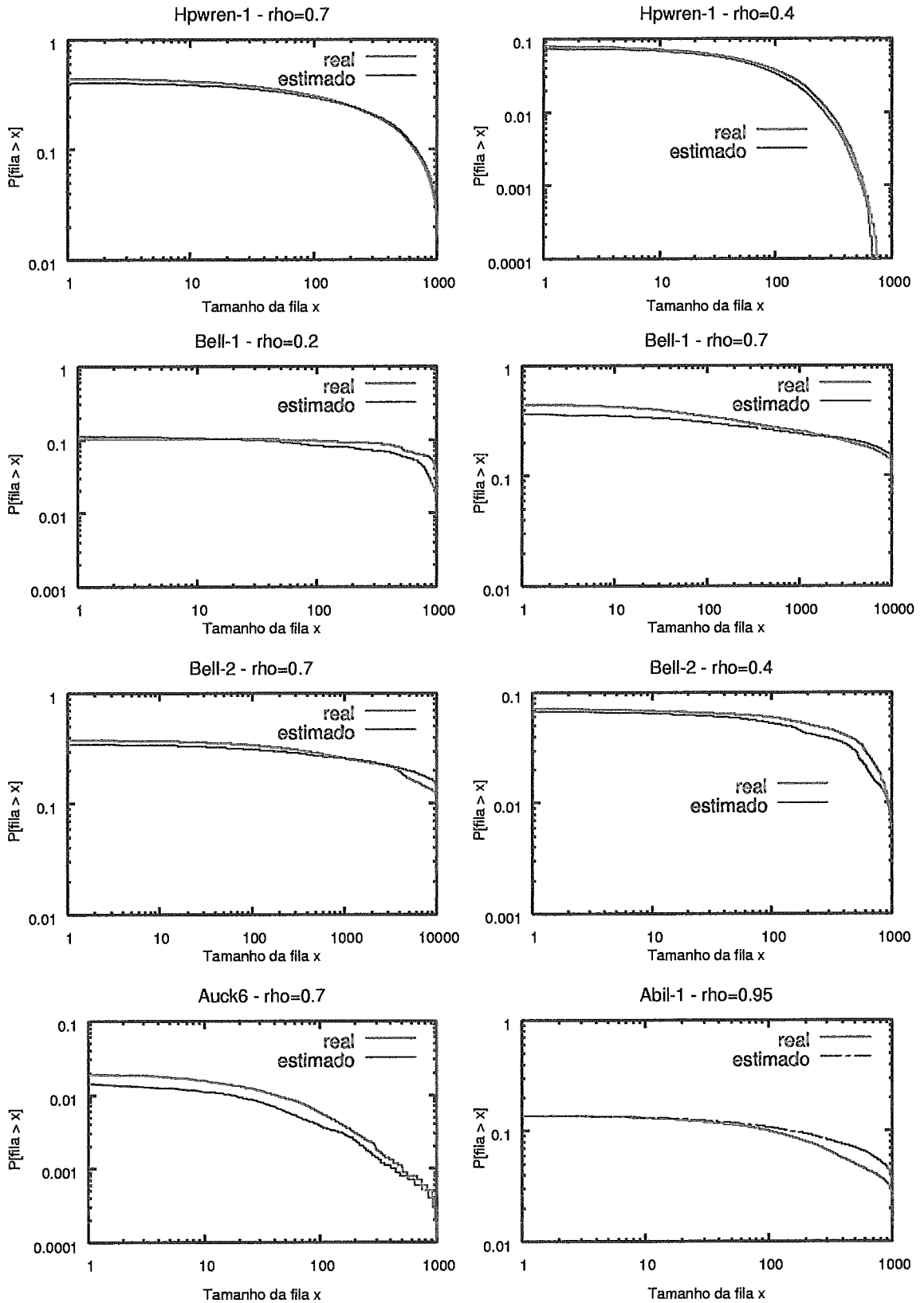


Figura 3.13: Distribuição do tamanho da fila para as seqüências real e estimada.

Os gráficos de autocorrelção mostram um decaimento lento da função, compatíveis com o parâmetro de $Hurst > 0.5$, conforme apresentados na Tabela 3.3.

As Figuras 3.13 e 3.14 apresentam o resultado da simulação de uma fila sendo alimentada pelas seqüências estimada e real. A distribuição do tamanho da fila obtida para a seqüência estimada prevê com bastante precisão a distribuição real. Para estas seqüências, a probabilidade do tamanho da fila ser maior que um certo valor é no máximo 5% maior ou menor do que a probabilidade obtida para a seqüência real. A exceção foi para as seqüências mais suavizadas *Auck6* e *Abil-1*, onde o método apresentou uma precisão menor na estimativa da distribuição do tamanho da fila. Uma possível explicação para este comportamento é que estes *traces* foram coletados em enlaces com menor utilização do que os outros. Portanto, a probabilidade de perda usada no algoritmo talvez esteja superestimada.

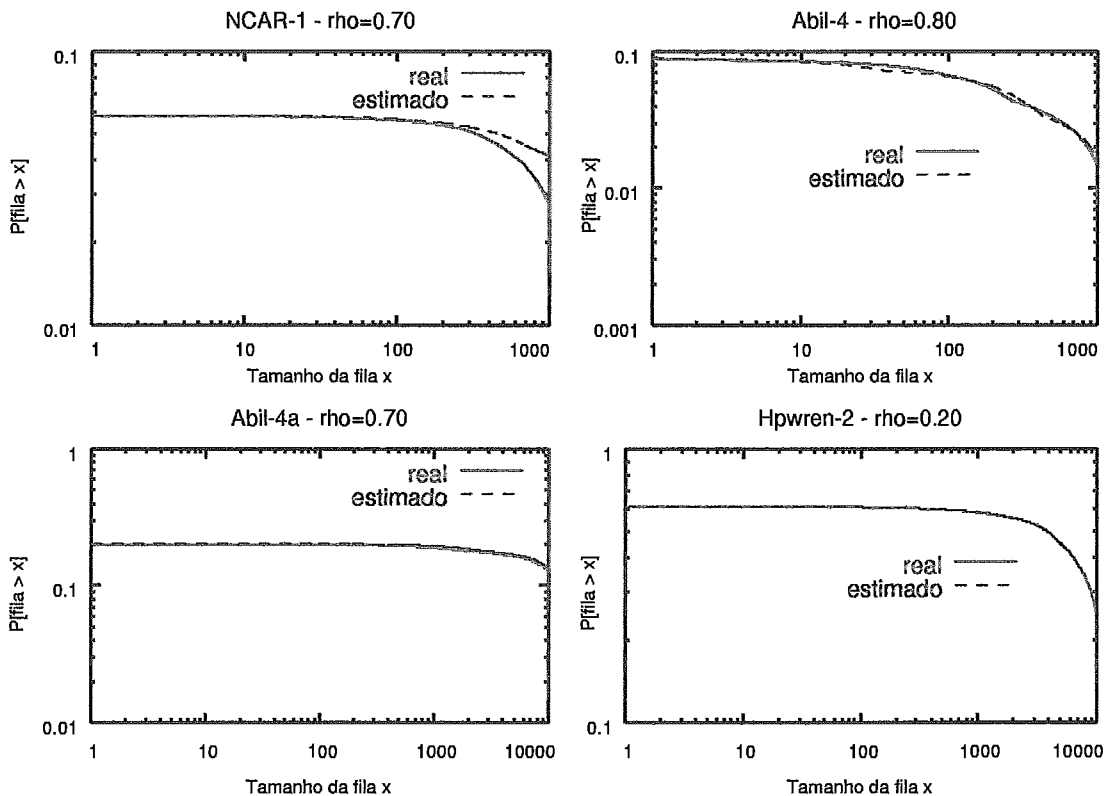


Figura 3.14: Distribuição do tamanho da fila para as seqüências real e estimada.

3.4.2 Resultados para o tempo entre chegadas de pacotes

Após a obtenção dos tempos de cada pacote, conforme procedimento em 3.2.6, calcula-se o tempo entre chegadas. Para verificar a precisão do método, comparamos estatísticas obtidas para o tempo entre chegadas do trace de pacotes real com o gerado pelo método. Os descritores calculados foram a média, variância, parâmetro de Hurst (estimado pelo método R/S do Octave), e a função de autocorrelação. Os resultados estão representados na Tabela 3.5 e na Figura 3.15

Trace	Média		Variância		Hurst	
	real	estim.	real	estim.	real	estim.
Bell-1	0.0012826	0.0016487	5.4900e-06	1.9039e-03	0.78436	0.67865
Bell-2	0.0040559	0.0049811	1.2041e-04	9.7035e-03	0.77134	0.66588
Hpwren-1	0.0031222	0.0031947	2.7158e-05	2.9567e-04	0.72593	0.66584
Abil-4	0.00013624	0.00043881	2.7142e-08	1.7317e-04	0.74538	0.71765

Tabela 3.5: Estatísticas obtidas para as seqüências real e estimada para o tempo entre chegadas dos pacotes.

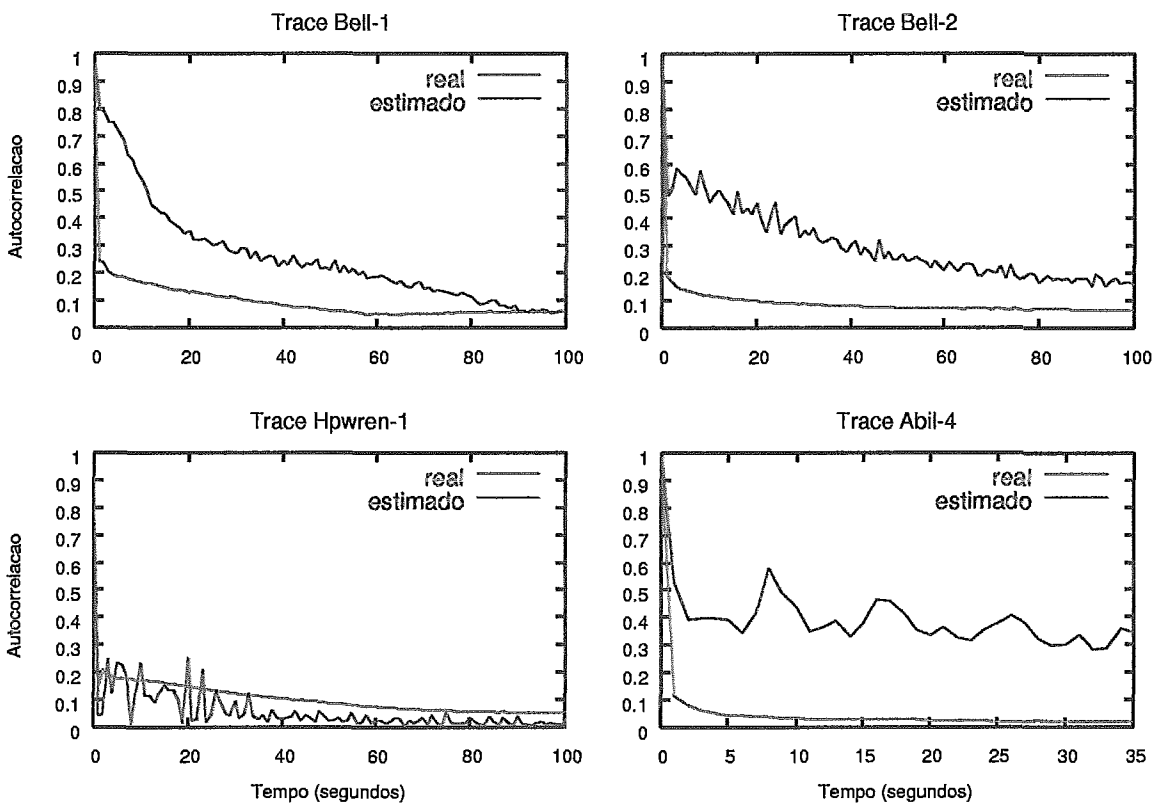


Figura 3.15: Função de autocorrelação para as seqüências real e estimada do tempo entre chegadas dos pacotes.

As estatísticas apresentadas mostram que a média dos tempos entre chegadas

determinada pelo método apresenta uma boa aproximação para os *links* de baixa velocidade *Bell-2* e *Hpwren-1*. Para *links* de alta velocidade, como é o caso do *Abil-4* de 2.5Gbps, os pacotes tendem a ficar concentrados no início do RTT [6]. Esta concentração faz com que a média dos tempos da seqüência real seja menor do que aquela calculada pelo método que leva em consideração a distribuição uniforme dos pacotes ao longo de todo o RTT. Esta distribuição uniforme dos pacotes ao longo do RTT também influencia na autocorrelação dos tempos entre chegadas, pois mantendo o mesmo espaçamento de tempo entre os pacotes a correlação entre eles aumenta. Isto pode ser observado na Figura 3.15 onde a seqüência estimada apresenta uma autocorrelação maior comparada com a seqüência real.

3.5 Análise do Erro Médio Produzido pelo Método

Na seção 3.4.1 mostramos que a seqüência de pacotes gerada pelo método reproduz com bastante precisão as estatísticas de primeira e de segunda ordem da seqüência real, assim como a distribuição da ocupação de uma fila para uma determinada escala de tempo. Nesta seção vamos medir o erro da seqüência estimada analisando cada intervalo de tempo conforme o procedimento utilizado em [59]. O número de pacotes em um intervalo de tempo da seqüência estimada (denotado por $N_{estimado}$), será comparado com o número de pacotes da seqüência real (N_{real}) para o mesmo intervalo de tempo. Para cada intervalo de tempo t_i , $i = 1, 2, \dots, n$ determinaremos o erro relativo em relação ao número real de pacotes. O erro médio será determinado pela seguinte expressão:

$$Erro = \frac{\sum_{i=1}^n \frac{|N_{estimado} - N_{real}|}{N_{real}}}{n}$$

Onde n é o total de intervalos da seqüência.

Calculamos o erro médio para as seqüências de 1s e 200ms dos *traces* da Tabela 3.2. Os resultados são apresentados na Figura 3.16 que mostra o erro em função da taxa média dos *traces*. Podemos observar que o erro das seqüências de 200ms é sempre maior que o erro das seqüências de 1s e que ambos diminuem com o aumento da

taxa média do *trace*.

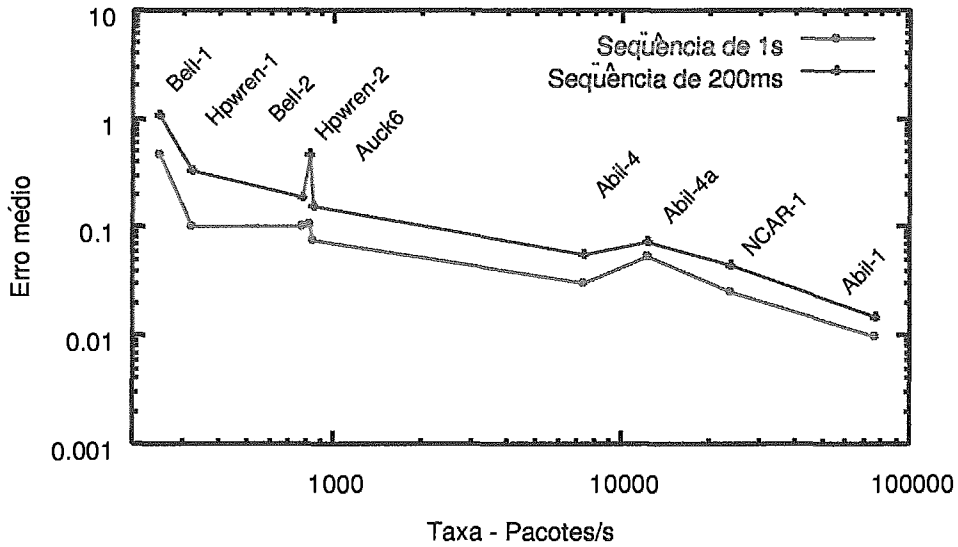


Figura 3.16: Erro médio para as seqüências estimadas de 1s e 200ms.

A diminuição do erro para taxas maiores é um forte indicador de que o método é escalável com o aumento da velocidade dos *links*. Desta forma, podem ser geradas seqüências com granularidades cada vez mais finas na medida em que trabalhamos com taxas mais altas. Podemos notar na Figura 3.16 que os pontos *Hpwwen-2* e *Abil-4a* não acompanham o decaimento da curva. Analisando a Figura 3.11 notamos que estes dois *traces* possuem picos de curta duração que elevam a taxa média muito acima do restante da seqüência. Ou seja, embora a taxa média do *trace* seja maior, o erro médio é influenciado pelo maior número de amostras da seqüência com taxas menores.

Um outro comportamento observado para o erro é com relação ao número de fluxos usados nas estimativas. Observamos que quanto maior o número de fluxos utilizados para formar a seqüência estimada, mais ela se aproxima da seqüência real. Realizamos o seguinte experimento para demonstrar esse comportamento: (i) Utilizando o *trace Abil-4*, separamos os fluxos que se estendem durante todos os 90 segundos. Foram selecionados 20 fluxos que correspondem a 7% do total de pacotes. (ii) Geramos uma seqüência de pacotes em intervalos de 1s para cada fluxo individualmente utilizando o método de geração de pacotes. Também coletamos a seqüência real produzida por cada fluxo no mesmo intervalo de tempo. (iii) Calculamos o erro

da seqüência gerada por cada fluxo individualmente. O erro médio para as seqüências de fluxo individual, na maioria dos casos, ficou muito acima do erro calculado para as seqüências que continham fluxos simultâneos. (iv) Agrupamos gradativamente os fluxos e calculamos o erro. O aumento do número de fluxos simultâneos faz com que o erro médio da seqüência diminua.

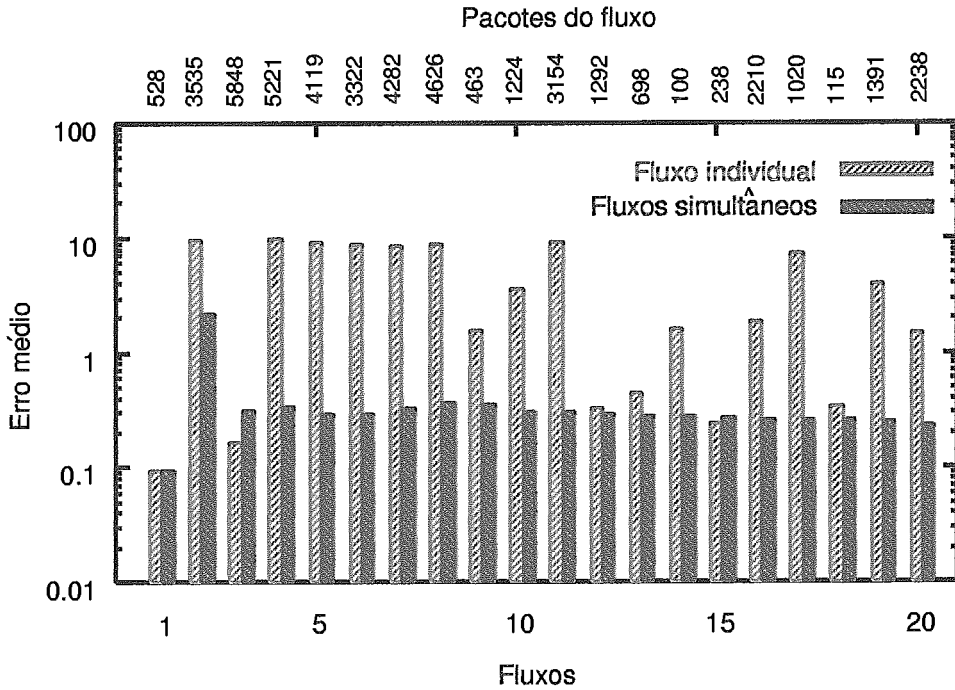


Figura 3.17: Erro médio das seqüências para fluxos individuais e simultâneos.

A Figura 3.17 mostra os 20 fluxos analisados. Para cada um é calculado o erro médio da seqüência. Também é mostrado o erro médio quando estes fluxos são agrupados. Para o fluxo número 5, por exemplo, temos que o erro da seqüência deste fluxo é de 9,2. Quando agrupamos ao fluxo 5 os quatro anteriores o erro cai para 0,30. Para os 20 fluxos simultâneos o erro da seqüência gerada diminui para 0,23.

3.6 Aplicação em Modelos de Tráfego

Nesta seção utilizamos o método proposto para gerar uma seqüência de pacotes e obter as estatísticas necessárias para a parametrização de dois modelos da literatura.

Após a parametrização dos modelos, comparamos o comportamento dos mesmos quando estes são usados para alimentar uma fila, com o comportamento da seqüência real. Utilizamos a seqüência de pacotes estimada para o *trace Bell-2* neste estudo.

A medida de desempenho usada para esta comparação é o número médio de pacotes na fila obtido para diferentes valores de taxa de serviço. Optamos por calcular esta medida, pois ela é usada em [22] para o dimensionamento do backbone de uma rede e será aplicada no estudo de caso apresentado no Capítulo 5.

Foram considerados dois modelos com diferentes requisitos de parametrização. O primeiro é o modelo MAP de 32 estados proposto em [37]. Um dos parâmetros deste modelo são os coeficientes wavelet de Haar. Para o cálculo deste parâmetro é necessário o tempo entre chegadas dos pacotes que foram estimados através do método apresentado na seção 3.2.6. O segundo é um modelo de cadeia de Markov oculta (*Hidden Markov Model - HMM*) proposto em [22]. A parametrização do modelo HMM necessita do número de pacotes coletados em intervalos pequenos de tempo para treinamento da cadeia de Markov. No Capítulo 4 veremos estes dois modelos com maiores detalhes.

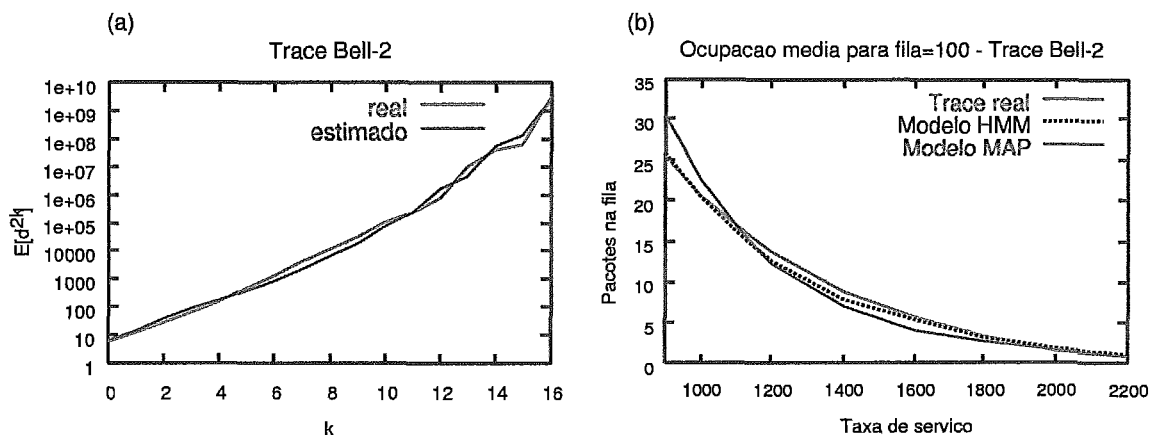


Figura 3.18: a) Segundo momento dos coeficientes wavelet de Haar para diferentes níveis de agregação k . b) Tamanho médio da fila para a seqüência real de pacotes e os modelos HMM e MAP.

A Figura 3.18a mostra o segundo momento dos coeficientes wavelet de Haar calculados para a seqüência real e a estimada. As curvas apresentadas na Figura 3.18a mostram que é possível reproduzir o comportamento deste descritor com boa precisão, usando a seqüência estimada pelo método.

A Figura 3.18b mostra a ocupação média de uma fila para diferentes valores de taxa de serviço. Os resultados para a seqüência real foram obtidos através de simulação. Já os resultados dos modelos foram calculados analiticamente, ambos usando a ferramenta Tangram-II. É importante salientar que pequenas diferenças na ocupação média da fila são provenientes das características dos próprios modelos em aproximar melhor ou não as características do tráfego real. Podemos observar que o modelo HMM parametrizado, usando a seqüência de pacotes gerada pelo método proposto, prediz com melhor precisão o comportamento real da fila.

3.7 Resumo e Conclusões

Neste capítulo é proposto um método simples e eficiente para a decomposição temporal dos fluxos em pacotes. O método utiliza os fluxos que possuem influência direta do comportamento dos usuários e distribui os pacotes baseado na dinâmica do TCP. Como resultado, temos uma distribuição de pacotes que reproduz as características dos níveis hierárquicos de rede, abrangendo as várias escalas de tempo.

O método foi validado usando-se *traces* da Internet. Os resultados mostraram que a seqüência de pacotes estimada pelo método se aproxima muito da seqüência real. Calculamos descritores de primeira e de segunda ordem e obtivemos a distribuição da ocupação de uma fila para diferentes níveis de utilização, para as seqüências real e estimada. As estatísticas calculadas para a seqüência gerada pelo método com o número de pacotes por intervalo de tempo, aproximaram com bastante precisão as estatísticas da seqüência real.

Embora o método adote o comportamento do TCP para todos os fluxos, constatamos que o resultado final da seqüência estimada, para as escalas de tempo analisadas, não é afetado por fluxos de outros protocolos como o UDP. O tráfego total é fortemente influenciado pela dinâmica do TCP, até mesmo para proporções de tráfego onde o TCP possui 50% do total. Também foi verificado o desempenho do método na determinação dos tempos de chegadas dos pacotes, mostrando uma boa aproximação da média para os *traces* derivados de *links* com velocidades até 45Mbps.

Já para os *links* de grande velocidade (2.5Gbps), os resultados não foram tão precisos indicando que o algoritmo pode ser otimizado. Outras melhorias, como reproduzir o comportamento de outros protocolos, estimar as perdas e variar o tamanho do buffer do receptor, também podem influenciar na precisão do método.

Analisamos o erro médio produzido pelo método na geração de seqüências com o número de pacotes por intervalo de tempo. O erro diminui com o aumento da taxa do tráfego. Isto possibilita trabalharmos com granularidades mais finas para taxas mais elevadas. Demonstramos que o aumento do número de fluxos simultâneos torna a seqüência gerada mais próxima da real.

A vantagem deste método em comparação com a coleta de pacotes é que podemos relacionar o mecanismo de geração de pacotes com as aplicações em uso na rede. Ou seja, é possível isolar os fluxos de uma transferência FTP, por exemplo, e analisar o comportamento dos pacotes na utilização dos recursos da rede. Podemos identificar quais fluxos são responsáveis por causar congestionamentos e quais são mais agressivos na ocupação da banda disponível. A monitoração de pacotes, por outro lado, não fornece a informação em nível de aplicação, além da complexidade na sua execução.

No capítulo seguinte apresentamos modelos de tráfego da literatura que foram parametrizados com o método proposto, possibilitando a utilização dos mesmos para o dimensionamento e o planejamento de capacidade.

Capítulo 4

Modelos de Tráfego

ESTE capítulo analisa o uso de modelos de tráfego para o dimensionamento de capacidade. São estudados quatro tipos de modelos Markovianos parametrizados por diferentes descritores de tráfego, com o objetivo de analisar as vantagens, desvantagens e seu desempenho em reproduzir corretamente a ocupação em uma fila para diferentes capacidades de armazenamento e taxas de serviço.

4.1 Introdução

Os modelos de tráfego possuem um papel fundamental no estudo do desempenho e dimensionamento das redes de comunicação. É através desses modelos que podemos prever os recursos de rede de forma a atingir a qualidade de serviço (QoS) desejada.

A utilização de modelos que reproduzam as características do tráfego em redes de alta velocidade tem sido alvo de constantes pesquisas, não existindo atualmente um consenso de modelo ideal que descreva o processo de geração de pacotes ou a multiplexação estatística do tráfego agregado.

Os modelos variam na sua capacidade de capturar melhor ou pior as propriedades relevantes na descrição do processo, ou seja, conforme o modelo, algumas características podem ser bem representadas enquanto que outras não. Desta forma, devemos

utilizar modelos que atendam as nossas necessidades conforme sua capacidade de reproduzir adequadamente os descritores desejados.

Nas aplicações envolvendo o dimensionamento da rede é necessário que tenhamos modelos que, além de reproduzirem com fidelidade as estatísticas de primeira e segunda ordem e a função de autocorrelação do tráfego real, possam alimentar uma fila e obter a mesma distribuição de ocupação obtida com a seqüência real. Isto deve ser válido para diferentes taxas de serviço e tamanhos de *buffer*. No Capítulo 5 iremos mostrar porque a distribuição da ocupação da fila é uma medida importante para o dimensionamento da rede.

Neste capítulo são apresentados quatro modelos Markovianos. Para analisar o desempenho dos modelos, foi realizada a simulação da seqüência real em uma fila com taxa de serviço determinística. Esta seqüência é formada pelo número de pacotes que chegam em intervalos de 1s. O tamanho médio da fila, gerado pela simulação, serviu como referência de comparação para as medidas geradas analiticamente pelos modelos. Esta análise foi feita utilizando-se a ferramenta TANGRAM-II [21].

4.2 Modelo de Sobrecarga e Folga

Em [20] é apresentado um modelo para caracterização do tráfego capaz de capturar com eficiência o comportamento das perdas em uma fila finita. São propostos dois novos descritores que representam quanto tempo o tráfego permanece fluindo acima de uma determinada taxa, chamado de *nível de rajada*. Estes descritores são: a distribuição da duração das rajadas e a duração média das rajadas. No modelagem apresentada, a taxa de serviço da fila é incorporada ao modelo e definida como *nível de sobrecarga*. Desta forma, as durações de sobrecarga (taxas acima do nível de sobrecarga) e folga (taxas abaixo do nível de sobrecarga) são usadas para descrever o tráfego aplicado nesta fila.

As distribuições das durações de sobrecarga e folga são então parametrizadas e modeladas usando-se distribuições *hiper-exponenciais* de três estados, com o objetivo de capturar o lento decaimento de $P[D > t]$, onde D é a variável aleatória

representando a duração que se deseja aproximar. O modelo resultante é Markoviano com recompensas de taxa, com três estados de sobrecarga e três de folga. Em [20] também é proposto um algoritmo para aproximar as distribuições de duração de sobrecarga e folga na hiper-exponencial de três estados. Cada estado i é aproximado por uma exponencial predominante para cada região, sendo as duas primeiras representando a parte inicial e a terceira a cauda da distribuição. A estratégia consiste em encontrar pontos b_1 e b_2 que delimitam as três regiões. De posse destes pontos, basta investigar o histograma das durações para encontrar as taxas das exponenciais.

Vale ressaltar que este modelo foi originalmente concebido para determinar as taxas de perda em uma fila para tráfego multimídia, especialmente para seqüências de vídeo, através de descritores que capturam este comportamento. Nesta seção vamos analisar o seu desempenho em determinar a ocupação média em uma fila para o tráfego de redes WAN (tráfego agregado).

4.2.1 Cadeia de Markov do modelo

Inicialmente utilizamos o modelo com três estados de sobrecarga e três de folga, com a parametrização ajustada para um nível de 90% de carga. Com esta configuração, o modelo não teve bom desempenho na determinação da ocupação da fila, principalmente para taxas de serviço onde a utilização da fila era inferior a 80%. Para estas taxas o modelo apresentou ocupação da fila igual zero, não acompanhando a seqüência real. Ajustando o nível de carga para 80% observamos uma melhoria na resposta do modelo. Mesmo assim, a ocupação da fila foi zero quando a utilização estava abaixo dos 60%.

Este comportamento pode ser explicado pelo fato do modelo ser parametrizado considerando-se um nível de taxa de serviço específico. Quando a taxa usada na parametrização é diferente da taxa de serviço da fila, o modelo não reproduz o comportamento correto. Ou seja, para taxas de serviço da fila maiores do que a taxa considerada na parametrização do modelo, a ocupação da fila tenderá a ser menor que a real, pois a utilização do canal está abaixo da utilização inicialmente

prevista.

O ideal é que a parametrização seja feita para vários níveis de taxa de serviço, dando origem a um modelo para cada nível de carga considerado. Entretanto, este procedimento não pode ser aplicado para o perfil de tráfego que estamos interessado, principalmente nos que tendem a ser mais suavizados, como é o caso do tráfego agregado das redes LAN e WAN. Nestes o nível de sobrecarga raramente fica abaixo de 80%, pois o número de amostras acima deste nível não é suficiente para determinar as distribuições das durações de sobrecarga e folga, para a granularidade considerada neste trabalho.

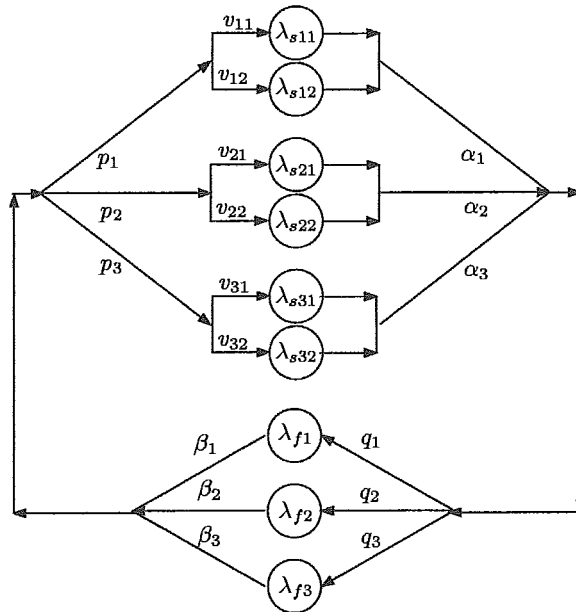


Figura 4.1: Modelo com 6 estados de sobrecarga e 3 de folga.

Para contornar esta limitação do modelo imposta pelo perfil do tráfego agregado, foi adotada uma variação aumentando-se o número de estados de sobrecarga visando aumentar a variabilidade do tráfego. Do modelo original, com três estados, cada estado de sobrecarga foi convertido em outros dois. Cada um destes dois estados recebeu uma recompensa de taxa diferente. Estas recompensas, ao serem ponderadas pelas probabilidades de entrada nos seus respectivos estados, equivalem a recompensa original. Os valores das recompensas foram obtidos analisando-se os histogramas de taxas condicionados a cada intervalo de tempo. Os histogramas para cada região (intervalo de tempo) foram divididos em duas sub-regiões, onde

a sub-região contendo as taxas maiores representa aproximadamente 25% do total das taxas para a região. Foram feitos testes também com a sub-região contendo as taxas maiores, representando aproximadamente 10% do total das taxas para a região. Vamos referenciar estas divisões em sub-regiões como sendo 25/75 e 10/90, respectivamente. Também foram executados testes aumentando-se o número de estados de folga para 6, conforme procedimento descrito anteriormente, mas não foram verificadas diferenças significativas para o modelo de 3 estados de folga. Desta forma, o modelo final ficou com a configuração mostrada na Figura 4.1, com seis estados de sobrecarga e três de folga, onde λ_{si} são as taxas de sobrecarga, λ_{fi} as taxas de folga, p_i a probabilidade de ocorrência do estado i de sobrecarga, q_i a probabilidade de ocorrência do estado i de folga, α_i as durações de sobrecarga e β_i as durações de folga. Os parâmetros v_{ij} representam a probabilidade para as sub-regiões do modelo de 6 estados de sobrecarga.

4.2.2 Parametrização

A parametrização do modelo com 6 estados de sobrecarga e 3 de folga, consiste nos seguintes passos:

1. A partir do *trace* com o número de pacotes por intervalo de tempo, obtém-se duas seqüências com base no nível de sobrecarga escolhido: uma seqüência com as durações de sobrecarga e outra com as durações de folga.
2. Obtém-se a distribuição complementar ($P[X > x]$) das durações das seqüências de sobrecarga e folga.
3. Obtém-se o histograma das durações das seqüências de sobrecarga e folga.
4. Com base na distribuição complementar e no histograma de cada seqüência, são determinados os pontos b_1 e b_2 que delimitam as regiões e a probabilidades p_i , v_i e q_i .
5. Calcula-se a taxas médias λ_{si} e λ_{fi} das exponenciais de cada região.
6. Calcula-se a durações média α_i e β_i

A parametrização do modelo com os nove estados envolveu um total de 27 parâmetros. Embora este número de variáveis seja elevado e torne o modelo trabalhoso de configurar, a grande vantagem está na sua eficiência computacional. Por se tratar de um modelo de fluido, o *buffer* não é explicitamente modelado, o que faz a cardinalidade do espaço de estados ser reduzida para 9.

4.2.3 Ocupação da fila

Para a obtenção da ocupação da fila em estado estacionário do modelo, utilizamos um método para cálculo da distribuição da recompensa acumulada em estado transiente do TANGRAM-II. Consideramos um tempo t de observação extremamente longo em relação aos demais tempos de ocorrência de eventos, ou seja, $t = 1000s$. Para este tempo foi observado que os valores da distribuição já estavam estabilizados. Após isto, a ocupação média da fila foi determinada pela seguinte expressão [46]:

$$E[x] = \int_0^{\infty} [1 - F_X(x)] dx, \quad X \geq 0$$

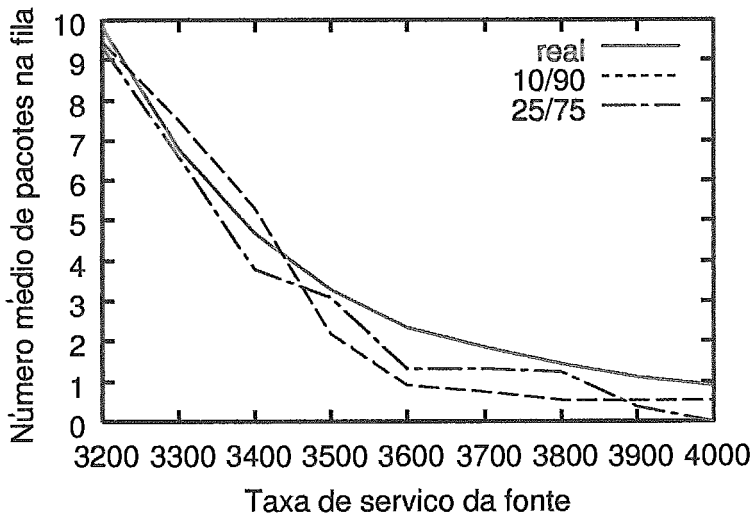


Figura 4.2: Desempenho do modelo de sobrecarga e folga na ocupação da fila.

A configuração que apresentou melhores resultados em todas as análises foi a configuração com 80% de carga e 25/75 de divisão do histograma. Contudo, o modelo não teve bom desempenho para baixos níveis de utilização, onde obtivemos valores de ocupação média da fila igual a zero. Uma análise do desempenho de ocupação da

fila pode ser verificada na Figura 4.2, onde são comparados as variações do modelo de seis estados com 80% de carga e divisão 10/90, e 80% de carga e divisão 25/75.

4.3 Modelo MAP Multifractal

A complexidade na caracterização do tráfego nas redes de pacotes está na invariância de suas propriedades ao longo do tempo, ou seja, da ausência de uma escala de tempo específica que defina as flutuações do processo estocástico em questão. Ao invés disto, é necessário descrever o comportamento ao longo de várias escalas de tempo, inicialmente em pequenas escalas, na ordem de centésimos de milisegundos, onde predomina os efeitos dos protocolos de transporte principalmente do TCP. Estudos atuais sugerem que o tráfego nestas escalas pode ser caracterizado por um tipo especial de fractal, denominado multifractal. Já nas escalas maiores de tempo, acima de 1s, predomina o comportamento monofractal ou auto-similar do tráfego [76].

Com o objetivo de capturar de forma mais abrangente o comportamento similar por várias escalas de tempo, a análise multifractal foi levada em consideração na elaboração de novos modelos que caracterizassem melhor as propriedades do tráfego. Riedi *et al* [67] propuseram um modelo baseado em wavelets para reproduzir o comportamento multifractal obtendo bons resultados em vários testes estatísticos, mas com alto custo computacional. Em [53] é apresentado um estudo comparativo para a modelagem do tráfego multifractal utilizando processos envelope. Estes processos podem representar com precisão o tráfego multifractal a um custo computacional reduzido.

A proposta de [37] adota um modelo Markoviano para aproximar o efeito multifractal. Para isto, foi utilizado um Processo de chegada Markoviano (*MAP - Markovian Arrival Processes*) com uma estrutura baseada no mecanismo de geração da transformada wavelet de Haar. Wavelets são usadas como a principal ferramenta matemática de análise multifractal. Nesta seção estudaremos o desempenho deste modelo em reproduzir o comportamento do tráfego real em uma fila.

4.3.1 Cadeia de Markov do modelo

Para exibir o comportamento multifractal foi utilizado uma cadeia de tempo contínuo empregando um Processo de Poisson modulados por Markov (*Markov Modulated Poisson Process - MMPP*), com estrutura simétrica de cubo de n dimensões, onde cada dimensão representa uma escala de tempo. As taxas geradas pela cadeia são determinadas de acordo com a variação do processo de chegada para as diferentes escalas de tempo. Esta estrutura assemelha-se com a geração da transformada wavelet de Haar. Basicamente esta transformada captura as variações da seqüência para os diferentes níveis de agregação ou escalas de tempo. Foi observado que a série formada com o segundo momento dos coeficientes de Haar pode capturar o comportamento multifractal desta seqüência [37].

Desta forma, o método utilizado para parametrizar o modelo procura aproximar o segundo momento dos coeficientes de Haar, com as variações de taxa produzidas pela cadeia MMPP nas diferentes escalas de tempo.

As diferentes escalas de tempo obedecem o seguinte critério: 1) A taxa média do modelo é igual a 1. 2) A próxima escala de tempo é $1/\lambda$ que é definida por uma cadeia MMPP de dois estados, com taxas de chegadas de $(1 - a_1)$ e $(1 + a_1)$ ($-1 \leq a_1 \leq 1$), de modo que a taxa média de chegada fique inalterada (igual a 1). 3) Para cada nova escala de tempo $1/\gamma\lambda$ onde $\gamma > 1$, é introduzida uma nova dimensão na cadeia, de forma que o comportamento das escalas anteriores não seja alterado. Para esta nova dimensão um novo parâmetro a_i é adicionado na composição das taxas. Estes parâmetros determinam a variância da taxa na respectiva escala de tempo.

As Figuras 4.3 e 4.4 mostram os modelos de 3 e 4 níveis respectivamente. Cada nível do modelo reproduz a geração de pacotes para uma certa escala de tempo. É importante notar a característica simétrica da cadeia e o aumento da dimensão quando passamos de 3 para 4 níveis. Ou seja, a cadeia de 4 níveis é formada por duas cadeias de 3 níveis, uma cadeia de 5 níveis é formada por duas cadeias de 4 níveis, e assim sucessivamente. Na Figura 4.3 são mostradas as taxas para cada

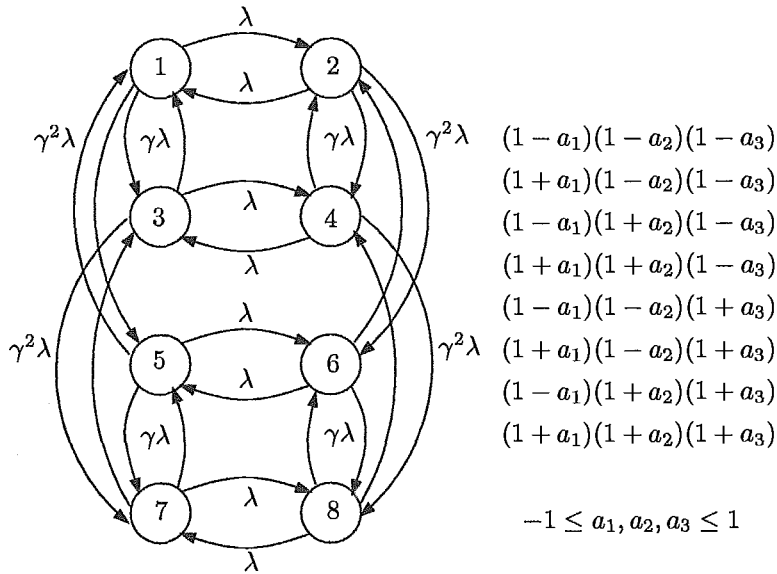


Figura 4.3: Cadeia do modelo Multifractal de 3 níveis e 8 estados com suas respectivas taxas.

estado da cadeia.

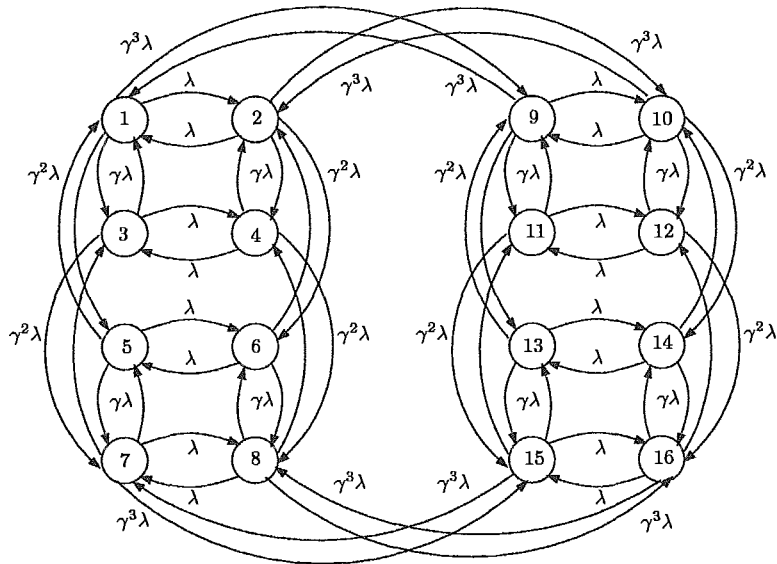


Figura 4.4: Cadeia do modelo Multifractal de 4 níveis e 16 estados.

Uma cadeia de n níveis é composta de 2^n estados e tem $n + 2$ parâmetros. Os parâmetros γ e λ definem as escalas de tempo que serão consideradas na parametrização, e os valores a_1, a_2, \dots, a_n determinam a variância do processo de chegada. Nos testes realizados utilizamos $n = 4$ e $n = 5$, resultando em uma cadeia de 16 e 32 estados respectivamente. Estes valores foram baseados no compromisso entre precisão, complexidade da cadeia e custo computacional.

As escalas de tempo são escolhidas por inspeção da seqüência. Através do próprio modelo temos que os valores da maior escala T_M , e a menor escala T_m são dados por:

$$T_M = 1/\lambda; T_m = \frac{1}{\gamma^n \lambda}$$

Os valores de T_M e T_m são calculados relativos ao tempo médio entre chegadas de pacotes. Por exemplo: Para um tempo médio entre chegadas de 5ms, um $T_m = 16$ significa o tempo médio de chegada de 16 pacotes, ou seja, 80ms. Tanto T_M como T_m são expressos em número de chegadas de pacotes. Para o cálculo de T_M adotamos um procedimento baseado no tempo em que a função de autocorrelação atinge zero.

4.3.2 Parametrização

Como este modelo captura as características do tráfego em escalas pequenas de tempo, ele necessita de medições mais detalhadas do tráfego obtidas através dos tempos entre chegadas (*interarrival times*) dos pacotes. Para isto, o *trace* deve conter os *time-stamps* e não apenas o número de pacotes por intervalo de tempo.

Em nossa análise utilizamos o seguinte procedimento para a obtenção dos parâmetros do modelo:

1. Usando o *trace*, gerado pelo algoritmo proposto, foi produzida uma seqüência com os tempos entre chegadas dos pacotes. Também geramos uma seqüência com a contagem dos pacotes em intervalos de 1s.
2. Determinamos a média do tempo entre chegadas dos pacotes $E[iat]$.
3. Medimos o tempo T_0 em que a função de autocorrelação da seqüência de amostras coletadas em intervalos de 1s atinge o valor zero (ou próximo disto).
4. Calculamos a maior escala de tempo em função do tempo médio entre chegadas $T_M = E[iat]T_0$. É importante que T_M seja aproximado para uma potência de base 2 (2^x) para coerência com os níveis de agregação (escalas de tempo) calculados pela transformada wavelet de Haar.

5. A menor escala de tempo T_m foi calculada em função dos valores de T_M e γ . Fixamos o valor de γ em 8 (sugestão do autor), e o número de níveis n usados foi de 4 ou 5, dependendo da cadeia escolhida.
6. A próxima etapa envolveu a determinação dos coeficientes wavelet de Haar. Para isto foi utilizada a seqüência *iat* e os coeficientes calculados para uma escala de tempo T_M . Por exemplo, $T_M = 2^{15}$ significa que os coeficientes serão calculados para 15 escalas de tempo.
7. Uma vez definidos os valores de T_M , T_m , λ , γ e n , os parâmetros a_1, a_2, \dots, a_n são calculados através de um algoritmo de otimização (*downhill simplex method*), de forma a minimizar a soma dos erros relativos ao segundo momento dos coeficientes wavelet de Haar.

O algoritmo implementado pelo autor calcula o segundo momento dos coeficientes wavelet de Haar gerado pela cadeia MMPP e os compara com os coeficientes da seqüência real. Isto é feito iterativamente atribuindo-se diferentes valores para a_1, a_2, \dots, a_n . Estes valores são otimizados pelo *downhill simplex method* de forma a minimizar a distância do erro para os coeficientes Haar da seqüência real. Ao final da otimização são retornados os parâmetros que determinam a estrutura da cadeia resultante. O custo computacional desta otimização cresce exponencialmente com o aumento da dimensão da cadeia MMPP. Para as cadeias de 16 e 32 estados analisadas ($n = 3$ e $n = 4$) o custo foi aceitável, na ordem de minutos.

4.3.3 Ocupação da fila

A cadeia resultante produz uma taxa média de pacotes de intensidade igual a 1. Desta forma, a taxa de serviço da fila deve ser ajustada de acordo. Para uma utilização de 80% a taxa de serviço deve ser de 1.25, por exemplo. Para análise do desempenho em uma fila utilizamos para o modelo cadeias MMPP de 16 e 32 estados, parametrizadas conforme procedimento anteriormente descrito.

Utilizando a ferramenta TANGRAM-II, foi gerado o modelo matemático e em-

pregado o método exato de solução analítica em estado estacionário *GTH no block*. Desta forma, a ocupação da fila foi calculada para diferentes taxas de serviço e e valores de *buffer*.

A análise do desempenho do modelo em uma fila foi feita através de medidas comparativas com o *trace* real simulado em uma fila com taxa de serviço determinística. O modelo de fonte com 32 estados apresentou melhor desempenho, como era de se esperar, comparado com o de 16 estados. O desempenho comparativo dos modelos de 16 e 32 estados é mostrado na Figura 4.5.

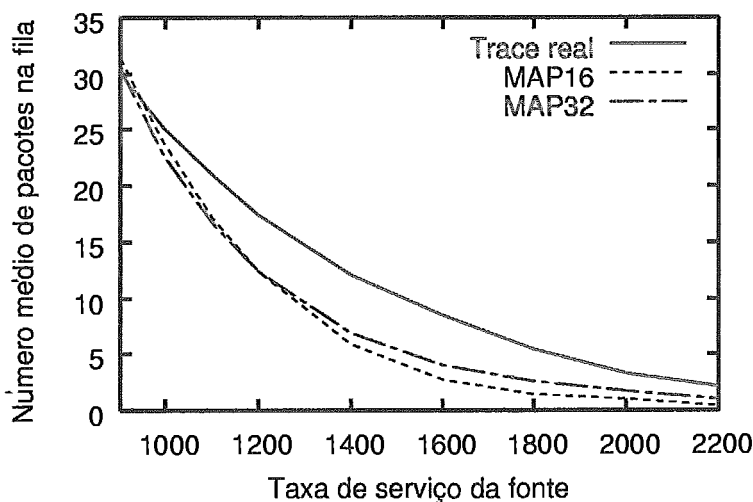


Figura 4.5: Desempenho do modelo MAP de 16 e 32 estados na ocupação da fila.

Um dado importante é que o custo computacional da resolução analítica deste modelo é proporcional ao número de estados da fonte e tamanho do *buffer*, pois neste modelo temos o *buffer* explicitamente representado. Por exemplo, um modelo de fonte com 32 estados alimentando uma fila de 2000 pacotes, produzirá um espaço de, aproximadamente, 64000 estados. Mesmo assim, o modelo se demonstrou computacionalmente eficiente em virtude de gerar uma taxa média igual a 1.

4.4 Modelo MMPP Hierárquico

Vários modelos MMPP hierárquicos tem sido propostos na literatura [3, 37, 68, 69]. Em [54] é apresentado um modelo de Poisson modulado por Markov (*Markov*

Modulated Poisson Process - MMPP) que mimetiza, até certo ponto, a interação entre usuários, protocolos e rede, fornecendo uma estrutura Markoviana que atua em três diferentes escalas de tempo: sessões, fluxos e pacotes. As sessões correspondem às solicitações dos usuários que são direcionadas a uma interface de rede por um determinado período de tempo, ou seja, todas as páginas acessadas por um usuário em um servidor *Web* ou uma conexão FTP para transferência de arquivos. Cada sessão, por sua vez, é formada por um ou mais fluxos que podem ser definidos por conexões TCP com sua respectiva sinalização de início e fim (SYN/FIN). (No Capítulo 2 apresentamos uma definição mais abrangente de fluxo. Todavia, associar cada conexão TCP como sendo um fluxo, embora mais restritiva, serve perfeitamente para o entendimento do modelo que vamos analisar.) E cada fluxo é formado por um conjunto de pacotes.

Desta forma, do ponto de vista hierárquico, temos que um usuário, ao iniciar uma ou mais sessões, irá gerar um conjunto de fluxos que, por sua vez, produzirá uma seqüência de pacotes. Cada camada possui sua escala de tempo característica. As sessões tem ocorrência mais lenta do que os fluxos e os fluxos ocorrem em menor frequência do que os pacotes.

4.4.1 Cadeia de Markov do modelo

Para a definição da cadeia de Markov do modelo utilizando este conceito hierárquico, os autores em [54] valeram-se das seguintes hipóteses: (i) As sessões são geradas conforme processo de Poisson com taxa λ_s . Cada sessão é iniciada com a chegada de um novo fluxo e terminada com a chegada do último fluxo daquela sessão. O número de fluxos gerados pela sessão é uma variável aleatória geométrica de média N_f . (ii) Os fluxos de uma mesma sessão são gerados conforme processo de Poisson com taxa λ_f . Todo o fluxo inicia com a geração de um pacote e termina após ter gerado N_p pacotes, que corresponde a média de uma distribuição geométrica. (iii) Os pacotes são gerados dentro de cada fluxo seguindo um processo de Poisson com média λ_p .

O modelo necessita dos seguintes parâmetros que são obtidos através de medidas do tráfego e simulação:

- λ_s : taxa de chegada de novas sessões;
- λ_f : taxa de chegada de fluxos por sessão ativa;
- λ_p : taxa de chegada de pacotes por fluxo ativo;
- N_f : número médio de fluxos por sessão;
- N_p : número médio de pacotes por fluxo.

A Figura 4.6 mostra o diagrama de estados do modelo. A cadeia de Markov em tempo contínuo (CTMC), que modula as chegadas dos fluxos e pacotes, é definida pela variável de estado $\bar{s} = (n_f, n_s)$, onde n_f e n_s representam o número de fluxos e sessões ativas. A taxa média de geração de pacotes é de $n_f \lambda_p$, enquanto que a taxa de geração de fluxos é de $n_s \lambda_f + \lambda_s$. Podemos definir ainda que $\beta = 1 - 1/N_f$ é a probabilidade de um fluxo não ser o último fluxo da sessão, e $\mu_f = \lambda_p/N_p$, a duração média dos fluxos.

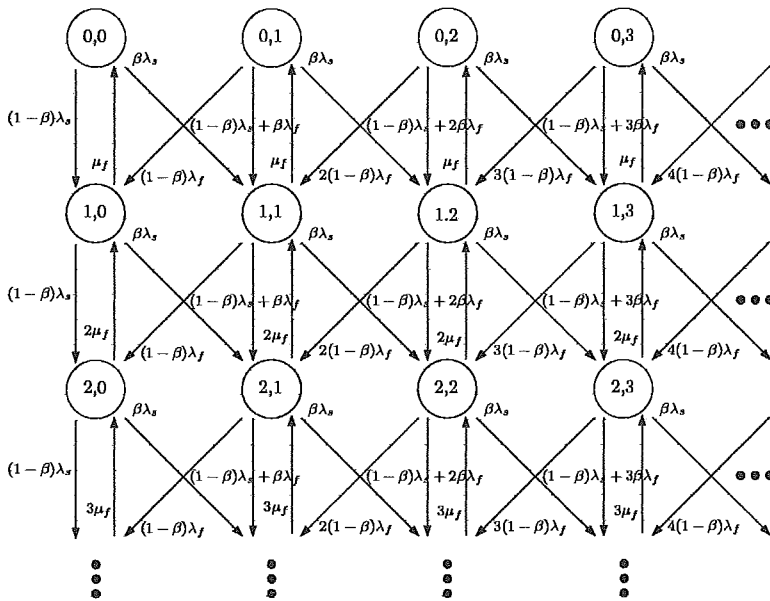


Figura 4.6: Cadeia do modelo MMPP Hierárquico.

As transições entre os estados da cadeia podem ser descritas da seguinte forma:

- Transições do estado (n_f, n_s) para o estado $(n_f + 1, n_s + 1)$ corresponde à chegada de um fluxo que é o primeiro de uma sessão. A taxa é de λ_s ;
- Transições do estado (n_f, n_s) para o estado $(n_f + 1, n_s)$ corresponde à chegada

de um fluxo que não é o primeiro nem o último da sessão. A taxa é de $(1 - \beta)\lambda_s + n_s\beta\lambda_f$;

- Transições do estado (n_f, n_s) para o estado $(n_f + 1, n_s - 1)$ corresponde à chegada de um fluxo que é o último da sessão. A taxa é de $n_s(1 - \beta)\lambda_f$;
- Transições do estado (n_f, n_s) para o estado $(n_f - 1, n_s)$ corresponde ao término de um fluxo que não é o último da sessão. A taxa é de $n_f\mu_f$.

4.4.2 Parametrização

A parametrização deste modelo é complexa, pois envolve a obtenção de dados em nível de pacotes, fluxos e sessões. Alguns parâmetros podem ser diretamente mapeados através das estatísticas de tráfego, outros precisam ser estimados, de forma indireta, com base em outros parâmetros e com uso de simulação. A seguir descreveremos o procedimento que foi empregado neste trabalho.

Inicialmente, utilizamos a coleta de fluxos realizada pelo *Netflow* [16] e disponível nos roteadores da rede em análise. Para gerar as estatísticas dos fluxos coletados utilizamos a ferramenta *Flow-tools* [32] onde são obtidas diversas medidas de tráfego, entre elas os parâmetros N_p , μ_f e a taxa global de geração de fluxos $\hat{\lambda}_f$. Também obtivemos o *trace* de fluxos contendo o tempo de início, duração, número de bytes e pacotes de cada fluxo, obtido através de modificações no programa *Flow-tools*. O arquivo de *trace* gerado foi, então, ordenado pelos tempos de início dos fluxos. A Figura 4.7 mostra o formato utilizado no arquivo *trace* de fluxos, onde o tempo de início segue o formato UNIX com resolução de um milissegundo (1ms). A duração dos fluxos também é expressa em milissegundos.

Este *trace* serviu como entrada para o método de Geração de Pacotes usado para obter o *trace* com a contagem do número de pacotes por intervalo de tempo. Com os dados obtidos com o *Netflow* e o *trace* de pacotes gerado, utilizamos o método de parametrização proposto em [54]:

1. Determinar N_p , $\lambda_p = \frac{N_p}{\mu_f}$ e $\hat{\lambda}_f$, através das estatísticas do *Netflow*;

2018979264;20632;17560;19
2018979272;5216;19496;18
2018979280;5928;10531;26
2018979284;384;416;6
2018979288;92;916;7
2018979324;400;2567;15
2018979332;16036;41662;35
.....

Figura 4.7: *trace* de fluxos obtidos com o *Netflow*.

2. Estimar o parâmetro de Hurst \hat{H}_f da contagem de fluxos que chegam em intervalos de 100ms e 1s e o parâmetro de Hurst do tempo entre chegadas dos fluxos.
3. Estimar o parâmetro de Hurst \hat{H}_p da contagem de pacotes que chegam em intervalos de 100ms e 1s e o parâmetro de Hurst do tempo entre chegadas dos pacotes.
4. Calcular $C = \lambda_s / \lambda_f$. Considerar inicialmente $C = 1$.
5. Calcular $\lambda_s = \frac{\hat{\lambda}_f}{N_f}$. Considerar inicialmente $N_f = 1$.
6. Simular o modelo durante o tempo de duração do *trace* de fluxos registrando as chegadas dos fluxos e pacotes. Neste passo é gerado um *trace* sintético com os tempos de chegadas dos fluxos e pacotes.
7. Estimar o parâmetro de Hurst H_f e H_p para os *traces* de fluxos e pacotes gerados pelo modelo.
8. Comparar os parâmetros Hurst dos *traces* reais com aqueles gerados pelo modelo (*traces* sintéticos). Caso estejam próximos o modelo está corretamente parametrizado. Caso negativo, atribua novos valores para N_f e C e retorne ao passo 4.

O algoritmo utilizado para fazer o ajuste do modelo com os parâmetros medidos está representado na Figura 4.8.

1. Determinar N_p , λ_p e $\hat{\lambda}_f$, através das estatísticas do *Netflow*;
2. Gerar o *trace* de pacotes para intervalos de 100ms e 1s usando o programa gerador de pacotes;
3. Estimar \hat{H}_f e \hat{H}_p para os intervalos de tempo de 100ms e 1s e para o tempo entre chegadas;
4. Fazer $N_f = 1$ e $C = 1$;
5. Calcular $\lambda_s = \frac{\hat{\lambda}_f}{N_f}$ e $\lambda_f = \frac{\lambda_s}{C}$
6. Simular o modelo, usando N_p , λ_p , N_f , λ_s e λ_f , e obter os *traces* dos tempos de chegadas dos fluxos e pacotes;
7. Estimar H_f e H_p para os *traces* gerados pelo modelo;
8. **while** ($|\hat{H}_f - H_f| > 0.05$) **and** ($|\hat{H}_p - H_p| > 0.05$)
9. **if** ($\hat{H}_f > H_f$)
10. $N_f = N_f + 5$;
11. **else if** ($\hat{H}_f < H_f$)
12. $N_f = N_f - 1$;
13. **else if** ($\hat{H}_p > H_p$)
14. $C = 2 * C$;
15. **else if** ($\hat{H}_p < H_p$)
16. $C = C/2$;
17. **end**
18. **end**

Figura 4.8: Algoritmo de ajuste do modelo HMM

Inicialmente, quando $N_f = 1$ e $C = 1$, o modelo gera fluxos na mesma taxa λ_s , fazendo com que a chegada de fluxos por intervalo de tempo apresente decaimento exponencial da função de autocorrelação com parâmetro $H_f = \frac{1}{2}$ e $H_p > \frac{1}{2}$, devido a memória do processo de geração de pacotes que é introduzida em função da hierarquia do modelo. Na medida que aumentamos N_f , há um crescimento nos valores de H_f e H_p , que também é verificado para o aumento de C , mas em menor proporção.

O autor argumenta que os parâmetros do modelo podem ser encontrados em menos de dez interações do algoritmo apresentado na Figura 4.8, mas nos testes realizados este processo de sintonia se demonstrou muito demorado, devido a necessidade de constantes simulações. Ou seja, para cada alteração nos parâmetros é necessário realizar nova simulação e analisar os resultados alcançados.

4.4.3 Ocupação da fila

O mecanismo hierárquico do modelo, embora intuitivo, faz com que a dinâmica de geração de sessões e fluxos seja explicitada no modelo, ocasionando um grande número de estados na medida em que as taxas envolvidas aumentam. Podemos classificar este crescimento demasiado como **explosão de estados**. O aumento excessivo na cardinalidade do espaço de estados traz como consequência o alto custo computacional, tornando inviável a solução matemática do modelo pelos métodos tradicionais.

Nos testes realizados a explosão de estados não permitiu que calculássemos a distribuição da ocupação da fila de forma analítica. Vale lembrar aqui que estamos interessados apenas na solução exata da ocupação da fila obtida analiticamente. Não estamos considerando o uso de simulação, pois, desta forma, bastaria utilizar a própria seqüência real, sem a necessidade do modelo de tráfego.

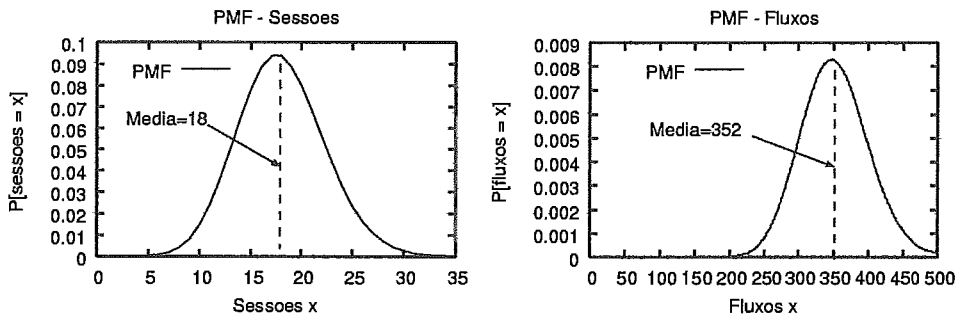


Figura 4.9: Distribuição do número de sessões e fluxos.

A Figura 4.9 mostra a distribuição do espaço de estados de sessões e fluxos do modelo para um tráfego com as seguintes características:

$$C = 2; N_f = 10; N_p = 34; \hat{\lambda}_f = 75.75; \lambda_s = 7.58; \lambda_f = 3.79; \lambda_p = 7.11;$$

A média de sessões é de 18 e a de fluxos é de 352. Com estes valores o espaço de estados médio é de aproximadamente $18 \times 352 = 6336$, podendo chegar a um máximo de $35 \times 500 = 17500$ estados. Observe que limitamos o modelo a estes valores máximos. A explosão de estados torna-se evidente quando adicionamos o tamanho do *buffer* ao modelo. Para um *buffer* de 100 o modelo gera aproximadamente 1800000 estados. Este fator, aliado ao trabalhoso processo de parametrização, faz

com que este modelo não seja adequado aos nossos propósitos.

4.5 Modelo de Markov Oculto

Uma cadeia de Markov oculta (*Hidden Markov Model - HMM*) representa um processo estocástico onde os estados não são diretamente observados, mas estão probabilisticamente associados a um outro conjunto de eventos (estados) que produzem a seqüência de observação.

Como exemplo, vamos considerar uma pessoa que identifica as condições do tempo como seco, úmido ou chuvoso, baseando-se apenas na intensidade de dor que sente nas articulações. Pessoas idosas atestam que a dor em suas articulações é probabilisticamente relacionada com o estado do tempo. Em dias úmidos ou chuvosos a sensação de dor é maior do que em dias secos. Neste caso temos dois tipos de estados: o estado observado (o estado das articulações) e o estado oculto (estado do tempo). Podemos, então, imaginar um algoritmo para que a pessoa identifique as condições do tempo sem realmente ver o que ocorre.

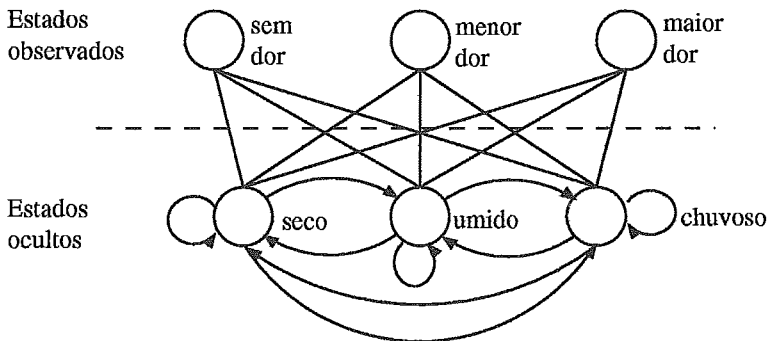


Figura 4.10: Estados observados e ocultos em uma HMM.

O diagrama da Figura 4.10 mostra os estados observáveis e ocultos deste exemplo. As conexões entre os estados observáveis e ocultos representam as probabilidades de geração de um estado observado, dado que o processo de Markov está em um estado oculto. Em [65] é apresentada uma completa introdução da teoria das cadeias de Markov ocultas.

Os elementos que compõe uma cadeia de Markov oculta são formalmente definidos

por 5 elementos [65]:

1. N: número de estados do modelo

Apesar de os estados serem ocultos, para muitas aplicações existe um significado físico relacionado com os estados do modelo ou a um conjunto deles. Os estados individuais são denotados por $S = \{S_1, S_2, \dots, S_N\}$, onde N é o número de estados e q_t é o estado no instante t .

2. M: número de símbolos distintos observados

Também conhecido como tamanho do alfabeto. Os símbolos representam as saídas físicas observadas do sistema que está sendo modelado. No caso da previsão das condições do tempo, os símbolos são a intensidade de dor nas articulações. Os símbolos são denotados por $V = \{v_1, v_2, \dots, v_M\}$.

3. A: matriz de probabilidade de transição

Indica a probabilidade do sistema sair de um estado para outro em um passo. A probabilidade de transição do estado i para o estado j em um passo é:

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N.$$

4. B: matriz de probabilidade de emissão

A probabilidade de emissão do símbolo k , dado que o sistema está no estado j é $b_j(k) = P[v_k \text{ no tempo } t | q_t = S_j]$.

5. π : distribuição do estado inicial

É a probabilidade do sistema estar no estado q_i no instante $t = 1$ (estado inicial). $\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N$.

Uma vez que um sistema possa ser modelado por uma HMM, três problemas devem ser resolvidos para que o modelo seja utilizado em aplicações reais. Os dois primeiros dizem respeito ao reconhecimento de padrões, e o terceiro e mais complexo trata do treinamento da cadeia. Os três problemas são: (i) avaliação da probabilidade da seqüência dado um modelo específico, (ii) a decodificação para determinar a mais provável seqüência dos estados ocultos q_1, q_2, \dots, q_N do modelo, dado a seqüência de observações. Ou seja, achar os estados ocultos que geraram

a saída observada. (iii) O treinamento, que consiste em otimizar os parâmetros do modelo HMM para melhor descrever a seqüência de observações. Isto é, dado a topologia da HMM e o conjunto de observações, deve-se ajustar os parâmetros do modelo para maximizar $P[\text{observações}|\text{modelo}]$. A seqüência de observações usada para ajustar o modelo é chamada de **seqüência de treinamento**, pois é usada para *treinar* a HMM. Este treinamento é feito de forma iterativa utilizando o algoritmo EM (*Expectation-Maximization*).

4.5.1 Cadeia de Markov do modelo

Modelos utilizando Cadeias de Markov Ocultas são largamente empregados em dispositivos de reconhecimento de voz ou em problemas relacionados a identificação de padrões diversos. Nosso interesse, entretanto, é usar o modelo de Markov oculto (HMM) para representar o tráfego agregado que passa em um canal de comunicação. Para esta tarefa, o primeiro passo é fazer a escolha da estrutura do modelo que melhor descreva a seqüência observada. A estrutura da cadeia pode ter influência no resultado obtido após o treinamento. Desta forma, em [5] foram analisadas três diferentes cadeias com diferentes números de estados. A cadeia escolhida possui estrutura tipo *Branch-Erlangian* e foi utilizada em [24] para a previsão de perdas de pacotes de áudio com bons resultados e é mostrada na Figura 4.11.

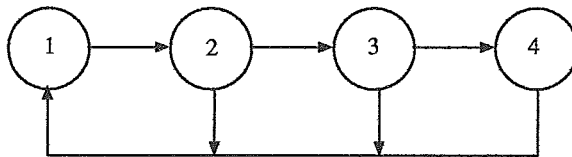


Figura 4.11: Estrutura utilizada na cadeia HMM.

Para cada estado da HMM está associada uma probabilidade de emissão de um símbolo. Neste caso, os símbolos são a quantidade de pacotes observados no enlace por intervalo de tempo. Foi utilizado o conceito de níveis, diminuindo o número total de símbolos que cada estado pode emitir. Ou seja, a seqüência real foi discretizada em níveis para reduzir o número de símbolos gerados em cada estado. O mapeamento das amostras originais e das amostras discretizadas pelos níveis é

obtido pela seguinte fórmula:

$$n_i = \left\lfloor \frac{O_i * qtdNiveis}{max} \right\rfloor$$

onde n_i é a amostra discretizada em número de níveis, O_i é a amostra real coletada no enlace, $qtdNiveis$ é a quantidade total de níveis e max é a maior amostra coletada. Em [5] foram feitos experimentos com o número máximo de níveis (max) mostrando que $max = 30$ níveis era a melhor relação custo benefício. Como cada estado HMM pode gerar até $qtdNiveis$ estados no modelo de Markov não oculto, o aumento do número de níveis representa um aumento na precisão do modelo, mas ocasiona também um aumento no custo computacional.

Com base na estrutura escolhida e nos parâmetros iniciais da cadeia é realizada a etapa de treinamento, com o objetivo de adaptar os parâmetros da HMM para a seqüência real. Como resultado, obtivemos o modelo de Markov não oculto (HMM'), que foi utilizado com a ferramenta TANGRAM-II, permitindo, assim, várias medidas de interesse, entre elas a distribuição de ocupação da fila.

4.5.2 Parametrização

O procedimento completo desde a parametrização até a geração da cadeia para o modelo HMM compreende as seguintes etapas:

1. Determinar a estrutura da cadeia que será usada para representar a seqüência de observações. No caso foi escolhida a estrutura da Figura 4.11;
2. Gerar a seqüência de observações discretizada em níveis n_i . A seqüência em níveis é obtida dividindo-se o número de pacotes por intervalo de tempo da seqüência original pelo seu maior valor dividido por 30.
3. Inicializar a cadeia HMM fornecendo as informações de número de estados (N), número de símbolos (M), distribuição do estado inicial (π), matriz de transição inicial (A) e a matriz de emissão inicial (B), que é uma matriz $M \times N$.

4. Treinar a cadeia com base nos parâmetros iniciais e na seqüência discretizada.
5. Gerar o modelo de Markov não oculto apartir da HMM para uso com o TANGRAM-II.

Em [24] foi desenvolvido um programa para geração do modelo de Markov não oculto apartir da HMM. Este programa recebe como entrada os parâmetros iniciais (N , M , π , A e B) e a seqüência de observações discretizada em níveis n_i . Após isto, o programa inicia a fase de treinamento da cadeia e produz como saída o modelo de Markov não oculto. Para o treinamento da cadeia utilizamos 1000 iterações do algoritmo de maximização.

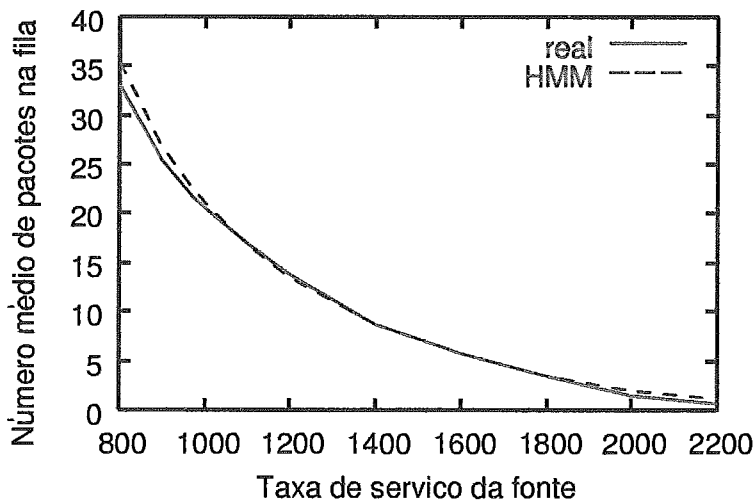


Figura 4.12: Desempenho do modelo HMM na ocupação da fila.

4.5.3 Ocupação da fila

A cadeia de Markov não oculta gerada possui 4 estados, cada um com no máximo 31 níveis de emissão, produzindo um total de 124 estados. Na prática este número de estados máximo fica abaixo deste valor, pois nem todos os estados possuem o número máximo de níveis. Para o *trace* analisado na Figura 4.12 o modelo final ficou com 70 estados e 7070 quando alimentando uma fila de tamanho 100. Esta análise foi feita utilizando-se a ferramenta TANGRAM-II que também serviu para calcular a ocupação da fila para diferentes taxas de serviço. A Figura 4.12 ilustra o desempenho

do modelo, demonstrando uma ótima aproximação com a curva produzida pela seqüência real. Para tamanhos de fila maiores (10000) ocorre o efeito da explosão de estados com grande aumento no custo computacional. Todavia, para o nosso método de dimensionamento, não usaremos filas superiores a 100.

4.6 Resumo e Conclusões

Este capítulo procurou mostrar a necessidade de um modelo de tráfego que reproduza o padrão estatístico de geração de pacotes. Além disto, devemos levar em consideração o espaço de estados do modelo e a complexidade de sua parametrização.

Com base nos testes realizados, verificamos que os modelos que melhor reproduzem o comportamento da ocupação em uma fila são os modelos MAP32 e o HMM. A determinação da ocupação na fila é fundamental para aplicarmos o método de dimensionamento que baseia-se no atraso médio dos pacotes na fila em função da capacidade do enlace. O modelo de sobrecarga e folga, embora apresente bons resultados na determinação de perdas, não teve bom desempenho na ocupação da fila, assim como o modelo MAP16. Já o modelo MMPP não pode ser resolvido analiticamente, apenas via simulação, o que o torna muito limitado para o dimensionamento. Desta forma, seria mais razoável fazermos a simulação com o próprio *trace* e obter a ocupação da fila, sendo desnecessário utilizar um modelo.

Os modelos markovianos podem ser resolvidos analiticamente através de métodos bastante conhecidos. Entretanto o número de estados deve ser reduzido para não comprometer seu custo computacional. Em nossos testes, os modelos MMPP e HMM passaram a ter elevado custo computacional quando eram incorporadas filas de tamanho acima de 500.

A tabela 4.1 sumariza as principais características dos modelos aqui apresentados. A coluna *Número de Estados* indica o número de estados do modelo sem considerar o tamanho da fila. Esta tabela também aponta os modelos que podem ser resolvidos analiticamente e o desempenho em determinar a ocupação real da fila.

<i>Modelo</i>	<i>Número de Estados</i>	<i>Total de parâmetros</i>	<i>Método de solução</i>	<i>Desempenho na fila</i>
Sobrecarga	9	27	Análítico	ruim
MAP 16	16	6	Análítico	médio
MAP 32	32	7	Análítico	bom
MMPP	17500	5	Simulação	–
<i>HMM'</i>	124	5	Análítico	muito bom

Tabela 4.1: Características dos modelos analisados

Um dos fatores importantes na escolha de um modelo de tráfego é a sua flexibilidade em permitir que seus parâmetros sejam ajustados para que novas condições de tráfego sejam analisadas. Esta característica facilita o processo de dimensionamento, pois torna a parametrização independente do tráfego coletado, permitindo prever o desempenho para futuras demandas de carga. O modelo HMM, embora aproxime com precisão as características do tráfego medido, não oferece meios de alteração de seus parâmetros. É necessária uma nova coleta de tráfego e repetir o treinamento da cadeia.

O modelo MAP multifractal permite uma certa flexibilidade no ajuste das escalas de tempo em que o tráfego apresenta maior ou menor variabilidade. Pode-se determinar, por exemplo, a duração das rajadas de longa ou curta duração e a dependência temporal de longo alcance (LRD). De forma análoga, o modelo de sobrecarga e folga permite alterar a distribuição das rajadas alterando o decaimento de $P[D > t]$, porém esta mudança não é muito intuitiva.

A grande vantagem do modelo MMPP é a sua analogia com o processo de geração de tráfego através da hierarquia de sessões, fluxos e pacotes, possibilitando um ajuste dos parâmetros de maneira bastante intuitiva e real. Os parâmetros podem ser alterados baseados no número de usuários, no seu comportamento e no tipo de aplicação predominante. Pode-se simular, por exemplo, qual será o impacto no desempenho da rede de uma nova aplicação, ou qual o tráfego que será gerado em um *site* após uma campanha de divulgação de comércio eletrônico.

No Capítulo 5 usaremos a modelagem do tráfego como parte integrante do processo de planejamento e gerenciamento de capacidade. No exemplo que será dado,

escolheremos o modelo de acordo com as considerações aqui apresentadas.

Capítulo 5

Dimensionamento de Capacidade Baseado em Medição de Fluxos: Um Estudo de Caso

ESTE capítulo reúne os conceitos e técnicas apresentados ao longo desta tese com o objetivo de integrá-los em um processo de dimensionamento de capacidade. Através de um estudo de caso, aplicaremos o método proposto no Capítulo 3 para dimensionar uma rede baseando-se apenas em medições de fluxos coletados em uma rede real.

5.1 Introdução

As redes IP agregam uma grande variedade de tráfego proveniente de inúmeros usuários que compartilham estatisticamente os recursos disponíveis. Em períodos de congestionamento ocorrem os retardos e as perdas de pacotes, resultando em queda de desempenho da aplicação que pode ser percebido pelo usuário. Vários são os requisitos de desempenho exigidos pelas aplicações. Em uma transferência FTP, por exemplo, o importante é que haja vazão suficiente para abreviar o tempo de espera. No acesso a páginas *Web*, o que é percebido é o tempo de resposta.

Logo, uma baixa latência é o objetivo. Já para as aplicações multimídia que geram *streaming* de vídeo e áudio, exigências como retardo e *jitter* passam a ter prioridade. É possível, portanto, identificar a necessidade de múltiplas categorias de serviços, tendo cada qual sua própria característica de tráfego e requisitos de desempenho.

Para atender a estas necessidades variadas, pode-se adotar uma arquitetura de QoS ou provisionar a capacidade da rede de forma que os requisitos de desempenho sejam preservados. Visto que o *best effort* é o serviço oferecido pela Internet e pela maioria das redes IP, vamos explorar o provisionamento como alternativa para garantir o desempenho das aplicações.

Neste capítulo descrevemos o processo de dimensionamento de capacidade, que contempla desde a monitoração do tráfego até a escolha correta dos enlaces de acordo com o nível de serviço requerido. A monitoração de fluxos, em conjunto com o método de geração de pacotes, é utilizada para a caracterização do tráfego coletado. Na seção 5.3 é apresentado um estudo de caso ilustrando a aplicação deste processo.

5.2 Dimensionamento de Capacidade

O dimensionamento do tráfego Internet atual ainda é uma questão em aberto. Na ausência de mecanismos que ajustem a taxa de tráfego que é injetada na rede, ou de um controle de admissão que atenda os requisitos de QoS de uma conexão, uma opção é adotar um provisionamento extra de capacidade ou *over provisioning* de forma que os usuários obtenham uma qualidade satisfatória. Esta técnica mantém a rede com uma capacidade ociosa na maior parte do tempo, mas adequada para suprir eventuais rajadas ou contingências de *links*. Portanto, todo o tráfego recebe o mesmo nível de serviço, que deve ser suficiente para atender a todas as aplicações. A medida que a utilização atinge um nível que degrade o desempenho oferecido aos usuários, a capacidade dos *links* pode ser aumentada trazendo a utilização novamente para níveis baixos. O aumento da capacidade pode ser feito pela adição de novos *links* ou com aumento da velocidade dos existentes.

Embora o *over provisioning* possa atender os requisitos de desempenho, os custos

desta solução podem ser bastante altos. Em função disto, é necessário desenvolver métodos para dimensionar a rede de maneira a otimizar os recursos, procurando evitar a capacidade ociosa em excesso e sem comprometer o desempenho. Para dimensionar a capacidade dentro deste cenário, dois fatores são fundamentais: (i) Determinar o provisionamento adicional, baseado no ganho da multiplexação estatística, e (ii) minimizar o custo da rede garantindo uma qualidade de serviço mínima para as aplicações. A qualidade de serviço mínima depende das necessidades dos usuários. Na maioria dos casos devemos atender os requisitos de vazão e retardo das aplicações. O balanceamento correto destas premissas proporcionará uma otimização dos recursos para atender os múltiplos serviços presentes na rede, sem a necessidade de mecanismos específicos de QoS.

O ganho com a multiplexação estatística vem do fato que, quanto maior o número de fontes de tráfego agregadas em um *link*, mais o tráfego se aproxima de um processo Gaussiano [2]. Em consequência, ele tende a ser mais suavizado, possibilitando elevar a utilização do *link*. A convergência para uma distribuição de taxa Gaussiana foi observada em *links* do backbone Internet com volume de tráfego acima de 50 Mbps [31].

Minimizar o custo da rede é o principal objetivo a ser atingido no dimensionamento de capacidade. Este custo inclui equipamentos, contratação de serviços, pessoal especializado, instalação, manutenção e demais custos financeiros. Neste estudo de caso definiremos uma *função de custo* a ser minimizada baseado em restrições definidas pela qualidade mínima de serviço a ser atendida.

O processo de dimensionamento que vamos utilizar relaciona todos estes aspectos, fornecendo ao mesmo tempo otimização dos recursos e desempenho a custos reduzidos.

5.2.1 O processo de Dimensionamento

O processo apresentado na Figura 5.1 mostra as etapas para determinar a capacidade dos enlaces atendendo as condições apresentadas. Este processo é usado

tanto para o planejamento quanto para o gerenciamento da capacidade [42].

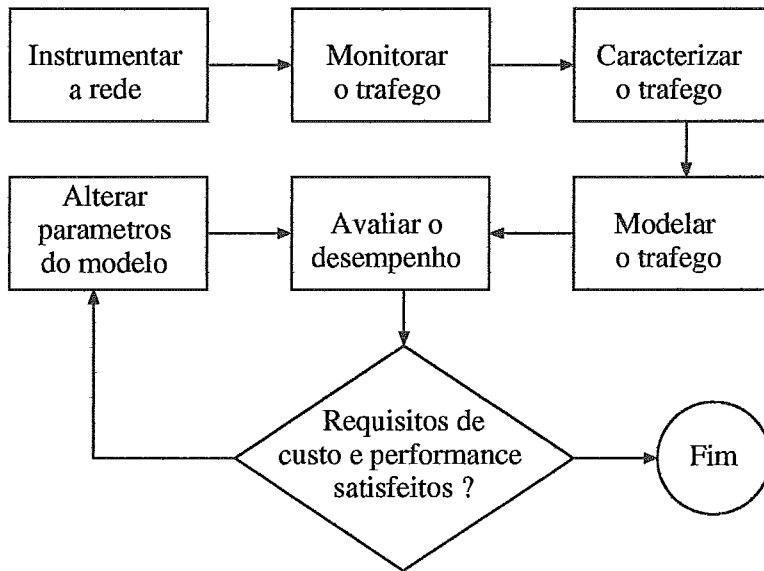


Figura 5.1: Processo de dimensionamento de capacidade

A primeira etapa consiste em instrumentar a rede, instalando os monitores e a infra-estrutura de medição do tráfego. Em nosso estudo de caso utilizaremos o *Netflow* instalado em um roteador da rede como instrumento de monitoração. A etapa seguinte é a coleta dos dados que normalmente é realizada em dias e horários diferentes, de forma a monitorar o tráfego em diversas condições de utilização do *link*. O intervalo de tempo da coleta varia conforme o volume de dados gerado. A caracterização e a modelagem servem para representar o tráfego medido de forma a aproximá-lo de um modelo estatístico. Nesta etapa será usado o método de geração de pacotes para extrair os descritores necessários na parametrização do modelo de tráfego. Através deste modelo é possível determinar o comportamento do tráfego em termos de vazão, perdas e retardo para diferentes condições de utilização do *link*. Os parâmetros do modelo, bem como as restrições de desempenho impostas pela rede podem ser alteradas em combinações que permitam avaliar a melhor alternativa que atenda aos requisitos de custo e desempenho das aplicações.

5.3 Estudo de Caso

Este estudo contempla medições reais de fluxos realizadas em uma rede corporativa no período de 10 de Setembro a 25 de Novembro de 2003. Os dados foram coletados por um roteador do nó principal da rede que concentra os *links* provenientes de várias localidades e o acesso para Internet. Para ilustrar o processo de dimensionamento, estaremos analisando o tráfego para dois *links* específicos desta rede.

5.3.1 Arquitetura da rede

A Figura 5.2 ilustra o ambiente de análise, onde estão evidenciados os *links* relacionados a coleta de tráfego que faremos. Não foram incluídos *links* de contingência e demais rotas alternativas.

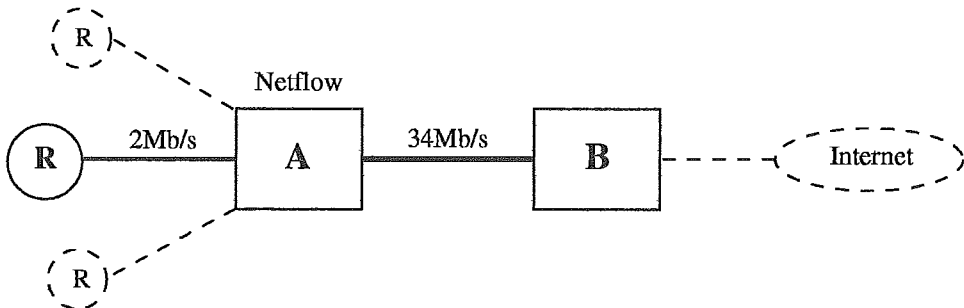


Figura 5.2: Rede de análise.

Esta rede é formada por dois nós principais localizados em cidades diferentes e conectados por um *link* de 34Mbps. Na localidade B estão concentrados todos os servidores de banco de dados, servidores de Intranet e demais servidores corporativos, além da saída para a Internet. Na localidade A existem 2700 usuários que acessam os servidores em B e Internet, além de concentrar *links* de outras localidades. A localidade R é um escritório remoto com 300 usuários e também utiliza os serviços em B e o acesso Internet. Os demais escritórios remotos totalizam 800 usuários.

No roteador da localidade A foi ativado o *Netflow* que captura e exporta os fluxos

para um servidor de armazenamento que está localizado na própria rede local em A. Neste servidor foi instalado o *Flow-tools* [32], responsável por armazenar, gerenciar e analisar os datagramas enviados pelo processo de exportação do *NetFlow*. Através da monitoração de fluxos feita pelo roteador em A é possível determinar o tráfego nas duas direções dos *links* de 34Mb/s e 2Mb/s. Os fluxos são coletados em todas as 4 interfaces do roteador, representando um volume diário de dados de 150 Mbytes.

Paralelamente, neste roteador, foi ativada a monitoração SNMP, contabilizando o histórico do tráfego de entrada e saída em cada uma das interfaces. Os administradores se valem desta monitoração para fazer o gerenciamento da rede, identificar períodos de congestionamento, gerar relatórios e atuar na ocorrência de problemas.

5.3.2 Monitoração do Tráfego

Após instrumentar a rede, iniciamos a monitoração com a coleta do tráfego em cada *link*. Esta coleta foi efetuada simultaneamente através dos contadores SNMP e pelos fluxos do *Netflow*.

Com o auxílio das medições SNMP, verificamos a utilização média típica dos *links*, procurando identificar os períodos de maior demanda de tráfego. A Figura 5.3 ilustra a medição SNMP observada durante o dia 25/11/2003 medido na interface de 34Mb/s do roteador em A. A medição é feita tanto para o *link* de entrada sentido B-A, como para o *link* de saída sentido A-B. A utilização do *link* nos dois sentidos fica em torno de 50% no período entre 09:00 e 18:00, atingindo picos de 62% de utilização. Nesta figura também é mostrado o histórico das medidas para a semana corrente e o acumulado mensal.

A monitoração dos fluxos ocorreu de forma contínua durante o período de análise, pois a ativação do *Netflow* não comprometeu o desempenho do roteador, e o volume de dados gerado pode ser facilmente gerenciado. Com esta monitoração foi possível obter estatísticas mais detalhadas do tráfego referente a sua composição. A Tabela 5.1 mostra a proporção do tráfego IP para os *links*, onde temos a total predominância do TCP. Analisando os fluxos com base nas portas TCP, verificamos que

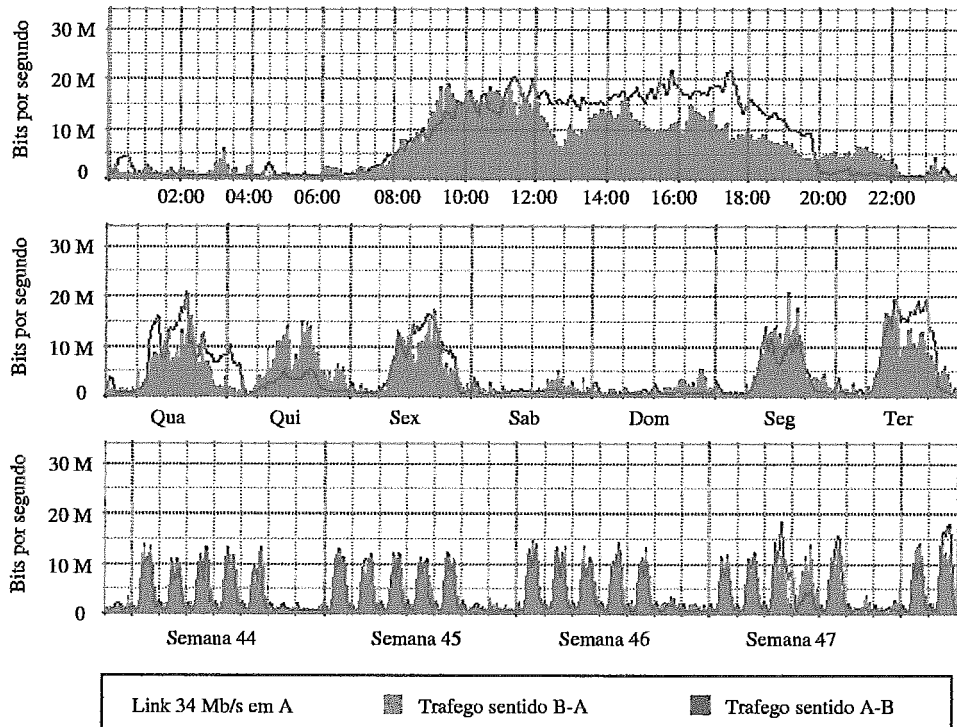


Figura 5.3: Monitoração SNMP para o *link* de 34Mb/s.

as aplicações *Web* (HTTP porta 80) são responsáveis pela maioria dos fluxos e do volume de tráfego. A exceção ocorre com o tráfego no sentido A-B, onde existe um grande número de acessos a aplicativos de banco de dados localizados em B. A proporção do tráfego *Web* pode ser vista na Tabela 5.2. Nestas duas tabelas o intervalo de observação foi das 08:00 as 20:00 do dia 25/11/2003, abrangendo o período de maior atividade de tráfego.

<i>Link</i>	% <i>TCP</i>	% <i>UDP</i>	% <i>Outros</i>
34Mb/s: sentido A-B	98.6	1.1	0.3
34Mb/s: sentido B-A	98.1	1.8	0.1
2Mb/s: sentido R-A	96.8	2.7	0.5
2Mb/s: sentido A-R	99.8	0.2	0.0

Tabela 5.1: Proporção de tráfego IP nos *links* da rede em análise.

De qualquer modo, a monitoração de fluxos e SNMP presentes nesta rede não fornecem a granularidade de tempo adequada para o dimensionamento de capacidade. Para contornar esta limitação, e mantendo a mesma infra-estrutura de medição, iremos caracterizar o tráfego em nível de pacotes aplicando o método de geração de pacotes proposto no Capítulo 3.

<i>Link</i>	<i>HTTP - porta 80</i>		
	<i>%Fluxos</i>	<i>%Bytes</i>	<i>%Pacotes</i>
34Mb/s: sentido A-B	47.5	10.2	15.7
34Mb/s: sentido B-A	56.2	52.4	41.2
2Mb/s: sentido R-A	66.5	48.3	66.6
2Mb/s: sentido A-R	56.4	87.34	78.32

Tabela 5.2: Proporção de tráfego *Web* nos *links* da rede em análise.

5.3.3 Caracterização e Modelagem do Tráfego

O objetivo desta etapa é representar o tráfego coletado na rede por um modelo que possa ser resolvido analiticamente. Para esta modelagem foram preparados quatro *traces* de pacotes obtidos com o método de geração de pacotes, a partir dos fluxos coletados pelo roteador. A Tabela 5.3 mostra o dia e o horário de coleta dos *traces*, que foram gerados representando o número de pacotes em intervalos de 200ms.

<i>Trace</i>	<i>Link</i>	<i>Dia</i>	<i>Horário</i>
1	34Mb/s: sentido A-B	25/11/2003	16:00-18:00
2	34Mb/s: sentido B-A	25/11/2003	09:00-11:00
3	2Mb/s: sentido R-A	23/10/2003	12:00-14:00
4	2Mb/s: sentido A-R	10/09/2003	10:00-12:00

Tabela 5.3: Especificação dos *traces* de pacotes usados na modelagem do tráfego.

A geração dos *traces* foi feita através do seguinte procedimento:

- Inicialmente utilizamos as medições SNMP para identificar o comportamento médio do tráfego nos *links* ao longo de vários dias. Foram selecionados os períodos de maior tráfego.
- Os fluxos coletados correspondentes aos períodos selecionados foram testados quanto a possíveis perdas em função do processo de exportação ser via UDP. As medidas realizadas mostraram que as perdas eram nulas ou desprezíveis, garantindo, assim, a integridade da coleta dos fluxos. A baixa perda verificada deve-se ao fato do equipamento coletor estar bem dimensionado e localizado em segmento isolado de rede.

- Através do programa *Flow-tools* foi gerado um trace de fluxos no formato a ser lido pelo gerador de pacotes. Foram feitas pequenas alterações neste programa de forma a fornecer o formato requerido. Um exemplo deste formato é apresentado na seção 4.4.2.
- Através do programa de geração de pacotes foi gerada uma sequência com o número de pacotes em intervalos 200ms.
- Esta sequência foi então usada para caracterização e modelagem do tráfego.

A análise do tráfego com uma granularidade de 200ms permitiu identificar rajadas que ultrapassam a capacidade do canal, podendo causar retardo e perda de pacotes. Como exemplo, vejamos a medição SNMP da Figura 5.3 no *link* sentido B-A entre 09:00 e 11:00, onde temos uma utilização moderada do canal em torno de 50% ou 17Mb/s. Através do trace de pacotes gerado, descobrimos que a utilização neste mesmo período frequentemente atinge 100% da capacidade do canal, conforme mostra a Figura 5.4.

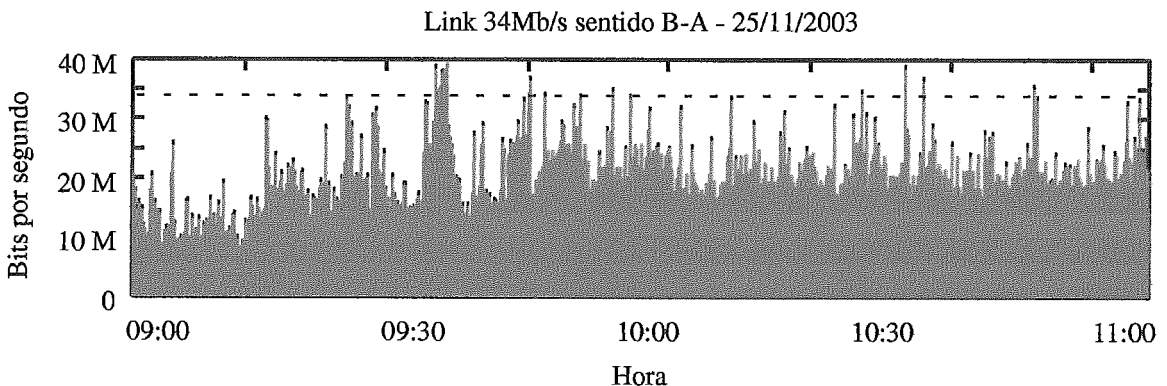


Figura 5.4: Tráfego com granularidade de 200ms medido no *link* de 34Mb/s.

A modelagem do tráfego é uma das etapas mais críticas no processo de dimensionamento, onde informações imprecisas podem levar a resultados que não atendam aos requisitos de custo e desempenho. Desta forma, a escolha do modelo foi baseada nas conclusões obtidas no Capítulo 4. Para o dimensionamento é necessário que o modelo reproduza corretamente a ocupação em uma fila e, conforme demonstrado, o modelo que melhor atende a esta característica é o HMM. Sendo assim, a parametrização consistiu em treinar a cadeia HMM com as sequências da Tabela 5.3. Para cada

cadeia HMM gerada foi adicionada uma fila de tamanho 100, completando, assim, o modelo de tráfego para cada enlace da rede. Com a ferramenta Tangram-II [21] resolvemos matematicamente os modelos, determinando o número médio de pacotes na fila para diversos valores de capacidade. Foram considerados apenas valores que resultaram em uma utilização inferior a 80%. O resultado é apresentado na Figura 5.5 que mostra a ocupação da fila em função da capacidade para cada *link* em estudo.

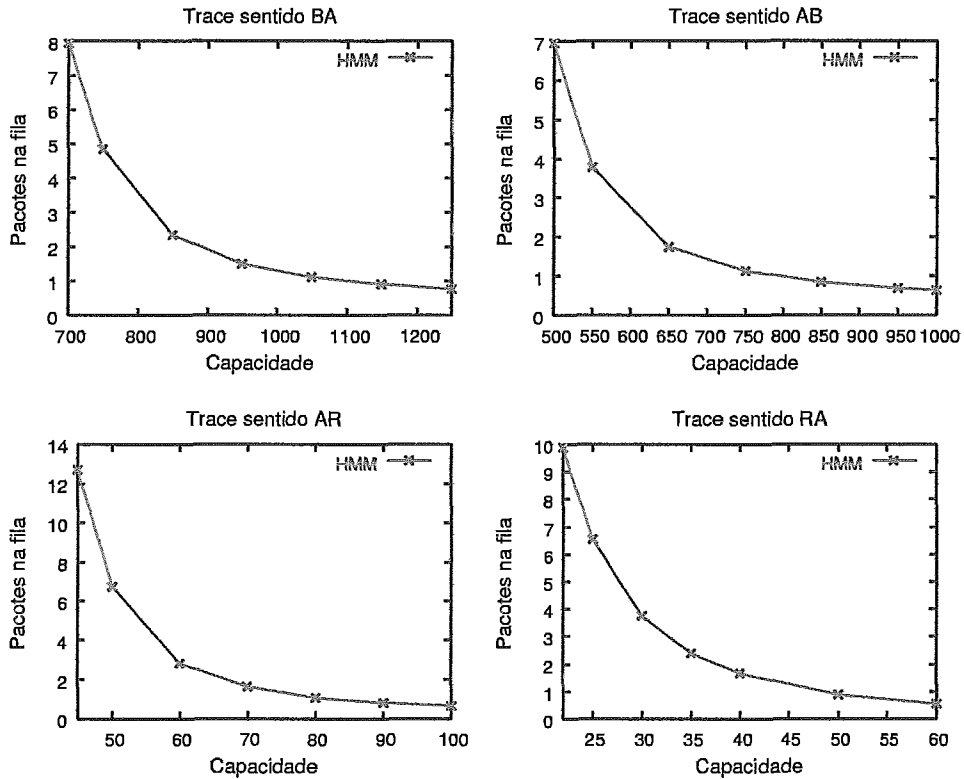


Figura 5.5: Ocupação média da fila para os quatro *links* em estudo.

5.3.4 Avaliação de Desempenho

Com base no tamanho das filas calculado na etapa anterior, iremos avaliar qual a capacidade ótima que devemos ter para atender aos requisitos de custo e desempenho. Esta avaliação será efetuada seguindo a metodologia empregada em [22] para o dimensionamento de redes, conforme abaixo:

- Mapear os requisitos de desempenho necessários ao bom desempenho das aplicações.

- Definir, a partir do modelo de tráfego, uma função de mapeamento atraso versus capacidade do enlace.
- Otimizar o custo da rede.

Mapeamento dos requisitos de desempenho: O primeiro passo contempla mapear os requisitos de QoS das aplicações em métricas de desempenho de rede. Vimos que as aplicações *Web* são predominantes, bem como o acesso a banco de dados com a transferência de grandes volumes de dados. Desta forma, para atender ao bom desempenho destas aplicações, estaremos mapeando a latência e a vazão em parâmetros configuráveis da rede como o atraso fim-a-fim (RTT) e a probabilidade de perda. O modelo que relaciona o requisito de vazão com o RTT e a probabilidade de perda é o modelo TCP proposto em [59] e utilizado na seção 3.2.4. Através da seguinte equação podemos calcular o RTT médio que atenda ao requisito de vazão e probabilidade de perda:

$$RTT_{vazao} = \min \left(\frac{W_m}{B}, \frac{1}{B(\sqrt{\frac{2bp}{3}} + k \min(1, 3\sqrt{\frac{3bp}{8}})p(1 + 32p^2))} \right)$$

Onde p representa a probabilidade de perda, b o número de pacotes confirmados por um ACK, $k = T_0/RTT$ a relação de *time out* do TCP sendo T_0 o tempo de *time out*, W_{max} a janela máxima do TCP e B a vazão média do fluxo. Substituindo na equação os valores utilizados no Capítulo 3, temos que $p = 0.005$, $b = 2$, $k = 1.1$ e $W_{max} = 20$. Sendo assim, o RTT mínimo para que a vazão B seja de 512Kb/s é de $RTT_{vazao} = 0.2582s$. Adotamos este valor de vazão para reduzir o tempo das transferências de arquivos grandes (acima de 10Mbytes) e o tempo de resposta no acesso aos bancos de dados.

O outro requisito de desempenho que devemos atender é a latência. Um interessante estudo conduzido por Fiona Nah [55] conclui que o tempo de espera aceitável para um usuário *Web* receber uma resposta é de 2s, quando este não tem informação sobre o progresso da solicitação. Para chegar a esta conclusão Nah submeteu um grupo de estudantes para procurar um produto específico em uma página *Web* que continha dez *links*, três dos quais não funcionavam. Quando um estudante acionava

um *link* inoperante e não recebia resposta, ele acionava outro ou o botão de parar (*stop*). O tempo entre estes dois acionamentos foi então definido como **tempo de espera**. Nah também identificou que, quando o usuário era informado de que a página *Web* estava sendo carregada, através de uma barra de progresso, os tempos de espera eram significativamente maiores.

Com base neste estudo, vamos adotar o tempo de 2s como sendo o tempo de resposta total. Considerando que o tempo de resposta do servidor para processar a página *Web* seja de 0.5s, o tempo de latência máximo da rede será de 1.5s. Em [23, 19] são apresentados estudos sobre a distribuição do tamanho dos arquivos presentes na *Web*, indicando que a grande maioria é menor que 20Kbytes ou 20 pacotes para um tamanho médio de segmento de 1000bytes. Em consequência, o TCP termina de enviá-los antes de sair da fase de *slow-start*. Isto pode ser comprovado pela equação apresentada em [14] que determina o número de pacotes enviados durante a fase de *slow-start* antes da ocorrência de perdas:

$$Edss = \frac{(1 - (1 - p)^d)(1 - p)}{p} + 1$$

Onde p é a probabilidade de perda e d o número de pacotes enviados. Sendo $p = 0.005$ e $d = 20$ pacotes, $Edss = 20$, o que significa que todos os pacotes são transmitidos na fase de *slow-start*.

Utilizando a equação apresentada na seção 3.2.3 podemos calcular o RTT em função da latência da fase de *slow-start*:

$$RTT_{latencia} = \frac{Tss}{\lceil \log_g \left(\frac{Edss+2}{C_1} \right) \rceil - 2}$$

onde $g = 1.618033989$ e $C_1 = 0.723606798$ são constantes de Fibonacci.

Fazendo $Tss = 1.5s$ e $Edss = 20pacotes$ temos que $RTT_{latencia} = 0.3s$

Assim, o RTT que deverá ser considerado para o projeto é:

$$RTT = \min(RTT_{vazio}, RTT_{latencia}) = 0.2582s$$

Este valor será utilizado para dimensionar a rede considerando o acesso aos servidores que estão na localidade **B**. Logo, este tempo é válido para acessos feitos a partir das localidades **A** e **R**.

Função de mapeamento Atraso versus Capacidade: Após mapear o requisito de QoS necessário vamos definir, com base na ocupação da fila obtida com o modelo, uma função que associe a capacidade dos enlaces à QoS requisitada. Usando a lei de Little podemos obter o atraso da fila em função destas capacidades.

Seja i equivalente a um enlace da rede da Figura 5.2 que representa um par origem-destino, ou seja, $i \in \{BA, AB, AR, RA\}$. Nosso objetivo é obter uma função $f(d_i)$, que mapeia o atraso d_i da fila do roteador na capacidade do canal C_i . Isto é possível, pois estamos considerando a probabilidade de perda de pacotes pequena (0,005). A Figura 5.6 ilustra a relação entre atraso e capacidade mostrando a curva ajustada da função $f(C_i)$ para os quatro enlaces.

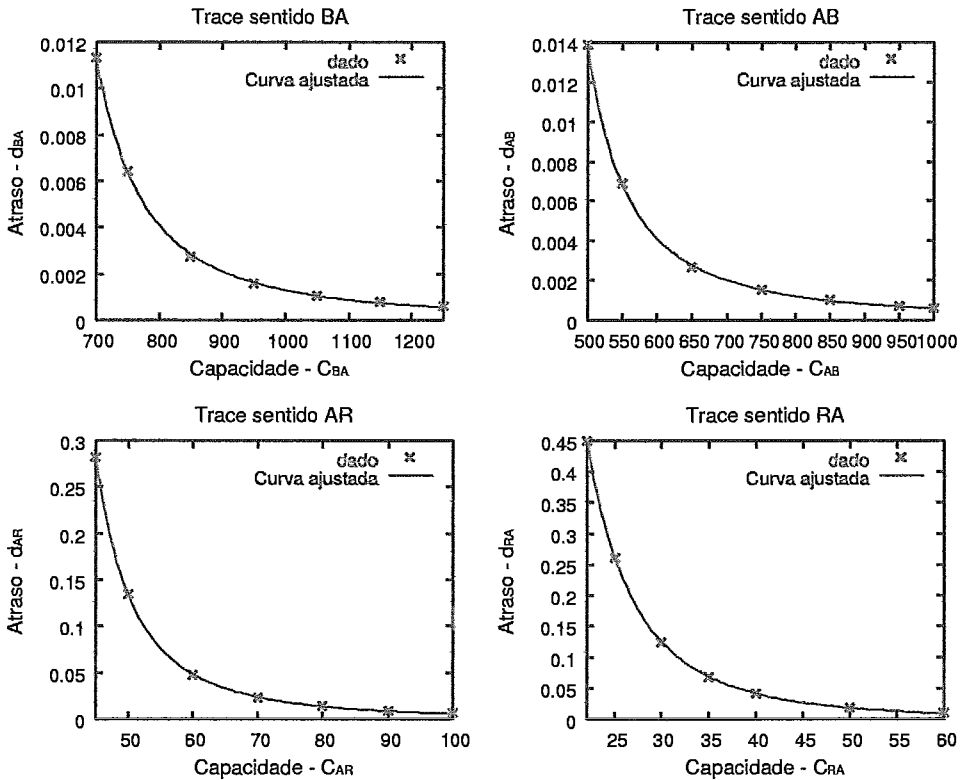


Figura 5.6: Curva ajustada para as funções de atraso versus capacidade dos enlaces.

As funções que mapeiam o tempo de espera na fila em função da capacidade foram obtidas através de programas de regressão não linear [63, 77] e foram dadas preferências para as funções de fácil inversão, definidas em todo o domínio e com derivada primeira negativa para o domínio em questão. As funções candidatas para o ajuste da curva aos dados foram a exponencial com deslocamento e a função potência com deslocamento.

$$\text{A) } f(C_i) = a(C_i - b)^c \quad \text{B) } f(C_i) = ae^{C_i b} + c$$

Figura 5.7: Funções candidatas. A) Potência, B) Exponencial.

A função candidata escolhida para todos os enlaces foi a função potência com deslocamento, pois demonstrou o melhor ajuste aos pontos da curva. As equações $f(C_i)$ para os enlaces com os parâmetros ajustados são:

$$\text{Enlace B-A: } f(C_{BA}) = 153.175(C_{BA} - 557.595)^{-1.91767}$$

$$\text{Enlace A-B: } f(C_{AB}) = 54.9648(C_{AB} - 397.269)^{-1.78762}$$

$$\text{Enlace A-R: } f(C_{AR}) = 158.481(C_{AR} - 31.1639)^{-2.40882}$$

$$\text{Enlace R-A: } f(C_{RA}) = 3832.92(C_{RA} - 5.58119)^{-3.23419}$$

O problema de otimização a ser formulado exige que a função custo seja expressa em função do atraso $f(d_i)$ em cada enlace, pois as restrições são expressas em função desta variável. Desta forma, trabalharemos com a inversa da função $f(C_i)$. Definiremos, então, a função $f(d_i) = f^{-1}(C_i)$ para cada enlace:

$$\text{Enlace B-A: } f(d_{BA}) = \left(\frac{d_{BA}}{153.175}\right)^{-\frac{1}{1.91767}} + 557.595$$

$$\text{Enlace A-B: } f(d_{AB}) = \left(\frac{d_{AB}}{54.9648}\right)^{-\frac{1}{1.78762}} + 397.269$$

$$\text{Enlace A-R: } f(d_{AR}) = \left(\frac{d_{AR}}{158.481}\right)^{-\frac{1}{2.40882}} + 31.1639$$

$$\text{Enlace R-A: } f(d_{RA}) = \left(\frac{d_{RA}}{3832.92}\right)^{-\frac{1}{3.23419}} + 5.58119$$

Otimização da capacidade:

Nesta etapa vamos definir uma função de custo para a capacidade dos enlaces. Vamos formular e resolver o problema de otimização para calcular as capacidades mínimas dos enlaces para satisfazer os requisitos de desempenho, sujeitos a uma função de restrição. O problema de otimização, denominado de **atribuição de capacidade (AC)**, relaciona uma função de custo U_i do enlace com a sua capacidade C_i . A função de custo representa o custo financeiro por unidade de capacidade C_i do enlace i com base na aquisição, implementação e manutenção da banda requerida. A capacidade do enlace é obtida em função do atraso na fila d_i . Podemos, então, formular o problema de otimização da seguinte forma:

Minimizar

$$AC = \sum_i f(d_i)U_i = f(d_{BA})U_{BA} + f(d_{AB})U_{AB} + f(d_{AR})U_{AR} + f(d_{RA})U_{RA}$$

Sujeito às seguintes restrições:

Restrição 1: $d_{RA} + d_{AB} + d_{BA} + d_{AR} < 0.2582$

Restrição 2: $d_{AB} + d_{BA} < 0.2582$

Restrição 3: $d_{AB} > 1/2000$

Restrição 4: $d_{BA} > 1/2000$

Restrição 5: $d_{RA} > 1/2000$

Restrição 6: $d_{AR} > 1/2000$

A restrição 1 considera que o RTT entre as localidades **R** e **B** deve ser menor que 0.2582, da mesma forma que a restrição 2, entre as localidades **A** e **B**. As demais restrições garantem que o atraso calculado para cada enlace não seja negativo.

Para resolver este problema optamos por um algoritmo de otimização não-linear primal-dual de barreira logarítmica (pontos interiores), pois este não supõe convexidade das funções. A solução do problema de otimização fornecerá o atraso d_i em cada um dos quatro enlaces. Através da função $f_i(d_i)$ podemos calcular a capacidade C_i de cada enlace para que tenha o atraso na fila d_i . Além disto, devemos considerar o tamanho médio do pacote p_i para cada enlace. As capacidades dos enlaces CL_i

foram, então, calculadas pelas seguintes expressões:

$$C_{BA} = f(d_{BA})p_{BA}$$

$$C_{AB} = f(d_{AB})p_{AB}$$

$$C_{AR} = f(d_{AR})p_{AR}$$

$$C_{RA} = f(d_{RA})p_{RA}$$

Resultados alcançados:

Parâmetros iniciais:

- Função custo dos enlaces U_i : O custo do Mb/s diminui a medida que a velocidade do enlace aumenta. Considerando que os enlaces AB e BA tem o custo do Mb/s 5 vezes menor que o custo dos enlaces AR e RA, temos:

$$U_{BA} = U_{AB} = 1, \quad U_{AR} = U_{RA} = 5$$

- Pacote médio em bytes p_i :

$$p_{BA} = 1016, \quad p_{AB} = 1124, \quad p_{AR} = 1011, \quad p_{RA} = 896$$

Utilizando o sistema de computação algébrica *Maple* versão 8.0 e o pacote de otimização não-linear disponível em [72], obtivemos os seguintes resultados:

- Atraso d_i nos enlaces que atendem às restrições impostas:

$$d_{BA} = 0.05481, \quad d_{AB} = 0.04815 \quad d_{AR} = 0.08095 \quad d_{RA} = 0.07428$$

- Capacidade C_i dos enlaces:

$$C_{BA} = 5.041594 \cdot 10^6 \quad C_{AB} = 4.033787 \cdot 10^6$$

$$C_{AR} = 0.440149 \cdot 10^6 \quad C_{RA} = 0.245374 \cdot 10^6$$

A capacidade C_i calculada corresponde ao intervalo de tempo de 200ms. Para obtermos a capacidade em Mb/s multiplicaremos este valor por 5:

$$C_{BA} = 25.207970 \text{ Mb/s} \quad C_{AB} = 20.168935 \text{ Mb/s}$$

$$C_{AR} = 2.200745 \text{ Mb/s} \quad C_{RA} = 1.22687 \text{ Mb/s}$$

Analisando estas capacidades podemos concluir que os enlaces entre as localidades AB estão bem dimensionados, pois atualmente contam com *links* de 34Mb/s. Já o enlace de 2Mb/s no sentido AR deve ser aumentado para atender à demanda solicitada. A Tabela 5.4 resume a capacidade atual e a capacidade dimensionada para atender aos requisitos de QoS.

<i>Link</i>	<i>Capacidade Atual</i>	<i>Capacidade Dimensionada</i>
Sentido A-B	34 Mb/s	20.17 Mb/s
Sentido B-A	34 Mb/s	25.21 Mb/s
Sentido R-A	2 Mb/s	1.23 Mb/s
Sentido A-R	2 Mb/s	2.20 Mb/s

Tabela 5.4: Capacidade atual e dimensionada para a rede em estudo.

5.4 Resumo e Conclusões

O projeto de uma rede de dados que atenda aos requisitos de QoS das aplicações depende de um correto dimensionamento dos recursos, principalmente quando não há mecanismos específicos que garantam o nível de serviço oferecido pela rede. Este capítulo apresentou um processo de dimensionamento que determina a capacidade necessária para acomodar as diferentes classes de tráfego que compartilham os recursos de um canal.

O processo de dimensionamento foi aplicado em um estudo de caso baseado em medições reais de rede. Neste estudo demonstramos que o método de geração de pacotes pode ser usado como ferramenta para caracterização do tráfego, valendo-se apenas da coleta de fluxos. Com isto, foi possível parametrizar o modelo de tráfego HMM para determinar a ocupação das filas dos roteadores. É importante salientar que tanto a monitoração dos fluxos quanto a modelagem do tráfego é feita de forma simples e rápida, pois foram utilizados métodos analíticos de solução.

O ponto-chave no planejamento de capacidade é a obtenção de uma rede que satisfaça aos requisitos das aplicações com o menor custo possível. Devido a isto utilizamos a metodologia empregada em [22] que otimiza a relação entre perdas,

atraso e capacidade do *link* para atender aos requisitos impostos pelas aplicações. Normalmente, estes requisitos são exigências dos usuários para um rápido tempo de resposta, seja para acesso a uma página *Web* ou para transferência de grandes arquivos. Com base nisto, nosso estudo de caso procurou atender tanto ao requisito de vazão como de latência.

O estudo de caso apontou a necessidade de aumento de banda para o enlace entre as localidades A e R. Já os enlaces entre as localidades A e B estão atendendo a demanda atual, podendo ainda suportar um crescimento do tráfego. Esta conclusão foi possível devido a precisão das medidas e da otimização empregada pelo método de dimensionamento. O processo de dimensionamento apresentado, juntamente com a monitoração de fluxos geralmente presente nas redes pode ser usado nas atividades de gerenciamento da rede e implementado pelos administradores, possibilitando otimizar o uso da rede e postergar novos investimentos.

Capítulo 6

Contribuições e Trabalhos Futuros

O gerenciamento e o planejamento de redes IP são otimizados quando baseados em medições de tráfego oriundas de diversas fontes de monitoração, com diferentes granularidades de tempo e níveis de agregação de tráfego [61]. É importante que o projetista tenha informações diversas que apontem características predominantes, tanto para curtos períodos de tempo, como para períodos maiores. É necessário conhecer padrões de uso que o ajudem a identificar possíveis congestionamentos, bem como aplicações que necessitem de garantias de desempenho. As medições passivas existentes possibilitam obter informações variadas sobre o tráfego com maior ou menor nível de detalhe, dependendo do método empregado.

A monitoração SNMP, uma das mais utilizadas na administração de redes, mede a utilização dos enlaces continuamente, mas em escalas de tempo na ordem de minutos, o que suaviza os picos do tráfego medido. A medição de fluxos fornece uma visão mais detalhada da composição do tráfego, capturando o comportamento de aplicações e usuários, podendo ser executada nos próprios roteadores. Contudo, a informação em pequenas escalas de tempo é perdida. Já a monitoração de pacotes, embora precisa em escalas menores, é disassociada da dinâmica das aplicações e usuários, além de necessitar de uma infra-estrutura complexa e pessoal especializado para a coleta dos dados.

O tema central desta tese foi mostrar que é possível obter medidas de tráfego

equivalentes à monitoração de pacotes a partir da coleta de fluxos. Nosso objetivo é possibilitar que meios mais simples de coleta forneçam informações adequadas para o dimensionamento de redes. Devemos olhar a monitoração de fluxos como uma alternativa viável para fornecer informações com diferentes granularidades de tempo e níveis de agregação de tráfego.

6.1 Contribuições

6.1.1 Método de geração de pacotes

No Capítulo 3 apresentamos o método de geração de pacotes a partir de informações de fluxos, que é baseado em modelos analíticos do TCP. O método mimetiza a evolução temporal da janela de congestionamento para cada fluxo individualmente, resultando no número de pacotes transmitidos em cada instante de tempo. Quando este processo de decomposição individual é executado para um conjunto grande de fluxos, a contagem de pacotes por intervalo de tempo se aproxima de uma contagem real de pacotes no mesmo intervalo. Também foi analisado o tempo entre chegadas através da geração do *time-stamp* de cada pacote. Os resultados obtidos foram confrontados com *traces* reais, evidenciando a grande precisão do método, principalmente para a contagem de pacotes por intervalo de tempo. Nossas contribuições foram:

- Mostramos que fluxos classificados por: (i) endereço de origem e destino, (ii) porta de origem e destino e (iii) protocolo podem ser decompostos para a obtenção da distribuição temporal dos pacotes. Nosso objetivo foi obter medidas de granularidade mais fina, equivalente à medição de pacotes.
- Desenvolvemos um algoritmo que implementa uma estrutura em *array* para executar a contagem de pacotes por intervalo de tempo de forma eficiente, a partir de um *trace* de fluxos. O intervalo de tempo pode ser ajustado pelo usuário conforme suas necessidades.

- Verificamos que a distribuição do tamanho dos pacotes obtida pelo método apresenta valores muito próximos à distribuição real. Isto possibilita determinar o tráfego em bytes ou bits por intervalo de tempo com boa precisão, dado que a capacidade do canal normalmente é expressa em bits por segundo (*bits/s*).
- Mostramos que o comportamento do tráfego agregado é fortemente influenciado pela dinâmica do TCP, mesmo para proporções com apenas 50% de tráfego TCP. Os fluxos UDP contribuem como parcela constante do tráfego total, não tendo influência em sua variabilidade para determinadas escalas de tempo. O comportamento CBR foi observado para intervalos de tempo de 1s.
- Constatamos que o erro produzido pelo método possui o seguinte comportamento: (i) quanto maior o número de fluxos agregados em um enlace mais a seqüência de pacotes por intervalo de tempo gerada se aproxima da real e (ii) quanto maior a taxa do *trace* menor é o erro da seqüência estimada.
- O método pode ser aplicado na parametrização de modelos de tráfego. Mostramos que os descritores obtidos pelo método são próximos dos oriundos de *traces* reais, o que possibilita configurar diferentes modelos de tráfego.

Nossas conclusões foram no sentido de possibilitar uma nova abordagem na utilização dos fluxos, tanto para fornecer um procedimento mais simples de medição de pacotes, como para conceber novos modelos de tráfego baseados somente na dinâmica dos fluxos.

6.1.2 Análise de Modelos de tráfego

No Capítulo 4 analisamos o desempenho de quatro modelos Markovianos em determinar a ocupação em uma fila. Comparamos o custo computacional para a resolução analítica dos modelos, com o objetivo de utilizá-los no dimensionamento de redes. As contribuições foram:

- Verificamos que o modelo de sobrecarga e folga não teve bom desempenho em determinar a ocupação da fila para valores de utilização do *link* abaixo de 60%. A grande vantagem apresentada foi o seu baixo custo computacional.
- Para o modelo MAP multifractal adotamos um procedimento alternativo de parametrização baseado no decaimento da função de autocorrelação.
- Utilizamos medidas de fluxos em conjunto com o método de geração de pacotes para a parametrização do modelo MMPP hierárquico. Mostramos que é possível parametrizar modelos de pacotes baseados inteiramente em medições de fluxos.
- Mostramos que o modelo HMM é o mais preciso na determinação da ocupação da fila, embora o custo computacional possa comprometer sua utilização para filas com tamanho acima de 1000.

Concluimos que, além da precisão na ocupação da fila, é importante que o modelo seja flexível para alterações em seus parâmetros. Isto é necessário para ajustes e análise de futuras demandas de carga.

6.1.3 Estudo de Caso

No Capítulo 5 apresentamos um estudo visando o planejamento de capacidade de uma rede. Este estudo foi baseado no método de geração de pacotes e na metodologia de dimensionamento proposta em [22]. O dimensionamento dos *links* possibilita minimizar o custo e otimizar a capacidade ociosa da rede, de forma a atender os requisitos das aplicações. Apresentamos um estudo de caso baseado em medições de rede reais. Desenvolvemos passo a passo a análise da capacidade dos *links* utilizando a monitoração SNMP, a medição de fluxos, o método de geração de pacotes e a metodologia de [22]. Uma síntese das contribuições foram:

- Apresentamos uma proposta para dimensionamento e gerenciamento de capacidade para atender os requisitos de custo e desempenho desejados.

- Mostramos que a monitoração SNMP não captura as rajadas de curta duração, porém, estas podem ser identificadas através do método de geração de pacotes. A análise do tráfego em escala de 200ms revelou curtos períodos de congestionamento, mesmo quando reportada pelo SNMP uma utilização do *link* em torno de 50%.
- Analisamos uma rede real através de um estudo de caso e efetuamos o dimensionamento de dois enlaces desta rede valendo-se da monitoração de fluxos.

A monitoração de fluxos e o método de geração de pacotes fornecem aos administradores uma importante ferramenta de análise do tráfego com diferentes granularidades de tempo e níveis de agregação. Através destas informações, e utilizando um processo de dimensionamento, é possível gerenciar eficientemente a capacidade e os custos da rede.

6.2 Trabalhos Futuros

Na medida em que a velocidade dos *links* aumenta, medições detalhadas do tráfego se tornam mais complexas e difíceis de serem executadas. A medição passiva de fluxos existente nos roteadores é uma alternativa viável para contornar esta limitação, contudo, ainda carece de uma maior padronização entre os diferentes fabricantes. Por outro lado, os fluxos podem ser monitorados por equipamentos externos específicos para esta tarefa, utilizando hardware mais simples e a custos menores.

O método de geração de pacotes proposto amplia as alternativas de pesquisa na utilização dos fluxos para a caracterização e modelagem do tráfego. Futuros modelos de tráfego podem ser desenvolvidos baseados unicamente na geração de fluxos produzidos pelos usuários através da interação com as diferentes aplicações. A decomposição de fluxos fará o elo entre os modelos de alto nível e a multiplexação estatística de pacotes nos canais de alta velocidade.

Trabalhos futuros podem explorar ainda mais a capacidade do método, melho-

rando sua precisão em reproduzir a distribuição dos pacotes em intervalos ainda menores de tempo. Para isto, novas medidas e comparações precisam ser feitas para intervalos de 100ms e 10ms. Acreditamos que, com o aumento da velocidade dos *links*, o método mantenha a mesma precisão das medidas para intervalos menores de tempo. Podemos estudar o erro produzido pelo método em relação a vários aspectos como perdas, velocidade e utilização dos *links*.

Outra área de importante pesquisa é a simulação de novos protocolos ou modificações do TCP. No Capítulo 3 mostramos que é possível a decomposição dos fluxos e a determinação da distribuição dos pacotes. Da mesma, forma podemos criar um novo mecanismo de *slow-start*, por exemplo, e analisar qual o impacto na rede em termos de autocorrelação e ocupação em uma fila. Podemos também estudar qual a relação dos fluxos longos e curtos com a característica auto-similar do tráfego. Em aplicações envolvendo tomografia de rede, podemos inferir qual será o comportamento do tráfego de um backbone, através da decomposição temporal dos fluxos presentes nos *links* de acesso.

Várias são as aplicações que podem utilizar a decomposição de fluxos como ferramenta de medição. Esta relação não é exaustiva e novas propostas podem surgir baseadas no conceito de que podemos obter medidas detalhadas do tráfego a partir de mecanismos mais simples e de fácil implementação.

Referências Bibliográficas

- [1] ABRY, P., BARANIUK, R., FLANDRIN, P., RIEDI, R., E VEITCH, D. The Multiscale Nature of Network Traffic. *IEEE Signal Processing Magazine* (May 2002), 28–46.
- [2] ADDIE, R., ZUKERMAN, M., E NEAME, T. Broadband Traffic Modeling: Simple Solutions to Hard Problems. *IEEE Communications Magazine* 36, 8 (1998), 88–95.
- [3] ANDERSEN, A., E NIELSEN, B. A Markovian approach for modeling packet traffic with long-range dependence. *IEEE Journal on Selected Areas in Communications* 16, 5 (June 1998), 719–732.
- [4] ARIFLER, D., DE VECIANA, G., E EVANS, B. Network tomography based on flow level measurements. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (May 2004), vol. 2, pp. 437–440.
- [5] AZEVEDO, J. A., DE SOUZA E SILVA, E., DUARTE, F. P., LEÃO, R. M. M., RIBEIRO, B. F., E DA SILVA, A. P. C. Metodologia de Dimensionamento de Redes: Detalhamento. Relatório técnico, Universidade Federal do Rio de Janeiro - COPPE, Dezembro 2003.
- [6] BACCELLI, F., E HONG, D. Flow level simulation of large IP networks. In *IEEE INFOCOM* (April 2003), vol. 3, pp. 1911–1921.
- [7] BALAKRISHNAN, H., PADMANABHAN, V. N., SESHAN, S., STEMM, M., E KATZ, R. H. TCP behavior of a busy Internet server: Analysis and improvements. In *IEEE INFOCOM* (April 1998), pp. 252–262.

- [8] BARAKAT, C., THIRAN, P., IANNACCONE, G., DIOT, C., E OWEZARSKI, P. Modeling Internet backbone traffic at the flow level. In *IEEE Transactions on Signal Processing - Special Issue on Networking* (August 2003), vol. 51.
- [9] BARFORD, P., E PLONKA, D. Characteristics of network traffic flow anomalies. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement* (San Francisco, CA, USA, 2001), ACM Press, pp. 69–73.
- [10] BERAN, J. *Statistics for long-memory processes*. Chapman & Hall, 1994.
- [11] BROWNLEE, N. Netramet. <http://www2.auckland.ac.nz/net/NeTraMet/>.
- [12] BROWNLEE, N., E CLAFFY, K. C. Understanding Internet traffic streams : Dragonflies and tortoises. *IEEE Communication Magazine* (October 2002), 110–117.
- [13] BROWNLEE, N., MILLS, C., E RUTH, G. RFC 2722 - Traffic Flow Measurement: Architecture, 1999. <http://rfc.net/rfc2722.html>.
- [14] CARDWELL, N., SAVAGE, S., E ANDERSON, T. Modeling TCP latency. In *IEEE INFOCOM* (March 2000), vol. 1, pp. 1742–1751.
- [15] CASE, J., FEDOR, M., SCHOFFSTALL, M., E DAVIN, J. RFC 1157 - A Simple Network Management Protocol (SNMP), 1990. <http://rfc.net/rfc1157.html>.
- [16] CISCO. Cisco Netflow. <http://www.cisco.com/warp/public/732/Tech/nmp/netflow>.
- [17] CISCO. Cisco NetFlow FlowCollector with Network Data Analyzer. <http://www.cisco.com/univercd/cc/td/doc/pcat/24.htm>.
- [18] CLAFFY, K. C., BRAUN, H.-W., E POLYZOS, G. C. A parameterizable methodology for Internet traffic flow profiling. *IEEE Journal on Selected Areas in Communications* 13, 8 (October 1995), 1481–1494.
- [19] CROVELLA, M., TAQQU, M., E BESTAVROS, A. Heavy-tailed probability distributions in the world wide web. In *A Pratical Guide To Heavy Tails*. Chapman & Hall, New York, 1998, ch. 1, pp. 3–26.

- [20] DE LUCENA, S. C. *Modelos de Fluido para Tráfego Multimídia e Aplicações*. PhD thesis, COPPE/UFRJ, Rio de Janeiro, 2004.
- [21] DE SOUZA E SILVA, E., E LEÃO, R. M. M. The Tangram-II Environment. In *Proceedings of 11th International Conference TOOLS2000* (2000), pp. 366–369. LNCS 1786.
- [22] DE SOUZA E SILVA, E., LEÃO, R. M. M., TRINDADE, M. B., DA SILVA, A. P. C., RIBEIRO, B. F., DUARTE, F. P., E AZEVEDO, J. A. Uma metodologia de dimensionamento com QoS usando cadeias de Markov ocultas. In *XXI Simpósio Brasileiro de Telecomunicações* (Setembro 2004).
- [23] DOWNEY, A. B. The structural cause of file size distributions. In *SIGMETRICS: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (2001), ACM Press, pp. 328–329.
- [24] DUARTE, F. P. Previsão de Perdas utilizando Cadeias de Markov Escondidas. Tese de Mestrado, COPPE/UFRJ, 2002.
- [25] EATON, J. W. Octave, 2002. <http://www.octave.org/>.
- [26] FELDMANN, A., GILBERT, A. C., E WILLINGER, W. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. In *ACM SIGCOMM* (1998), ACM Press, pp. 42–55.
- [27] FIGUEIREDO, D. R., LIU, B., MISRA, V., E TOWSLEY, D. On the Auto-correlation Structure of TCP Traffic. *Computer Networks: Special Issue on Advances in Modeling and Engineering of Long-Range Dependent Traffic* 40, 3 (October 2002), 339–361.
- [28] FIROIU, V., BOUDEC, J. L., TOWSLEY, D., E ZHANG, Z. Theories and models for Internet quality of service. In *Proceedings of the IEEE* (May 2002).
- [29] FRALEIGH, C., DIOT, C., LYLES, B., MOON, S., OWEZARSKI, P., PAPA-GIANNAKI, K., E TOBAGI, F. Design and Deployment of a Passive Monitoring

- Infrastructure. In *Passive and Active Measurement Workshop* (Amsterdam, NL, April 2001).
- [30] FRALEIGH, C., MOON, S., LYLES, B., COTTON, C., KHAN, M., MOLL, D., ROCKELL, R., SEELY, T., E DIOT, C. Packet-level traffic measurements from the Sprint IP backbone. In *IEEE Network Magazine* (November 2003), vol. 17, pp. 6–16.
- [31] FRALEIGH, C., TOBAGI, F., E DIOT, C. Provisioning IP Backbone Networks to Support Latency Sensitive Traffic. In *IEEE INFOCOM* (San Francisco, USA, March 2003).
- [32] FULLMER, M. Flow-tools. <http://www.splintered.net/sw/flow-tools/>.
- [33] GROSSGLAUSER, M., E BOLOT, J.-C. On the relevance of long-range dependence in network traffic. In *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications* (August 1996), ACM Press, pp. 15–24.
- [34] GROUP, I. W. Internet Protocol Flow Information (IPFIX). <http://net.doit.wisc.edu/ipfix/>.
- [35] GUO, L., E MATTA, I. The War between Mice and Elephants. In *IEEE International Conference on Network Protocols* (November 2001).
- [36] HEYMAN, D. P., E LAKSHMAN, T. V. What are the implications of long-range dependence for VBR-video traffic engineering? *IEEE/ACM Trans. Netw.* 4, 3 (June 1996), 301–317.
- [37] HORVÁTH, A., E TELEK, M. A Markovian point process exhibiting multifractal behaviour and its application to traffic modeling. In *Proceedings of the 4th International Conference on Matrix-Analytic Methods in Stochastic models* (Adelaide, Australia, July 2002).
- [38] IANNACCONE, G., DIOT, C., GRAHAM, I., E MCKEOWN, N. Monitoring very high speed links. In *ACM SIGCOMM Internet Measurement* (San Francisco, USA, November 2001), pp. 267–272.

- [39] IETF. The Internet Engineering Task Force. <http://www.ietf.org/>.
- [40] INFORMATION SCIENCES INSTITUTE. The Network Simulator - NS-2. <http://www.isi.edu/nsnam/ns/>.
- [41] JAIN, R., E ROUTHIER, S. A. Packet Trains-Measurements and a New Model for Computer Network Traffic. *IEEE Journal on Selected Areas in Communications Sac-4*, 6 (September 1986), 986–995.
- [42] JAIN, R. K. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley and Sons, April 1991.
- [43] JUNIPER. Juniper Networks. <http://www.juniper.net/>.
- [44] KARAGIANNIS, T., FALOUTSOS, M., E MOLLE, M. A User-Friendly Self-Similarity Analysis Tool. In *ACM SIGCOMM - Special Section on Tools and Technologies for Networking Research and Education* (2003).
- [45] KARAGIANNIS, T., FALOUTSOS, M., E RIEDI, R. Long-Range Dependence: Now you see it now you don't! In *IEEE GLOBECOM* (November 2002).
- [46] KLEINROCK, L. *Queueing Systems*, vol. 1. Wiley-Interscience, 1975.
- [47] KLEINROCK, L. Creating a Mathematical Theory of Computer Networks. *INFORMS-Operations Research* 50, 1 (2002), 125–131.
- [48] LEINER, B. M., CERF, V. G., CLARK, D. D., KAHN, R. E., KLEINROCK, L., LYNCH, D. C., POSTEL, J., ROBERTS, L. G., E WOLFF, S. S. The past and future history of the Internet. *Communications of the ACM* 40, 2 (1997), 102–108.
- [49] LELAND, W. E., TAQQU, M. S., WILLINGER, W., E WILSON, D. V. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Netw.* 2, 1 (February 1994), 1–15.
- [50] M. MELLIA, I. STOICA, H. Z. TCP model for short lived flows. In *IEEE Communications Letters* (February 2002), vol. 6, pp. 85–87.

- [51] MATHIS, M., SEMKE, J., E MAHDAVI, J. The macroscopic behavior of the TCP congestion avoidance algorithm. In *ACM SIGCOMM* (July 1997), vol. 27, pp. 67–82.
- [52] MCCLOGHRIE, K., E ROSE, M. RFC 1213 - Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, 1991. <http://rfc.net/rfc1213.html>.
- [53] MELO, C. A. V., E FONSECA, N. L. S. Um estudo comparativo de processos envelope para modelagem de tráfego multifractal. In *Congresso da Sociedade Brasileira de Computação* (2003), pp. 1–10.
- [54] MUSCARIELLO, L., MELLIA, M., MEO, M., AJMONE, M., E CIGNO, R. L. An MMPP-based hierarchical model of Internet traffic. In *IEEE International Conference on Communications* (June 2004), vol. 4, pp. 2143–2147.
- [55] NAH, F. F. A Study on Tolerable Waiting Time: How Long Are Web Users Willing to Wait? *Behaviour and Information Technology* 23, 3 (2004), 153–163.
- [56] NGUYEN, H., THIRAN, P., E BARAKAT, C. On the correlation of TCP traffic in backbone networks. In *Proceedings of the International Symposium on Circuits and Systems* (May 2004), vol. 5, pp. 481–484.
- [57] NLANR. Passive Measurement and Analysis (PMA). <http://pma.nlanr.net/>.
- [58] OETIKER, T., E RAND, D. Multi Router Traffic Grapher - MRTG. <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>.
- [59] PADHYE, J., FIROIU, V., TOWSLEY, D. F., E KUROSE, J. F. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Trans. Netw.* 8, 2 (April 2000), 133–145.
- [60] PAPAGIANNAKI, D., TAFT, N., ZHANG, Z., E DIOT, C. Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models. In *IEEE INFOCOM* (April 2003).
- [61] PAPAGIANNAKI, K. *Provisioning IP Backbone Networks Based on Measurements*. PhD thesis, University of London, London, 2003.

- [62] PAPAGIANNAKI, K., TAFT, N., E DIOT, C. Impact of Flow Dynamics on Traffic Engineering Design Principles. In *IEEE INFOCOM* (March 2004).
- [63] PHILLIPS, J. R. ZunZun. <http://zunzun.com/>.
- [64] PLONKA, D. FlowScan: A Network Traffic Flow Reporting and Visualization Tool. In *Proceedings of the 14th Conference on Systems Administration (LISA)* (Berkley, CA, USA, December 2000), USENIX Association, pp. 305–318.
- [65] RABINER, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. In *IEEE* (February 1989), pp. 257–286.
- [66] RIEDI, R., E WILLINGER, W. Toward an Improved Understanding of Network Traffic Dynamics. In *Self-similar Network Traffic and Performance Evaluation*, Park e Willinger, Eds. Wiley, 2000, pp. 507–530.
- [67] RIEDI, R. H., CROUSE, M. S., RIBEIRO, V. J., E BARANIUK, R. G. A Multi-fractal Wavelet Model with Application to Network Traffic. *IEEE Transactions on Information Theory* 45, 4 (1999), 992–1018.
- [68] ROBERT, S., E BOUDEC, J.-Y. L. On a Markov modulated chain exhibiting self-similarities over finite timescale. *Performance Evaluation* 27-28 (1996), 159–173.
- [69] ROBERT, S., E BOUDEC, J.-Y. L. New models for pseudo self-similar traffic. *Performance Evaluation* 30 (1997), 57–68.
- [70] RTG. Real Traffic Grabber. <http://rtg.sourceforge.net/>.
- [71] RYU, B. K., E ELWALID, A. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications* (August 1996), ACM Press, pp. 3–14.
- [72] SCHATTMAN, J. The NonlinearProgramming Package. <http://www.adeptscience.co.uk/products/mathsim/maple/powertools/optimization/>.

- [73] SOMMER, R., E FELDMANN, A. NetFlow: information loss or win? In *ACM SIGCOMM* (Marseille, France, November 2002), pp. 173–174.
- [74] SOMMER, R., E FELDMANN, A. NetFlow: information loss or win? Relatório técnico, Saarland University Germany, 2002.
- [75] SOULE, A., SALAMATIA, K., TAFT, N., EMILION, R., E PAPAGIANNAKI, K. Flow Classification by Histograms: or How to Go on Safari in the Internet. In *ACM SIGMETRICS* (New York, USA, June 2004), vol. 32, pp. 49–60.
- [76] TIAN, X., WU, J., E JI, C. A Unified Framework for Understanding Network Traffic Using Independent Wavelet Models. In *IEEE INFOCOM* (2002).
- [77] WILLIAMS, T., E KELLEY, C. GNUPLOT, 2002. <http://gnuplot.sourceforge.net/>.
- [78] XUDONG WU NIKOLAIDIS, I. On the advantages of lifetime and RTT classification schemes for TCP flows. In *IEEE Computing, and Communications Conference* (April 2003), pp. 301–308.
- [79] ZHANG, Y., BRESLAU, L., PAXSON, V., E SHENKER, S. On the Characteristics and Origins of Internet Flow Rates. In *ACM SIGCOMM* (Pittsburgh, USA, August 2002).
- [80] ZHENG, D., LAZAROU, G. Y., E HU, R. A stochastic model for short-lived TCP flows. In *IEEE International Conference on Communications* (May 2003), vol. 26, pp. 76–81.