

**Avaliação e Simulação de Mecanismos de
Gerenciamento de Capacidade para
Suporte ao Tráfego Não Orientado a
Conexão na RDSI-FL**

Ronaldo Alves Ferreira

Dissertação de Mestrado

Avaliação e Simulação de Mecanismos de Gerenciamento de Capacidade para Suporte ao Tráfego Não Orientado a Conexão na RDSI-FL

Ronaldo Alves Ferreira¹

Julho de 1998

Banca Examinadora:

- Prof. Dr. Célio Cardoso Guimarães (Orientador)
- Prof. Dr. Maurício Ferreira Magalhães
Faculdade de Engenharia Elétrica e de Computação - UNICAMP
- Prof. Dr. Nelson Luís Saldanha da Fonseca
Instituto de Computação - UNICAMP
- Prof. Dr. Paulo Lício de Geus (suplente)
Instituto de Computação - UNICAMP

¹Trabalho financiado pelo CNPq.



UNIDADE	BC
N.º CHAMADA:	UNICAMP
	F413a
Ex.	
TOMBO BC/	34940
FE. S.	395/98
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	04/09/98
N.º CPJ	

CM-00116071-9

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Ferreira, Ronaldo Alves

F413a Avaliação e simulação de mecanismos de gerenciamento de capacidade para suporte ao tráfego não orientado a conexão na RDSI-FL / Ronaldo Alves Ferreira -- Campinas, [S.P. :s.n.], 1998.

Orientador : Célio Cardoso Guimarães

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Rede digital de serviços integrados. 2. Telecomunicações - Tráfego. 3. Fractais. 4. Redes de computação. I. Guimarães, Célio Cardoso. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Avaliação e Simulação de Mecanismos de Gerenciamento de Capacidade para Suporte ao Tráfego Não Orientado a Conexão na RDSI-FL

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Ronaldo Alves Ferreira e aprovada pela Banca Examinadora.

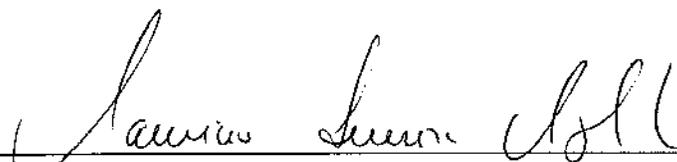
Campinas, 17 de Julho de 1998.



Prof. Dr. Célio Cardoso Guimarães
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Tese de Mestrado defendida e aprovada em 17 de julho de 1998
pela Banca Examinadora composta pelos Professores Doutores



Prof. Dr. Maurício Ferreira Magalhães



Prof. Dr. Nelson Luís Saldanha da Fonseca



Prof. Dr. Célio Cardoso Guimarães

© Ronaldo Alves Ferreira, 1998.
Todos os direitos reservados.

Resumo

Um grande desafio a ser superado para a implantação da Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) é o suporte ao serviço não orientado a conexão. Esse serviço serve de base para a construção de redes corporativas nas quais os equipamentos dos usuários podem estar separados por grandes distâncias. A dificuldade para que esse serviço seja disponibilizado está no gerenciamento eficiente da capacidade alocada às conexões que transportam o tráfego não orientado a conexão.

Há várias propostas na literatura de mecanismos para gerenciamento de capacidade de canais que transportam tráfego não orientado a conexão. As avaliações desses mecanismos são, geralmente, feitas através de simulações e o tráfego não orientado a conexão é tradicionalmente representado pelo modelo de tráfego *On-Off* com períodos de transmissão (*On*) e de silêncio (*Off*) representados por variáveis aleatórias independentes com distribuição exponencial. Trabalhos recentes, entretanto, mostram que o comportamento do tráfego gerado em uma rede local, ou até mesmo o tráfego que sai de uma rede local para uma rede de longa distância, é melhor representado por processos auto-semelhantes.

Os objetivos deste trabalho são: apresentar uma visão geral dos diversos mecanismos de gerenciamento de capacidade existentes, estudar as principais características do modelo de tráfego auto-semelhante, avaliar através de simulações o comportamento dos mecanismos de gerenciamento de capacidade quando alimentados com tráfego auto-semelhante e comparar os resultados com os resultados produzidos pelo modelo tradicional.

Abstract

A great challenge in the deployment of B-ISDN is the support of connectionless services. This kind of service can be used to build wide area virtual private networks. The difficulty in the deployment of this service is the efficient management of the bandwidth assigned to connections which transport connectionless traffic.

There are several bandwidth management mechanisms for connectionless traffic support in the literature. These mechanisms are usually evaluated by simulation and the connectionless traffic is traditionally represented by the On-Off model where the transmission (On) and silence (Off) periods are exponentially distributed independent random variables. Recent works, however, show that LAN traffic, or even the traffic sent to an internet, is best represented by self-similar processes.

The goals of this work are: to present an overview of bandwidth management mechanisms, to study the main aspects of self-similar traffic model and to evaluate by simulations how the different bandwidth management mechanisms for connectionless traffic support react to self-similar traffic and compare the results with results produced by the traditional model.

À minha querida mãe.

Agradecimentos

Ao CNPq pelo suporte financeiro.

Ao meu orientador, Prof. Célio Guimarães, pela paciência, pela confiança depositada e pelos valiosos comentários que muito contribuíram para a conclusão deste trabalho.

Aos colegas do Departamento de Computação e Estatística da Universidade Federal de Mato Grosso do Sul pela excelente convivência nos últimos anos e pelo apoio para que eu pudesse concluir este trabalho.

Aos inúmeros pesquisadores brasileiros e estrangeiros que gentilmente enviaram-me seus trabalhos via Internet ou de forma impressa. Em particular, agradeço a José Augusto Suruagy Monteiro, Mark Garret, Vitorio Trecordi, Murad Taqqu e Walter Willinger.

Ao amigo Marcelo pelos incentivos constantes para que eu terminasse este trabalho.

À minha família por todo apoio recebido.

Ao Valdeir que é muito mais que um irmão, é o pai que conheci e que guiou boa parte dos meus passos.

Conteúdo

1	Introdução	1
1.1	Motivação e Objetivos	1
1.2	Evolução das Redes de Telecomunicações	3
1.2.1	Rede Digital de Serviços Integrados	5
1.2.2	Comutação Rápida de Pacotes	6
1.3	Organização da Dissertação	8
2	Rede Digital de Serviços Integrados de Faixa Larga	9
2.1	Características Técnicas da RDSI-FL	9
2.2	Modo de Transferência Assíncrono - ATM	12
2.3	Capacidades de Controle de Congestionamento da RDSI-FL	17
2.4	Classificação dos Esquemas de Controle de Congestionamento	20
2.5	Abordagens Utilizadas no Controle de Congestionamento	21
2.5.1	Mecanismos Preventivos	22
2.5.2	Mecanismos Reativos	25
2.5.3	Controle de Admissão	26
2.6	Suporte a Serviços Não Orientados a Conexões	28
3	Mecanismos de Gerenciamento de Capacidade	31
3.1	Definição do Problema	32
3.1.1	Critérios de Avaliação	33
3.2	Mecanismo de Reservas	35
3.2.1	Mecanismo para Monitoramento de Aumento do Tráfego	36
3.2.2	Mecanismo para Monitoramento de Diminuição do Tráfego	38
3.2.3	Avaliação	39

3.3	Mecanismo de Anúncio de Capacidade	40
3.3.1	Esquemas de Anúncio de Capacidade	42
3.3.2	Controle de Acesso	42
3.3.3	Avaliação	43
3.4	Mecanismo Baseado em Estatística de Atraso Fim a Fim	45
3.4.1	Estimativa da Capacidade Disponível	45
3.4.2	Controle de Solicitação e Liberação de Capacidade	47
3.4.3	Avaliação	47
3.5	Protocolo de Reserva Rápida de Buffer	47
3.5.1	Avaliação	50
4	Modelos de Tráfegos	52
4.1	Modelo Genérico de Tráfego para ATM	55
4.2	Modelo Auto-Semelhante	57
4.2.1	Propriedades dos Processos Auto-Semelhantes	59
4.2.2	Métodos Estatísticos para Teste de Auto-Semelhança	62
4.2.3	Processos Auto-Semelhantes	77
4.2.4	Geração de <i>Traces</i> de Processos Auto-Semelhantes	78
4.2.5	Método da FFT	79
4.2.6	Algoritmo RMD	82
5	Ambiente de Simulação	86
5.1	Modelo de Simulação e Ambiente de Implementação	86
5.2	Descrição dos Componentes	87
5.2.1	Parâmetros da Simulação	89
5.2.2	Comutador ATM	89
5.2.3	Enlace	92
5.2.4	Gateway	93
5.2.5	Aplicação	97
6	Comparação dos Mecanismos	102
6.1	Resultados Preliminares	102
6.2	Modelo de Configuração da Rede e Parâmetros Adotados	109
6.3	Resultados Obtidos	114

7 Conclusões e Trabalhos Futuros	124
7.1 Resumo e Contribuições	124
7.2 Extensões e Trabalhos Futuros	125
Bibliografia	133
A Transformações Lineares e o Parâmetro de Hurst	134
B Dependência em Longos Intervalos do FGN	137
C Linguagem para Entrada de Dados do Simulador	139
C.1 Especificação Formal	139

Lista de Figuras

1.1	Principais hierarquias de sinais digitais	4
1.2	Hierarquia de sinais SONET/SDH	4
1.3	Utilização da capacidade do enlace pelo STM.	6
1.4	Comutação usando pacotes com tamanhos variáveis.	7
1.5	Comutação usando células.	7
2.1	Classificação dos serviços de faixa larga da RDSI-FL.	10
2.2	Classes de serviços da RDSI-FL.	11
2.3	Utilização da capacidade do enlace pelo ATM.	13
2.4	Estrutura da célula ATM.	14
2.5	Formato do cabeçalho da célula ATM na interface Usuário-Rede.	15
2.6	Formato do cabeçalho da célula ATM na interface Rede-Rede.	15
2.7	Modelo de referência dos protocolos da RDSI-FL.	16
2.8	Configuração recomendada pelo ITU-T para o controle de tráfego na RDSI-FL.	18
2.9	Classificação por escala de tempo dos esquemas de controle de congestionamento.	21
2.10	Suporte indireto a serviços não orientados a conexões.	29
2.11	Suporte direto a serviços não orientados a conexões.	30
3.1	Interconexão de redes locais e metropolitanas através da RDSI-FL.	32
3.2	Dimensionamento dos Limiares no Buffer de Recepção.	38
3.3	Funcionalidade do Mecanismo de Anúncio de Capacidade	43
3.4	Esquema de policiamento do tráfego com dois baldes furados em série.	44
3.5	Mecanismo de Reserva de Buffer	49
4.1	Tráfego Ethernet coletado na <i>Bellcore</i> em diferentes escalas de tempo.	54

4.2	Níveis de Tráfegos em ATM.	55
4.3	Tráfego em Rajada e Silêncio.	56
4.4	Autocorrelações para os dados gerados no experimento.	64
4.5	Autocorrelações para os dados Ethernet.	65
4.6	Espectro de potências estimado do tráfego Poisson gerado no experimento.	66
4.7	Espectro de potências estimado do tráfego Ethernet.	67
4.8	Gráfico $\log_{10}(Var[X^{(m)}])$ x $\log_{10}(m)$ para o tráfego Poisson gerado no experimento. A inclinação dos pontos calculados é igual a -1 o que resulta em uma estimativa de H igual a 0,5.	68
4.9	Gráfico $\log_{10}(Var[X^{(m)}])$ x $\log_{10}(m)$ para o tráfego Ethernet. A inclinação dos pontos calculados é igual a -0,36 o que resulta em uma estimativa de H igual a 0,82.	69
4.10	<i>Pox Diagram</i> ($\log_{10}(R/S)$ x $\log_{10}(d)$) para o tráfego Ethernet. A inclinação da reta ajustada aos pontos resulta em uma estimativa para H igual a 0,81.	72
4.11	<i>Pox Diagram</i> ($\log_{10}(R/S)$ x $\log_{10}(d)$) para o tráfego Poisson gerado no experimento. A inclinação da reta ajustada aos pontos resulta em uma estimativa para H igual a 0,5.	73
4.12	Periodograma do tráfego Poisson gerado no experimento. $H = 0,5$ estimado pela reta ajustada a 10% das menores frequências.	75
4.13	Periodograma do tráfego Ethernet. $H = 0,81$ estimado pela reta ajustada a 10% das menores frequências.	76
4.14	Seqüência de dados de um processo FGN gerada pelo Método da FFT. Os parâmetros utilizados foram $n = 512$ e $H = 0,5$; $H = 0,7$ e $H = 0,9$	81
4.15	Primeiros 4 estágios do algoritmo RMD.	84
5.1	Definição de um nó do grafo de simulação.	88
5.2	Parâmetros globais da simulação.	90
5.3	Especificação de um comutador ATM.	91
5.4	Especificação dos enlaces de uma rede a ser simulada.	92
5.5	Especificação de um gateway.	94
5.6	Especificação de uma fonte VBR.	98
6.1	Taxa de perda de células ocasionada pelo tráfego Ethernet e pelo tráfego gerado sinteticamente pelo método da FFT.	105

6.2	Taxa de perda de células utilizando o método da FFT	106
6.3	Taxa de perda de células utilizando o algoritmo RMD	107
6.4	Atraso médio enfrentado pelas células no buffer do comutador ATM. O tamanho do buffer foi fixado em 200 células.	108
6.5	Configuração de rede para a simulação	109
6.6	Modelo simplificado da rede para a simulação	110
6.7	Fonte <i>On-Off</i>	110
6.8	Taxa de perda de células em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,5.	116
6.9	Taxa de perda de células em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,85.	117
6.10	Taxa de perda de células do mecanismo de reservas para os valores de H iguais a 0,5 e 0,85.	118
6.11	Taxa de perda de células do mecanismo de anúncio de capacidades híbrido para os valores de H iguais a 0,5 e 0,85.	118
6.12	Taxa de perda de células do mecanismo de anúncio de capacidades <i>on-the-fly</i> para os valores de H iguais a 0,5 e 0,85.	119
6.13	Atraso médio enfrentado pelo tráfego principal em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,85.	120
6.14	Atraso médio enfrentado pelo tráfego de retaguarda em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,85.	120
6.15	Atraso médio enfrentado pelo tráfego principal no mecanismo de reservas para os valores de H iguais a 0,5 e 0,85.	121
6.16	Atraso médio enfrentado pelo tráfego principal no mecanismo de anúncio de capacidades híbrido para os valores de H iguais a 0,5 e 0,85.	122
6.17	Atraso médio enfrentado pelo tráfego principal no mecanismo de anúncio de capacidades <i>On-the-fly</i> para os valores de H iguais a 0,5 e 0,85.	122
6.18	Atraso médio enfrentado pelo tráfego de retaguarda no mecanismo de anúncio de capacidades híbrido para os valores de H iguais a 0,5 e 0,85.	123
6.19	Atraso médio enfrentado pelo tráfego de retaguarda no mecanismo de anúncio de capacidades <i>On-the-fly</i> para os valores de H iguais a 0,5 e 0,85.	123

Lista de Tabelas

6.1	Valores estimados de H dos algoritmos FFT e RMD.	103
6.2	Parâmetros utilizados para o dimensionamento dos três mecanismos simulados.	115

Capítulo 1

Introdução

O objetivo deste trabalho é apresentar um estudo comparativo de mecanismos de gerenciamento de capacidade para suporte ao tráfego não orientado a conexão na Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL).

A seção 1.1 apresenta a motivação para a realização deste trabalho e os objetivos a serem atingidos. Inicialmente, serão apresentados vários termos sem suas definições, uma vez que eles serão definidos e detalhados nos capítulos subseqüentes. A seção 1.2 apresenta uma breve evolução das redes de telecomunicações visando introduzir a Rede Digital de Serviços Integrados (RDSI) e servir como base para o estudo do capítulo 2. A seção final mostra como está organizada esta dissertação.

1.1 Motivação e Objetivos

Existe uma clara tendência para que o ambiente de comunicação a altas velocidades do futuro seja heterogêneo. Várias tecnologias de redes operando a velocidades na ordem de 100Mbps têm surgido e começam a ser implantadas nos mais variados ambientes de trabalho, sejam eles acadêmicos, comerciais ou industriais. No âmbito de redes locais, tecnologias como FDDI e Fast Ethernet já são realidades. No âmbito de redes de longa distância, o ITU-T e o Forum ATM estão trabalhando intensamente para definir padrões para uma nova rede que visa integrar diferentes mídias de comunicação e que opere a taxas na ordem de centenas de megabits e até mesmo gigabits, a Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL). Espera-se que a RDSI-FL venha a fazer a interligação desses diferentes ambientes de forma eficiente, possibilitando, assim, a implantação de

redes corporativas mesmo com subredes separadas por grandes distâncias. O maior problema para a realização dessa tarefa é que a RDSI-FL terá de transportar o tráfego não orientado a conexão, proveniente das redes locais, no Modo de Transmissão Assíncrono (ATM - *Asynchronous Transfer Mode*), sistema de transmissão orientado a conexão que foi escolhido para sua implementação. Dispositivos apropriados (*gateways*) serão necessários para fazer a interligação dos dois tipos de ambientes.

Como discutido em [Par90], os *gateways* entre ambientes orientados a conexões e ambientes não orientados a conexões podem variar, em termos de complexidade, de esquemas simples e ineficientes, que estabelecem e liberam conexões para cada datagrama enviado, a esquemas complexos, que estabelecem uma conexão para um datagrama e a mantém por um determinado período de tempo tentando antecipar a transmissão de futuros datagramas. Este último esquema requer um mecanismo inteligente de temporização para o estabelecimento e a liberação das conexões. Para fazer a interface entre as diferentes tecnologias de redes locais e a RDSI-FL, os *gateways*, entre esses dois ambientes, também encontrarão esse problema e terão de decidir quando e o quanto alocar de capacidade nas conexões que transportarão o tráfego não orientado a conexão dentro da RDSI-FL, ou seja, eles terão de gerenciar a capacidade utilizada na RDSI-FL para suporte ao tráfego não orientado a conexão.

Vários mecanismos foram propostos, na literatura, para gerenciar a capacidade alocada às conexões na RDSI-FL para o transporte de tráfego não orientado a conexão [MFT91, GTG93, FM92, Tur92]. Os esquemas de alocação de capacidade nesses mecanismos variam significativamente de uma proposta para outra, utilizando diferentes idéias. Entretanto, o que todos eles procuram é utilizar de maneira eficiente os recursos da RDSI-FL.

Para a análise de suas propostas, os autores apresentam resultados baseados em simulações e assumem hipóteses ultrapassadas para o modelo de tráfego utilizado para modelar o tráfego não orientado a conexão. A hipótese normalmente assumida para o tráfego que alimenta as simulações é que o tempo entre chegadas e os comprimentos dos pacotes são variáveis aleatórias independentes e com distribuição exponencial, ou seja, a chegada de pacotes constitui um processo de Poisson. Trabalhos recentes [LTWW94, PF95], entretanto, argumentam de forma convincente que o tráfego gerado em uma rede local é melhor representado por processos auto-semelhantes (*self-similar*), ou seja, esse tipo de tráfego é *fractal* por natureza.

O objetivo deste trabalho é fazer uma análise, baseada em simulações, do compor-

tamento de alguns mecanismos de gerenciamento de capacidade para suporte ao tráfego não orientado a conexão na RDSI-FL quando alimentados com um modelo de tráfego auto-semelhante e também explorar as principais características desse tipo de tráfego.

A seguir será feito um breve resumo de como as redes de telecomunicações evoluíram nos últimos anos até chegar na RDSI-FL.

1.2 Evolução das Redes de Telecomunicações

Para se comunicar o homem sempre procurou desenvolver novos métodos e novas tecnologias. Ao surgir um novo dispositivo de comunicação iniciava-se uma busca de métodos de conexão desses novos dispositivos. Com isso surgiram diferentes redes de comunicação, redes especializadas, com o objetivo de interligar dispositivos semelhantes. Exemplos típicos de redes especializadas são as redes de telefonia, de telex e de transmissão de dados.

A rede de comunicação que mais se desenvolveu e está presente no dia a dia de quase todo ser humano é a rede de telefonia. Com ela é possível se comunicar com pessoas localizadas em, praticamente, qualquer lugar do globo terrestre. Entretanto, países diferentes utilizam tecnologias e formas de acessos diferentes em suas redes. Desde a digitalização, surgiram várias propostas para se organizar a transmissão em redes de telecomunicações.

As transmissões digitais nas redes de telecomunicações estão estruturadas segundo uma hierarquia de taxas de transmissão. Esta hierarquia não é única no mundo, havendo diferenças entre as taxas de acordo com o país. As três principais hierarquias são a americana, a japonesa e a européia, figura 1.1. O Brasil utiliza a hierarquia européia.

Com a utilização de taxas diferentes de acessos, a comunicação entre países que utilizam hierarquias diferentes fica prejudicada devido ao processamento adicional para a conversão de taxas. Tendo como motivação esta incompatibilidade, a *Bellcore* (*Bell Communications Research*) propôs a SONET (*Synchronous Optical Network*) [BC89], uma nova hierarquia de sinais que visa unificar as comunicações e também acrescentar taxas maiores de acesso. Usando a SONET como base, o então CCITT (*Comité Consultatif Télégraphique et Téléphonique*), hoje ITU-T (*International Telecommunication Union - Telecommunication Standardization Sector*), padronizou a Hierarquia Digital Síncrona (SDH - *Synchronous Digital Hierarchy*) [KMU90] para tentar unificar as comunicações no mundo. A figura 1.2 apresenta as taxas de acesso segundo a hierarquia SONET/SDH.

A popularização dos computadores, presentes cada vez mais em pequenos negócios e

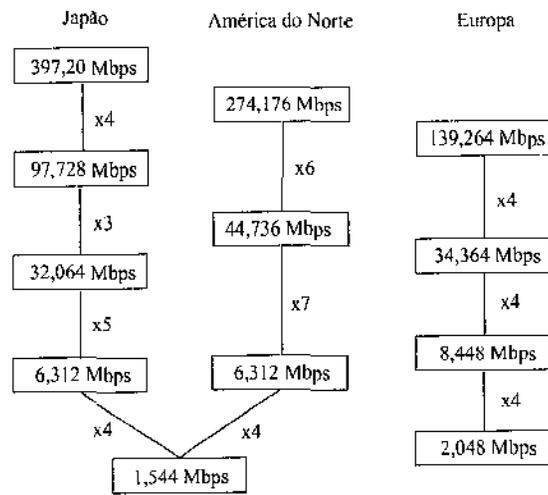


Figura 1.1: Principais hierarquias de sinais digitais

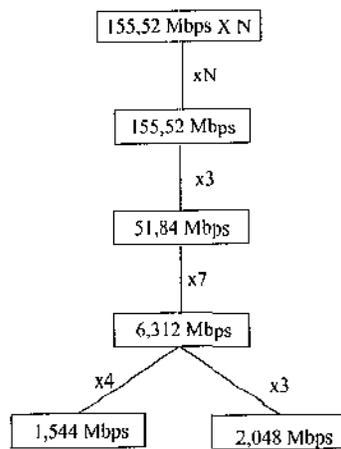


Figura 1.2: Hierarquia de sinais SONET/SDH

em residências, deu início à busca de formas de se fazer com que eles se comunicassem. O que se percebeu há algum tempo atrás foi a crescente utilização de dispositivos para conexão de computadores a rede de telefonia (*modems*), aumentando significativamente o tráfego de dados nesta rede. Esse fato motivou o CCITT, nos meados dos anos 70, a trabalhar na padronização de uma rede que unificasse as comunicações entre dispositivos heterogêneos. A nova rede, batizada de Rede Digital de Serviços Integrados (RDSI), deveria dar acesso a diferentes dispositivos de comunicação de forma transparente, através de uma interface de acesso unificada.

1.2.1 Rede Digital de Serviços Integrados

Na sua fase inicial de implantação a RDSI ofereceria apenas interfaces padronizadas, utilizando, entretanto, redes dedicadas para oferecer serviços distintos. A idéia era prover uma tomada universal de informação [Mon90], assim como a oferecida pela rede de energia elétrica, de modo que dispositivos completamente distintos, como por exemplo telefone, telex, computador, etc., pudessem se conectar diretamente à rede.

Inicialmente foram propostas duas interfaces de acesso à RDSI: a interface básica com dois canais B de comunicação de 64 Kbps e um canal D de sinalização de 16 Kbps ($2B + D$); e a interface primária com n canais B de comunicação ($n = 23$ nos EUA e $n = 30$ na Europa e Japão) e um canal D de sinalização de 64 Kbps ($nB + D$).

As transmissões na RDSI são implementadas utilizando o Modo de Transferência Síncrono (STM - *Synchronous Transfer Mode*). O STM, para transmitir informações, utiliza um quadro de bits que se repete periodicamente. Para se obter um canal de comunicação no STM é necessário reservar alguns bits dentro desse quadro. O canal é então alocado em uma posição fixa dentro do quadro, possuindo tantos bits quantos forem necessários para se atingir a taxa de transmissão desejada. A figura 1.3 ilustra a utilização da capacidade do enlace pelo STM.

O que pode ser percebido no sistema de transmissão do STM é que, mesmo que não haja dados a serem transmitidos por um determinado canal, a capacidade do canal está reservada e não pode ser utilizada por novos canais ou por outros já estabelecidos. Portanto, o STM é bastante útil para se garantir a transmissão de dados contínuos (por exemplo, voz), mas pouco vantajoso para as transmissões que alternam períodos de silêncio com períodos de transmissão intensa. Entretanto, o que se tem percebido nos últimos tempos é o surgimento de aplicações com características completamente distintas das

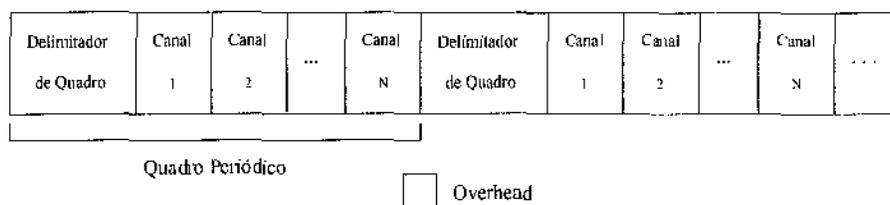


Figura 1.3: Utilização da capacidade do enlace pelo STM.

características da transmissão de voz.

Aplicações como teleconferência, em que os sinais de vídeo e som são compactados alternando períodos de transmissão intensa com períodos de baixa transmissão de dados, estão ficando cada vez mais comuns. Isso tem motivado a procura de novos métodos de transmissão de dados. Uma das idéias é utilizar a comutação por pacotes para se ter o ganho da multiplexação estatística, mas retirar o máximo possível de informações de controle dos protocolos para possibilitar um processamento rápido das unidades de transmissão. A essa nova técnica dá-se o nome de *Comutação Rápida de Pacotes* [Par94].

1.2.2 Comutação Rápida de Pacotes

Segundo Turner [Tur86], dois fatores estão dirigindo as telecomunicações para a comutação rápida de pacotes. O primeiro fator são as diferentes necessidades dos novos serviços de comunicação, tais como comunicações multiponto e multimídia, e comunicação de áudio e vídeo de alta qualidade. Para atingir esses novos requisitos, a rede tem de ter algumas capacidades primárias tais como sinalização multiponto, alocação dinâmica de recursos da rede, sincronização entre cadeias de informações multimídia, etc.

O segundo fator é o desenvolvimento de novas tecnologias. O custo decrescente e o aumento da capacidade de fibras ópticas tornam real a comunicação a altas velocidades. O desenvolvimento de circuitos integrados e tecnologias de computadores oferecem a possibilidade de produzir sistemas de comutação rápida de pacotes.

Comutação por Células

Uma das tendências importantes na comutação rápida de pacotes é o interesse crescente em uma nova tecnologia conhecida como comutação por células. A premissa na comutação por células é que todos os dados devem ser transmitidos em *pequenos pacotes de tamanhos*

fixos, chamados *células*.

Para se definir o tamanho de uma célula algumas questões técnicas devem ser consideradas. Se as células forem muito pequenas a informação de controle deve ser minimizada, para se evitar o excesso de *overhead*. A utilização de células pequenas, entretanto, reduz o desperdício de capacidade, já que o número de bytes sem informações úteis é pequeno.

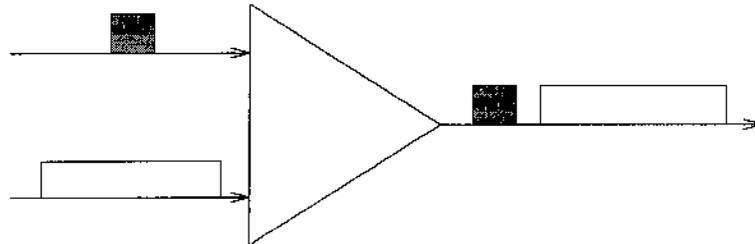


Figura 1.4: Comutação usando pacotes com tamanhos variáveis.

Um dos objetivos no desenvolvimento de novas tecnologias de comunicação é permitir que diferentes tráfegos (voz, vídeo e dados), com diferentes características e diferentes necessidades, sejam transmitidos em uma mesma rede. Em uma rede em que há mistura de tráfegos, alguns tipos de tráfegos necessitarão que seus dados sejam entregues dentro de limites severos de tempo. A escolha de um tamanho pequeno para a célula pode ajudar a minimizar os atrasos na rede (atrasos de serialização) e com isso garantir os tempos de entrega. A figura 1.4 dá uma idéia do que pode ocorrer em uma rede com pacotes grandes. No caso, se o pacote pequeno estiver transportando dados de tempo real (por exemplo, voz), muito provavelmente, ele chegará atrasado ao seu destino, pelo fato de ter chegado no comutador alguns instantes depois de um pacote grande. Isso já não ocorre no exemplo da figura 1.5, porque o pacote grande foi quebrado em várias células de menor tamanho.

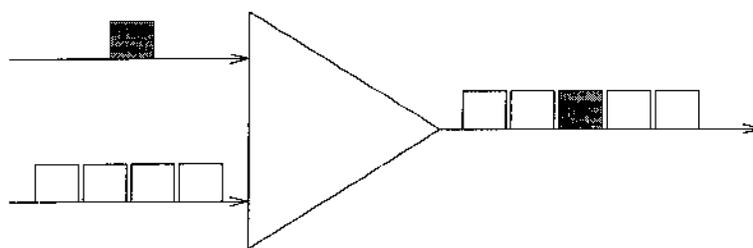


Figura 1.5: Comutação usando células.

No próximo capítulo será estudado um método específico de comutação por células que foi padronizado pelo ITU-T para a RDSI-FL. Nessa padronização, a célula foi definida com um tamanho de 53 bytes, sendo 5 bytes para o cabeçalho e 48 para os dados. Para uma discussão mais detalhada sobre as vantagens de se utilizar células para o transporte de informações em redes de alta velocidade veja [Par94].

1.3 Organização da Dissertação

Este trabalho está dividido da seguinte maneira:

- O capítulo 2 apresenta as principais características da RDSI-FL e do ATM, enfocando, principalmente, os mecanismos de controle de tráfego e de congestionamento propostos na literatura.
- O capítulo 3 apresenta e avalia qualitativamente as diversas soluções propostas na literatura para o problema de gerenciamento de capacidade para o transporte de dados não orientados a conexões na RDSI-FL.
- O capítulo 4 discute o problema da modelagem de tráfego de dados para estudos de redes de computadores. A ênfase no capítulo é dada ao entendimento dos conceitos e das principais características presentes no modelo, recentemente proposto, auto-semelhante.
- O capítulo 5 descreve um ambiente de simulação desenvolvido para avaliar o desempenho quantitativo dos mecanismos que serão estudados no capítulo 3.
- O capítulo 6 faz uma avaliação quantitativa de alguns mecanismos de gerenciamento de capacidade estudados no capítulo 3. Além disso, o capítulo apresenta alguns resultados preliminares relacionados à geração de *traces* de tráfego auto-semelhante e à influência desse tipo de tráfego em uma rede ATM.
- O capítulo 7 resume as principais contribuições e sugere algumas extensões futuras deste trabalho.

Capítulo 2

Rede Digital de Serviços Integrados de Faixa Larga

Antes de serem apresentados mecanismos específicos de gerenciamento de capacidade para suporte ao tráfego não orientado a conexão para a RDSI-FL, este capítulo apresentará detalhes a respeito do ATM e da RDSI-FL que são pertinentes a este trabalho. Vários aspectos relacionados ao controle de congestionamento e ao controle de tráfego na RDSI-FL serão apresentados.

Uma introdução geral à RDSI-FL é apresentada na seção 2.1, seguida por uma introdução ao ATM na seção 2.2. As capacidades de controle de congestionamento previstas para a RDSI-FL são apresentadas na seção 2.3. Ao se projetar um esquema de controle de congestionamento é essencial considerar a escala de tempo em que o esquema irá atuar. Na seção 2.4 é apresentada uma classificação de esquemas de controle de congestionamento baseada na escala de tempo em que o controle atua. A seção 2.5 apresenta um estudo de algumas abordagens propostas na literatura para o controle de congestionamento em redes de alta velocidade. Um tema de particular interesse para este trabalho é o suporte ao serviço não orientado a conexão na RDSI-FL; a seção 2.6 apresenta as duas propostas do ITU-T para o oferecimento desse serviço.

2.1 Características Técnicas da RDSI-FL

As características técnicas da RDSI-FL podem ser divididas logicamente em quatro grupos:

- Os serviços suportados pela RDSI-FL e seus impactos nas necessidades da rede.
- As camadas de protocolos necessárias para construir uma RDSI-FL para suportar os serviços definidos para a rede.
- A natureza da interface Usuário-Rede (UNI - *User Network Interface*).
- Os aspectos de controle de redes necessários para permitir que os protocolos e a UNI suportem os serviços RDSI-FL. Isso inclui sinalização, gerenciamento de recursos, desempenho, operação e manutenção.

Os aspectos de serviços da RDSI-FL abrangem uma grande variedade de serviços, como voz, texto, imagem, dados e vídeo. Para simplificar esses serviços, o ITU-T os classificou em tipos diferentes como mostrado na figura 2.1 [IT93a]. Além dos serviços interativos bi-direcionais disponíveis nas redes de telecomunicações atuais, a RDSI-FL também suportará vários serviços distributivos. Isso permitirá que a RDSI-FL transporte os serviços de vídeo oferecidos atualmente pelas redes de televisão a cabo. O serviço de televisão a cabo é um exemplo de serviço distributivo sem controle de apresentação pelo usuário. Vídeo sob demanda representa um potencial serviço distributivo com controle de apresentação pelo usuário.

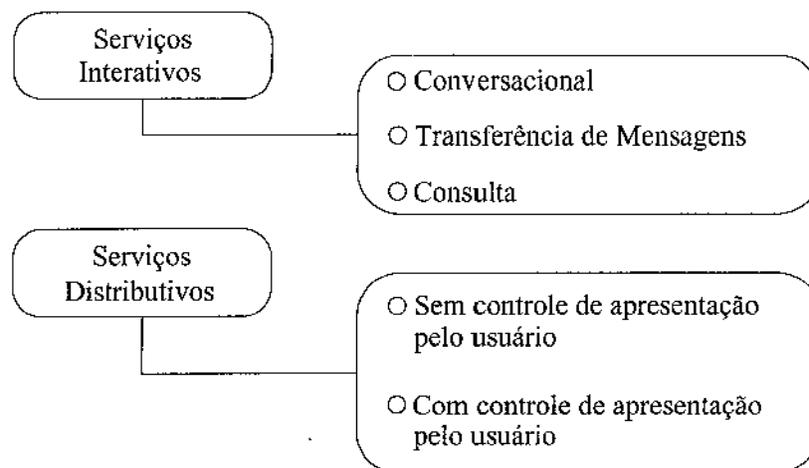


Figura 2.1: Classificação dos serviços de faixa larga da RDSI-FL.

Para determinar a associação com as capacidades de rede, o ITU-T definiu duas categorias de serviços transportadores para a RDSI-FL: serviço transportador orientado a conexão e serviço transportador não orientado a conexão.

- **Serviços orientados a conexões**

Esses serviços têm como característica a exigência de que uma conexão seja estabelecida antes do início da comunicação. A transmissão de vídeo a taxa variável de bits e a transmissão de voz são exemplos de serviços orientados a conexões.

- **Serviços não orientados a conexões**

A natureza desses serviços permite que a conexão entre os usuários seja estabelecida e liberada ao mesmo tempo em que a informação é transmitida. Exemplo deste serviço é a interconexão de redes locais.

O ITU-T recomendou a utilização de quatro classes de serviços [Mon94] com base em três critérios: exigência de sincronismo de tempo entre fonte e destino, taxa de bits (constante ou variável) e o modo de conexão (orientado ou não a conexão). A figura 2.2 mostra as quatro classes recomendadas e também suas características.

Classes Critérios	Classe A	Classe B	Classe C	Classe D
Tempo entre Fonte e Destino	Relacionado		Sem Relação	
Taxa de bits	Constante	Variável		
Modo de Conexão	Orientado a conexão			Sem Conexão

Figura 2.2: Classes de serviços da RDSI-FL.

- **Classe A:** Serviço orientado a conexão com taxa constante de bits ou emulação de circuito.

As taxas de bits podem variar desde alguns Kbps a vários Mbps. Os serviços nesta classe serão semelhantes aos serviços de voz e vídeo oferecidos atualmente pelas operadoras telefônicas e pelas operadoras de televisão a cabo respectivamente. Esses serviços requerem sincronização de tempo entre fonte e destino e será transmitido pelo serviço transportador orientado a conexão da RDSI-FL.

- **Classe B:** Serviços com taxas variáveis de bits e com sincronização de tempo entre emissor e receptor.

Os serviços nesta classe permitirão que a RDSI-FL transporte serviços de tempo real tais como voz e vídeo sobre conexões com taxas variáveis de bits. A principal razão para isso é tirar vantagem da capacidade de multiplexação estatística do ATM. Esta classe de serviços também será transportada utilizando o serviço transportador orientado a conexão.

- **Classe C:** Serviços de dados orientados a conexões.

Serviços de dados que não requerem sincronização de tempo entre fonte e destino. Esses serviços, como o X.25 atual, suportarão tráfegos a taxas variáveis de bits. Os serviços da classe C serão transmitidos pelo serviço transportador orientado a conexão da RDSI-FL.

- **Classe D:** Serviços de dados não orientados a conexões.

Estes serviços, como os atuais serviços de datagramas, suportarão tráfego a taxas variáveis de bits que não requerem o estabelecimento prévio de uma conexão entre fonte e destino. Estes serviços serão transmitidos pelo serviço transportador não orientado a conexão da RDSI-FL.

2.2 Modo de Transferência Assíncrono - ATM

ATM define um serviço de comutação rápida de pacotes orientado a conexão baseado na transmissão de pequenos pacotes de tamanhos fixos denominados células [IT92, Bou92, Min89]. Pelo fato do ATM ser baseado na comutação de pacotes, ele pode suportar uma grande variedade de velocidades de enlaces e permite que enlaces de diferentes velocidades possam se comunicar, um aspecto importante em expansão de sistemas. Além disso, uma outra vantagem do ATM é o ganho oferecido pela multiplexação estatística que possibilita que capacidades disponíveis nos enlaces possam ser utilizadas por outros serviços, diferente do STM que só possibilita canais de tamanhos fixos e não alocáveis a outros serviços. A figura 2.3 mostra a utilização da capacidade de um enlace pelo ATM. Interfaces físicas para o ATM atualmente definidas variam de 45 Mbps a 622 Mbps. Estudos para interfaces com velocidades superiores a 2 Gbps estão atualmente em andamento.

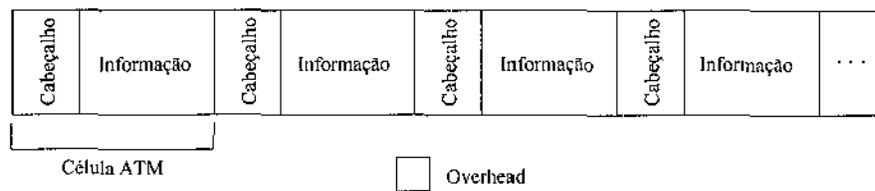


Figura 2.3: Utilização da capacidade do enlace pelo ATM.

O padrão ATM é também aplicável a redes locais e privadas. As tecnologias atuais de redes apresentam grandes diferenças tecnológicas no interfaceamento entre as redes locais e as redes de longa distância. Computadores ATM para redes locais já oferecem uma solução eficiente e barata para o oferecimento de altas velocidades e também para a interconexão com as redes públicas [SJ95].

Uma conexão ATM pode ser *comutada*, *semi-permanente* ou *permanente*. Para o estabelecimento de chamadas comutadas, são necessários procedimentos de sinalização. *Sinalização de rede* é o processo de transferência de informação de controle entre componentes de uma rede de comunicação para estabelecer, manter, e liberar conexões e também a transferência de informações de gerenciamento. As conexões semi-permanentes e permanentes são estabelecidas por procedimentos administrativos [Bou92].

Cada conexão ATM opera sob certa qualidade de serviço (QOS - *Quality of Service*). A qualidade de serviço de uma conexão em particular é caracterizada por parâmetros tais como: taxa de perda de células (CLR - *Cell Loss Rate*), atraso de transferência de célula (CTD - *Cell Transfer Delay*), variação de atraso de célula (CDV - *Cell Delay Variation*), tolerância de rajada (BT - *Burst Tolerance*), taxa de pico de células (PCR - *Peak Cell Rate*) e taxa média de células (SCR - *Sustained Cell Rate*). A qualidade de serviço é negociada entre o usuário e a rede quando a conexão é estabelecida. O ATM opera na base do maior empenho possível, ou seja, ele não garante que todas as células serão entregues, células com erros ou que encontrem congestionamento são descartadas sem aviso aos participantes da conexão. É responsabilidade das camadas superiores da pilha de protocolos perceber a ausência das células descartadas e, se necessário, retransmitir as células perdidas. O ATM, entretanto, garante a seqüência das células, ou seja, as células serão recebidas na mesma ordem em que foram enviadas.

Há dois tipos de conexões no ATM: *ponto-a-ponto* e *multi-ponto*. Conexões ponto-a-ponto são como chamadas telefônicas normais, elas conectam dois usuários do serviço

ATM. Conexões multi-ponto podem ser utilizadas para prover facilidades de *multicast*.

A base do ATM é uma célula de 53 bytes consistindo de um campo de 48 bytes para transporte de informações dos usuários do serviço ATM e um cabeçalho de 5 bytes com informações do próprio ATM, veja figura 2.4.

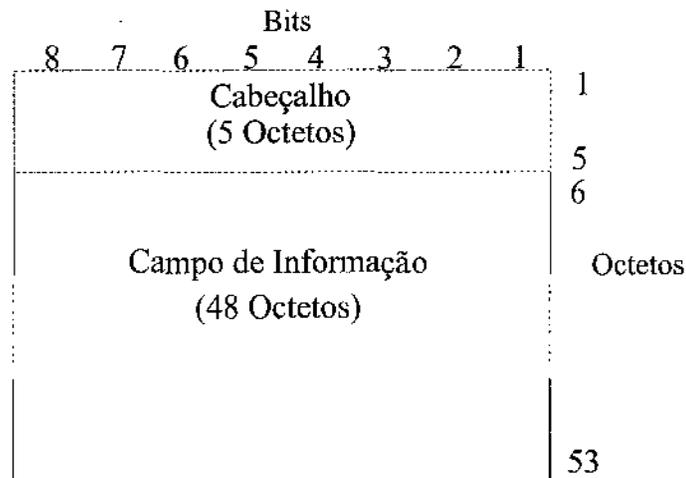


Figura 2.4: Estrutura da célula ATM.

Um elemento chave na padronização do ATM são os campos do cabeçalho associado com cada célula. O ITU-T, através da recomendação I.361 [IT92], especifica os seguintes campos para o cabeçalho da célula utilizada na interface Usuário-Rede, figura 2.5:

- GFC (*Generic Flow Control*) - 4 bits: utilizado na interface Usuário-Rede para controlar o fluxo de informações do usuário que saem da rede do usuário em direção à rede pública.
- VCI (*Virtual Channel Identifier*) - 16 bits: utilizado para direcionar células dentro da rede ATM.
- VPI (*Virtual Path Identifier*) - 8 bits: utilizado para direcionar células dentro da rede ATM, logo abaixo é explicado em maiores detalhes a utilização dos campos VCI e VPI.
- PT (*Payload Type*) - 3 bits: identifica tipos diferentes de dados, tanto dados do usuário quanto dados administrativos, transportados pelas células.

- CLP (*Cell Loss Priority*) - 1 bit: Se este bit for igual a 1, a célula possui baixa prioridade e pode ser descartada quando a rede estiver congestionada. Se ele estiver zerado, a célula é de alta prioridade e não deve ser descartada.
- HEC (*Header Error Control*) - 8 bits: O valor do campo HEC é gerado e inserido pela camada física. Ele serve como uma soma verificadora para os 4 primeiros octetos do cabeçalho da célula ATM. Ele consegue corrigir erros em um bit e detectar erros em vários bits.

8	7	6	5	4	3	2	1	Bit Octeto
GFC				VPI				1
VPI				VCI				2
VCI								3
VCI				PT		CLP		4
HEC								5

Figura 2.5: Formato do cabeçalho da célula ATM na interface Usuário-Rede.

O formato do cabeçalho da célula ATM dentro da rede (entre nós da rede) não é o mesmo do utilizado na interface Usuário-Rede. A figura 2.6 mostra o cabeçalho utilizado na interface Rede-Rede. A diferença consiste na eliminação do campo GFC e na expansão do campo VPI de 8 para 12 bits.

8	7	6	5	4	3	2	1	Bit Octeto
VPI								1
VPI				VCI				2
VCI								3
VCI				PT		CLP		4
HEC								5

Figura 2.6: Formato do cabeçalho da célula ATM na interface Rede-Rede.

As células ATM circulam ao longo de entidades conhecidas como canais virtuais (VC). Um canal virtual é identificado pelo seu identificador de canal virtual (VCI). Todas as células em um determinado canal virtual seguem a mesma rota dentro da rede e são entregues na mesma ordem em que foram transmitidas. Canais virtuais são transportados dentro de caminhos virtuais (VPs). Um caminho virtual é identificado pelo seu identificador de caminho virtual (VPI). Os caminhos virtuais são utilizados para agregar canais virtuais, para, entre outras coisas, facilitar a comutação de canais virtuais com o mesmo destino.

Para se transportar unidades de dados maiores do que 48 octetos em células ATM, é necessário uma camada de adaptação. O ITU-T propôs para a RDSI-FL um modelo de referência de protocolos em camadas. Uma inovação nesse modelo de referência, em relação, por exemplo, ao modelo OSI-ISO, é a presença de três planos [Mon94]: plano do usuário, plano de controle e plano de gerenciamento, veja figura 2.7.

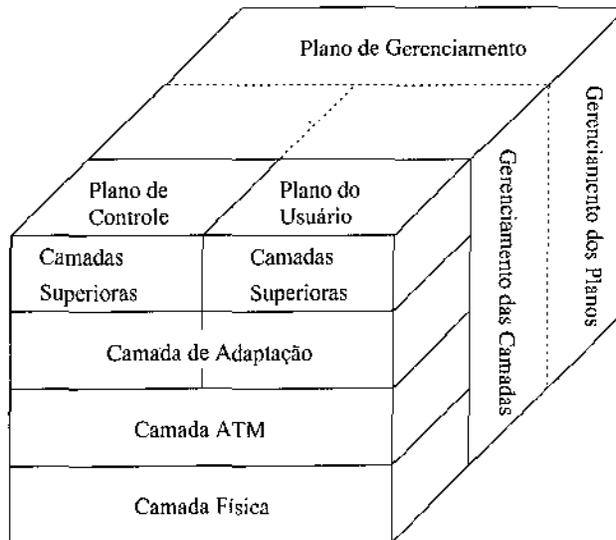


Figura 2.7: Modelo de referência dos protocolos da RDSI-FL.

O plano do usuário é responsável pela transferência de informações do usuário e do controle associado a esta transferência, tais como controle de fluxo e recuperação de erros.

O plano de controle é responsável pelo controle da chamada e pelas funções de controle das conexões. Ele cuida de toda a sinalização referente ao estabelecimento, supervisão e liberação de chamadas e conexões.

O plano de gerenciamento é responsável pelo gerenciamento dos planos e das camadas. As funções de gerenciamento do plano são relativas ao sistema como um todo e coordenação entre os planos. As funções de gerenciamento das camadas correspondem à sinalização referente aos parâmetros residentes nas suas entidades de protocolo. O gerenciamento das camadas trata dos fluxos de informação de operação e manutenção específicos de cada camada.

No modelo de referência da RDSI-FL, a camada física diz respeito aos aspectos mais básicos da transmissão dos bits, tais como codificação, alinhamento, etc. Sua função básica é fornecer à camada ATM uma interface independente do meio físico de transmissão.

A camada ATM é independente do meio físico e suporta a transferência de células para todos os tipos de serviços, sejam eles orientados ou não a conexões, com taxa de transmissão constante ou variável.

A camada de adaptação ATM (AAL - *ATM Adaptation Layer*) faz o trabalho de segmentação e remontagem das unidades de dados das camadas superiores e também faz a detecção de erros na transmissão. Como a camada ATM simplesmente transporta células sem se preocupar com seus conteúdos, diferentes AALs podem ser utilizadas em uma única interface ATM. Os extremos de cada conexão devem entrar em acordo sobre qual AAL eles utilizarão, mas a rede ATM não precisa ficar sabendo disso. Inicialmente cinco AALs estavam sendo consideradas pelo ITU-T [IT93b]. A AAL 1 é voltada para serviços com taxas constantes de bits (serviços classe A). Os serviços classe B serão oferecidos pela AAL 2. As camadas de adaptação 3, 4 e 5 são responsáveis pelo oferecimento dos serviços classe C e D. Devido às semelhanças entre as camadas 3 e 4, elas foram fundidas em uma única camada denominada 3/4. A AAL 5 é a camada de menor *overhead* entre as camadas utilizando um único bit por célula para a implementação do protocolo, ela está voltada para o transporte de dados não orientados a conexões e sem características de tempo real.

2.3 Capacidades de Controle de Congestionamento da RDSI-FL

Uma configuração de referência para o controle de tráfego, controle de congestionamento e gerenciamento de recursos foi recomendada pelo ITU-T na recomendação, ainda em fase de aprovação, I.371 [IT93c], veja figura 2.8. Para informações mais recentes sobre gerenciamento de tráfego em ATM, veja a especificação de gerenciamento de tráfego do

Forum ATM versão 4.0 [For96].

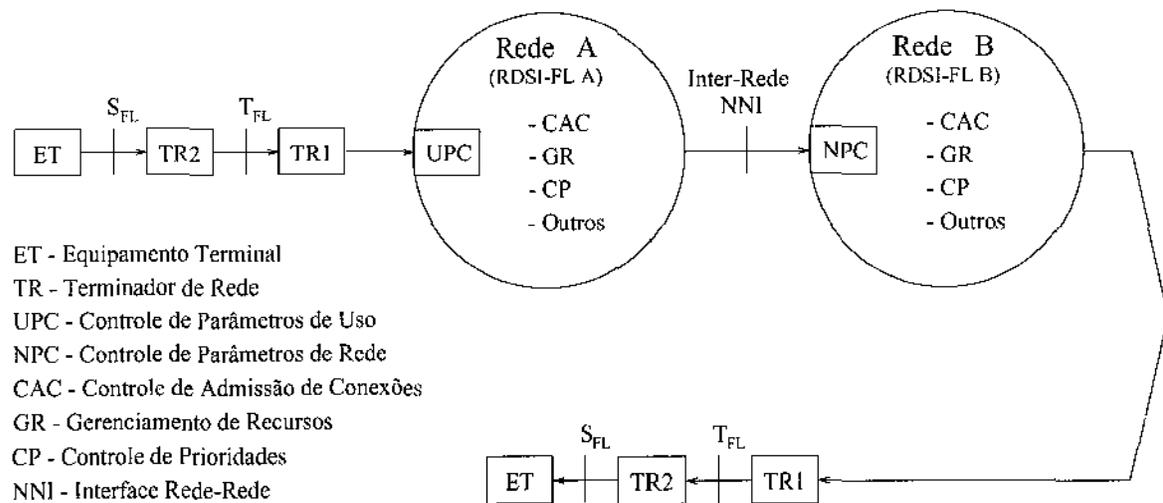


Figura 2.8: Configuração recomendada pelo ITU-T para o controle de tráfego na RDSI-FL.

Os principais elementos de controle de congestionamento que podem ser implementados na configuração de referência são:

- **Controle de Admissão de Conexões**

Esta função controla o acesso aos serviços da rede para assegurar que existem recursos suficientes para suportar o serviço exigido. O controle de admissão de conexões foi definido na recomendação I.371 como sendo o conjunto de ações que são tomadas durante a fase de estabelecimento da conexão de modo a determinar se esta conexão de canal virtual ou caminho virtual pode ser aceita ou deve ser rejeitada.

- **Controle de Parâmetros de Uso (UPC) - Policiamento**

Esta função monitora e controla as características do tráfego do usuário para assegurar que elas estejam consistentes com as características solicitadas no estabelecimento da conexão. O policiamento é utilizado para proteger a rede de comportamentos maliciosos que podem afetar a qualidade de serviço de outras conexões já estabelecidas na rede.

- **Controle de Parâmetros de Rede (NPC) - Policiamento**

Esta função é implementada na interface entre nós de rede e controla a utilização

dos recursos quando redes diferentes são interconectadas. O objetivo, novamente, é proteger os recursos locais dos efeitos que podem causar o tráfego de outras redes.

- **Função de Descarte de Células**

Esta função é baseada no bit CLP do cabeçalho da célula ATM. Ele é utilizado para dividir células em classes de alta e baixa prioridades. Na ocorrência de congestionamento, as células de baixa prioridade podem ser descartadas dentro da rede.

- **Descarte Seletivo de Células**

Um elemento de rede que esteja sofrendo congestionamento pode descartar seletivamente células de acordo com as seguintes condições:

- Células pertencentes a conexões que não estão seguindo os parâmetros pré-estabelecidos com a rede.
- Células que possuem o bit CLP com valor 1, ou seja, células de baixa prioridade.

- **Moldagem de Tráfego**

Moldagem de tráfego é um mecanismo que altera as características do tráfego de uma conexão para se obter maior eficiência da rede e para assegurar os parâmetros pré-estabelecido pela conexão. Exemplos de moldagem de tráfego são a redução da taxa de pico, a limitação do comprimento de rajada, e a redução da variação do atraso de células. Todos esses objetivos são, normalmente, atingidos através do armazenamento e do posterior espaçamento apropriado no tempo das células.

- **Descarte de Pacotes**

Se um elemento de rede tiver de descartar células, é muito mais efetivo, na maioria dos casos, fazer o descarte no nível de pacotes do que no nível de célula, ou seja, é melhor descartar células que pertençam a um mesmo pacote do que fazer o descarte aleatório. Essa medida é justificada pela, possível, retransmissão do pacote incompleto. O termo pacote é relativo à unidade de dados do protocolo da camada de adaptação ATM (AAL).

- **Indicação Explícita Adiante de Congestionamento (EFICI)**

Um elemento de rede que esteja sofrendo congestionamento pode indicar esta ocorrência através de um bit do cabeçalho da célula ATM, de modo que, essa indicação

será examinada pelo destinatário. Neste caso, o destinatário pode usar essa informação para implementar um protocolo que adaptativamente diminua a taxa de células da conexão enquanto durar o congestionamento.

2.4 Classificação dos Esquemas de Controle de Congestionamento

Como apontado por Jain [Jai90, SJ95], um planejamento de controle e prevenção de congestionamento consiste de esquemas de controle que operem em várias escalas de tempo. Alguns desses esquemas estão ilustrados na figura 2.9

Controles *rápidos* ou de curta duração atuam no extremo inferior da escala de tempo, ou seja, no tempo de transmissão de um pacote. Estes mecanismos incluem esquemas preventivos tais como gerenciamento de buffer ou escalonamento de pacotes, e esquemas reativos tais como o descarte de pacotes.

Controles de *média duração* são geralmente controles reativos baseados em realimentação da rede. Eles confiam em informações a respeito do estado da rede e ajustam as fontes de tráfego nos extremos das conexões para se enquadrarem dentro de condições favoráveis para a rede. Normalmente, atuam em escalas de tempo da ordem do atraso fim-a-fim da rede.

Controles *lentos* ou de *longa duração* atuam em uma escala de tempo que é aproximadamente o tempo de duração da chamada. No estabelecimento da chamada, a rede seleciona rotas entre fontes e destinos tentando balancear o tráfego total entre os vários caminhos existentes para cada par fonte-destino. Se os recursos ao longo de todos os caminhos não são suficientes, a chamada pode ser bloqueada por esses controles. Se houver congestionamento na rede, devido à, por exemplo, quebra de algum componente, os controles de longa duração podem estabelecer novas rotas para as chamadas e em alguns casos podem até mesmo desconectar chamadas existentes para reduzir a carga total da rede.

Além dos controles discutidos até agora, para se ter uma estratégia eficiente de controle de congestionamento, são necessários procedimentos de planejamento de rede de longa duração para projetar mudanças no tráfego e também antecipar falhas nos elementos da rede.

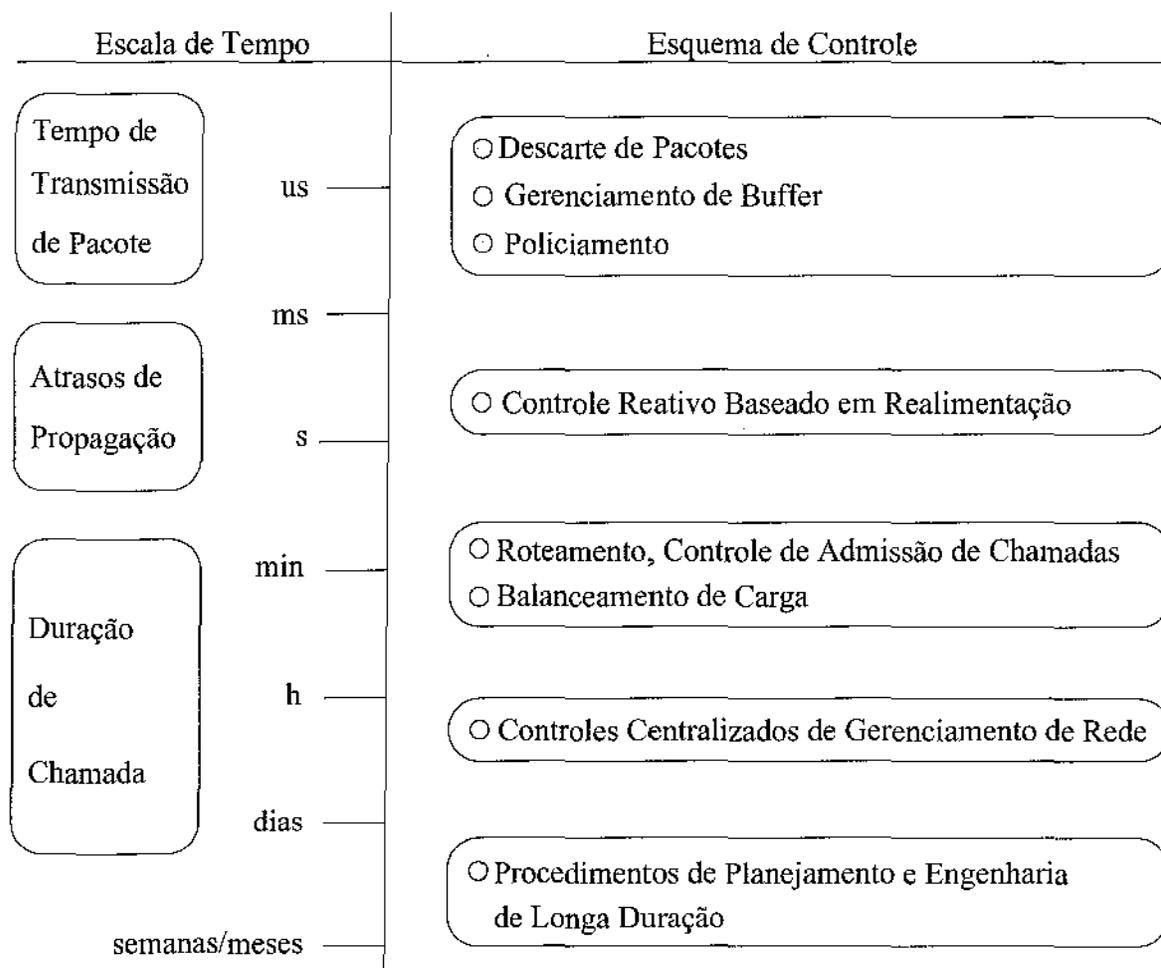


Figura 2.9: Classificação por escala de tempo dos esquemas de controle de congestionamento.

2.5 Abordagens Utilizadas no Controle de Congestionamento

Controle de congestionamento se preocupa primeiramente com a redução do tráfego para reduzir a sobrecarga da rede. Os mecanismos para controle de congestionamento podem ser classificados em *reativos* e *preventivos*. Mecanismos de controle reativos reagem ao congestionamento depois que ele acontece e tenta diminuir o grau de congestionamento para um nível aceitável. Mecanismos de controle preventivo, por outro lado, tentam prevenir o congestionamento antes que ele aconteça. O objetivo do controle preventivo é

assegurar que o tráfego da rede não atingirá níveis que causarão congestionamento.

Mecanismos de controle de admissão de conexões podem ser classificados como preventivos ou reativos dependendo da estratégia utilizada. Entretanto, o controle de admissão opera em uma escala de tempo da ordem de duração da chamada, que é maior do que a escala de tempo dos mecanismos preventivos e reativos. O controle de admissão pode, portanto, ser aplicado em conjunto com mecanismos preventivos e reativos que operam em uma escala de tempo menor. Portanto, as questões relacionadas ao controle de admissão de conexões serão mencionadas em uma seção a parte sem especificá-las como preventivas ou reativas.

2.5.1 Mecanismos Preventivos

Mecanismos preventivos podem ser subdivididos em mecanismos de controle de curta duração e mecanismos de controle de média duração de acordo com a escala de tempo em que eles operam. Nos mecanismos de curta duração estão incluídos os mecanismos de controle de uso da capacidade ou policiamento. Mecanismos de controle de média duração são geralmente os esquemas de gerenciamento de recursos.

Controle dos Parâmetros de Uso/Rede - Policiamento

Este método de controle de congestionamento tenta evitar congestionamento empregando uma política de admissão de célula por conexão na entrada da rede. Vários mecanismos foram propostos para a realização de policiamento na RDSI-FL, sendo que os mais conhecidos são o balde furado e suas variantes, a janela saltitante e a janela deslizante [Jr93, Mon90, LRG92].

O *Balde Furado (Leaky Bucket)*, proposto por Turner [Tur86], é, sem dúvida, o mecanismo de policiamento mais conhecido e mais estudado [Jr93, Mon90, LRG92, CFT92]. O algoritmo é bastante simples e está baseado em três parâmetros: um contador, um valor máximo N para o contador e uma constante B_e que representa a taxa de decremento do contador.

O algoritmo do balde furado se baseia em uma pseudo-fila implementada através do contador. O contador é incrementado toda vez que chega uma célula da fonte monitorada e, se positivo, decrementado na taxa B_e . O valor máximo que esse contador pode atingir é N , que representa o tamanho da pseudo-fila. As células que chegam e encontram o

contador com valor N , ou seja, a pseudo-fila cheia, são descartadas.

Na realidade, nenhuma célula é realmente enfileirada, daí o nome pseudo-fila, o que implica que o fluxo de células bem comportadas não é alterado pelo mecanismo e também não sofre nenhum tipo de retardo.

Uma proposta de modificação para o balde furado foi apresentada em [GRF89]. No algoritmo proposto, denominado *Balde Furado Virtual* (*Virtual Leaky Bucket*), as células mal comportadas, células que ultrapassam a taxa declarada, não são descartadas e sim marcadas como de baixa prioridade. As células marcadas serão descartadas dentro da rede somente se houver necessidade.

O mecanismo da *Janela Saltitante* (*Jumping Window*) [Jr93, LRG92] monitora as células de uma conexão em intervalos de tempo T consecutivos. Ele limita o número de células que uma fonte pode emitir dentro de um intervalo T , que representa o tamanho da janela, a um valor máximo N . Para controlar as células, ele utiliza um contador que é incrementado toda vez que chega uma célula da fonte monitorada. As células são marcadas quando o contador ultrapassa o valor máximo permitido N . Um novo intervalo de monitoração começa sempre após o término do anterior e o contador é sempre zerado para monitorar o tráfego no novo intervalo. Portanto, o intervalo de tempo em que uma célula contribui para o valor do contador varia de zero ao tamanho da janela.

A principal desvantagem da janela saltitante é que, com o início de uma nova janela, o mecanismo perde a história anterior do tráfego monitorado.

No mecanismo da *Janela Deslizante* (*Moving Window*) [Jr93, LRG92], de modo semelhante à janela saltitante, o número de células que chegam em um intervalo de tempo T é limitado. A principal diferença entre os dois mecanismos é que na janela deslizante cada célula deve ser lembrada por um período de tempo equivalente ao tamanho da janela, T . Isso quer dizer que o mecanismo necessita saber qual foi o tempo em que a célula chegou. A cada chegada de uma nova célula, o contador é incrementado em uma unidade. As células que chegam e encontram o contador com o seu limite máximo (N) são descartadas ou marcadas. Após o período de tempo T em que houve a aceitação de uma célula, o contador é decrementado em uma unidade. Daí a analogia com uma janela que desliza no tempo.

Em [Jr93] Frazão apresenta um estudo detalhado desses mecanismos abordando a eficiência deles no policiamento de fontes com comportamento de pior caso. Além disso, são apresentadas as complexidades de implementação dos mecanismos aqui citados e de

outros mecanismos propostos na literatura.

Gerenciamento de Recursos

Gerenciamento de recursos na RDSI-FL inclui um conjunto de tópicos complexos e relacionados, localizados em vários níveis da rede, como por exemplo, nível de célula, de chamada e de rede, tais como [GRV90, BD91]:

- **Utilização de caminhos virtuais**

O ITU-T na recomendação I.371 [IT93c] sugere a utilização de caminhos virtuais para:

- simplificar o controle de admissão de chamadas;
- implementar uma forma de controle de prioridade separando tipos de tráfegos com diferentes exigências de qualidade de serviço;
- distribuir eficientemente mensagens para a operação de esquemas de controle de tráfego, como por exemplo, para indicar congestionamento da rede através de uma única mensagem em um caminho virtual compreendido por vários canais virtuais;
- agregar serviços do usuário de modo que o policiamento (UPC/NPC) possa ser aplicado ao tráfego agregado;
- agregar as capacidades de rede de modo que o policiamento (NPC) possa ser aplicado ao tráfego agregado.

- **Gerenciamento de Buffers**

Os buffers dos comutadores constituem recursos de alta importância em uma rede, pois o gerenciamento adequado pode possibilitar diferentes níveis de qualidade de serviço para as conexões. Entre as tarefas a serem realizadas em um buffer estão a disciplina de atendimento e o estabelecimento de prioridades dentro do buffer.

Para o atendimento nos buffers algumas disciplinas foram propostas na literatura [HB92]:

- *HOLP (Head of Line Priority)*: O cabeça da fila tem prioridade sobre os demais. Neste esquema a fila com prioridade mais alta é sempre examinada primeiro.

- WRR (*Weighted Round-Robin*): A capacidade do canal é dividida entre diversas filas de acordo com valores de pesos pré-estabelecidos. As filas são servidas ciclicamente numa ordem pré-fixada, sendo cada fila examinada um número de vezes proporcional ao seu peso em cada ciclo.

O ATM estabelece prioridades para células através de um bit no cabeçalho. Portanto, pode haver em uma rede células de alta e de baixa prioridade. Os comutadores podem implementar mecanismos para armazenar células com diferentes prioridades em seus buffers, os mais difundidos na literatura são [Mon94, BM92, GTG93]:

- *Push-out*: Se existir uma célula de baixa prioridade em um buffer já saturado aguardando transmissão, uma célula de alta prioridade pode ser armazenada e uma célula de baixa prioridade é removida do buffer e descartada.
- Limiar (*Threshold*): Quando o buffer atinge uma certa ocupação (limiar), apenas células de alta-prioridade são armazenadas, as demais são descartadas.
- *L Push-out*: Neste mecanismo, o buffer é particionado em dois. Na primeira parte do buffer, não é considerada a prioridade das células, as células de baixa prioridade podem ser armazenadas normalmente. Na segunda parte do buffer é utilizado o mecanismo *Push-out*, ou seja, células de baixa prioridade podem ser removidas do buffer e descartadas.

2.5.2 Mecanismos Reativos

Nos últimos anos, pesquisas em controles reativos têm recebido muita atenção. Há vários mecanismos propostos na literatura [Jai96]. Uma abordagem simples é utilizar o bit de prioridade de célula (CLP) no cabeçalho da célula ATM que permite que as aplicações do usuário indiquem quais células são menos importantes do que outras. Essas células de menor prioridade podem ser descartadas em qualquer instante que haja congestionamento na rede. O descarte dessas células não afeta severamente a aplicação, já que elas foram marcadas pela própria aplicação.

Uma outra proposta para mecanismos reativos de controle de congestionamento é colocar o valor um em um bit específico do cabeçalho da célula ATM e passar adiante, até o destino, a célula marcada. Quando a célula chega ao destino com o bit com valor um, significa que ela encontrou congestionamento ao longo da conexão. O nó destino

pode, então, sinalizar para a origem a ocorrência do congestionamento e fazer com que ela dispare alguma ação apropriada para corrigir a situação. Este mecanismo é chamado de Notificação Explícita Adiante de Congestionamento (EFCN - *Explicit Forward Congestion Notification*) [Jai96]. Uma outra possibilidade é a geração explícita de uma célula de gerenciamento de recursos no nó dentro da rede que está sofrendo congestionamento e o envio desta célula para a origem. Este último mecanismo é conhecido por Notificação Explícita para Trás de Congestionamento (EBCN - *Explicit Backward Congestion Notification*) [Jai96].

Em oposição às abordagens de realimentação negativa adotadas pelos mecanismos EFCN e EBCN, o Algoritmo de Taxa Proporcional (PRCA - *Proportional Rate Algorithm*) [Jai96, For96] é baseado em um paradigma de controle de congestionamento com realimentação positiva. No algoritmo de taxa proporcional, a fonte aumenta sua taxa de transmissão somente se ela receber uma indicação explícita e positiva do destinatário, senão, na ausência de tal indicação positiva, a fonte reduzirá continuamente sua taxa de transmissão. Os incrementos e decrementos na taxa de transmissão para cada conexão são proporcionais à taxa atual. O Forum ATM, através do grupo de trabalho de gerenciamento de tráfego, adotou uma variante do algoritmo de taxa proporcional como o mecanismo proposto para o gerenciamento de tráfego baseado na taxa de transmissão (*Rate-Based Traffic Management*) [For96].

2.5.3 Controle de Admissão

O controle de admissão de conexões deve decidir se aceita ou rejeita uma nova conexão com base no estado atual da rede e se o desempenho solicitado pela conexão pode ser mantido. Em [FMGV92] é apresentada uma descrição de alguns esquemas de controle de admissão de conexões propostos para redes ATM. As principais questões estudadas em controle de admissão de conexões são:

- Os parâmetros para o tráfego (descritores de tráfego) necessários para se prever de forma precisa o desempenho da rede;
- Os critérios utilizados pela rede para aceitar novas conexões;
- A dependência do desempenho da rede em relação aos vários parâmetros de tráfego.

A determinação de um conjunto apropriado de descritores de tráfego e a estimativa da capacidade necessária para uma conexão são problemas desafiadores e ainda em aberto. Prever as características do tráfego de uma conexão é uma tarefa muito difícil. O princípio básico do controle de admissão de conexões é: se a soma de todas as capacidades estimadas das conexões compartilhando um determinado enlace é menor que um valor máximo pré-determinado, então a nova conexão pode ser estabelecida [GRV90]. A capacidade estimada da nova conexão então se torna a quantidade de capacidade alocada para a conexão, e espera-se do usuário que ele mantenha a transmissão dentro dos limites deste *contrato* ou então sofrerá as conseqüências em termos de degradação do serviço.

O Forum ATM vem trabalhando para definir um conjunto de classes de serviços, de modo que o usuário no estabelecimento da conexão deve escolher uma classe específica para a conexão. As classes de serviços serão utilizadas para diferenciar tipos específicos de conexões, cada uma com sua qualidade de serviço. O conjunto atual de classes de serviços consideradas pelo Forum ATM são [For96]:

- CBR - (*Continuous Bit Rate*): Os usuários utilizarão conexões CBR para transportar tráfegos com taxas constantes de bits e com relacionamento de tempo fixo entre as amostras de dados.
- VBR-RT - (*Variable Bit Rate - Real Time*): A classe de serviço VBR-RT será utilizada para transportar tráfegos com taxas variáveis de bits, nos quais haja relacionamento de tempo fixo entre as amostras de dados. Aplicações que farão uso desta classe são as aplicações de transmissão de vídeo com compressão.
- VBR-NRT - (*Variable Bit Rate - Non-Real Time*): A classe de serviço VBR-NRT será utilizada para transportar tráfegos com taxas variáveis de bits, nos quais não haja relacionamento fixo de tempo entre as amostras de dados, mas haja necessidade de garantia da qualidade de serviço.
- ABR - (*Available Bit Rate*): Da mesma maneira que a classe VBR-NRT, a classe ABR será utilizada para transportar tráfegos com taxas variáveis de bits, nos quais não haja relacionamento fixo de tempo entre as amostras de dados. O fator que diferencia esta classe da anterior é a não garantia da qualidade de serviço. Mais especificamente, a rede atuará na base do maior empenho possível para transportar dados desta classe. Os dados serão transmitidos desde que não haja congestionamento na rede e baseado em informações de realimentação. Entretanto, com base

nas informações obtidas, a rede pode controlar a quantidade de tráfego transmitida visando minimizar a perda de células.

- UBR - (*Unspecified Bit Rate*): O serviço UBR não oferece qualquer tipo de garantia de serviço. O usuário pode enviar qualquer quantidade de dados até um valor máximo especificado. A rede não oferece garantias quanto a taxa de perda de células, atrasos ou variações de atrasos.

O ITU-T definiu até o momento apenas as classes CBR e VBR, sendo que esta última é considerada uma única classe não havendo a diferenciação entre VBR-RT e VBR-NRT [IT93c].

2.6 Suporte a Serviços Não Orientados a Conexões

Muito embora se espere da RDSI-FL o oferecimento de serviços orientados a conexões com garantia de serviços, há também a necessidade de se oferecer serviços não orientados a conexões, pois há uma grande quantidade de protocolos não orientados a conexões atualmente que terão de interagir com a rede numa fase inicial, por exemplo o protocolo IP (*Internet Protocol*) que é a base de toda comunicação na Internet e é um protocolo não orientado a conexão.

O ITU-T, através da Recomendação I-211 [IT93a], identifica duas formas de suporte a serviços não orientados a conexões: *indiretamente* através da utilização das conexões ATM convencionais da rede e *diretamente* através de servidores específicos dentro da própria RDSI-FL.

A figura 2.10 mostra como foi sugerido pelo ITU-T o oferecimento indireto do serviço não orientado a conexão. A RDSI-FL oferecerá conexões ATM normais entre os usuários, ou seja, um serviço orientado a conexão. É responsabilidade dos usuários compatibilizar os dois tipos de protocolos (orientado e não orientado a conexão). Esse esquema possui um problema sério que é a necessidade de se manter um grande número de conexões entre os vários usuários, a menos que sejam utilizados explicitamente e externamente à rede servidores do serviço não orientado a conexão. Além desse, um outro problema nesta configuração é que os dispositivos de acesso à rede orientada a conexão terão de decidir quanta capacidade tem de ser alocada para as conexões e quando. Este último problema será estudado em detalhes no capítulo 3.

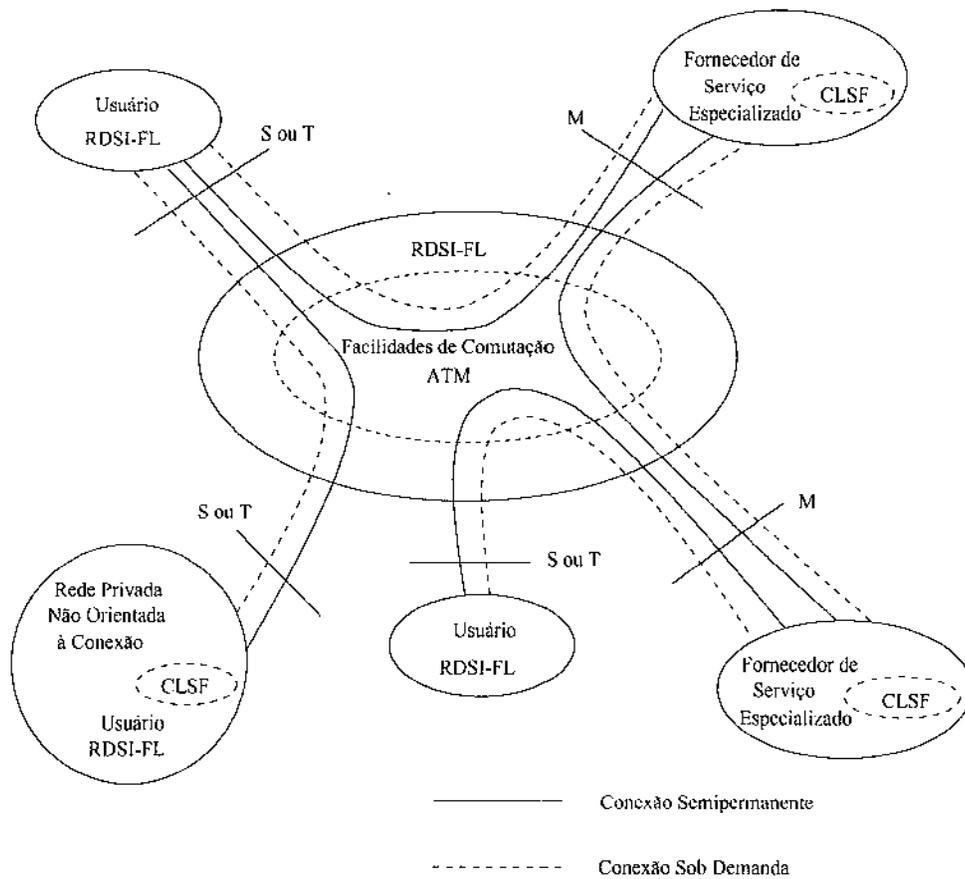


Figura 2.10: Suporte indireto a serviços não orientados a conexões.

No caso do suporte direto ao serviço não orientado a conexão, a RDSI-FL se encarregará de implementar, dentro da própria rede, servidores com o propósito específico de administrar esse serviço. O serviço será suportado através das funções de comutação ATM e das funções de serviço não orientado a conexões (CLSF - *Connectionless Server Functions*). Estas últimas podem ser implementadas num equipamento distinto ou então no próprio comutador ATM. A figura 2.11 apresenta a configuração de referência para o suporte direto ao serviço não orientado à conexão na RDSI-FL [IT95]. As letras S, T, M e P nas figuras 2.10 e 2.11 indicam os pontos de referência dos modelos.

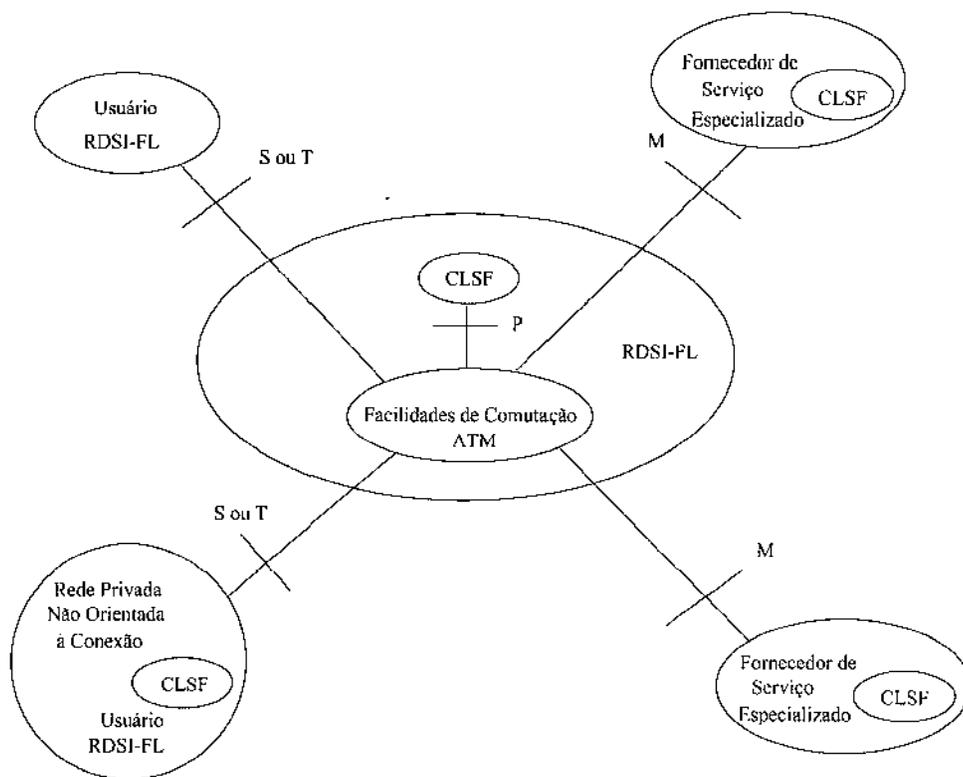


Figura 2.11: Suporte direto a serviços não orientados a conexões.

Capítulo 3

Mecanismos de Gerenciamento de Capacidade

O protocolo proposto para a RDSI-FL é orientado a conexão. Tal escolha é justificada por considerações de desempenho e pela necessidade de se garantir qualidade de serviço a clientes específicos. Por outro lado, a maioria das redes locais operam num modo não orientado a conexão e necessitarão interagir com a nova rede. Essa interação constitui um dos maiores desafios a ser superado para a implantação da RDSI-FL [Par90, GTG93, MFT91, Boy90, Tur92, FM92, Par90]. Dispositivos apropriados de interconexão, *gateways*, serão necessários para interfacear as duas categorias de redes, sendo que esses dispositivos terão de gerenciar e controlar os fluxos de informações entre os equipamentos dos usuários e a rede de longa distância. Os *gateways* terão de decidir quando e quanto alocar de capacidade às conexões que os interligam, principalmente porque é esperado que as taxas de transmissão das futuras redes locais sejam da mesma ordem de grandeza daquelas previstas para a RDSI-FL, de modo que atribuir a taxa de pico a cada conexão resultaria num desperdício significativo de recursos. O desafio nesse novo ambiente é combinar as características desconhecidas do tráfego não orientado a conexão, gerado pelas redes locais, com as características do modo de transmissão orientado a conexão escolhido para a RDSI-FL.

Uma solução de consenso para esse problema, segundo vários trabalhos apresentados na literatura [GTG93, MFT91, FM92], é prover aos dispositivos de interconexão inteligência para gerenciar dinamicamente as capacidades alocadas às conexões na RDSI-FL que transportam dados não orientados a conexões.

Neste capítulo serão estudadas algumas propostas para se gerenciar a capacidade dos enlaces de uma rede, com o objetivo de se transportar de maneira eficiente tráfegos em rajadas. Na seção 1 serão apresentados uma visão particular do problema e um conjunto de critérios para avaliar as propostas estudadas. Nas seções 2, 3, 4, e 5 serão apresentadas propostas específicas de vários autores, sendo que para cada proposta será apresentada uma avaliação qualitativa de cada uma delas. As avaliações quantitativas de algumas propostas selecionadas serão dadas no capítulo 6.

3.1 Definição do Problema

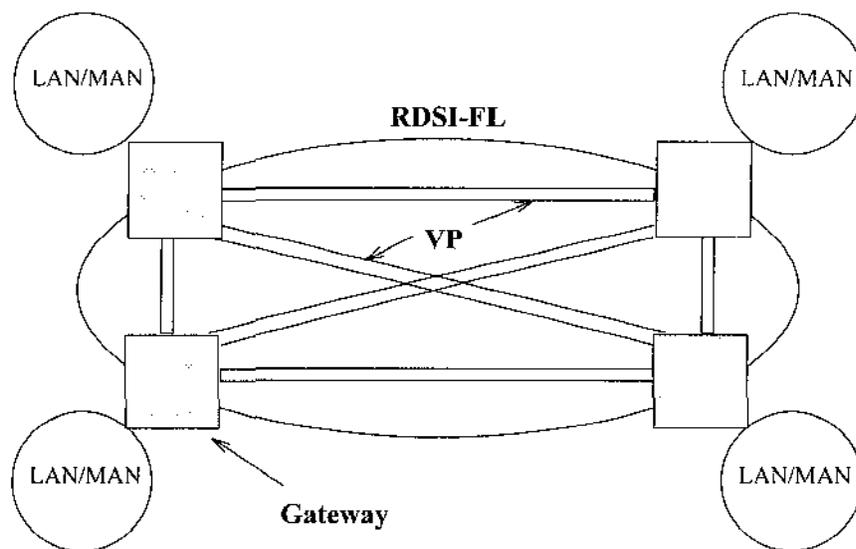


Figura 3.1: Interconexão de redes locais e metropolitanas através da RDSI-FL.

O suporte ao serviço não orientado a conexão em uma rede ATM pode ser estudado de várias maneiras, como por exemplo a emulação de redes locais definida pelo Forum ATM [For94, Kav95] e o suporte direto através de servidores específicos [IT95]. Neste trabalho serão consideradas, basicamente, as definições do ITU-T para a RDSI-FL. O trabalho concentrar-se-á na interconexão de redes locais (LANs) através da RDSI-FL. A rede local será considerada como sendo simplesmente um concentrador de tráfego não isócrono e sem características de tempo real alimentando a interface Usuário-Rede da RDSI-FL. As facilidades da RDSI-FL serão utilizadas para construir uma rede local estendida, que é

uma coleção de LANs conectadas por dispositivos de interconexão transparentes, doravante referenciados *gateways*. Devido à grande incidência da palavra *gateway* no texto e também pela familiaridade do termo na literatura portuguesa não será mais utilizada a forma enfatizada.

A figura 3.1 mostra como o problema será abordado. A RDSI-FL simplesmente oferecerá os meios de interconexão para se ter uma rede local estendida. Entretanto, algumas propostas introduzem novas características nos comutadores da RDSI-FL visando agilizar o transporte do tráfego entre os gateways.

3.1.1 Critérios de Avaliação

Frost e Mullen [FM92] enumeram um conjunto de atributos que os gateways entre as redes locais e a RDSI-FL e os seus mecanismos de gerenciamento de capacidade associados devem possuir. Esses critérios podem ser vistos, mais genericamente, como atributos entre dispositivos que fazem a ligação entre protocolos não orientados a conexões e protocolos orientados a conexões. Os autores também mencionam como conclusão o que deve ser feito para atingir esses atributos. Abaixo são citadas as características desejáveis e as conclusões de Frost e Mullen:

- **Os gateways não devem exigir dos comutadores da RDSI-FL atributos especiais para gerenciar as conexões que transportam tráfego não orientado a conexão.**

Seria irreal desenvolver mecanismos de alocação dinâmica de capacidade que exijam tratamento especial em todos os comutadores ao longo do caminho entre os gateways origem e destino. Essa exigência aumentaria a complexidade dos comutadores, além de afetar diretamente as funções de operação e manutenção.

Conclusão: alocação dinâmica de capacidade deve ser feita numa base fim-a-fim no ponto de acesso.

- **Os gateways devem minimizar o armazenamento no buffer do gateway de origem.**

O armazenamento no gateway origem aumentará o atraso e a perda de pacotes. Um armazenamento mínimo implica que a carga oferecida ao gateway é atendida

Conclusão: os gateways devem ser robustos com relação às variações estatísticas do tráfego oferecido.

- **Os gateways devem estar aptos a operar a velocidades na ordem de Gigabits por segundo nas duas interfaces.**

Os pacotes que chegam nos gateways devem ser processados e reformatados para transmissão no enlace de saída. O tempo necessário para realizar essas funções será crítico para o gateway operar a taxas na ordem de Gigabits por segundo.

Conclusão: todas as funções e algoritmos dos gateways devem ser simples e passíveis de serem implementados em redes operando a Gbps.

pela rede a maior parte do tempo. Assim algumas rajadas de pacotes podem ser injetadas na rede orientada a conexão a uma taxa superior à taxa negociada.

Conclusão: permitir que o gateway ocasionalmente exceda a taxa negociada com a rede.

- **Os gateways não devem solicitar com frequência interação através de sinalização com a rede orientada a conexão.**

Alterar a taxa alocada irá exigir sinalização entre o gateway e a rede orientada a conexão. A renegociação normalmente é feita com base em informações relacionadas com a demanda atual do tráfego e a taxa de pico alocada. Tais informações estarão sujeitas a imprecisões e, como em qualquer mecanismo de realimentação, pode haver oscilações.

Conclusão: as solicitações de renegociações devem ocorrer com baixa frequência.

- **Os gateways não devem ser sensíveis às variações estatísticas do tráfego.**

A natureza estatística do tráfego é desconhecida na RDSI-FL. Qualquer esquema de alocação dinâmica de capacidade deve ser aplicável a uma grande variedade de características de tráfego.

Conclusão: os gateways devem ser robustos com relação às variações estatísticas do tráfego oferecido.

- **Os gateways devem estar aptos a operar a velocidades na ordem de Gigabits por segundo nas duas interfaces.**

Os pacotes que chegam nos gateways devem ser processados e reformatados para transmissão no enlace de saída. O tempo necessário para realizar essas funções será crítico para o gateway operar a taxas na ordem de Gigabits por segundo.

Conclusão: todas as funções e algoritmos dos gateways devem ser simples e passíveis de serem implementados em redes operando a Gbps.

- **Os gateways não devem ser sensíveis a latências relativas em redes operando na ordem de Gigabits por segundo.**

Esquemas de alocação dinâmica de capacidade e outras funções de controle dependem, normalmente, de informações de realimentação da rede. O processo de

renegociação será estimulado pela coleção de informações com o relacionamento entre as exigências atuais do tráfego, a taxa de pico alocada, e o estado da rede. A relevância dessas informações tendem a degradar com o tempo. Ao passo que as taxas dos enlaces aumentam, o efeito desses atrasos pode se tornar significativo, levando a mecanismos ineficazes.

Conclusão: todos algoritmos devem ser robustos às latências relativas de redes operando a Gbps.

Muito embora esses atributos sejam direcionados à apresentação de uma proposta específica pelos autores, veja seção 3.4, eles servem como bons parâmetros para se ter uma idéia da qualidade dos mecanismos que serão apresentados. A avaliação das propostas não se restringirá a esses parâmetros, porque senão o mecanismo proposto pelos autores estaria sendo beneficiado. Outros critérios serão apresentados dependendo da natureza do mecanismo proposto.

3.2 Mecanismo de Reservas

O trabalho de Mongiovi e outros [MFT91] foi um dos primeiros a se preocupar com o problema de alocação de capacidades para conexões entre redes locais na RDSI-FL. A idéia básica do mecanismo é armazenar as rajadas de tráfego provenientes das redes locais no gateway e renegociar gradualmente a capacidade alocada na rede ATM conforme o tráfego armazenado atinja certos limiares no buffer do gateway. O mecanismo procura, com isso, encontrar uma capacidade ideal para a conexão através de sucessivas tentativas.

Cada gateway tem de implementar um algoritmo apropriado para atualizar a quantidade de capacidade atribuída às conexões entre as redes locais na rede ATM de acordo com as características do tráfego em cada conexão. Dois mecanismos distintos de monitoração de tráfego foram propostos para ajustar a capacidade alocada: um para monitorar o aumento de tráfego e um outro para monitorar o decréscimo de tráfego. O primeiro tem por objetivo disparar mensagens de sinalização para Solicitações de Incrementos de Capacidade (SIC), quando o volume de tráfego numa conexão aumenta além de um certo limiar. O segundo é responsável por um ajuste fino de alocação a longo prazo e por disparar mensagens de Sinalização para Liberação de Capacidade (SLC). As mudanças na capacidade podem ocorrer somente na forma discreta de incrementos ou decrementos chamados "Passos de Capacidade".

3.2.1 Mecanismo para Monitoramento de Aumento do Tráfego

Supõe-se que a cada conexão entre gateways está associado um buffer. Cada buffer é particionado em limiares apropriados. O cálculo dos limiares é apresentado abaixo. A necessidade de aumento da capacidade alocada é disparada quando um limiar é excedido no buffer associado a conexão. A solicitação de um estabelecimento ou renegociação de conexão em uma rede ATM pode resultar em uma resposta positiva ou negativa. No caso da resposta ser positiva, o gateway incrementa a capacidade alocada por um valor igual ao “Passo de Capacidade”. No caso da resposta ser negativa, o gateway começa a descartar os novos quadros direcionados ao enlace congestionado, desabilita o mecanismo de solicitações de incrementos de capacidade, e notifica a fonte do tráfego usando procedimentos específicos da rede local, por exemplo, não ligando o bit de copiado do quadro FDDI em uma rede FDDI [Ros86]. O descarte de quadros produz um esvaziamento progressivo do buffer, porque o buffer continua sendo esvaziado pela taxa alocada. Quando a ocupação do buffer cai para baixo do limiar que causou a solicitação, o mecanismo de solicitações de incrementos de capacidade é reabilitado.

Os limiares no buffer devem ser definidos para compensar a diferença entre a capacidade recebida e a capacidade transmitida pelo gateway. Considere as definições abaixo:

- T_{na} : tempo em segundos (s) entre uma solicitação de incremento de capacidade e a resposta da rede (Aceitação ou Rejeição).
- Tr_i : posição em bits do i -ésimo limiar.
- T_i : tempo em segundos (s) para o gateway absorver a quantidade de bits resultante da diferença entre a taxa recebida e a capacidade transmitida entre os limiares Tr_{i-1} e Tr_i .
- A_i : tamanho em bits (b) do buffer entre os limiares Tr_i e Tr_{i+1} .
- B : maior quantidade em bits por segundo (bps) de capacidade que pode chegar ao gateway, *throughput* da rede local.
- B_i : quantidade em bits por segundo (bps) da capacidade alugada. O mecanismo assume que existe sempre uma capacidade mínima alocada para a conexão.
- B_i : quantidade em bits por segundo (bps) do i -ésimo “Passo de Capacidade”.

O tamanho total do buffer é dado por:

$$A = \sum_{i=0}^N A_i = \sum_{i=1}^N (T_i + T_{na})(B - B_i - \sum_{j=2}^i B_j)$$

Se $T_i = T_j = T_1$ e $B_i = B_j = B_l = \frac{B}{N+1}$ para cada i, j :

$$A = B(T_1 + T_{na}) \sum_{i=1}^N \frac{N - i + 1}{N + 1} = \frac{NB}{2}(T_1 + T_{na})$$

As posições dos limiares são calculadas por:

$$Tr_0 = 0$$

$$Tr_1 = T_1 B \frac{N}{N+1}$$

$$Tr_i = Tr_{i-1} + B \left[(T_1 + T_{na}) \frac{N - i + 1}{N + 1} - \frac{T_1}{N + 1} \right]$$

$$Tr_{N+1} = A$$

A figura 3.2 mostra o relacionamento entre os vários parâmetros utilizados nos cálculos.

Devido à grande dificuldade de se dimensionar adequadamente os limiares no buffer e também visando diminuir o número de renegociações com a rede, Gerla e outros em [GTG93] propuseram uma variante para este mecanismo. Nesta variante, ao invés das renegociações serem disparadas por limiares no buffer, o tráfego de entrada é monitorado e caso a taxa de chegada de dados permaneça acima da taxa alocada para o canal por um determinado período (t_{crit-a}), é enviada na rede uma solicitação de incremento de capacidade. Diferente da proposta original que solicita vários incrementos de capacidade até atingir a taxa de chegada, na variante apresentada, a solicitação de aumento já é baseada em medições do tráfego, com isso, ela compensa mais rapidamente a diferença entre a taxa de chegada e a capacidade alocada. As medições são feitas através de um esquema de média móvel que será detalhado mais adiante.

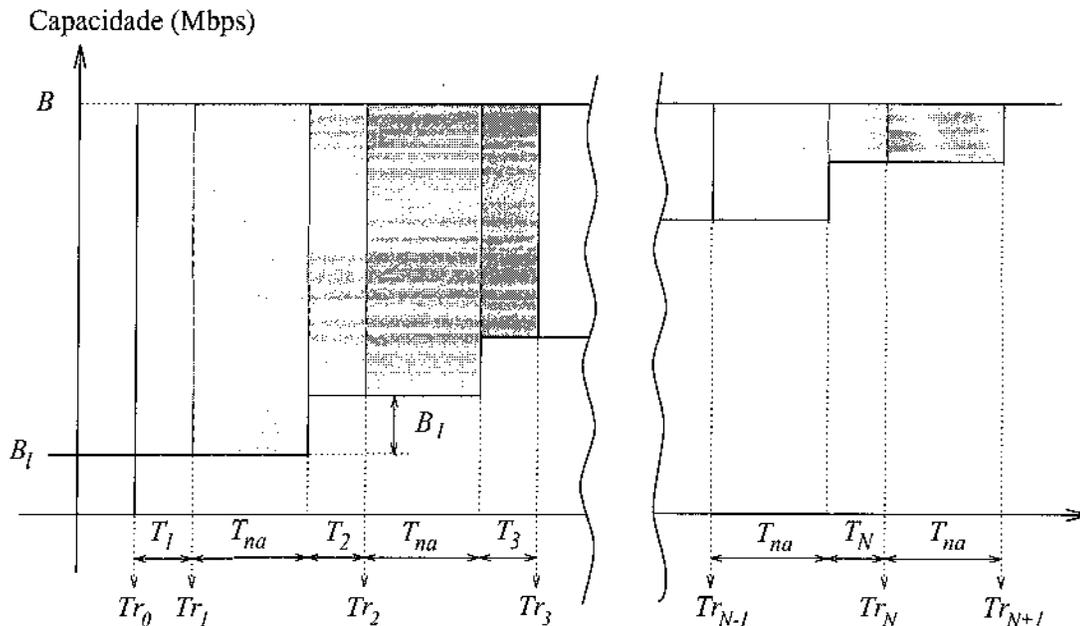


Figura 3.2: Dimensionamento dos Limiares no Buffer de Recepção.

3.2.2 Mecanismo para Monitoramento de Diminuição do Tráfego

Além do mecanismo de solicitação de capacidade, o gateway precisa de uma maneira de liberar capacidade quando ela estiver em excesso. Basicamente, isso pode ocorrer por um decréscimo no tráfego ou por uma alocação acima da necessária, devido à natureza discreta das solicitações de capacidades.

Existem várias maneiras de implementar essa tarefa. Mongiovi e outros [MFT91] propuseram dois mecanismos simples.

O primeiro mecanismo, e o mais simples, solicita a liberação de um “Passo de Capacidade”, menor do que aquele solicitado durante a alocação, toda vez que o buffer da conexão é esvaziado. Se a capacidade alocada for maior do que a taxa de chegada, o buffer continuará a esvaziar e liberações sucessivas ocorrerão. Com as sucessivas solicitações de liberação, a capacidade alocada cairá abaixo da taxa de chegada de dados, levando a um novo período de enchimento do buffer. Então o algoritmo de alocação irá recuperar a capacidade liberada impropriamente. Este mecanismo é chamado de *histerese*, por fazer a alocação de capacidade oscilar ao redor da necessidade real do tráfego.

O segundo mecanismo de liberação é um pouco mais sofisticado e se baseia em medições para prever o comportamento futuro do tráfego. Com as medições, procura-se atingir

uma alocação de recursos mais eficiente e conseqüentemente atrasos menores. Com esse propósito em mente, um mecanismo simples baseado no monitoramento da taxa média do tráfego que entra no buffer foi proposto. Quando um buffer se esvazia o gateway começa a monitorar o tráfego de entrada. Isso é feito determinando a média móvel do tráfego. O gateway faz medições em intervalos T de tempo e divide o número de bits que chegaram no intervalo pelo tempo T . O valor final usado para realocar a capacidade é obtido da média da última medição com as $N - 1$ medições anteriores. Quando o número de amostras atinge N o gateway começa a deslocar os valores, descartando o valor calculado mais antigo toda vez que uma nova amostra é calculada.

No final de cada período de medição, o gateway compara a média calculada com a capacidade alocada. Se a capacidade alocada excede a média calculada por um valor maior que um determinado percentual, por exemplo 20%, a capacidade alocada é reduzida para a capacidade medida mais uma margem de segurança. O percentual, a margem de segurança, o tempo T e o número de amostras N são parâmetros do esquema.

O processo de monitoração do tráfego é zerado e interrompido, toda vez que há uma nova solicitação de capacidade na conexão. Ele reiniciará apenas se houver um esvaziamento completo do buffer.

3.2.3 Avaliação

O mecanismo de reservas segue a filosofia utilizada nas definições das características e dos protocolos da RDSI-FL, qual seja, a migração da inteligência e do maior volume de processamento para fora da rede. Ele possui como principais vantagens a não exigência de características adicionais aos comutadores ATM e a garantia de que nenhuma célula é descartada dentro da rede ATM, pelo fato do mecanismo transmitir dados somente na taxa alocada.

As principais desvantagens do mecanismo podem ser resumidas nos seguintes pontos:

- Uma grande quantidade de memória é necessária para armazenar as rajadas no gateway;
- Atrasos consideráveis são impostos aos dados de todos os usuários pelo fato do tamanho do buffer do gateway poder crescer até valores consideráveis;
- Os processadores de controle dos comutadores ATM são sobrecarregados pelas inúmeras solicitações de incremento e liberação de capacidade;

- As capacidades dos enlaces pertencentes a conexão não são utilizadas eficientemente por dois motivos: a) alguma capacidade é desperdiçada no processo de alocação e liberação, principalmente se o comprimento da rajada recebida for comparável ao tempo de negociação com a rede; b) a alocação de capacidade baseada na taxa de pico não explora as possíveis vantagens da multiplexação estatística;
- A escolha dos limiares no buffer do gateway é um processo complexo e depende fortemente das características do tráfego recebido. Portanto, os parâmetros do algoritmo devem ser ajustados a aplicações e a ambientes específicos.

3.3 Mecanismo de Anúncio de Capacidade

A abordagem proposta por Gerla e outros em [GTG93, GTG92, GTMG91] consiste em permitir que rajadas de tráfego entrem na rede ATM diretamente, ao invés de serem acumuladas no buffer de entrada, desde que haja capacidade livre suficiente no caminho da conexão. Para tanto, é definido um procedimento para relatar aos gateways informações de capacidades disponíveis nos enlaces, *anúncio de capacidade*. As *células abusivas*, ou seja, células que excedem a taxa de pico alocada ao canal virtual (VP - *Virtual Path*), têm grande chance de chegar aos seus destinos devido ao conhecimento que os gateways possuem das capacidades residuais. Os resultados dessa proposta de transmissão denominada *on-the-fly* aliada ao esquema de anúncio de capacidade, segundo os autores, são filas menores e atrasos menores, além do aumento da eficiência da rede.

Para se entender o mecanismo proposto, são necessárias algumas definições iniciais. A Capacidade Disponível num Enlace ATM é definida como:

$$CDE = C - CAE$$

em que C = capacidade do enlace e CAE = Capacidade Alocada no Enlace para os diversos circuitos virtuais. Note que a alocação de capacidade em um enlace é tipicamente uma mistura de alocação de taxa de pico, para conexões CBR, e alocação estatística, para conexões VBR. Assim sendo, CDE é a quantidade de capacidade que o comutador pode contar para satisfazer solicitações. Tipicamente, CDE só altera de valor quando há estabelecimento de novas conexões ou liberação de conexões existentes. A *capacidade disponível anunciada* pelo mecanismo de anúncio de capacidade é o mínimo entre as capacidades disponíveis nos enlaces ao longo do caminho da conexão.

Gerla e outros supõem que há um mecanismo de anúncio de capacidade na rede ATM que alimenta os gateways com informações atualizadas a respeito das capacidades disponíveis nos enlaces. Na seção 3.3.1 serão discutidos alguns mecanismos de anúncio de capacidade.

Quando a interconexão LAN/RDSI-FL/LAN é inicializada, são estabelecidos caminhos virtuais entre todos os gateways com uma capacidade alocada mínima, por exemplo 1 Mbps. No esquema básico, denominado *on-the-fly*, a capacidade da conexão não se altera. Entretanto, pode-se pensar em uma variante para o esquema básico que consistiria em solicitar mais capacidade se o tráfego da conexão aumentar muito. A capacidade seria solicitada baseada em medições de média e longa duração do tráfego. De maneira análoga, a capacidade alocada pode ser reduzida se o tráfego diminuir.

Essa variante, denominada *esquema híbrido*, tenta se ajustar às necessidades do usuário renegociando dinamicamente a capacidade alocada como no mecanismo de reserva. Uma diferença significativa em relação ao esquema de reserva, no mecanismo híbrido, é que quando ocorre uma rajada acima da taxa alocada ela não é armazenada no gateway para esperar a alocação de mais capacidade, ao invés disso, ela pode ser transmitida na taxa anunciada como disponível pela rede. Se a condição de sobrecarga, causada pela rajada, persistir além de um determinado tempo (t_{crit_a}) escolhido como sendo pelo menos uma ordem de grandeza maior que o tempo de renegociação de capacidade, o gateway solicita um incremento de capacidade adequado para acomodar o tráfego extra observado durante o período de sobrecarga.

Uma outra diferença é que a renegociação de capacidade não é disparada pela ocupação no buffer de entrada, mas sim por medições no tráfego. Além disso, o gateway sabe quanta capacidade está disponível no caminho da conexão, pelo mecanismo de anúncio, e, portanto, pode solicitar toda a capacidade necessária de uma só vez, desde que ela esteja disponível, ao invés de solicitar vários pequenos incrementos.

Mesmo prevendo a renegociação de capacidade, o mecanismo híbrido procura explorar ao máximo a capacidade disponível na rede. Uma solicitação de capacidade é disparada somente se a taxa recebida for superior a um determinado percentual (*Allocation Threshold*) da capacidade anunciada como disponível. Por exemplo, se a capacidade disponível anunciada for igual a 50Mbps e o percentual for de 20%, o tráfego recebido pode exceder a taxa alocada até o valor de 10Mbps e será transmitido na rede sem disparar uma nova solicitação de capacidade. Como mencionado anteriormente, a solicitação de capacidade

será disparada somente se essa situação permanecer por um período superior a t_{crit_a} .

A liberação de capacidade é feita sempre que o mecanismo percebe que a taxa alocada é superior à taxa recebida. Uma solicitação de liberação de capacidade é disparada sempre que a taxa recebida fica abaixo da taxa alocada por um período de tempo igual a t_{crit_d} . A quantidade de capacidade a ser liberada é definida como um parâmetro do esquema e corresponde a um percentual da taxa alocada (*Deallocation Threshold*).

A seguir serão discutidos os elementos básicos do mecanismo de anúncio de capacidade: esquemas de anúncio de capacidade e controle de acesso. Maiores detalhes sobre o dimensionamento e características adicionais deste mecanismo serão dados nos capítulos 5 e 6.

3.3.1 Esquemas de Anúncio de Capacidade

As informações das capacidades devem ser passadas aos gateways periodicamente e eficientemente pela rede para auxiliá-los no gerenciamento do tráfego não orientado a conexão. Duas técnicas são apresentadas em [GTG93]: sondagem e difusão (*broadcast*) de capacidades.

No *esquema de sondagem de capacidades (bandwidth probing)*, supõe-se que existe um canal de controle dedicado entre os gateways. O gateway destino envia periodicamente, em direção ao gateway origem, uma célula de gerenciamento de recursos, pelo canal de controle, e os comutadores ATM, ao longo da conexão, atualizam essa célula para que ela transporte a menor capacidade disponível encontrada. Considere, por exemplo, os enlaces e as capacidades disponíveis mostrados na figura 3.3. Neste caso, a capacidade que deve ser transportada pela célula de gerenciamento de recursos, capacidade mínima no caminho da conexão, é a de 5 Mbps disponível no enlace 3.

No *esquema de difusão de capacidades*, os comutadores da rede ATM enviam periodicamente a todos os demais comutadores da rede as capacidades disponíveis nos enlaces que passam por eles. Os gateways devem manter uma tabela com os enlaces pertencentes às conexões e atualizá-la de acordo com as informações recebidas dos comutadores da rede.

3.3.2 Controle de Acesso

A informação da capacidade disponível é utilizada pelo gateway origem e pela rede, para controlar o acesso do tráfego não orientado à conexão. A idéia básica é permitir a trans-

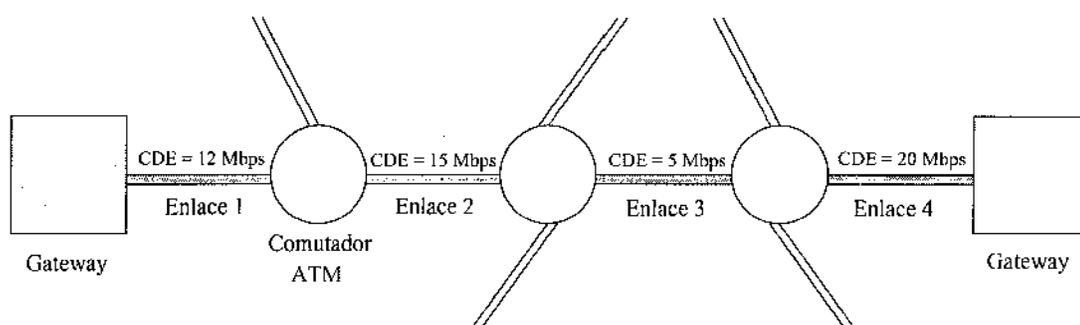


Figura 3.3: Funcionalidade do Mecanismo de Anúncio de Capacidade

missão de células abusivas, células que excedem a taxa alocada, desde que elas sejam marcadas como de baixa prioridade ($CLP = 1$), e desde que a taxa total transmitida (células com capacidade garantida + células abusivas) fique abaixo da taxa de transmissão permitida pelo mecanismo (taxa alocada + taxa anunciada como disponível na rede).

O controle de acesso do tráfego é feito por dois limitadores de taxa de pico, por exemplo, balde furado [Tur86, Jr93, Par94], em série como mostrado na figura 3.4. Cada limitador deve possuir um pequeno buffer. Além disso, é necessário um buffer antes do policiador de dois estágios. O primeiro controlador é ajustado para policiar a taxa de transmissão permitida pelo mecanismo (taxa alocada + taxa anunciada). Ele armazena, ou descarta se o buffer estiver cheio, todas as células que excedem a taxa de transmissão permitida. O segundo controlador é ajustado para policiar a taxa que foi pré-alocada à conexão. Ele marca como abusiva todas as células que excedem esta taxa.

Dentro da rede ATM, o controle também é feito por dois policiadores em série. Assume-se que os comutadores interceptaram o mecanismo de anúncio para a conexão em questão e que foram calculadas as duas taxas dos policiadores. O comutador deve verificar se a taxa total transmitida está abaixo do primeiro limite, caso não esteja, as células devem ser descartadas. Ele também deve verificar se a taxa de células não marcadas está abaixo da taxa alocada, caso não esteja, as células devem ser marcadas.

3.3.3 Avaliação

Diferente do mecanismo de reservas, o mecanismo de anúncio de capacidade exige que sejam implementadas características adicionais na RDSI-FL. Apesar disso, o mecanismo

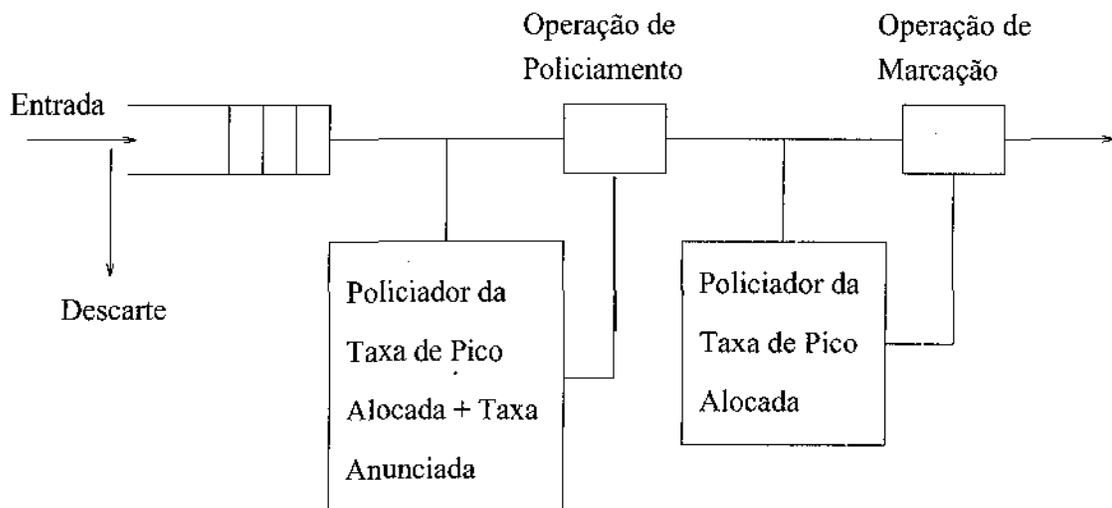


Figura 3.4: Esquema de policiamento do tráfego com dois baldes furados em série.

apresenta vários pontos positivos, sendo que os principais estão relacionados abaixo.

- Dispensa a utilização de grande quantidade de memória no gateway para armazenar os dados provenientes do ambiente não orientado a conexão;
- Provoca atrasos menores aos dados que trafegam pelo gateway, pelo fato de evitar armazenamentos no gateway;
- Diminui sensivelmente a carga de processamento de controle nos comutadores, por não renegociar a capacidade alocada na rede ATM;
- Utiliza melhor a capacidade da rede ATM por não causar grandes *overheads* com renegociações da capacidade alocada e também por explorar melhor a multiplexação estatística ao longo da conexão.

Além da necessidade da alteração de algumas características da RDSI-FL, alguns outros pontos negativos podem ser ressaltados no mecanismo de anúncio de capacidades:

- Exige características adicionais nos comutadores ATM, pelo fato de utilizar um mecanismo de policiamento diferenciado de dois estágios e utilizar o mecanismo de anúncio de capacidade;

- Provoca perda de células abusivas dentro da rede ATM. Por esse fato, a rede deve prever um esquema diferenciado de tarifação para o tráfego não orientado a conexão;
- Introduce atrasos adicionais ao restante do tráfego transportado na rede ATM, tráfego orientado a conexão, pelo fato de transmitir dados além da capacidade alocada à sua conexão.

3.4 Mecanismo Baseado em Estatística de Atraso Fim a Fim

Este esquema, proposto por Frost e Mullen [FM92], é uma modificação da técnica de anúncio de capacidades proposta em [GTG93] e discutida anteriormente. Neste esquema, ao invés de se exigir dos comutadores o envio das capacidades livres nos enlaces, uma estimativa das capacidades disponíveis é feita com base em medidas de atrasos das células.

Como na proposta de anúncio, um mecanismo com dois baldes furados em série é utilizado, veja figura 3.4. As células que excedem a taxa de pico negociada, detectadas pelo segundo balde furado, serão marcadas como de baixa prioridade, portanto passíveis de serem descartadas, mas serão transmitidas na rede ATM a uma taxa igual a do primeiro balde furado.

O primeiro balde furado é ajustado para policiar uma taxa igual a estimativa da capacidade livre na rede mais a taxa de pico negociada. Isso assegura que as células nunca serão inseridas no enlace a uma taxa superior a melhor estimativa da capacidade disponível na rede.

Capacidades serão liberadas de acordo com o mecanismo de liberação baseado na média das últimas observações proposto por Mongiovi e outros [MFT91], veja seção 3.2.

3.4.1 Estimativa da Capacidade Disponível

A diferença crítica entre o esquema proposto por Frost e Mullen [FM92] e o proposto em [GTG93] é o uso de uma estimativa da capacidade livre disponível no enlace ao invés da transmissão explícita desta informação pelos comutadores ao longo do caminho da origem ao destino. No esquema de Frost e Mullen, células de sondagem também são utilizadas, mas a função dessas células é transportar uma amostra do atraso fim-a-fim da origem ao destino. A origem controla a taxa de geração de células de sondagem. Taxas sugeridas

pelos autores são em torno de 5 % da carga oferecida. As células de sondagem possuem uma marca de tempo e são transmitidas da origem para o destino. O destino usa a marca de tempo para calcular o atraso enfrentado pela célula, que é então transmitida de volta para a origem. Na origem, essas amostras são utilizadas para construir uma estimativa do atraso médio.

Para o cálculo da estimativa, os autores utilizam um modelo de fila $M/D/1$ para representar o processo de armazenamento das células ao longo da conexão. Claramente o modelo $M/D/1$ não reflete o processo de armazenamento nas várias filas. Um modelo $M/D/1$ (um único servidor) assume que todo o armazenamento ocorre em um único buffer no caminho da transmissão. Pelo fato do armazenamento ocorrer em várias filas, cada uma com o seu próprio servidor, esta abordagem resultará numa estimativa conservadora para a capacidade livre disponível. A capacidade livre disponível é estimada através da seguinte fórmula:

$$CDE = (r_{max} - r_e)C$$

$$r_e = \frac{2(r - 1)}{2r - 1} \qquad r = \frac{[D - T_p - (T_t N_h)]}{T_t}$$

- r_{max} – Utilização Máxima
- r_e – Utilização Estimada
- C – Capacidade do Enlace em bps
- r – Estimativa Normalizada do Atraso de Fila
- D – Atraso Fim-a-Fim
- T_p – Atraso de Propagação do Enlace
- T_t – Tempo de Transmissão
- N_h – Número de Nós entre os Gateways Origem e Destino

3.4.2 Controle de Solicitação e Liberação de Capacidade

Um dos objetivos do esquema de alocação dinâmica de capacidade é combinar a capacidade negociada com as necessidades de transmissão. Para atingir esse objetivo, a taxa de chegada do tráfego na origem deve ser monitorada e a capacidade negociada deve ser aumentada ou diminuída se o tráfego aumentar ou diminuir. Para o processo de monitoração é utilizado um esquema de média móvel idêntico ao proposto em [MFT91]. Tanto o mecanismo de solicitação quanto o de liberação de capacidade são idênticos aos seus correspondentes do mecanismo de anúncio de capacidade híbrido.

3.4.3 Avaliação

Por se tratar de uma variante do mecanismo de anúncio de capacidades, o mecanismo baseado em estatística de atraso fim-a-fim possui os mesmos pontos positivos e negativos do mecanismo original, com a diferença de que não possui como ponto negativo a exigência de um mecanismo explícito de anúncio de capacidades. Um ponto crítico no mecanismo baseado em estatística de atraso fim-a-fim é a estimativa da capacidade disponível. Essa estimativa é feita tendo-se como base um modelo simplificado da rede, o modelo $M/D/1$, ou seja, assume-se que o processo de entrada é Markoviano e que todo o armazenamento ocorre em uma única fila. Como o objetivo deste trabalho é avaliar o desempenho de mecanismos de gerenciamento de capacidade utilizando o modelo de tráfego auto-semelhante e considerando que este modelo possui características diferentes de modelos Markovianos, este mecanismo não será mais considerado nos capítulos subseqüentes, pois não se tem idéia de quão boa seria a estimativa neste novo cenário.

3.5 Protocolo de Reserva Rápida de Buffer

Um problema crucial com a maioria das propostas para gerenciamento de capacidade e controle de congestionamento é que elas não abordam diretamente a necessidade de se alocar recursos de rede visando preservar a integridade da rajada como um todo. Uma exceção pode ser encontrada na proposta de Boyer [Boy90, BT92], que utiliza um mecanismo de reserva rápida de capacidade para lidar com tráfegos em rajadas com baixas taxas de pico.

Turner [Tur92] adota uma abordagem semelhante a de Boyer, mas que se aplica ao caso

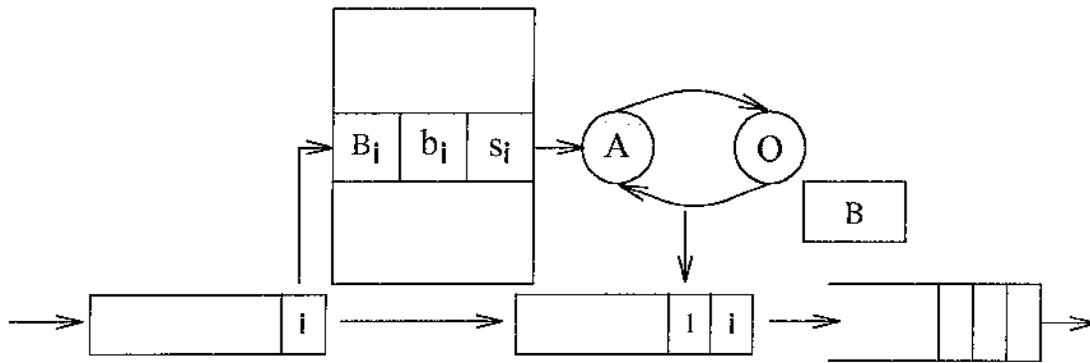
mais difícil do tráfego em rajadas com taxas de pico que podem ser uma grande fração da capacidade do enlace. Ele também propõe uma implementação na qual a reserva é feita conforme os dados são enviados. Isso elimina a necessidade de mensagens explícitas de controle, simplificando a implementação e permitindo respostas mais rápidas ao tráfego do usuário.

Para preservar a integridade das rajadas, a rede deve detectar e monitorar a atividade em diferentes circuitos virtuais. Esse objetivo é atingido associando a cada circuito virtual de um determinado enlace uma máquina de estados, com dois estados. Os dois estados da máquina são ATIVO e OCIOSO. Além disso, o buffer do comutador é visto como um conjunto de *slots*, sendo que cada *slot* corresponde a uma célula. Quando um determinado circuito virtual está ativo, é alocado um número pré-estabelecido de *slots* no buffer e é garantido o acesso aos *slots* até que o circuito se torne inativo. Um circuito virtual se torna inativo quando ocorre uma transição do estado ATIVO para o estado OCIOSO. As transições entre os estados ocorrem na recepção de células do usuário marcadas como sendo início de rajada ou final de rajada. Outros possíveis tipos de células são células de meio de rajada e células solitárias. A determinação dos diferentes tipos de células é feita com base em informações das camadas superiores. As células solitárias são consideradas de baixa prioridade e são transmitidas apenas se houver *slots* não alocados disponíveis. Uma transição forçada do estado ATIVO para o OCIOSO ocorre, também, se não houver recepção de células em um circuito virtual por um período fixo de tempo (*timeout*).

O mecanismo de reserva de buffer é mostrado na figura 3.5. Para um determinado circuito virtual i , o mecanismo armazena o número de *slots* necessários quando o circuito está ativo (B_i), o número de *slots* atualmente em uso por células não marcadas (b_i) e uma variável de estado (s_i : ATIVO, OCIOSO). O mecanismo também mantém o número de *slots* não alocados no buffer (B).

A máquina de estados, mostrada na figura, exerce um papel chave no mecanismo e sua operação é descrita a seguir.

- Quando uma célula de início de rajada é recebida e o circuito virtual está no estado OCIOSO ($s_i = \text{OCIOSO}$), duas situações podem ocorrer:
 - Se $B - B_i < 0$, ou seja o número de *slots* disponíveis é menor do que o exigido pela conexão, a célula é descartada.
 - Se $B - B_i \geq 0$, ou seja há *slots* suficientes para a conexão, s_i passa do estado

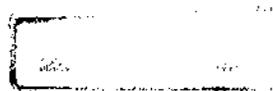


B : Número de slots não alocados no buffer B_i : Número de slots para o VC i quando ativo
 s_i : Estado do VC i (A = Ativo ou O = Ocioso) b_i : Número de slots em uso pelo VC i

Figura 3.5: Mecanismo de Reserva de Buffer

OCIOSO para o estado ATIVO, um temporizador para a conexão é inicializado e B_i é subtraído de B . Se $b_i < B_i$, b_i é incrementado e a célula é colocada no buffer. Se $b_i = B_i$, a célula é marcada e colocada no buffer.

- Se uma célula de início ou meio de rajada é recebida quando o circuito virtual está no estado ATIVO, o temporizador é reinicializado e a célula é colocada no buffer, podendo ser marcada ou não. A célula é marcada se, quando ela tiver chegado, b_i for igual a B_i , ou seja a conexão já utilizou todos os *slots* reservados para ela, caso contrário ela não é marcada e b_i é incrementado.
- Se uma célula de meio ou final de rajada é recebida quando o circuito virtual está no estado OCIOSO, ela é descartada. Esta situação ocorre devido ou a perda da célula de início de rajada ou pela impossibilidade da rede atender a solicitação de transmissão da rajada.
- Se uma célula de final de rajada for recebida quando o circuito virtual está no estado ATIVO ou se o temporizador atingir o seu valor máximo, s_i passa para o estado OCIOSO e B_i é adicionado a B , ou seja os *slots* utilizados pela conexão ficam disponíveis no buffer.
- Se uma célula solitária for recebida, ela é marcada e colocada no buffer. Células



solitárias são sempre consideradas de baixa prioridade.

- Toda vez que uma célula não marcada é retirada do buffer e transmitida no enlace, o valor de b_i da conexão é decrementado.
- O tempo máximo do temporizador (*timeout*), que força a transição do estado ATIVO para o estado OCIOSO, é determinado por variações de atrasos da rede. Turner sugere um tempo igual ao tempo de transmissão de algumas centenas de células no enlace. Isso daria um tempo na casa de alguns milissegundos em um enlace com taxa superior a 100Mbps.

No caso mais comum de transmissões na rede, ocasionado pela transmissão de unidades de dados de protocolos de camadas superiores, as células são transmitidas na forma $IM \dots MF$, em que I denota a célula de início da rajada, M células do meio da rajada e F a célula de final da rajada. Neste caso, se quando a célula de início chega ao comutador não há *slots* disponíveis para a rajada, a rajada inteira é descartada. A máquina de estados foi projetada para detectar outras situações também. Por exemplo, se rajadas nas formas $II \dots IF$ e $IM \dots MIM \dots MIM \dots MF$ são recebidas, o mecanismo reinicializa suas variáveis a cada célula de início e a parte final da rajada, correspondente a uma unidade completa da camada superior, é transmitida sem problemas. Rajadas desses tipos podem ser geradas em decorrência de retransmissões de unidades de dados das camadas superiores.

3.5.1 Avaliação

A grande vantagem do mecanismo de reserva rápida de buffer é a manutenção da integridade de uma rajada como um todo. Como o mecanismo deixa a cargo dos comutadores ATM o descarte de células abusivas, ele possui como vantagem adicional a não formação de grandes filas na origem. Entretanto, o mecanismo exige características adicionais nos comutadores da rede, sendo que a inclusão dessas características (máquina de estados e determinação dos diferentes tipos de células) complicaria fortemente o projeto dos mesmos. Além disso, o mecanismo está associado a um esquema sofisticado de controle de admissão de conexões [Tur92]. Para que um circuito virtual seja estabelecido, a fonte deve especificar a taxa de pico no circuito. Com base nessa informação e no número de circuitos já estabelecidos, a rede decide se a conexão é aceita ou não, só após o aceite da conexão é que as transmissões são monitoradas pela máquina de estado.

Como o objetivo deste trabalho não é analisar esquemas de controle de admissão de conexões e pelo fato da eficácia deste mecanismo estar diretamente associada a um esquema específico de admissão, este mecanismo não será mais considerado nos capítulos seguintes.

Capítulo 4

Modelos de Tráfegos

Um dos principais componentes na realização de simulações de redes e que dá credibilidade à análise efetuada é o modelo de tráfego utilizado para alimentar enlaces, comutadores, roteadores, etc. Até o presente momento, não há modelos precisos para todas as espécies de aplicações que podem existir em uma rede. Muitos trabalhos têm sido realizados com o objetivo de se entender melhor as características estatísticas dos tráfegos gerados por novas aplicações [Gar93, GW94, LTWW94, BSTW95], visando tornar o estudo do comportamento das partes constituintes de uma rede mais preciso e mais próximo da realidade.

Na maioria dos trabalhos envolvendo análise de desempenho de redes, assume-se que as chegadas de pacotes são processos de Poisson, devido às propriedades teóricas atrativas que tais processos possuem [FM94]. Trabalhos recentes, entretanto, argumentam de forma convincente que o tráfego gerado por uma rede local é melhor representado por processos auto-semelhantes (*self-similar*) [LTWW94], ou seja, é *fractal* por natureza. Os processos auto-semelhantes possuem propriedades teóricas bem diferentes daquelas apresentadas por processos de Poisson. Intuitivamente, a característica crítica do tráfego auto-semelhante é que não existe um comprimento natural de rajada, o tráfego é explosivo (*bursty*) e possui uma aparência semelhante em qualquer escala de tempo, desde alguns milissegundos a minutos e horas. Em um outro trabalho, Paxson e Floyd [PF95] sugerem que as mesmas características, evidenciadas em [LTWW94], estão presentes no tráfego que sai de uma LAN e é enviado para uma rede de longa distância.

Conforme mostrado em [LTWW94], o tráfego auto-semelhante se comporta de maneira bem diferente daquela pregada por modelos para tráfego de pacotes atualmente encon-

trados na literatura. A hipótese normalmente assumida de que o tráfego se torna menos explosivo quando se aumenta o número de fontes agregadas está bem longe da realidade. De fato, a explosividade (*burstiness*) de um tráfego auto-semelhante aumenta quando o número de fontes ativas aumenta, característica evidenciada no tráfego Ethernet [LTWW94].

Leland e outros [LTWW94] argumentam que o tráfego gerado em uma rede Ethernet é auto-semelhante com base em estudos realizados em uma enorme quantidade de dados coletados no *Bellcore Morris Research and Engineering Center*. Em um dos experimentos os dados foram registrados ao longo de 27 horas. A coleta dos dados foi feita através de um monitor de boa precisão que registrava todos os pacotes Ethernet enviados na rede [FL91]. Para maiores detalhes sobre o ambiente de trabalho do centro de pesquisa e sobre o monitor, veja [FL91, LW91, LTWW94]. A figura 4.1 mostra um dos *traces* coletados na *Bellcore* em diferentes níveis de agregação. É possível perceber claramente, pela figura, a natureza explosiva desse tipo de tráfego.

Vídeo sob demanda é considerado um dos serviços que deve popularizar a RDSI-FL. Para entender melhor o comportamento do tráfego gerado por esse serviço, alguns trabalhos têm sido realizados [GW94, BSTW95] visando caracterizar estatisticamente os dados gerados por um codificador de vídeo. Garret e Willinger [GW94] apresentam um estudo detalhado sobre as características do tráfego gerado na transmissão de vídeo usando compressão de dados, e argumentam que este tráfego também apresenta características de auto-semelhança. Essas evidências são reforçadas por Beran e outros [BSTW95] que apresentam estudos de vários *traces* de vídeos obtidos utilizando diferentes técnicas de compressão. Tais estudos indicam que o tráfego gerado para reprodução de vídeo apresenta características auto-semelhantes independente da técnica de compressão utilizada. Esta aplicação reforça a necessidade de um estudo detalhado de processos auto-semelhantes e de suas implicações na análise de desempenho de redes de computadores.

Este capítulo está organizado da maneira a seguir. A seção 4.1 apresenta como a modelagem de tráfego está sendo vista no âmbito de redes ATM e, também, o modelo mais utilizado em trabalhos teóricos. A seção 4.2 revisa a teoria de processos auto-semelhantes, apresenta as principais ferramentas estatísticas para se trabalhar com tais processos e descreve dois algoritmos para geração sintética de dados com características de auto-semelhança.

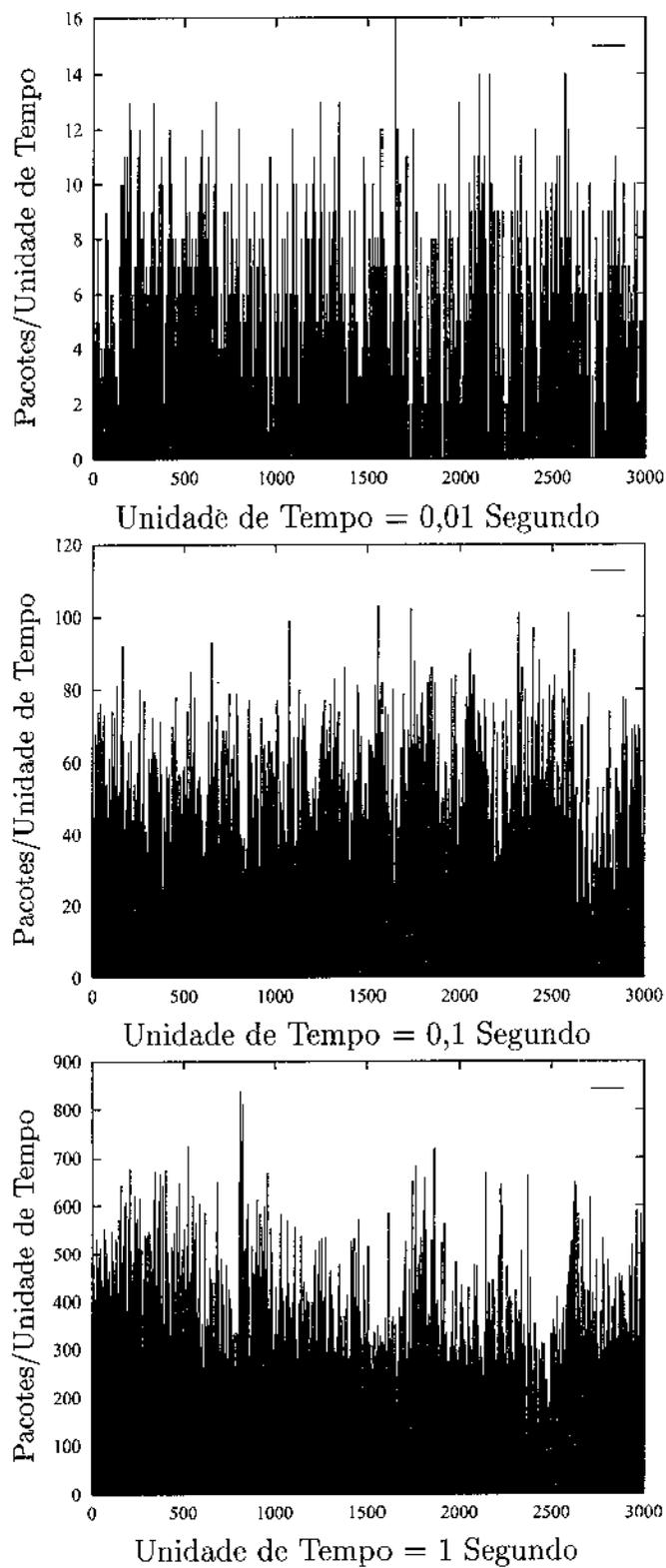


Figura 4.1: Tráfego Ethernet coletado na *Bellcore* em diferentes escalas de tempo.

4.1 Modelo Genérico de Tráfego para ATM

Estatisticamente, fontes de tráfego ATM podem ser caracterizadas em três níveis [Mon90]: no nível de chamada, no nível de rajada, e no nível de chegada de células como mostrado na figura 4.2.

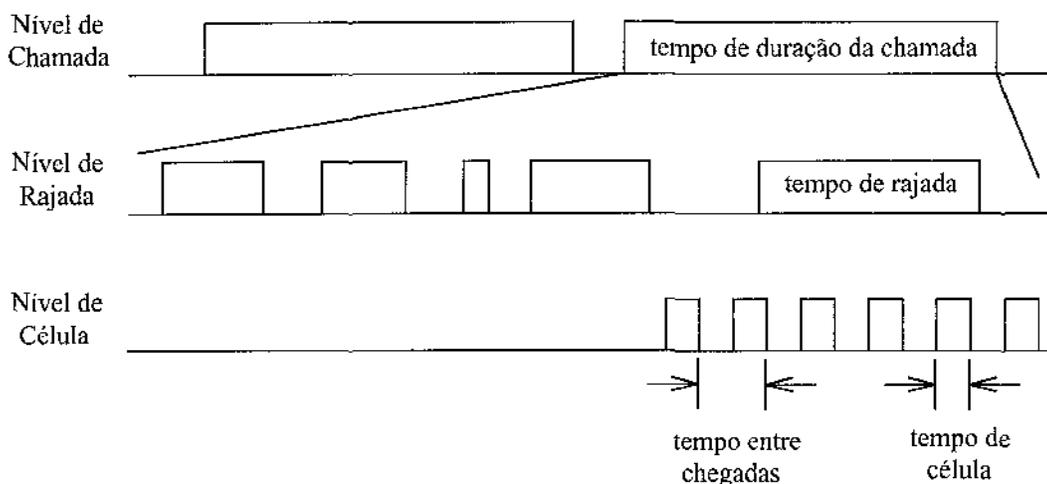


Figura 4.2: Níveis de Tráfegos em ATM.

O tempo de duração de chamada, no nível de chamada, é o tempo entre o momento em que a chamada é estabelecida e o momento em que a chamada é liberada. O tempo de rajada, no nível de rajada, é definido como a duração da transmissão de um único pedaço de informação (por exemplo, um quadro de vídeo ou um arquivo de dados). O tempo de célula, no nível de célula, é o tempo gasto para se transmitir uma única célula na rede ATM. Como no ATM as células possuem tamanhos fixos, cada tempo de transmissão de célula é constante e depende da taxa de transmissão do enlace.

Uma representação Markoviana de dois estados foi proposta por vários pesquisadores [BS91, Mon90] para modelar o tráfego em rajada, tanto no nível de rajada quanto no nível de célula. No modelo de dois estados, um estado representa o período ativo (em que a fonte transmite na taxa de pico) e o outro o período ocioso (em que não há transmissão, silêncio). Várias fontes de tráfegos podem ser caracterizadas como tráfego em rajada. Os exemplos mais conhecidos são os serviços de vídeo com compactação, transmissão de dados, e transmissão de voz em pacotes codificada com detecção de silêncio.

Seguindo a notação de [GRF89], uma fonte em rajada é caracterizada pelos seguintes

parâmetros: a taxa de pico em bits por segundo (B_p), a taxa média em bits por segundo (B_m), e a duração média de rajada em segundos (T). O valor $b = B_p/B_m$ denota o grau de explosividade (*burstiness*) das fontes. Quanto mais explosiva for a fonte, maior será o grau de explosividade. A unidade de tempo normalmente empregada neste modelo é o tempo de transmissão de uma célula em um enlace da rede, o que, obviamente, depende do enlace.

Modelos de tráfegos de dados em ATM não estão claramente definidos e nem há um consenso na literatura sobre o melhor modelo a ser empregado. Em vários trabalhos [GRF89, BS91, Mon90] a caracterização do tráfego de dados é feita por um processo de Poisson, em que tanto o período ativo quanto o período de silêncio são distribuídos exponencialmente com médias T e $S = T(b - 1)$, respectivamente. Portanto, uma única fonte em rajada pode ser modelada através de uma cadeia de Markov com dois estados como a mostrada na figura 4.3, em que $\lambda = 1/S$ e $\mu = 1/T$ [Mon90].

O comprimento médio de rajada em número de células, denotado por L , se relaciona com os outros parâmetros pela equação:

$$L = \frac{B_p T}{n_{cel}} \tag{4.1}$$

em que n_{cel} é o comprimento em bits de uma célula ATM e é igual a 424 bits.

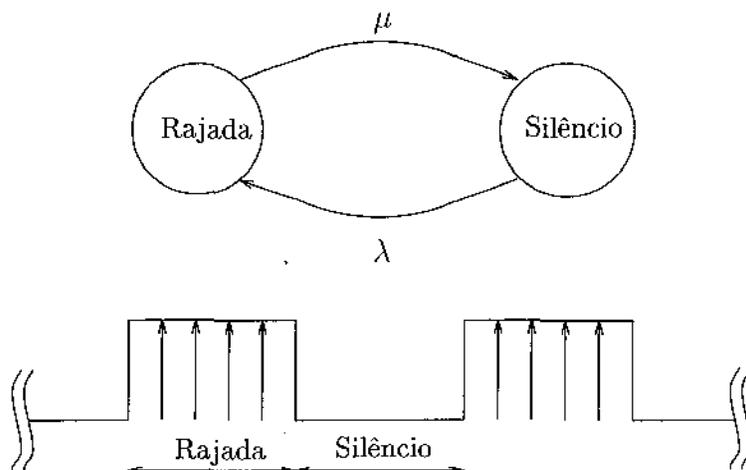


Figura 4.3: Tráfego em Rajada e Silêncio.

4.2 Modelo Auto-Semelhante

Para o melhor entendimento de auto-semelhança (*self-similarity*) e as características dos processos auto-semelhantes serão apresentados inicialmente alguns conceitos elementares de estatística e de séries temporais. Será dada apenas uma visão resumida desses conceitos, mesmo porque são assuntos extensos e com farta bibliografia, para maiores detalhes veja [Pri81, PPS86].

Suponha que se esteja estudando alguma quantidade X que varia ao longo do tempo, seja X_t esta quantidade no tempo t . A cada ponto no tempo t , X_t é uma variável aleatória e a forma completa de X_t , conforme t varia ao longo do tempo, é chamada de *processo aleatório* ou *processo estocástico*.

Um processo estocástico X é *estacionário* se as propriedades estatísticas do processo não mudam ao longo do tempo, ou seja, se suas funções de distribuições finito-dimensionais são invariantes por translações no tempo, isto é,

$$F(X(t_1 + \tau), \dots, X(t_n + \tau)) \stackrel{d}{=} F(X(t_1), \dots, X(t_n)) \quad (4.2)$$

o operador $\stackrel{d}{=}$ indica igualdade das distribuições finito-dimensionais.

Um processo estocástico é *estacionário em covariância*, ou *estacionário em sentido amplo*, ou *estacionário de segunda ordem* se as propriedades estatísticas de primeira e segunda ordem não mudam ao longo do tempo. Mais especificamente, um processo aleatório estacionário em covariância, $X_t (t = 1, 2, \dots)$ é um processo com média constante,

$$E[X_t] = \mu \quad (4.3)$$

com variância finita,

$$Var[X_t] < \infty \quad Var[X_t] = E[(X_t - \mu)^2] \quad (4.4)$$

e com uma função de autocorrelação,

$$\rho(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{E[(X_t - \mu)^2]} \quad (4.5)$$

que depende apenas de τ . A dependência apenas em τ significa que não importa o ponto no processo em que começamos analisar a correlação e sim a distância entre os pontos analisados.

Uma *série temporal* é uma realização ou trajetória de um processo aleatório. Ao conjunto de todas as possíveis realizações de um processo aleatório, dá-se o nome de *ensemble* [Pri81].

Com essas definições, é possível iniciar a discussão de auto-semelhança. As definições iniciais seguem o mesmo formato apresentado em [LTWW94]. Para um tratamento mais rigoroso do assunto veja [Cox84] e a extensa bibliografia em [WTE96].

Seja X um processo estocástico estacionário em covariância com média μ , variância σ^2 e função de autocorrelação $\rho(\tau)$. Em particular, assuma que $\rho(\tau)$ é da forma

$$\rho(\tau) \sim \tau^{-\beta} L(\tau), \quad \text{quando } \tau \rightarrow \infty \quad (4.6)$$

em que $0 < \beta < 1$ e $L(\tau)$ é uma função que varia muito pouco no infinito, ou seja,

$$\lim_{\tau \rightarrow \infty} \frac{L(\tau x)}{L(\tau)} = 1, \quad \text{para todo } x > 0. \quad (4.7)$$

Exemplos de funções com esse comportamento são logaritmos e constantes. Seja $X^{(m)} = (X_k^{(m)} : k = 1, 2, 3, \dots; m = 1, 2, 3, \dots)$ uma nova série temporal obtida a partir de X da seguinte maneira

$$X_k^{(m)} = \frac{X_{km-m+1} + \dots + X_{km}}{m}, \quad k = 1, 2, 3, \dots \quad (4.8)$$

$X_k^{(m)}$ é a média de m elementos de X , cada $X_k^{(m)}$ é obtido a partir de valores de X que não se sobrepõem. Seja $\rho^{(m)}(\tau)$ a função de autocorrelação de $X^{(m)}$.

O processo X é dito (*exatamente*) *auto-semelhante* com parâmetro de auto-semelhança $H = 1 - \frac{\beta}{2}$ se o processo agregado correspondente $X^{(m)}$ possui a mesma estrutura de correlação de X ,

$$\rho^{(m)}(\tau) = \rho(\tau) \quad (4.9)$$

para todo $m = 1, 2, 3, \dots; (\tau = 1, 2, 3, \dots)$. Em outras palavras, X é exatamente auto-semelhante se os processos agregados $X^{(m)}$ são indistinguíveis de X , pelo menos com relação as suas propriedades de segunda ordem [LTWW94].

Os processos auto-semelhantes foram introduzidos de forma teórica por Kolmogorov [Ber94]. Mandelbrot e seus colegas de trabalho [MN68, MW69a, MW69c] fizeram uso extensivo e popularizaram a utilização de processos auto-semelhantes em várias áreas de pesquisas, entre elas economia e hidrologia.

A idéia básica de auto-semelhança é bastante antiga, como pode ser visto em [Man83], e está presente também na computação gráfica [FFC92]. Uma forma geométrica é dita

deterministicamente auto-semelhante se as mesmas estruturas geométricas são observadas, independente da distância que a forma é observada.

No contexto de processos estocásticos, auto-semelhança é definida em termos da distribuição do processo. Seja X_t um processo estocástico com parâmetro tempo t contínuo. X_t é dito auto-semelhante com parâmetro de auto-semelhança H , se para qualquer fator positivo c , o processo reescalado com escala de tempo ct , $c^{-H}X_{ct}$, é idêntico em distribuição ao processo original X_t .

Isso significa que, para qualquer seqüência de pontos t_1, \dots, t_k , e qualquer constante positiva c , $c^{-H}(X_{ct_1}, X_{ct_2}, \dots, X_{ct_k})$ possui a mesma distribuição que $(X_{t_1}, X_{t_2}, \dots, X_{t_k})$. Portanto, amostras típicas de um processo auto-semelhante parecem qualitativamente a mesma, independente da distância em que são observadas. Ao contrário da auto-semelhança determinística, isso não significa que a mesma figura se repete exatamente conforme é observada mais proximamente e sim o que se tem é a mesma impressão para a figura independente da distância que é observada.

4.2.1 Propriedades dos Processos Auto-Semelhantes

Matematicamente, auto-semelhança se manifesta de várias maneiras equivalentes, a saber [LTWW94]:

1. *decrécimo lento das variâncias*: a variância da média amostral decai mais lentamente do que o recíproco do tamanho da amostra;
2. *dependência em longos intervalos*: a função de autocorrelação decai de forma hiperbólica ao invés de exponencial, isso implica que a soma das autocorrelações é infinita. Como será mostrado a seguir, essa característica tem conseqüências drásticas para testes estatísticos tradicionais e para cálculos de intervalos de confiança; e
3. *ruído 1/f*: a função de densidade espectral se comporta como $1/f$ próxima à origem, f indica a freqüência. Esta característica é na realidade uma manifestação de processos com dependências em longos intervalos.

Desde que essas são as principais características de processos auto-semelhantes, elas podem ser exploradas para se estimar os parâmetros de auto-semelhança, dando origem a abordagens tanto no domínio do tempo, análise R/S e análise das variâncias, quanto no

domínio da frequência, análise baseada no periodograma. Esses métodos serão apresentados na próxima seção.

Decréscimo Lento das Variâncias

Um dos principais resultados ensinado em um curso introdutório em estatística é que a variância da média amostral é igual a variância de uma observação dividida pelo tamanho da amostra. Em outras palavras, se X_1, \dots, X_n são observações com média comum $\mu = E[X_t]$ e variância $\sigma^2 = Var[X_t] = E[(X_t - \mu)^2]$, então a variância de $\bar{X} = \frac{1}{n} \sum_{t=1}^n X_t$ é igual a

$$Var[\bar{X}] = \frac{\sigma^2}{n} \quad (4.10)$$

Um segundo resultado elementar estudado é que a média populacional é estimada por \bar{X} , e para amostras suficientemente grandes o intervalo de confiança $(1 - \alpha)$ para μ é dado por

$$\bar{X} \pm Z_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \quad (4.11)$$

se σ^2 é conhecida e

$$\bar{X} \pm Z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}} \quad (4.12)$$

se σ^2 tem de ser estimada. Sendo que $s^2 = \frac{1}{n-1} \sum_{t=1}^n (X_t - \bar{X})^2$ é a variância amostral e $Z_{\frac{\alpha}{2}}$ é o quantil superior $(1 - \frac{\alpha}{2})$ da distribuição normal padrão.

O conjunto de condições, normalmente assumido verdadeiro, para que (4.10) seja válida é:

1. A média populacional $\mu = E[X_t]$ existe e é finita;
2. A variância populacional $\sigma^2 = Var[X_t]$ existe e é finita; e
3. X_1, \dots, X_n são não correlacionados, isto é $\rho(i, j) = 0$ para $i \neq j$, sendo que $\rho(i, j) = \frac{\gamma(i, j)}{\sigma^2}$ é a autocorrelação entre X_i e X_j , e $\gamma(i, j) = E[(X_i - \mu)(X_j - \mu)]$ é a autocovariância entre X_i e X_j .

O que é necessário saber é se uma ou mais dessas hipóteses não forem válidas até que ponto (4.10), (4.11) e (4.12) estarão erradas e como elas poderão ser corrigidas.

A hipótese de maior interesse para este estudo é a de número 3, e, portanto, é a que será estudada em maiores detalhes. Dessa forma, assume-se, a partir de agora, que as hipóteses 1 e 2 são válidas. Para maiores detalhes sobre processos com variâncias infinitas veja [ST94].

O que se deseja saber é o que acontece a (4.10) quando as observações são correlacionadas. Para \bar{X}_n ser significativo, assume-se que $E[X_t] = \mu$ é constante. A variância de \bar{X} é igual a

$$\text{Var}[\bar{X}] = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \gamma(i, j) = \frac{\sigma^2}{n^2} \sum_{i=1}^n \sum_{j=1}^n \rho(i, j) \quad (4.13)$$

Se a soma de todas correlações para $i \neq j$ for igual a zero, então

$$\sum_{i=1}^n \sum_{j=1}^n \gamma(i, j) = \sum_{i=1}^n \gamma(i, i) = \sum_{i=1}^n \sigma^2 = n\sigma^2 \quad (4.14)$$

Isso significa que, se

$$\sum_{i \neq j}^n \rho(i, j) = 0 \quad (4.15)$$

então (4.10) é válida. Em particular, este é o caso se X_1, \dots, X_n são mutuamente não correlacionados. Caso contrário, se (4.15) não vale, então a variância de \bar{X} é igual a

$$\frac{\sigma^2}{n} [1 + \delta_n(\rho)] \quad (4.16)$$

com o termo de correção

$$\delta_n(\rho) = \frac{1}{n} \sum_{i \neq j}^n \rho(i, j) \quad (4.17)$$

Se as correlações $\rho(i, j)$ dependem apenas da diferença $|i - j|$, *lag*, então (4.17) pode ser simplificada para [Ber94]

$$\delta_n(\rho) = 2 \sum_{\tau=1}^{n-1} \left(1 - \frac{\tau}{n}\right) \rho(\tau) \quad (4.18)$$

Neste estudo, somente processos com esse tipo de correlação serão considerados, ou seja, somente processos estacionários em covariância.

Pela definição de processos auto-semelhantes tem-se que as correlações desses processos é da forma (4.6) o que implica que a soma de todas as correlações quando $\tau \rightarrow \infty$ tende a infinito, ou seja,

$$\sum_{\tau=-\infty}^{\infty} \rho(\tau) = \infty. \quad (4.19)$$

Portanto, para processos com esse tipo de correlação diz-se que a variância da média amostral decai mais lentamente do que o recíproco do tamanho da amostra. As consequências para o cálculo de intervalos de confiança são desastrosas, como pode ser visto por (4.18), já que (4.16) tem de ser corrigida por um fator que tende a infinito.

Processos tradicionais, por outro lado, são caracterizados por uma função de autocorrelação da forma

$$\rho(\tau) \sim a^\tau \quad \text{quando } \tau \rightarrow \infty \quad (4.20)$$

e a é uma constante, $0 < a < 1$. É fácil de ver que quando τ tende a infinito (4.20) tende a zero e além disso a soma de todas as correlações é um valor finito. Portanto, (4.16) tem de ser corrigida por um fator finito, mas o resultado final da variância continua sendo proporcional ao inverso do comprimento da amostra.

Dependências em Longos Intervalos

Seja X_t um processo estacionário para o qual a seguinte afirmação é válida: Existe um número real α , $0 < \alpha < 1$, e uma constante $c_\rho > 0$ tal que

$$\lim_{\tau \rightarrow \infty} \frac{\rho(\tau)}{c_\rho \tau^{-\alpha}} = 1 \quad (4.21)$$

então X_t é chamado de *processo estacionário com memória, com dependências em longos intervalos*, ou um *processo estacionário com decréscimo lento das correlações*.

O conhecimento das covariâncias é equivalente ao conhecimento da densidade espectral f no domínio das frequências [Pri81]. Portanto, dependência em longos intervalos pode, também, ser definida impondo uma condição na densidade espectral.

Seja X_t um processo estacionário para o qual a seguinte afirmação é válida: Existe um número real β , $0 < \beta < 1$, e uma constante $c_f > 0$ tal que

$$\lim_{\lambda \rightarrow 0} \frac{f(\lambda)}{c_f \lambda^{-\beta}} = 1 \quad (4.22)$$

então X_t é chamado de *processo estacionário com memória, com dependências em longos intervalos*, ou um *processo estacionário com decréscimo lento das correlações*.

4.2.2 Métodos Estatísticos para Teste de Auto-Semelhança

De um ponto de vista teórico, variâncias que decaem lentamente, dependência em longos intervalos e uma densidade espectral da forma (4.22) são manifestações diferentes de

uma mesma propriedade, auto-semelhança, do processo estacionário em covariância X [LTWW94]. Com isso, a verificação da presença de auto-semelhança em um processo pode ser feita de três maneiras diferentes: (1) análise das variância dos processos agregados $X^{(m)}$, (2) análise no domínio do tempo baseada na estatística R/S, e (3) análise do periodograma no domínio da frequência. Nesta seção serão apresentados alguns testes para se inferir o grau de auto-semelhança de um processo estocástico qualquer.

Para ilustrar as características dos testes que serão apresentados e para salientar as diferenças entre dados que exibem características de auto-semelhança de dados gerados por processos convencionais serão utilizados dois *traces* de dados representativos dos dois tipos de processos mencionados. Para o caso auto-semelhante, será utilizado um dos *traces* de tráfego Ethernet da *Bellcore*, que, como mostrado em [LTWW94], exibe nitidamente características de processos auto-semelhantes. Este *trace* é público e está disponível na Internet via ftp em `ftp://thumper.bellcore.com/pub/world/wel/lan_traffic`.

Para representar os processos convencionais, foi gerado um *trace* baseado no modelo de Poisson apresentado na seção 4.1. Inicialmente, foram gerados cem *traces*, sendo que os parâmetros utilizados para cada *trace* são os seguintes: taxa de pico (B_p) igual a 10Mbps, grau de explosividade (b) igual a 10 e tamanho médio dos pacotes igual a 4800 bits. Os tempos entre chegadas de pacotes e os comprimentos dos pacotes foram gerados a partir de variáveis aleatórias independentes e com distribuição exponencial. Em seguida os cem *traces* foram agregados para formar um novo *trace* em que os dados representam o número de pacotes recebidos em intervalos de 10ms.

Uma outra característica, além das mencionadas, que pode ser apreciada em dados amostrais, e que é normalmente utilizada por estatísticos, é a forma da função de autocorrelação do processo que se deseja estudar. Como visto anteriormente, a autocorrelação de processos auto-semelhantes possui uma característica bem diferenciada de processos convencionais, ela tende a zero de forma hiperbólica quando o *lag* aumenta ao contrário dos processos convencionais em que a correlação tende a zero de forma exponencial. A figura 4.4 mostra claramente que as correlações calculadas para os dados gerados no experimento caem rapidamente para zero, ao passo que as correlações calculadas para os dados reais possuem um comportamento bem mais complicado, como mostrado na figura 4.5. Uma análise inicial dos gráficos sugeriria a um estatístico a exigência de um estudo mais detalhado da presença de outras características, o que levaria, como de fato levou, à conclusão de auto-semelhança.

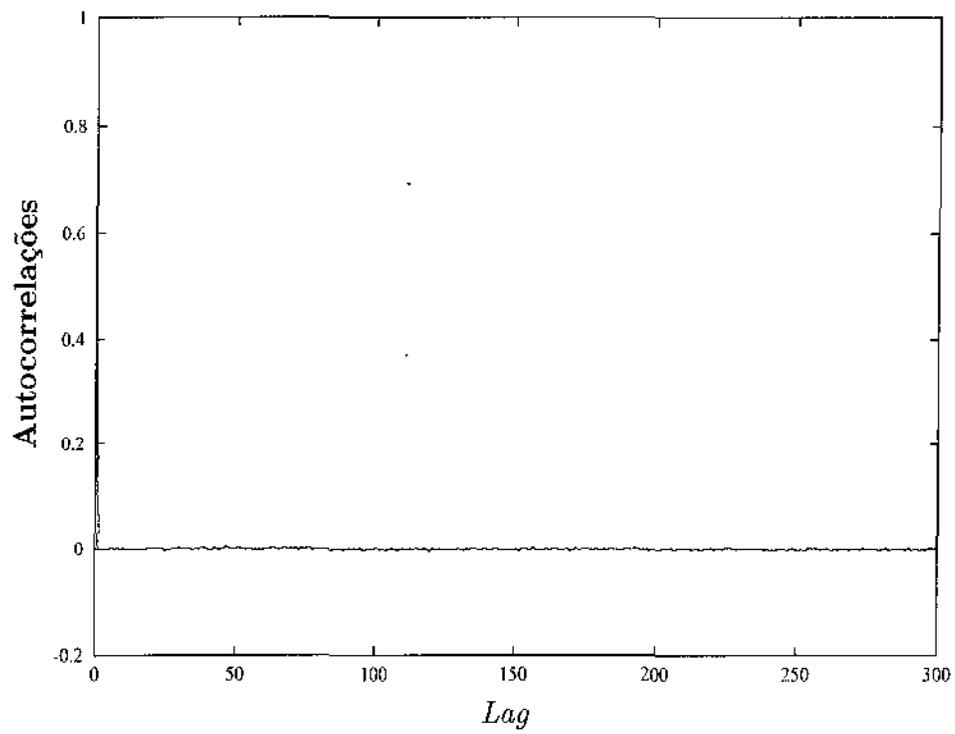


Figura 4.4: Autocorrelações para os dados gerados no experimento.

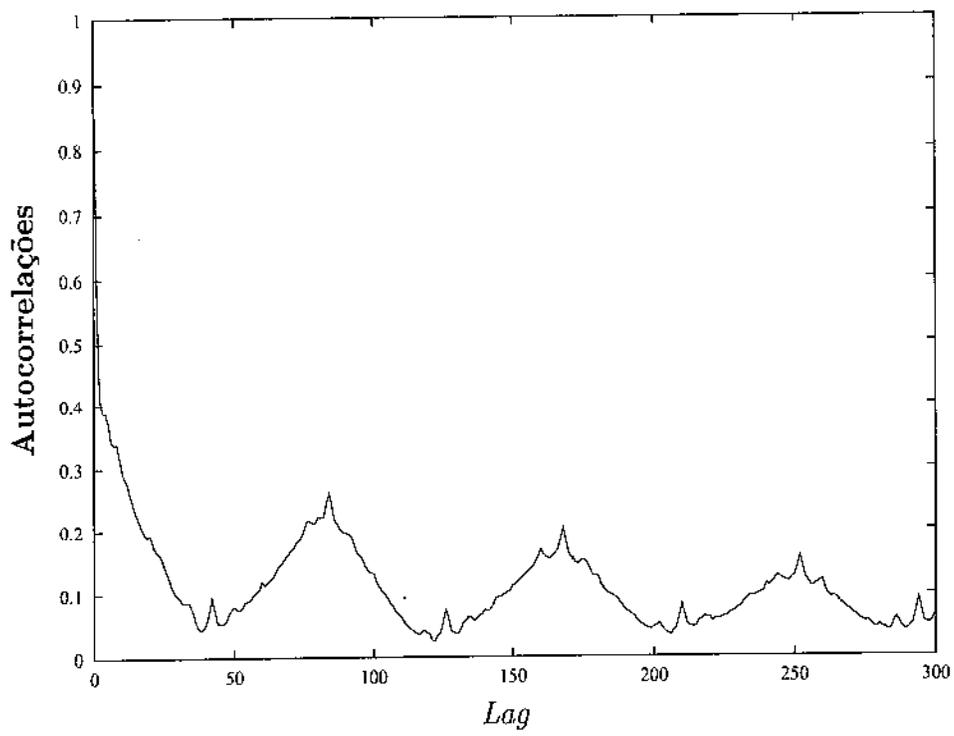


Figura 4.5: Autocorrelações para os dados Ethernet.

De maneira análoga à utilização da função de autocorrelação, o espectro de potências pode ser utilizado para se ter uma idéia inicial dos dados. As figuras 4.6 e 4.7 mostram os espectros estimados para os dois tráfegos. É possível visualizar o comportamento uniforme próximo à origem do espectro do tráfego do experimento. No caso do tráfego Ethernet, o espectro apresenta singularidade próximo à origem (*ruído 1/f*).

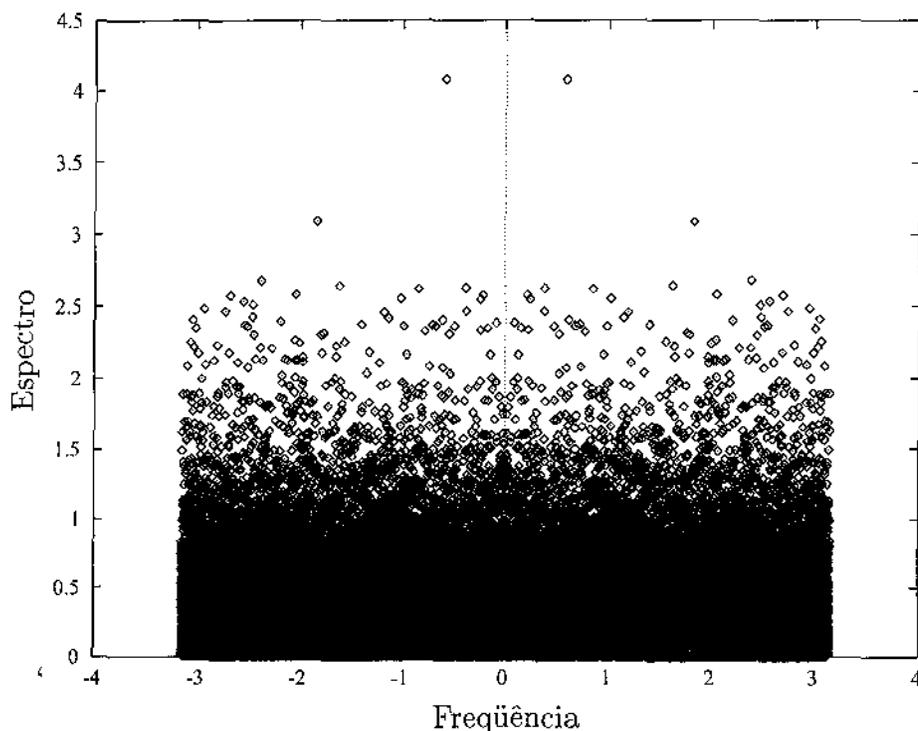


Figura 4.6: Espectro de potências estimado do tráfego Poisson gerado no experimento.

Gráfico da Variância X Tempo

Como visto na seção 4.2.1, uma das propriedades de processos auto-selhantes é que a variância da média amostral converge mais lentamente para zero do que n^{-1} , mais especificamente:

$$\text{Var}[\bar{X}_n] = cn^{-\beta} \quad (4.23)$$

com $c > 0$ e $0 < \beta < 1$. Esse resultado sugere o seguinte método para estimar o grau de auto-semelhança H :

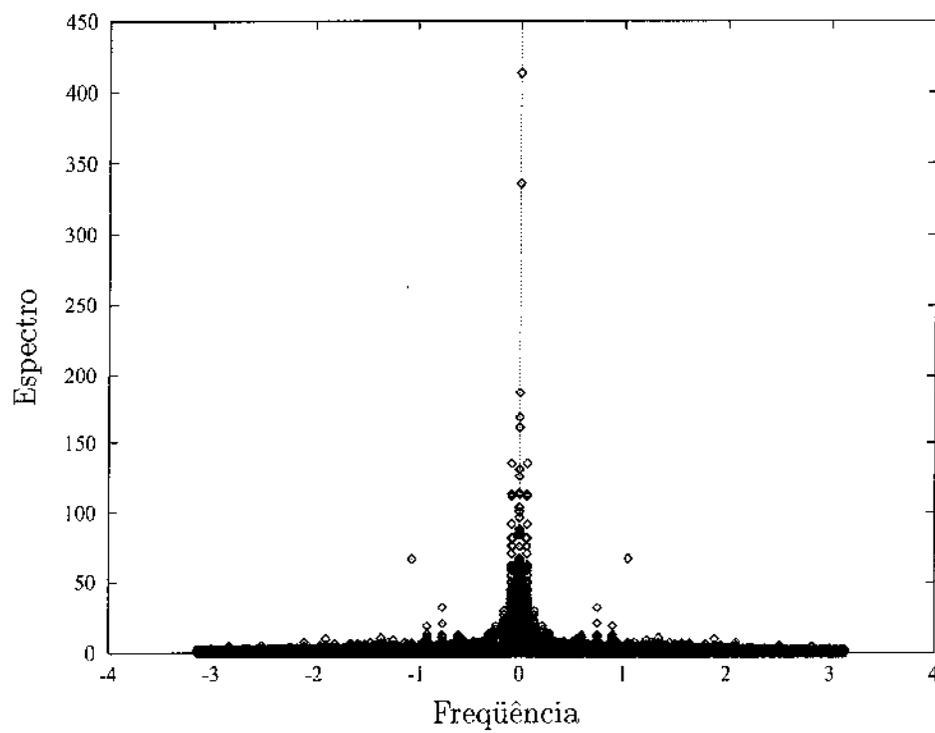


Figura 4.7: Espectro de potências estimado do tráfego Ethernet.

1. Para cada processo agregado $X^{(m)}$, como definido em (4.8), calcula-se a variância de $X^{(m)}$, $Var[X^{(m)}]$;
2. Constrói-se um gráfico para $\log_{10}(Var[X^{(m)}])$ por $\log_{10}(m)$ para vários valores de m ;
3. Ajusta-se uma reta aos pontos do gráfico por um método simples de regressão, por exemplo, mínimos quadrados [PTVF92], obtendo-se $\hat{\beta}$ a inclinação negada da reta ajustada;
4. Estima-se H por $\hat{H} = 1 - \frac{\hat{\beta}}{2}$.

As figuras 4.8 e 4.9 mostram os valores estimados de H para os dois tipos de dados. As retas presentes nas figuras possuem inclinação -1 e servem como referência.

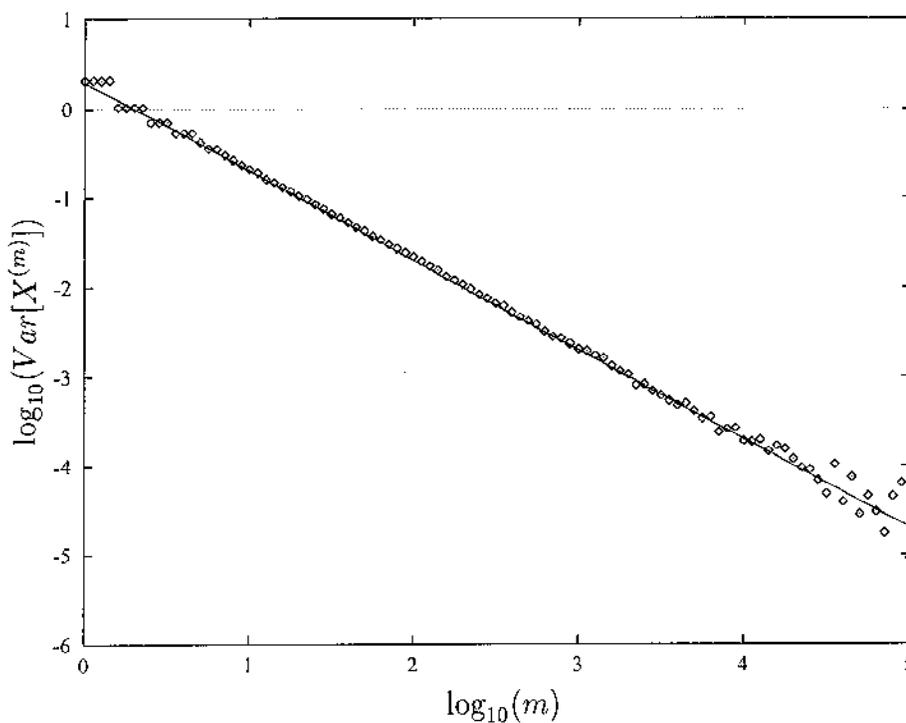


Figura 4.8: Gráfico $\log_{10}(Var[X^{(m)}])$ x $\log_{10}(m)$ para o tráfego Poisson gerado no experimento. A inclinação dos pontos calculados é igual a -1 o que resulta em uma estimativa de H igual a 0,5.

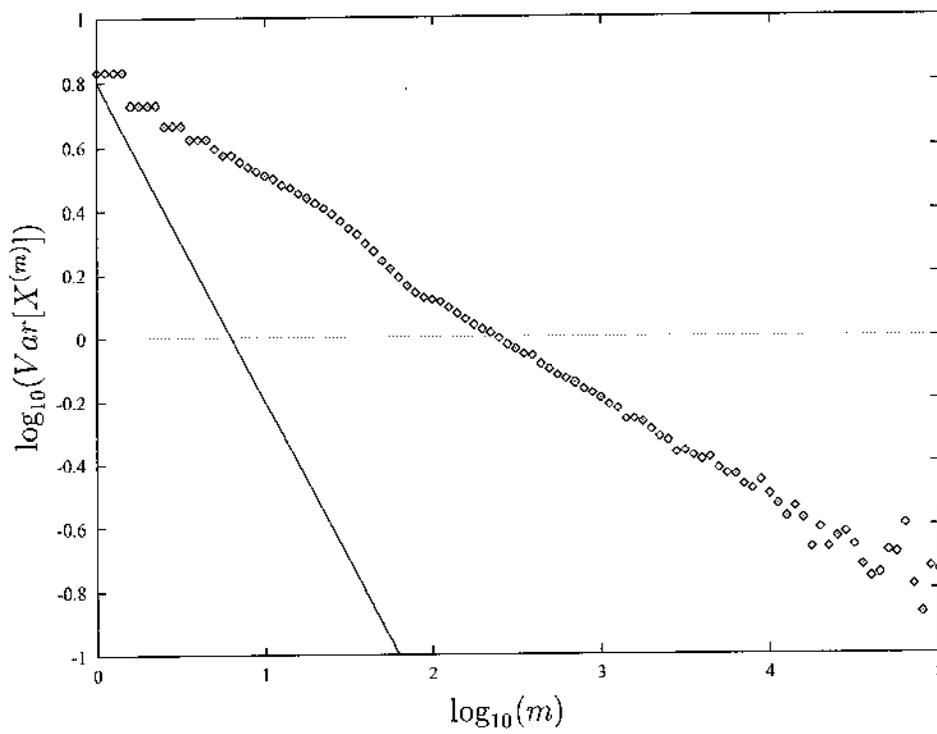


Figura 4.9: Gráfico $\log_{10}(\text{Var}[X^{(m)}])$ x $\log_{10}(m)$ para o tráfego Ethernet. A inclinação dos pontos calculados é igual a $-0,36$ o que resulta em uma estimativa de H igual a $0,82$.

Análise R/S

H. E. Hurst viveu em Cairo no Egito e estudou dados sobre o rio Nilo por mais de três décadas (iniciando nos anos quarenta). Suas descobertas em processos com dependências em longos intervalos levou à construção da represa Aswan High (Aswan High Dawn), completada em 1970 (Hurst morreu aos 98 anos em 1978). Ao longo de seu trabalho, ele desenvolveu uma abordagem estatística para se determinar a melhor capacidade de um reservatório de água. O objetivo é escolher um tamanho que não resulte transbordamento e nem tão pouco esgotamento d'água a partir de um registro empírico que exibía características de dependências em longos intervalos.

Mais especificamente, sua descoberta pode ser descrita da seguinte maneira: suponha que se queira calcular a capacidade de um reservatório tal que ela seja ideal para o intervalo de tempo entre t e $t+k$. Por questões de simplicidade, assumamos que o tempo é discreto e que não há perda no armazenamento (causada por evaporação, infiltração, etc.). Capacidade ideal, nesse contexto, significa que o que se quer atingir é um fluxo de saída uniforme, que no tempo $t+k$ o reservatório esteja com o mesmo armazenamento que no tempo t , e que o reservatório nunca transborde. Esse problema é análogo ao problema de dimensionar um buffer em uma rede de pacotes.

Seja X_i o fluxo de entrada no tempo i e $Y_j = \sum_{i=1}^j X_i$ o fluxo de entrada acumulado até o tempo j . A capacidade ideal é definida por

$$R(t, k) = \max_{0 \leq i \leq k} \left[Y_{t+i} - Y_t - \frac{i}{k} (Y_{t+k} - Y_t) \right] - \min_{0 \leq i \leq k} \left[Y_{t+i} - Y_t - \frac{i}{k} (Y_{t+k} - Y_t) \right] \quad (4.24)$$

$R(t, k)$ é chamado de intervalo ajustado (*adjusted range*). Para se estudar as propriedades que são independentes da escala, $R(t, k)$ é normalizado por

$$S(t, k) = \sqrt{\frac{1}{k} \sum_{i=t+1}^{t+k} (X_i - \bar{X}_{t,k})^2} \quad (4.25)$$

em que $\bar{X}_{t,k} = \frac{1}{k} \sum_{i=t+1}^{t+k} X_i$. A razão

$$\frac{R}{S} = \frac{R(t, k)}{S(t, k)} \quad (4.26)$$

é chamada de intervalo ajustado reescalado (*rescaled adjusted range*) ou estatística R/S . Hurst construiu vários gráficos do logaritmo de R/S para vários valores de k . Ele observou que para grandes valores de k , $\log_{10} \frac{R}{S}$ estava distribuído ao longo de uma linha reta com

inclinação superior a $\frac{1}{2}$. Em terminologia probabilística, isso significa que para valores grandes de k ,

$$\log_{10} E [R/S] = a + H \log_{10} k \quad (4.27)$$

com $H > \frac{1}{2}$ e a constante. Essa descoberta empírica contradizia resultados para processos Markovianos e outros processos estocásticos que eram normalmente considerados naquela época. Para qualquer processo estacionário sem memória, R/S deve se comportar assintoticamente como uma constante vezes $k^{\frac{1}{2}}$ [MW69a]. Portanto, para grandes valores de k , $\log_{10} R/S$ deve estar distribuído aleatoriamente ao longo de uma linha reta com inclinação $\frac{1}{2}$. Hurst descobriu que para os dados do rio Nilo, e para muitos outros dados hidrológicos, climatológicos e geofísicos, R/S se comporta como uma constante vezes k^H para algum $H > \frac{1}{2}$, tal fenômeno é conhecido por *efeito de Hurst*. A análise R/S foi utilizada para uma grande variedade de fenômenos geofísicos [MW69c] e sua robustez como medida estatística comprovada em [MW69b].

A análise R/S é baseada numa abordagem gráfica heurística, descrita em detalhes em [MW69c], que tenta explorar a maior quantidade possível de informações presentes em um dado conjunto de dados.

A estatística R/S é calculada da seguinte maneira: a partir de uma seqüência de observações, $\{X_1, X_2, \dots, X_n\}$, define-se uma seqüência de somas parciais ajustadas,

$$W_j = \sum_{i=1}^j (X_i - E[X]) \quad (4.28)$$

o intervalo ajustado é o maior valor entre os observados até o tempo d ,

$$R(d) = \max(0, W_1, W_2, \dots, W_d) - \min(0, W_1, W_2, \dots, W_d) \quad (4.29)$$

Normalizando o intervalo pelo desvio padrão amostral, $S(d)$, calculado a partir das mesmas observações no mesmo intervalo $\{X_1, \dots, X_n\}$, obtém-se o intervalo ajustado reescalonado, estatística R/S .

Quando se trabalha com dados empíricos, a medida $R(d)/S(d)$ resulta em apenas um ponto de dado. Para se obter uma estatística mais robusta, a série temporal (de comprimento n) é particionada em k blocos de comprimento $M = \lfloor \frac{n}{k} \rfloor$. O último bloco parcial, se existir, é truncado. Calcula-se $R_k(d)$ e $S_k(d)$ a partir das d primeiras observações no bloco K , produzindo K valores de $R(d)/S(d)$. Quando o *lag* excede o tamanho do bloco, $n > M$, os conjuntos de observações podem se sobrepor, e o último conjunto é descartado,

por causa do número insuficiente de pontos (o *trace* não pode ser tratado de forma circular, tomando o início novamente). Os dados resultantes são plotados em um diagrama (“*Pox Diagram*”), em coordenadas logarítmicas, H pode ser estimado como sendo a inclinação de uma linha reta.

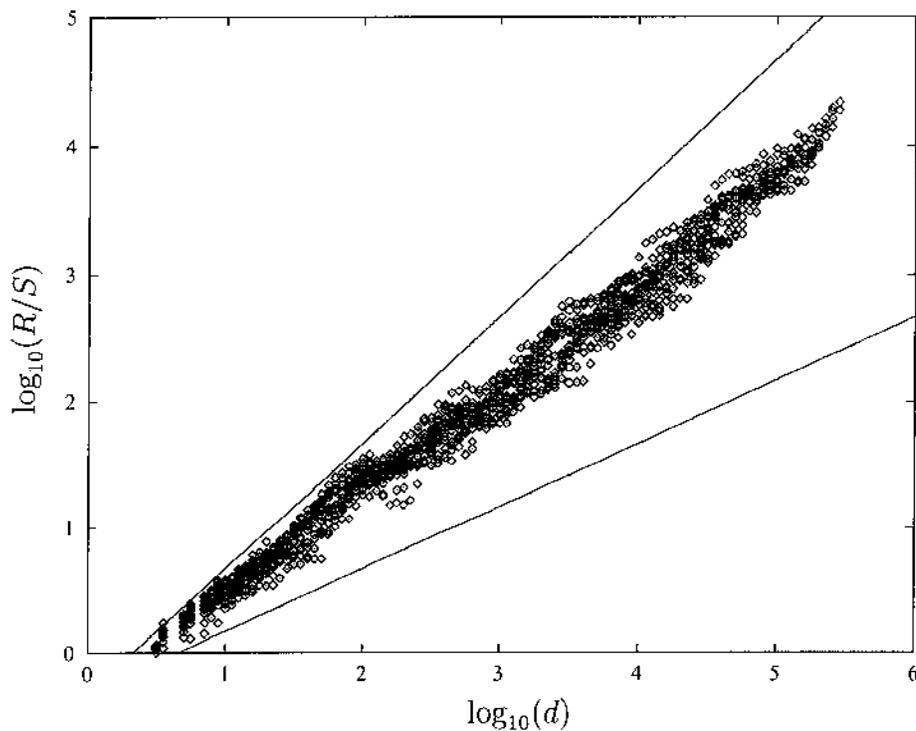


Figura 4.10: *Pox Diagram* ($\log_{10}(R/S)$ x $\log_{10}(d)$) para o tráfego Ethernet. A inclinação da reta ajustada aos pontos resulta em uma estimativa para H igual a 0,81.

As figuras 4.10 e 4.11 mostram os dados obtidos através da análise R/S para os dois tipos de tráfego. Para esses gráficos, como sugerido em [Gar93], os *lags* são espaçados igualmente em escala logarítmica com $k = 10$ blocos. Os pontos para *lags* pequenos são desconsiderados, por causa da influência do comportamento de curta memória na medida do parâmetro de longa memória. Os *lags* extremamente grandes são também desconsiderados devido ao pequeno número de observações. As retas mostradas nos gráficos servem como referência, sendo que a reta inferior possui inclinação 0,5 e a superior possui inclinação 1,0.

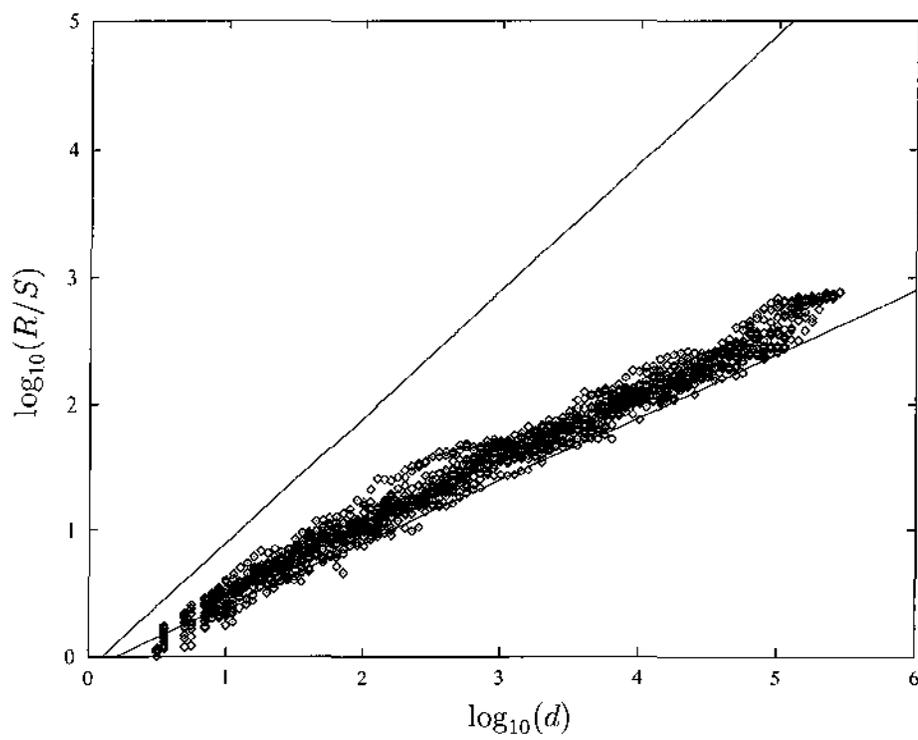


Figura 4.11: *Pox Diagram* ($\log_{10}(R/S)$ x $\log_{10}(d)$) para o tráfego Poisson gerado no experimento. A inclinação da reta ajustada aos pontos resulta em uma estimativa para H igual a 0,5.

Análise do Periodograma

O gráfico da Variância x Tempo e a análise R/S são bons estimadores pontuais para H . Entretanto, quando se necessita de uma análise mais detalhada dos dados, como por exemplo o intervalo de confiança para H , eles são de pouca utilidade. Serão consideradas agora técnicas para se estimar H e seu intervalo de confiança baseadas nas propriedades assintóticas do periodograma.

O periodograma, $I(\lambda)$, é calculado a partir dos dados por,

$$I(\lambda) = \frac{1}{2\pi N} \left| \sum_{k=1}^N X_k e^{ik\lambda} \right|^2 \quad (4.30)$$

com i sendo a unidade imaginária e N o número de pontos. Sabe-se da Análise Espectral que o periodograma é, assintoticamente, um estimador não viciado para f (função de densidade espectral), ou seja,

$$\lim_{N \rightarrow \infty} E[I(\lambda)] = f(\lambda). \quad (4.31)$$

Para se ter uma estimativa do grau de auto-semelhança H de dados empíricos, o periodograma pode ser utilizado explorando a forma da função de densidade espectral próxima à origem:

$$f(\lambda) = c_f |\lambda|^{-\beta} \quad , |\lambda| \rightarrow 0 \quad (4.32)$$

que pode ser reescrita como

$$\log_{10} f(\lambda) = \log_{10} c_f - \beta \log_{10} |\lambda| \quad (4.33)$$

em que $\beta = 2H - 1$ [LTWW94]. Geralmente, $I(\lambda)$ é calculado nas frequências de Fourier

$$\lambda_k = \frac{2\pi k}{N}, k = 1, \dots, m \quad (4.34)$$

em que $m = \frac{n-1}{2}$.

Para conjuntos grandes de dados, H pode ser estimado através de regressão por mínimos quadrados aplicada aos pontos $(\log_{10}(I(\lambda)), \log_{10} \lambda)$, λ como em (4.34). Pelo fato de (4.32) ser uma propriedade assintótica quando $\lambda \rightarrow 0$, em [LTWW94] é sugerida, para a estimativa de H , a utilização de apenas uma parcela das menores frequências (no caso particular dos dados Ethernet utilizou-se apenas 10% das menores frequências). As figuras 4.12 e 4.13 mostram os valores estimados para H a partir do espectro estimado.

Uma outra abordagem proposta em [BSTW95] é se estimar H através das séries agregadas do processo original, já que as séries agregadas possuem as mesmas estruturas de correlação que a série original.

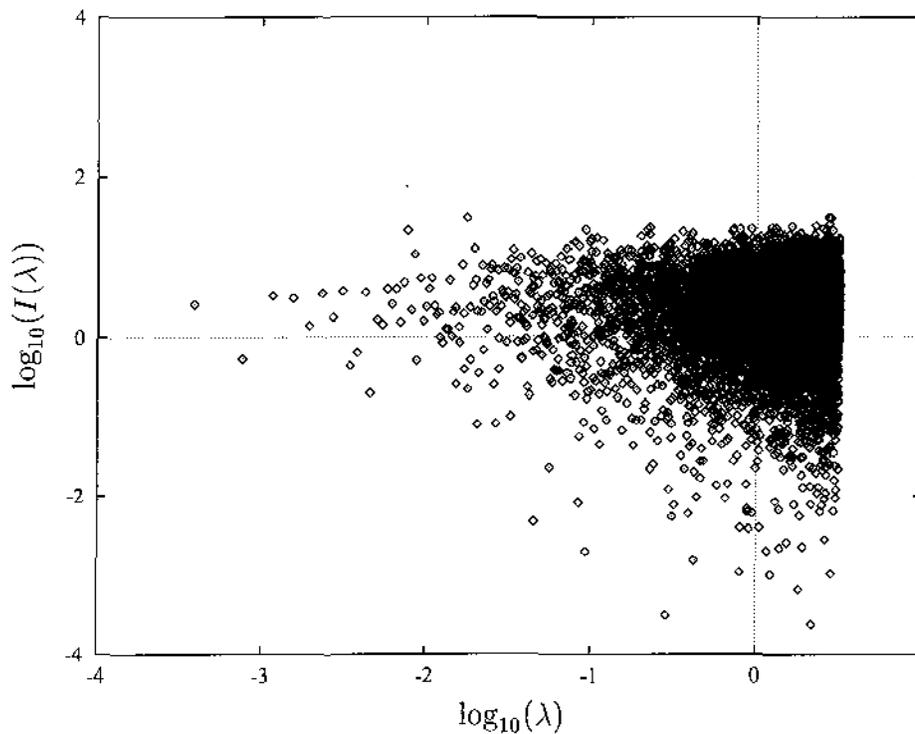


Figura 4.12: Periodograma do tráfego Poisson gerado no experimento. $H = 0,5$ estimado pela reta ajustada a 10% das menores frequências.

Essa forma de se estimar H , baseada na inclinação da reta ajustada ao periodograma próximo à origem, também fornece uma estimativa pontual para H . Para o cálculo do intervalo de confiança para H , o estimador mais utilizado na literatura é o *Estimador de Whittle* que também é baseado nas propriedades do periodograma. Suponha que $\{x_t\}$ seja uma amostra de um processo auto-semelhante X para o qual se deseja estimar H . Seja $f(\lambda, H)$ a função de densidade espectral de X normalizada para se ter variância 1, e $I(\lambda)$ o periodograma de $\{x_t\}$. Para se estimar H , deve ser encontrado um valor \hat{H} que minimize:

$$g(\hat{H}) = \int_{-\pi}^{\pi} \frac{I(\lambda)}{f(\lambda, \hat{H})} d\lambda. \quad (4.35)$$

Se $\{x_t\}$ possui comprimento N , então a integral acima é convertida em uma soma discreta nas frequências $\lambda = \frac{2\pi}{N}, \frac{4\pi}{N}, \dots, 2\pi$.

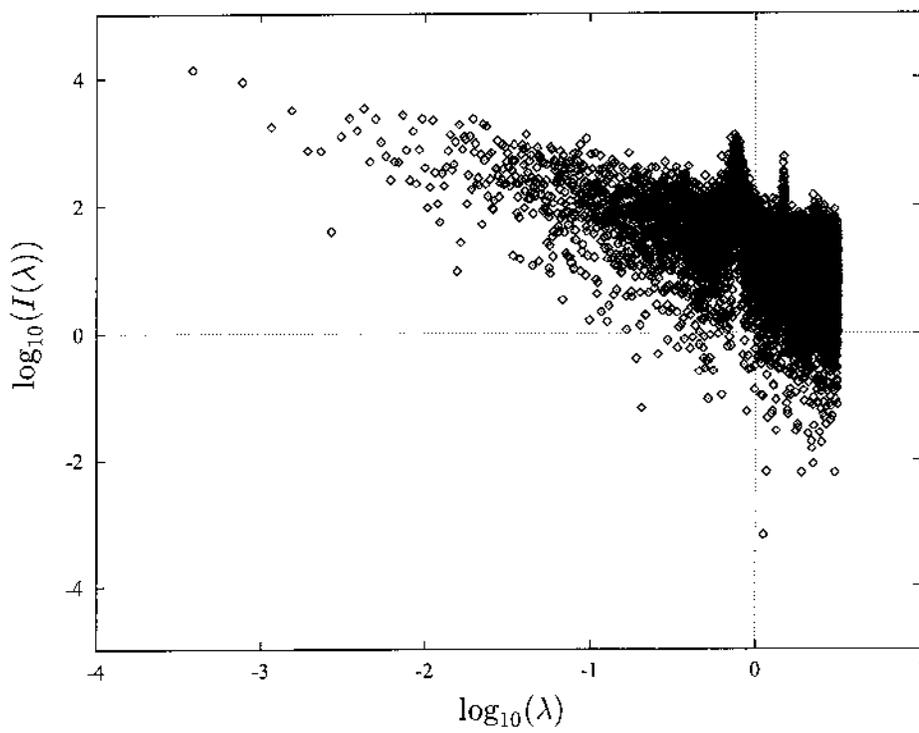


Figura 4.13: Periodograma do tráfego Ethernet. $H = 0,81$ estimado pela reta ajustada a 10% das menores frequências.

Junto com o estimador \hat{H} , pode-se calcular sua variância $\hat{\sigma}_H^2$ por [Pax97]

$$\sigma_H^2 = 4\pi \left[\int_{-\pi}^{\pi} \left(\frac{\partial \log f(\lambda)}{\partial H} \right)^2 d\lambda \right]^{-1} \quad (4.36)$$

O intervalo de confiança para H é dado por $H \pm 1,96\hat{\sigma}_H$ [LTWW94].

4.2.3 Processos Auto-Semelhantes

De um ponto de vista de modelagem de tráfego, as características estatísticas evidenciadas no tráfego Ethernet poderiam sugerir a necessidade de processos complexos e com um número elevado de parâmetros. Entretanto, a presença de auto-semelhança sugere uma modelagem compacta e parcimoniosa (número pequeno de parâmetros) [LTWW94] baseada em processos amplamente estudados, por exemplo, Movimento Browniano Fracionário e Ruído Gaussiano Fracionário. A principal característica do modelo, auto-semelhança, é especificada por apenas um parâmetro, parâmetro de Hurst H .

Movimento Browniano Fracionário e Ruído Gaussiano Fracionário

Movimento Browniano é um fenômeno bem conhecido na física que descreve o movimento aleatório de partículas microscópicas suspensas em um líquido ou gás. Ele foi observado experimentalmente pela primeira vez pelo botânico inglês Robert Brown em 1827.

Movimento Browniano é um processo Gaussiano contínuo, $\{B = B(t), t > 0\}$, com as seguintes propriedades:

- i) $B(0) = 0$
- ii) $B(t)$ possui incrementos independentes
- iii) $E[B(t) - B(s)] = 0$
- iv) $Var[B(t) - B(s)] = \sigma^2 |t - s|$

Movimento Browniano Fracionário é um processo Gaussiano contínuo, $\{B_H = (B_H(t), t > 0, 0 < H < 1\}$ com média zero e com função de autocorrelação $\rho(s, t) = \frac{1}{2}(s^{2H} + t^{2H} - |s - t|^{2H})$. Ele possui incrementos estacionários e é auto-semelhante com parâmetro de auto-semelhança H , isto é, para todo $a > 0$,

$$B_H(at) \stackrel{d}{=} a^H B_H(t) \quad (4.37)$$

O processo incremento do processo Movimento Browniano Fracionário,

$$\{X_H = (X_H(\tau) = B_H(\tau + 1) - B_H(\tau), \tau \geq 0) \quad (4.38)$$

é chamado Ruído Gaussiano Fracionário (FGN - *Fractional Gaussian Noise*) e é um processo Gaussiano estacionário com função de autocorrelação

$$\rho(\tau) = \frac{1}{2}(|\tau + 1|^{2H} - 2\tau^{2H} + |\tau - 1|^{2H}), \tau \geq 0 \quad (4.39)$$

Assintoticamente, $\rho(\tau) = H(2H - 1) |\tau|^{2H-2}$, ou seja

$$\lim_{\tau \rightarrow \infty} \frac{\rho(\tau)}{\tau^{2H-2}} = H(2H - 1) \quad (4.40)$$

Portanto, X_H exibe dependência em longos intervalos para $\frac{1}{2} < H < 1$. O apêndice B apresenta uma demonstração para (4.40).

4.2.4 Geração de *Traces* de Processos Auto-Semelhantes

A utilidade de um monitor Ethernet como o construído nos laboratórios da *Bellcore* é inquestionável, mas nem sempre é possível ter acesso a um monitor desse tipo. Para a realização de estudos de simulação, uma grande quantidade de *traces* de dados é necessária, e a aquisição e o armazenamento desses dados são caros tanto em termos de tempo quanto em termos de armazenamento. Existe também a questão se os usuários de uma rede Ethernet se sentiriam confortáveis tendo o tráfego de sua rede monitorado e registrado. O que é necessário, então, é um método rápido de geração sintética de *traces* de tráfegos que possam ser utilizados em simulações.

Auto-semelhança não é um fenômeno novo e foi estudado extensivamente em vários campos, incluindo hidrologia e economia [MN68, MW69a, MW69c]. Devido a esses estudos, vários modelos matemáticos formais que exibem auto-semelhança foram desenvolvidos. Entre esses modelos, estão o FARIMA (*Fractional Auto-Regressive Integrated Moving Average*), o Ruído Gaussiano Fracionário e o Movimento Browniano Fracionário.

A seguir são apresentados dois algoritmos para geração sintética de *traces* com características de auto-semelhança baseados no Ruído Gaussiano Fracionário e no Movimento Browniano Fracionário.

4.2.5 Método da FFT

O algoritmo de Paxson [Pax97] se baseia em um resultado bem conhecido da Análise Espectral, que diz que o periodograma é um bom estimador (não viciado) para a função de densidade espectral (espectro de potências) [Pri81]. O algoritmo também utiliza a Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*), método que acelera muito o cálculo do periodograma. Por esta razão, o algoritmo é conhecido por Método da FFT.

O Método da FFT consiste em obter uma série no domínio do tempo a partir de uma função de densidade espectral conhecida de um processo que exiba auto-semelhança. Como visto na seção 4.2.3, um processo bastante estudado por suas propriedades de auto-semelhança e com uma função de densidade espectral conhecida é o Ruído Gaussiano Fracionário (FGN - *Fractional Gaussian Noise*). A idéia do algoritmo consiste em construir uma seqüência de números complexos Z_i correspondente à função de densidade espectral de um processo FGN. Em seguida aplicar a transformada inversa discreta de Fourier para se obter X_i , a série equivalente a Z_i no domínio do tempo. Como X_i possui, por construção, o espectro de potências de um processo FGN, e como autocorrelação e espectro de potências formam um par de Fourier [Pri81], garante-se que X_i possui as mesmas propriedades de autocorrelação de um processo FGN que, para a maioria dos propósitos, é a sua característica mais importante.

O algoritmo possui apenas dois parâmetros: o número de elementos, n ; e o parâmetro de Hurst, H ; e segue os passos abaixo:

1. Construa uma seqüência de valores $\{f_1, \dots, f_{n/2}\}$, em que $f_j = f(\frac{2\pi j}{n}, H)$, corresponde ao espectro de potências de um processo FGN para as frequências de $\frac{2\pi}{n}$ até π . A função de densidade espectral de um processo FGN é dada por [Pax97]:

$$f(\lambda, H) = 2\sin(\pi H)\Gamma(2H + 1)(1 - \cos\lambda) [|\lambda|^{-2H-1} + B(\lambda, H)]$$

$$B(\lambda, H) = \sum_{j=1}^{\infty} [|(2\pi j + \lambda)|^{-2H-1} + |(2\pi j - \lambda)|^{-2H-1}]$$

2. Seja \hat{f} uma nova seqüência, obtida a partir de f . Os elementos \hat{f}_i são obtidos pela multiplicação dos elementos $\{f_j\}$ por uma variável aleatória independente com distribuição exponencial e média 1. Este passo do algoritmo é justificado por um

outro resultado bem conhecido da Análise Espectral que diz que os valores do periodograma (potência para uma determinada frequência) são independentes e distribuídos assintoticamente como uma variável aleatória exponencial, com média igual à potência total [Pri81].

3. Construa $\{Z_1, \dots, Z_{n/2}\}$, uma seqüência de números complexos tais que $|Z_i| = \sqrt{\hat{f}_i}$ e a fase de Z_i está distribuída uniformemente entre 0 e 2π . A aleatorização da fase preserva a função de densidade espectral correspondente a $\{\hat{f}_i\}$ e assegura que amostras diferentes geradas pelo método sejam independentes. Ela também faz com que o resultado final possua distribuição normal [Pax97], uma característica dos processos FGN.
4. Construa $\{Z'_0, \dots, Z'_{n-1}\}$, uma versão expandida de $\{Z_1, \dots, Z_{n/2}\}$ em que:

$$Z'_j = \begin{cases} 0, & \text{se } j = 0 \\ Z_j, & \text{se } 0 < j \leq n/2, \text{ e} \\ \bar{Z}_{n-j}, & \text{se } n/2 < j < n \end{cases}$$

\bar{Z}_{n-j} é o conjugado complexo de Z_{n-j} . $\{Z'_j\}$ possui o mesmo espectro utilizado na construção de $\{Z_i\}$ e devido a simetria em torno de $Z'_{n/2}$, Z'_j corresponde à transformada de Fourier de uma amostra de valores reais [Pax97].

5. Aplique a transformada inversa de Fourier a $\{Z'_j\}$ para obter uma amostra aproximada de um processo FGN $\{X_i\}$.

A figura 4.14 apresenta alguns exemplos de dados gerados pelo Método da FFT. Percebe-se claramente, pela figura, a aparência de ruído branco para $H = 0,5$ e o fenômeno de persistência para $H = 0,7$ e $H = 0,9$. A qualidade dos dados gerados e a qualidade do algoritmo serão discutidas no capítulo 6.

Na realidade, o algoritmo de Paxson gera uma seqüência de números reais contendo valores positivos e negativos. Para ser utilizado como um gerador de tráfego é necessário realizar uma transformação nos dados para se obter valores não negativos. No capítulo 5 serão discutidas algumas transformações propostas na literatura. A transformação mais utilizada é a transformação linear, já que esta não altera o parâmetro de Hurst. O apêndice A apresenta uma demonstração para essa afirmação.

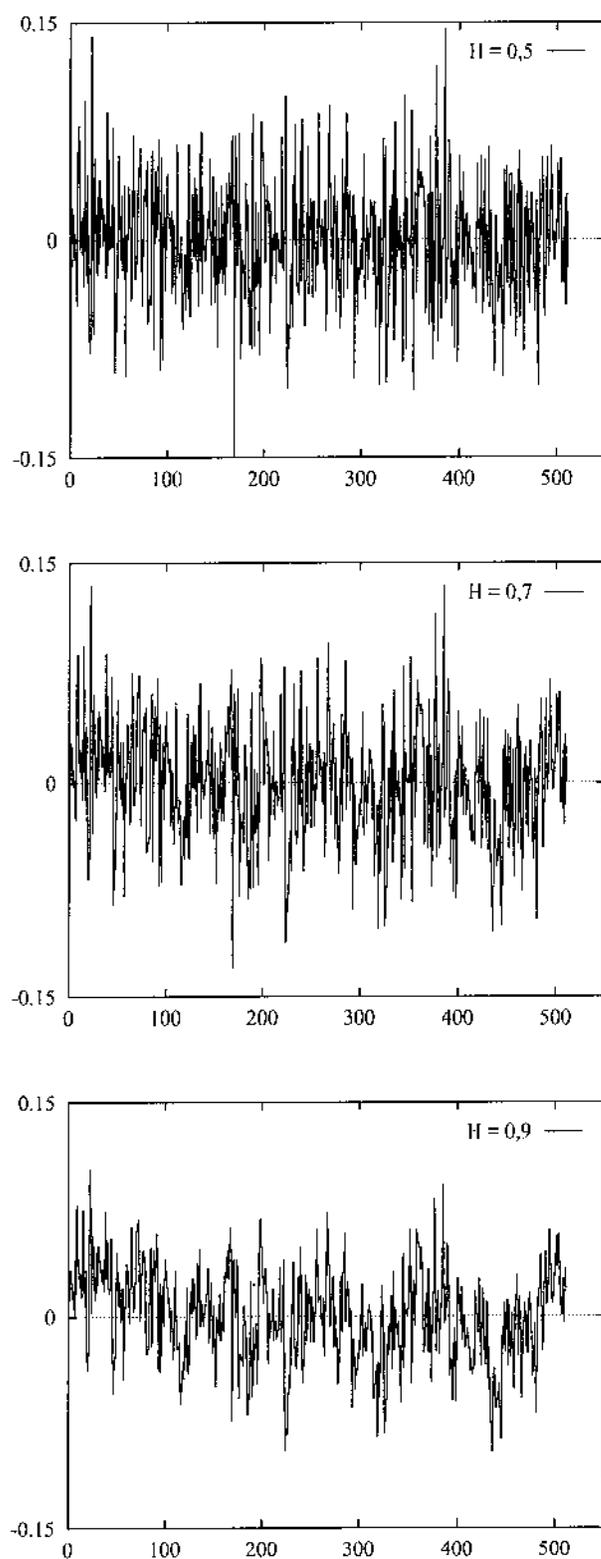


Figura 4.14: Seqüência de dados de um processo FGN gerada pelo Método da FFT. Os parâmetros utilizados foram $n = 512$ e $H = 0,5$; $H = 0,7$ e $H = 0,9$.

Um outro ponto que deve ser salientado no algoritmo proposto é a utilização de uma aproximação para a soma infinita existente no espectro. Paxson propõe a seguinte aproximação:

$$B(\lambda, H) \approx a_1^d + b_1^d + a_2^d + b_2^d + a_3^d + b_3^d + \frac{a_3^{d'} + b_3^{d'} + a_4^{d'} + b_4^{d'}}{8H\pi}$$

em que

$$\begin{aligned} d &= -2H - 1 \\ d' &= -2H \\ a_k &= 2k\pi + \lambda \\ b_k &= 2k\pi - \lambda \end{aligned}$$

Como mostrado pelo próprio autor, essa aproximação causa uma pequena perda de precisão nos dados gerados. Na maioria dos casos, o parâmetro H estimado a partir dos dados gerados diverge por uma pequena margem do parâmetro fornecido. Entretanto, a utilização da aproximação oferece um ganho em relação ao tempo de execução quando comparado com um outro algoritmo que utiliza a mesma idéia e foi proposto por Beran [Pax97, Ber94]. Neste algoritmo são utilizados os 200 primeiros termos da soma infinita.

4.2.6 Algoritmo RMD

O algoritmo RMD (*Random Midpoint Displacement*) é um algoritmo de aproximação para a geração de *traces* baseados em processo Movimento Browniano Fracionário (FBM - *Fractional Brownian Motion*) e foi proposto em [LEWW95]. O algoritmo consiste em se obter um *trace* de um processo FBM no intervalo de tempo $[0, T]$, ou seja, obter o número de células geradas em subintervalos do intervalo $[0, T]$. A idéia básica do algoritmo é avançar para dentro do intervalo $[0, T]$, subdividindo-o recursivamente e construindo os valores do processo nos pontos médios a partir dos valores dos extremos. A observação chave para o algoritmo, na construção do valor do ponto central $Z(\frac{a+b}{2})$ de um intervalo $[a, b]$ a partir dos valores dos extremos $Z(a)$ e $Z(b)$, é que se Z fosse verdadeiramente FBM, então o deslocamento do ponto central $Z(\frac{a+b}{2}) - \frac{Z(a)+Z(b)}{2}$ seria independente do incremento $Z(b) - Z(a)$ sobre todo o intervalo. Esse deslocamento teria uma distribuição Gaussiana com média zero e variância da forma calculada adiante.

A simplificação chave do algoritmo RMD, que resulta em um cálculo rápido mas em perda de exatidão, é a escolha de todos os deslocamentos independentes, quando eles são necessários na construção. Essa hipótese é válida para Movimento Browniano, para FBM,

exceto por esta introdução de independência, a construção utilizada no algoritmo é válida [LEWW95]. O problema inicial no algoritmo é determinar o desvio padrão utilizado em cada um dos passos.

Seja s_k o desvio padrão utilizado na geração dos pontos médios no passo k . Seja σ_0 o desvio padrão do deslocamento na escala de tempo T , ou seja, $\sigma_0^2 = \text{Var}[B_H(T) - B_H(0)]$, e pelas propriedades dos processos FBM, $\sigma_0 = T^H$. Assume-se que $T = 2^n$. Pelas propriedades dos processos FBM, tem-se

$$\text{Var} \left[B_H \left(\frac{1}{2^k} \right) - B_H(0) \right] = \left(\frac{1}{2^k} \right)^{2H} \sigma_0^2 \quad (4.41)$$

Por outro lado, pelo mecanismo de construção utilizado no algoritmo,

$$B_H \left(\frac{a+b}{2} \right) = \frac{B_H(a) + B_H(b)}{2} + \Delta, \Delta \text{ é o deslocamento, tem-se:}$$

$$\begin{aligned} \text{Var} \left[B_H \left(\frac{1}{2^k} \right) - B_H(0) \right] &= \text{Var} \left[\frac{B_H \left(\frac{1}{2^{k-1}} \right) + B_H(0)}{2} - B_H(0) + \Delta \right] \\ &= \text{Var} \left[\frac{B_H \left(\frac{1}{2^{k-1}} \right) + B_H(0) - 2B_H(0) + 2\Delta}{2} \right] \\ &= \frac{1}{4} \left\{ \text{Var} \left[B_H \left(\frac{1}{2^{k-1}} \right) - B_H(0) \right] + \text{Var} [2\Delta] \right\} \\ &= \frac{1}{4} \text{Var} \left[B_H \left(\frac{1}{2^{k-1}} \right) - B_H(0) \right] + s_k^2 \end{aligned}$$

substituindo a equação acima na equação 4.41, tem-se

$$\begin{aligned} \left(\frac{1}{2^k} \right)^{2H} \sigma_0^2 &= \frac{1}{4} \left(\frac{1}{2^{k-1}} \right)^{2H} \sigma_0^2 + s_k^2 \\ s_k^2 &= \sigma_0^2 \left[\left(\frac{1}{2^k} \right)^{2H} - \frac{1}{4} \left(\frac{1}{2^{k-1}} \right)^{2H} \right] \\ &= \sigma_0^2 \left[\left(\frac{1}{2^k} \right)^{2H} - \frac{2^{2H}}{2^2} \left(\frac{1}{2^k} \right)^{2H} \right] \\ &= \sigma_0^2 \left(\frac{1}{2^k} \right)^{2H} (1 - 2^{2H-2}) \\ s_k &= \left(\frac{1}{2^k} \right)^H \sigma_0 \sqrt{1 - 2^{2H-2}} \\ &\quad e \\ s_k &= \frac{1}{2^H} s_{k-1} \end{aligned}$$

o valor de s_0 é definido como sendo igual a $\sqrt{1 - 2^{2H-2}}$. Mais especificamente, se o processo FBM $Z(t)$ deve ser construído no intervalo $[0, T]$, então o passo inicial é fazer

$Z(0) = 0$ e $Z(T)$ é obtido como uma amostra de uma distribuição normal com média 0 e variância T^{2H} . Em seguida, $Z(\frac{T}{2})$ é construído como a média de $Z(0)$ e $Z(T)$, $(\frac{Z(0)+Z(T)}{2})$ mais um deslocamento. O deslocamento é uma variável aleatória normal com um desvio padrão dado por T^H vezes o fator de escala inicial $s_1 = \frac{1}{2^H} s_0 = \frac{1}{2^H} \sqrt{1 - 2^{2H-2}}$. Em seguida o fator de escala é reduzido por $\frac{1}{2^H}$, e os dois intervalos de 0 a $\frac{T}{2}$ e de $\frac{T}{2}$ a T são subdivididos, e assim por diante.

A figura 4.15 mostra os 4 primeiros estágios do algoritmo. As retas tracejadas mostram os deslocamentos adicionados aos pontos médios. O ponto 1 é calculado como sendo a média dos valores dos pontos 0 mais um deslocamento dado por uma variável aleatória com distribuição normal, com média zero e desvio padrão s_1 . Os pontos 2 são calculados de maneira análoga utilizando os pontos 0 e 1 como extremos. O processo se repete da mesma maneira para os demais pontos do intervalo.

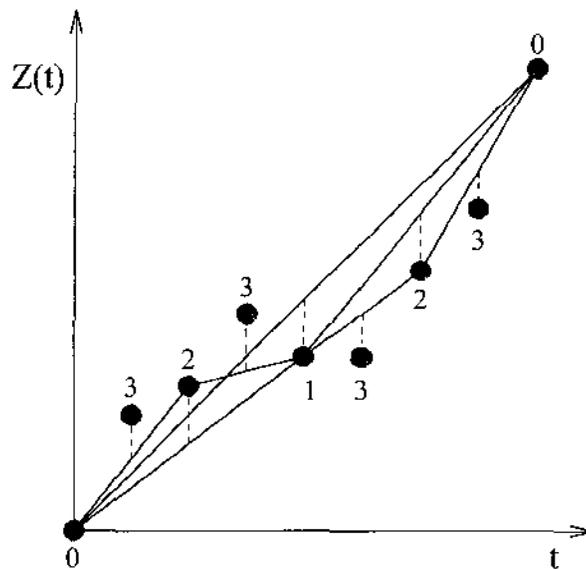


Figura 4.15: Primeiros 4 estágios do algoritmo RMD.

Os dados gerados pelo processo FBM aproximado ($Z(t)$) a partir do algoritmo RMD podem ser interpretados como um processo de chegada acumulado $A(t)$, como introduzido em [Nor94]:

$$A(t) = Mt + \sqrt{aM}Z(t) \quad (4.42)$$

em que M é a taxa média, a é definido como sendo a razão entre a variância e a média do número de células em um intervalo de tempo unitário. O processo incremento, que dá

o número de chegadas por unidade de tempo, do tempo t até $t + 1$ é definido por:

$$\hat{A}(t) = M + \sqrt{aM} [Z(t + 1) - Z(t)] \quad (4.43)$$

A entrada do algoritmo para geração do tráfego é, portanto, M , a , e H .

Capítulo 5

Ambiente de Simulação

Neste capítulo será descrito o simulador desenvolvido para analisar os mecanismos de gerenciamento de capacidade apresentados no capítulo 3. A abordagem adotada no desenvolvimento teve por objetivo tornar o simulador o mais genérico possível. O simulador permite que sejam criadas diferentes topologias e que todos os parâmetros necessários à operação de um determinado componente sejam fornecidos. Para a entrada de dados foi desenvolvida uma linguagem e um interpretador para facilitar o fornecimento dos dados dos componentes da rede que se deseja simular. O simulador pode ser utilizado para a análise de outros aspectos de uma rede ATM diferentes daqueles que serão analisados no capítulo 6, constituindo-se, portanto, em uma ferramenta para a análise de desempenho de redes ATM.

Este capítulo está organizado da seguinte maneira. A seção 5.1 apresenta as considerações sobre o modelo de simulação escolhido e sobre o ambiente em que foi realizada a implementação. Na seção 5.2 são descritos os componentes implementados no simulador e que estão disponíveis para serem utilizados em simulações.

5.1 Modelo de Simulação e Ambiente de Implementação

As características e forma de funcionamento dos diversos componentes do simulador apontaram para o modelo de simulação discreto orientado a eventos como o mais indicado para o seu desenvolvimento. A simulação discreta aplica-se a modelos em que as alterações no estado das variáveis das quais ele depende ocorrem discretamente em pontos específicos

do tempo simulado (ao contrário da simulação contínua em que as variáveis dependentes podem variar continuamente ao longo do tempo). Na sua forma orientada a eventos, essas variações são ocasionadas por eventos que ocorrem no decorrer do tempo simulado. Entre eventos, o estado das variáveis não se altera [Nee87, Soa90]. No simulador em questão, por exemplo, a chegada de uma célula a um comutador é um evento que pode gerar outros eventos, como, por exemplo, a solicitação de capacidade adicional em uma conexão virtual já estabelecida.

O tratamento de um evento além de ocasionar mudanças no estado corrente do componente, está sujeito a um atraso que é característico do componente e que é responsável pelo consumo do tempo simulado. Cada unidade do tempo simulado pode corresponder à qualquer unidade do tempo real. Devido aos baixos tempos envolvidos na simulação de redes ATM, na ordem de nanosegundos, e pela necessidade de se efetuar longas simulações, na ordem de minutos, optou-se pela utilização do segundo e dos seus submúltiplos, sendo que o relógio do simulador foi implementado utilizando-se uma variável de ponto flutuante de dupla precisão. Essa escolha é justificada principalmente pela limitação de variáveis do tipo inteiro na linguagem C, essas variáveis podem possuir no máximo 32 bits. Considerando uma variável de 32 bits sem sinal e uma unidade de tempo simulado igual a 10 nanosegundos, seria possível simular apenas 42,9 segundos de operação de uma rede.

As implementações do simulador e dos programas para síntese e análise de *traces* auto-semelhantes foram feitas inteiramente na linguagem C, em estações de trabalho SUN. O simulador contém um gerenciador de eventos (escalonador) que é responsável pelo escalonamento e pela execução das rotinas tratadoras dos eventos. O escalonador é responsável também pela manutenção do calendário de eventos. O calendário é utilizado para manter os eventos ordenados pelo tempo.

Para geração de variáveis aleatórias foi implementado o gerador de números aleatórios sugerido em [Lav83]. Com essa rotina é possível controlar as sementes utilizadas em diferentes replicações, assegurando, assim, a realização de replicações independentes.

5.2 Descrição dos Componentes

Para o desenvolvimento de um experimento de simulação, o simulador deve receber um arquivo com a descrição da rede que se espera simular. Uma rede é modelada como um

grafo em que os vértices possuem funções específicas como, por exemplo, comutadores, gateways ou aplicações e as arestas são os enlaces que interligam esses componentes. A figura 5.1 mostra como deve ser especificado um nó do grafo de simulação. Todo nó deve possuir um número, esse número é utilizado para definir a interligação do nó com os demais componentes da rede a ser simulada. Além disso, pode-se definir um nome, através do parâmetro **name**, que será utilizado para identificar o nó nos relatórios gerados pelo simulador. O parâmetro **function** indica a funcionalidade do nó (comutador, gateway ou aplicação). A definição das arestas (enlaces) do grafo de simulação será apresentada mais abaixo.

```
node 1 {
    name="Fonte CBR";           // Nome utilizado no relatório de saída
    function=CBR {             // Função do nó
        peak_rate=10000000;    // Taxa de transmissão da fonte
        duration_time=10;      // Tempo que a fonte permanece ativa
        connection_type=PERMANENT; // Conexão permanente
        destination_node=4;     // Nó para onde devem ser enviados
                                // os dados
    } /* CBR */
    plot=true; // Registra as estatísticas periodicamente
} /* node 1 */
```

Figura 5.1: Definição de um nó do grafo de simulação.

Além da topologia e das características de cada um dos componentes, podem ser especificados parâmetros globais relativos à execução da simulação. O tempo em que a simulação deve terminar e o nome do arquivo de resultados são alguns exemplos de parâmetros globais da simulação. Além disso, a linguagem definida para a entrada de dados permite a inclusão de comentários no arquivo de definição. Comentários são definidos como na linguagem C++, a seqüência `//` define um comentário de linha e as seqüências `/*` e `*/` delimitam um texto de uma ou mais linhas em comentário.

A seguir serão detalhados os componentes existentes no simulador e seus parâmetros. Um parâmetro é definido normalmente pela atribuição de um valor a uma palavra reservada da linguagem que representa o parâmetro. Os parâmetros podem ser especificados tanto em letras maiúsculas quanto em letras minúsculas. A sintaxe completa da linguagem

está definida no apêndice C.

5.2.1 Parâmetros da Simulação

O simulador reserva uma seção no início do arquivo de entrada para que possam ser especificados parâmetros globais da simulação. Os possíveis parâmetros são os seguintes:

- **log_file**: Nome do arquivo em que serão gravados os resultados. Se este parâmetro não for especificado, é utilizada a saída padrão.
- **end_simulation** (*segundos*): Tempo do término da simulação. Se este parâmetro não for especificado, a simulação terminará quando terminarem os eventos no calendário de eventos.
- **print_interval** (*segundos*): Tempo utilizado para gravar as estatísticas durante a simulação. A cada **print_interval** as estatísticas correntes dos componentes da simulação são registradas no arquivo de resultados.
- **plot** (*true | false*): Indica se as estatísticas serão registradas periodicamente ou não. Se for dado o valor *true*, as estatísticas periódicas serão registradas. Se for dado o valor *false*, as estatísticas periódicas não serão registradas, mas as estatísticas finais serão. O valor padrão é *false*. Cada componente do simulador pode redefinir este parâmetro para indicar se as estatísticas específicas do componente devem ou não ser registradas.

A figura 5.2 apresenta um exemplo de especificação desta seção do simulador.

5.2.2 Comutador ATM

O comutador ATM é o componente utilizado para comutar células de vários enlaces pertencentes a diferentes canais virtuais. O comutador é ligado aos diferentes enlaces através de portas de comunicação. Essas portas podem ser tanto de entrada como de saída. O número de portas de cada comutador é variável e depende do número de enlaces ligados ao comutador.

Os comutadores ATM implementados no simulador são não bloqueantes e com enfileiramento na saída [Kes96], sendo que não foi implementada nenhuma estrutura complexa de comutação (*switching fabric*), pois o interesse maior não é avaliar o desempenho

```
simulation_parameters {
  log_file="network1.output";
  end_simulation=100; // A simulação termina quando o tempo
                    // simulado atingir 100 segundos
  print_interval=10; // Grava estatísticas a cada 10 segundos
  plot=true;        // Registra as estatísticas durante a
                    // simulação
} /* simulation_parameters */
```

Figura 5.2: Parâmetros globais da simulação.

dessas estruturas e sim o comportamento de uma rede ATM como um todo. Para uma descrição das diferentes estruturas de comutação propostas na literatura para os comutadores ATM e uma comparação das estratégias de enfileiramento (entrada ou saída), veja [Par94, Kes96]. A função básica desempenhada por um comutador é, na chegada de uma célula proveniente de um enlace físico em uma porta, consultar sua tabela de roteamento para descobrir a partir dos valores dos identificadores de caminho virtual e canal virtual os novos valores para esses identificadores e a porta para onde a célula deve ser enviada. Essa operação acarreta um atraso de processamento, que é constante, a cada célula comutada. Se o enlace de saída estiver ocupado, o comutador deve armazenar a célula em uma fila destinada àquele enlace e a célula será enviada somente quando houver disponibilidade do enlace.

Os parâmetros monitorados em um comutador incluem número de células recebidas, número de células marcadas como de baixa prioridade pelo mecanismo de policiamento, número de células de baixa prioridade descartadas, número de células de alta prioridade descartadas, comprimento médio da fila de saída e atraso médio enfrentado pelas células na fila de saída.

Os parâmetros de entrada estão detalhados a seguir. A figura 5.3 apresenta um exemplo de especificação de um comutador ATM com sua respectiva sintaxe.

Parâmetros

Um comutador ATM é especificado como sendo uma função **ATM_SWITCH** de um nó do grafo de simulação. Os parâmetros que devem ser especificados para o comutador são

```
node 2 {
    name="Comutador ATM 1"; // Nome utilizado no relatório de saída
    function=ATM_SWITCH { // Função do nó
        delay=0.000023; // Tempo de processamento de uma célula
        output_queue_size=50; // Tamanho da fila de saída
        queue_threshold=40; // Limiar para descarte
        buffer_management=PUSHOUT; // Política de gerenciamento
        policing=YES; // Efetua policiamento
    } /* ATM_SWITCH */
    plot=true; // Registra as estatísticas periodicamente
} /* node 2 */
```

Figura 5.3: Especificação de um comutador ATM.

os seguintes:

- **delay** (*segundos*): Tempo gasto pelo comutador ATM para transferir uma célula de uma porta de entrada para uma porta de saída.
- **output_queue_size**: Tamanho do buffer de saída do comutador. Este parâmetro indica o número máximo de células que podem ser armazenadas à espera de transmissão em um enlace.
- **queue_threshold**: Limiar a partir do qual células de baixa prioridade não são aceitas no buffer no mecanismo de gerenciamento de buffer *Threshold*. Este valor também aponta a região perigosa para as células de baixa prioridade no mecanismo *L-Pushout*, ou seja, a região em que as células de baixa prioridade podem ser expulsas do buffer.
- **buffer_management**: Mecanismo utilizado para gerenciar o espaço ocupado no buffer. Os possíveis valores são: **THRESHOLD**, **PUSHOUT** e **L-PUSHOUT**. Para maiores detalhes sobre esses mecanismos, veja o capítulo 2.
- **policing**: Este parâmetro indica se o comutador fará ou não o policiamento das células das conexões virtuais. Os possíveis valores são **YES** ou **NO**, correspondendo, respectivamente, a sim e não.

5.2.3 Enlace

Um enlace corresponde a uma aresta do grafo de simulação e simula o meio físico no qual as células são transmitidas. Os parâmetros que devem ser fornecidos são simplesmente a velocidade do enlace e o atraso de propagação, além, obviamente, dos nós interligados pelo enlace. O único parâmetro monitorado nos enlaces é a utilização dos mesmos.

A figura 5.4 mostra a especificação dos enlaces de uma rede a ser simulada. Como, normalmente, os enlaces existentes em uma rede possuem as mesmas características, o simulador possibilita a especificação de valores padrão para todos os enlaces, bastando em seguida serem redefinidas as características diferentes de cada enlace, normalmente os extremos da conexão.

```
links {
  default {
    bandwidth=150000000;
    latency=0.001;
  } /* default */
  { 1 -> 2; }
  { 2 -> 3; bandwidth=2000000; latency=0.010; }
} /* links */
```

Figura 5.4: Especificação dos enlaces de uma rede a ser simulada.

Parâmetros

- **links**: Seção para definição dos enlaces utilizados na simulação.
- **default**: Seção para definição dos valores que são comuns à maioria dos enlaces. Cada enlace pode redefinir seus próprios parâmetros.
- **bandwidth** (*bps*): Capacidade de transmissão do enlace.
- **latency** (*segundos*): Atraso de propagação imposto pelo enlace.

Os atrasos de propagação são utilizados como pesos das arestas do grafo para a determinação do melhor caminho entre os diversos nós. O roteamento é estático e centralizado e

foi implementado através do algoritmo de Floyd-Warshall, *all pairs shortest paths*, baseado no pseudo código apresentado em [BG92].

5.2.4 Gateway

O principal objetivo do simulador é possibilitar a simulação dos mecanismos de gerenciamento de capacidade estudados no capítulo 3. Sendo assim, o simulador obrigatoriamente deve oferecer componentes para interfacear redes locais com redes ATM. Um gateway, da mesma maneira como um comutador ATM, é implementado como uma função de um nó do grafo de simulação.

Um gateway é semelhante a um comutador ATM no que tange a estrutura interna, ou seja, ele também é constituído por um conjunto de portas que o interligam aos demais componentes da rede. A principal diferença entre os gateways e os comutadores ATM é a presença de uma porta utilizada para fazer a interface de um ambiente não orientado à conexão com a rede ATM. É no buffer desta porta que são armazenadas as rajadas do tráfego proveniente do ambiente não orientado à conexão. Uma outra diferença é que os gateways utilizam estruturas de dados adicionais para armazenar as informações necessárias à operação dos diferentes mecanismos de gerenciamento de capacidade.

A figura 5.5 mostra a especificação de um gateway que utiliza o mecanismo de Reservas para gerenciar a capacidade utilizada para o transporte do tráfego não orientado à conexão.

Parâmetros

- **buffer_size**: Tamanho do buffer da porta que liga o gateway ao ambiente não orientado à conexão. Este parâmetro pode receber o valor **INFINITY** que indica que o buffer possui tamanho infinito.
- **output_queue_size** (*células*): Tamanho do buffer da porta que liga o gateway à rede ATM.
- **queue_threshold** (*células*): Limiar a partir do qual células de baixa prioridade não são aceitas no buffer da porta que liga o gateway à rede ATM.
- **buffer_management**: Mecanismo utilizado para gerenciar o espaço ocupado no buffer da porta que liga o gateway à rede ATM. Este parâmetro possui o mesmo

```
node 3 {
  name="Gateway - LAN<->ATM";
  function=GATEWAY {
    buffer_size=INFINITY;
    output_queue_size=50;
    queue_threshold=40;
    buffer_management=PUSHOUT;
    leased_bandwidth=2000000;
    destination_nodes= 5, 7, 8;
    bandwidth_management=RESERVATION {
      maximum_incoming_load=90000000;
      T1=0.01;
      estimated_Tna=0.0034;
      increment_step=10000000;
      number_of_steps=9;
      bandwidth_release=MOVING_AVERAGE {
        window_size=0.1;
        number_of_samples=10;
      } /* MOVING_AVERAGE */
    } /* RESERVATION */
  } /* GATEWAY */
} /* node 3 */
```

Figura 5.5: Especificação de um gateway.

significado do parâmetro homônimo descrito anteriormente na especificação do computador ATM.

- **leased_bandwidth** (*bps*): Capacidade alocada permanentemente no canal virtual da rede ATM utilizado pelo gateway para o transporte do tráfego não orientado à conexão.
- **destination_nodes**: Indica a quais outros gateways o gateway que está sendo definido estará conectado através de canais virtuais permanentes. No exemplo da figura 5.5, o gateway 3 estará ligado aos gateways 5, 7 e 8. No início da simulação, portanto, devem ser estabelecidos os canais virtuais que interligam todos os gateways da rede a ser simulada.
- **token_pool_size**: Indica o tamanho do buffer do mecanismo de policiamento utili-

zado para policiar a capacidade alocada ao canal virtual que interliga os gateways.

- **bandwidth_management**: Indica qual estratégia será utilizada para gerenciar as capacidades nos enlaces de comunicação entre os gateways: Anúncio de Capacidades *On-The-Fly* ou Híbrido, Reservas ou Estatística de Atraso Fim a Fim. Os possíveis valores para este campo são respectivamente **BA_ON_THE_FLY**, **BA_HYBRID**, **RESERVATION** e **RESERVATION2**. O valor **RESERVATION2** é utilizado para especificar a variante para o mecanismo de reservas apresentada em [GTG93]. De acordo com a escolha, novos parâmetros terão de ser definidos para dimensionar o mecanismo de gerenciamento de capacidade escolhido. A seguir são definidos os parâmetros específicos de cada um dos mecanismos.

No caso do mecanismo de Reservas é necessário definir os seguintes parâmetros:

- **maximum_incoming_load** (*bps*): Capacidade máxima que pode ser enviada para o gateway a partir da rede local. Por exemplo, no caso de uma rede FDDI, essa capacidade fica em torno de 90Mbps.
- **T1** (*segundos*): Tempo utilizado para se disparar a primeira solicitação de incremento de capacidade. Este parâmetro especifica o tempo necessário para o buffer de entrada do gateway ser ocupado até o primeiro limiar. Basicamente, ele dimensiona o espaço necessário para armazenar o tráfego resultante da diferença entre a taxa de chegada e a taxa alocada permanentemente.
- **estimated_Tna** (*segundos*): Estimativa inicial do tempo gasto pela rede para responder uma solicitação de incremento de capacidade de um canal virtual já estabelecido. Este parâmetro é utilizado no cálculo dos limites no buffer do gateway.
- **increment_step**: Quantidade de capacidade adicional solicitada à rede quando a ocupação do buffer de entrada do gateway atinge um determinado limiar.
- **number_of_steps**: Número de passos utilizados para o mecanismo de reserva de capacidades atingir a alocação máxima de capacidade. A alocação máxima corresponde a uma capacidade suficiente para transmitir a maior carga proveniente da rede local.
- **bandwidth_release**: Este parâmetro indica como deve ser feita a liberação de capacidade quando o gateway percebe que há mais capacidade alocada do

que a necessária. Os possíveis valores são **HALF_INCREMENT**, que indica que deve ser liberada uma capacidade igual à metade de um “passo de capacidade”, e **MOVING_AVERAGE**, que indica que a capacidade deve ser liberada de acordo com as medições do tráfego. No caso da liberação de capacidade ser baseada nas medições do tráfego (**MOVING_AVERAGE**), é necessário definir os seguintes parâmetros:

- * **window_size** (*segundos*): Tempo em segundos utilizado para registrar a taxa de bits recebida. A taxa recebida corresponde a uma amostra que será utilizada para calcular a média móvel do tráfego.
- * **number_of_samples**: Número de amostras utilizadas para o cálculo da média do tráfego.

Para utilizar a variante proposta em [GTG93] (**RESERVATION2**), os parâmetros relativos ao dimensionamento dos limiares no buffer não são especificados. Todavia, deve-se especificar o tempo que a taxa de chegada deve ficar acima da taxa alocada para ser disparada uma renegociação. Isso é feito através do parâmetro **crit_A** apresentado logo abaixo.

O mecanismo de Anúncio de Capacidade Híbrido é especificado através dos seguintes parâmetros:

- **crit_A**: Tempo que o tráfego deve ficar acima da taxa alocada para ser disparada uma solicitação de incremento de capacidade.
- **crit_D**: Tempo que o tráfego deve ficar abaixo da taxa alocada para ser disparada uma solicitação de decremento de capacidade.
- **allocation_threshold**: Percentual da taxa residual permitido para que o tráfego excedente possa ser transmitido. Por exemplo, se a capacidade residual for de 50Mbps e **allocation_threshold** for igual a 10 %, células abusivas com taxas até 5Mbps podem ser transmitidas sem disparar o mecanismo de solicitação de incremento de capacidade.
- **deallocation_threshold**: Percentual da taxa alocada que deve ser liberado quando o tráfego fica abaixo da taxa alocada por um período igual a **crit_D**.
- **probe_interval**: Indica o intervalo de tempo entre células de sondagem. Mais especificamente, a cada **probe_interval** o gateway destino envia uma célula ao

longo da conexão até o gateway origem para determinar a menor capacidade disponível nos enlaces pertencentes à conexão.

Como o mecanismo de Anúncio de Capacidade Híbrido faz alocação e liberação de capacidade baseadas em medições do tráfego, ou seja, na média móvel, os parâmetros **window_size** e **number_of_samples** também devem ser fornecidos. Esses parâmetros possuem os mesmos significados dos descritos no mecanismo de Reservas.

O mecanismo Anúncio de Capacidade *On-The-Fly* é o mais simples, em termos de parâmetros, entre os mecanismos estudados e é especificado através de um único parâmetro: **probe_interval** que é equivalente ao parâmetro de mesmo nome presente no mecanismo Híbrido.

5.2.5 Aplicação

Uma aplicação é o componente do simulador que emula o comportamento de uma aplicação do usuário em um equipamento terminal. Ela pode ser considerada como um gerador de tráfego, com taxa de bits constante ou variável. Uma aplicação é definida como uma função de um nó do grafo de simulação. O nó de uma aplicação é diferenciado dos demais nós pelo fato de existirem dois tipos básicos de aplicações, aplicações **CBR**, com taxa constante de bits, e aplicações **VBR**, com taxa variável de bits.

A figura 5.6 apresenta a especificação de uma aplicação que produz tráfego com taxa variável de bits. O exemplo mostra uma fonte com os tempos de chegada de rajadas de dados e de duração de rajadas determinados por uma variável aleatória com distribuição exponencial, esse tipo de fonte foi denominado de **POISSON**. A seguir serão apresentados os diferentes tipos de fontes de dados implementados no simulador e os parâmetros necessários para suas especificações.

Parâmetros

Uma fonte com taxa constante de bits é definida como uma função **CBR** de um nó do grafo de simulação e possui os seguintes parâmetros:

- **peak_rate** (*bps*): Número de bits que devem ser transmitidos por segundo.
- **start_time** (*segundos*): Tempo em que a fonte deve iniciar a geração de dados.
- **duration_time**(*segundos*): Tempo que a fonte permanece ativa.

```
node 5 {
  name="Fonte 1";
  function=VBR {
    POISSON {
      peak_rate=10000000;
      burstiness=10;
      mean_burst_length=4500;
      on_seed=377003613;
      off_seed=1752629996;
    } /* POISSON */
    start_time=10;
    duration_time=100;
    connection_type=SWITCHED;
    source_type=CD;
    destination_node=6;
  } /* VBR */
} /* node 5 */
```

Figura 5.6: Especificação de uma fonte VBR.

- **number_of_bytes** (*bytes*): Número de bytes a ser transmitido pela fonte. Esse parâmetro pode ser especificado em substituição ao parâmetro **duration_time**.
- **destination_node**: Nó para onde devem ser enviados os dados.
- **connection_type**: Indica se deve ser estabelecida uma conexão permanente (**PERMANENT**) ou comutada (**SWITCHED**). Se for indicado o valor **PERMANENT**, a conexão é estabelecida no início da simulação; caso contrário, a conexão será estabelecida somente no início da atividade da fonte.

No caso de fontes com taxas variáveis de bits (fontes **VBR**), é possível escolher dentro de um conjunto pré-estabelecido de funções qual será o comportamento do tráfego gerado. A seguir são descritas as funções implementadas (os parâmetros específicos de cada fonte serão abordados mais abaixo):

- **POISSON**: Indica que deve ser utilizada a distribuição exponencial, tanto para o comprimento de rajada quanto para o período de silêncio.

- **FFT**: Indica uma fonte de dados auto-semelhante na qual os dados são gerados através do algoritmo da FFT de Vernon Paxson [Pax97].
- **RMD**: Da mesma maneira que a declaração FFT, especifica uma fonte de dados auto-semelhante. Entretanto, os dados são gerados através do algoritmo RMD proposto em [LEWW95].
- **TRACE**: Indica que os dados da fonte são obtidos a partir de um arquivo em disco. O arquivo deve conter dois parâmetros por linha: o tempo de chegada e o número de bytes do dado.
- **AG_TRACE**: A maioria dos estudos de auto-semelhança é baseada em dados agregados. Com este tipo de fonte, o simulador possibilita a utilização desses dados, bastando que seja fornecido o nome do arquivo em disco que os contém. O arquivo deve conter o número de bytes que deve ser gerado em um determinado intervalo de tempo.

Além do comportamento da fonte, uma especificação VBR deve conter os parâmetros **start_time**, **duration_time**, **destination_node** e **connection_type**. Esses parâmetros possuem os mesmos significados dos parâmetros descritos acima na especificação de fontes CBR. As fontes VBR são utilizadas, também, para modelar os dados gerados por redes locais, necessários para análise dos mecanismos de gerenciamento de capacidade implementados nos gateways. Para utilizar esta funcionalidade, deve ser especificado o seguinte parâmetro:

- **source_type**: Indica se a fonte será orientada à conexão (**CO**) ou não orientada à conexão (**CL**). No caso de fontes não orientadas a conexões, os dados são sempre enviados para um gateway. No caso de fontes orientadas a conexões, os dados podem ser enviados diretamente para um comutador ATM.

Uma fonte do tipo POISSON é especificada pelos seguintes parâmetros:

- **peak_rate** (*bps*): Taxa de bits que deve ser transmitida quando a fonte está no período ativo (taxa de pico).
- **mean_burst_length** (*bytes*): Comprimento médio de rajadas em bytes.
- **burstiness**: Grau de explosividade da fonte.

- **on_seed**: Semente utilizada para a geração dos dados correspondentes à variável aleatória que modela o período ativo da fonte.
- **off_seed**: Semente utilizada para a geração dos dados correspondentes à variável aleatória que modela o período de silêncio da fonte.

As fontes dos tipos FFT e RMD devem possuir os seguintes parâmetros:

- **byte_average** (*bytes*): Média de bytes por intervalo amostral. Intervalo amostral é o tempo em que os dados são registrados quando coletados em um ambiente real. Neste caso, os algoritmos de geração deverão gerar amostras com média igual a este parâmetro.
- **byte_variance**: Variância dos dados gerados por intervalo amostral.
- **seed**: Semente utilizada para geração de números aleatórios.
- **time_interval**: Tempo corresponde ao intervalo de coleta de dados quando os dados são coletados em um ambiente real.
- **transformation**: Indica qual transformação deve ser aplicada aos dados gerados. As técnicas de geração de *traces* auto-semelhantes discutidas no capítulo 4 geram dados que podem possuir valores negativos e valores maiores do que a taxa máxima possível. Isso ocorre principalmente quando a variância dos dados é alta. Existem algumas abordagens para se evitar valores negativos e para se limitar os valores dentro de um máximo permitido. A seguir serão discutidas algumas dessas abordagens, sendo que as que foram implementadas são as transformações exponencial (**EXPONENTIAL**) e a que substitui um valor negativo por zero (**INT**).

A primeira abordagem consiste em transformar a seqüência auto-semelhante x_1, \dots, x_n em uma nova seqüência e^{x_1}, \dots, e^{x_n} e utilizar esta última na simulação [Pax97, KVJ⁺98]. Os valores e^{x_i} são arredondados para o inteiro mais próximo. Este método sempre gera números positivos ou zero. Os novos dados ainda exibem dependências em longos intervalos, mas não constituem mais um Ruído Gaussiano Fracionário, como os dados originais [Pax97]. Um problema com este método é que a média da nova seqüência não guarda uma relação exponencial com a média da seqüência original, o que torna bastante difícil obter uma seqüência com uma média específica.

Uma segunda abordagem consiste em evitar a exponenciação, mas simplesmente truncar os valores negativos para zero. Esta abordagem também possui a desvantagem de alterar a média, aumentando sempre o valor da mesma nos dados transformados.

Uma terceira abordagem consiste em simplesmente ignorar valores negativos e valores maiores do que o máximo permitido [KVJ⁺98]. Um dos problemas com esta técnica é que como não se tem inicialmente uma idéia da quantidade de valores negativos existentes, não há como se determinar a priori o tamanho da seqüência necessária para se simular um tempo específico. Além disso, como nas abordagens anteriores, a média e a variância dos dados transformados não guardam nenhuma relação com os dados originais.

As fontes cujos dados são *traces* do mundo real devem ser especificadas através de um único parâmetro:

- **trace_file**: Nome do arquivo em disco no qual os dados estão armazenados.

No caso dos *traces* em que os dados foram agregados (**AG_TRACE**), é necessário, além do nome do arquivo (**trace_file**), especificar o parâmetro **time_interval**. Esse parâmetro possui o mesmo significado do parâmetro homônimo descrito anteriormente.

Capítulo 6

Comparação dos Mecanismos

Neste capítulo serão descritos os experimentos de simulação que foram desenvolvidos para investigar como os mecanismos de suporte ao tráfego não orientado a conexão se comportam quando alimentados com um modelo de tráfego bem próximo da realidade e também comparar os resultados obtidos com os resultados produzidos pelo modelo convencional de Poisson. A escolha da simulação foi motivada principalmente pela ausência de uma rede ATM real, com as características necessárias à análise, e também pelas limitações das técnicas analíticas. Uma característica atrativa da simulação é a sua flexibilidade para explorar vários parâmetros e vários modelos.

Este capítulo está organizado da maneira a seguir. A seção 6.1 apresenta alguns resultados preliminares relativos ao tráfego auto-semelhante discutido no capítulo 4. A seção 6.2 descreve a rede a ser simulada e os parâmetros adotados. A seção 6.3 discute os resultados obtidos.

6.1 Resultados Preliminares

Antes de serem apresentados os resultados relativos aos mecanismos estudados no capítulo 3, serão apresentados alguns resultados sobre os algoritmos de geração de *traces* auto-semelhantes e a influência do parâmetro H nesse tipo de tráfego.

O primeiro ponto analisado foi a capacidade dos algoritmos gerarem dados de acordo com os parâmetros fornecidos. Como ambos algoritmos (FFT e RMD) são baseados em aproximações, os parâmetros estimados a partir dos dados gerados divergiram dos parâmetros fornecidos. Para cada algoritmo, o parâmetro H foi variado de 0,5 a 0,95

em incrementos de 0,05, sendo que para cada valor de H foram gerados 10 traces com diferentes sementes, resultando em um total de 200 traces, cada um com 32768 amostras. A tabela 6.1 mostra os parâmetros estimados. Os parâmetros foram estimados utilizando os estimadores R/S e o Gráfico da Variância, ambos produziram resultados equivalentes. Cada valor de \hat{H} corresponde à média dos 10 valores obtidos por um dos estimadores. Para cada valor de H , é mostrado, também, o intervalo dos valores obtidos. É possível observar pelos dados que ambos algoritmos produziram estimativas bem próximas aos parâmetros fornecidos para os valores de H até 0,75; a partir deste valor a diferença torna-se significativa, sendo que o valor fornecido de H não se encontra nem mesmo no intervalo dos valores estimados. Pelos intervalos mostrados, é possível observar, também, que a qualidade dos dados gerados dependem fortemente da semente escolhida. Por exemplo, no caso $H = 0,5$ houve um caso em que os dados gerados apresentou uma estimativa para H de 0,46, um valor bem distante do valor fornecido, apesar da média dos valores estimados ficar próxima ao valor fornecido para H . Esse fator deve ser levado em consideração quando os dados forem utilizados para alimentar algum tipo de simulação.

H	FGN		RMD	
	\hat{H}	Intervalo	\hat{H}	Intervalo
0,50	0,4940	0,460 - 0,513	0,4936	0,470 - 0,513
0,55	0,5493	0,531 - 0,577	0,5398	0,522 - 0,560
0,60	0,5981	0,577 - 0,625	0,5865	0,560 - 0,608
0,65	0,6463	0,624 - 0,671	0,6326	0,605 - 0,655
0,70	0,7052	0,689 - 0,714	0,7035	0,671 - 0,736
0,75	0,7459	0,730 - 0,753	0,7451	0,712 - 0,780
0,80	0,7879	0,774 - 0,792	0,7851	0,760 - 0,822
0,85	0,8245	0,812 - 0,849	0,8231	0,789 - 0,862
0,90	0,8608	0,847 - 0,882	0,8578	0,824 - 0,897
0,95	0,8906	0,876 - 0,910	0,8888	0,856 - 0,927

Tabela 6.1: Valores estimados de H dos algoritmos FGN e RMD.

O segundo ponto analisado foi a qualidade do tráfego gerado pelo método da FFT como modelo para os dados do *trace* Ethernet discutido no capítulo 4. Basicamente o que se procurou verificar foi quais os prejuízos causados à rede, em relação à perda de células, pelo tráfego Ethernet e comparar os resultados obtidos com o tráfego gerado sinteticamente. Os dados foram utilizados para alimentar uma rede com um único comutador ATM,

sendo que as células foram direcionadas para um enlace de baixa capacidade (2Mbps). O tamanho do buffer (B) que liga o comutador ao enlace de saída foi variado de 50 a 3200 células e as taxas de perda de células registradas para cada valor de B .

Os parâmetros utilizados para a geração sintética dos dados foram extraídos do *trace* da *Bellcore*. Como já discutido no capítulo 4, este *trace* foi coletado a partir de uma rede Ethernet real. O *trace* é formado por um milhão de observações, sendo que cada observação corresponde ao tempo de chegada e o tamanho em bytes de um quadro Ethernet. Os parâmetros são os seguintes: parâmetro de Hurst H igual a 0,82; 1342 bytes por intervalo amostral (10 milissegundos), o que corresponde a uma taxa de 1073600 bits por segundo, e variância igual a 4770242. Cabe salientar que os parâmetros utilizados na geração sintética do *trace* foram ajustados para que o algoritmo gerasse dados que produzissem os valores mencionados, e não utilizados cegamente como entrada para o algoritmo de geração. Esse cuidado é necessário pelo fato do algoritmo gerar dados cujos parâmetros estimados divergem dos parâmetros fornecidos. A figura 6.1 mostra os valores obtidos.

Os dados gerados sinteticamente apresentaram resultados praticamente idênticos aos do *trace* real para os valores de B igual ou superior a 200 e divergiram em 11,29% para $B = 50$ e 1,15% para $B = 100$. Essa comparação foi feita com o *trace* Ethernet agregado em intervalos de 10 milissegundos. A figura 6.1 também mostra a diferença entre a taxa de perda causada pelo *trace* real sem agregação em relação a taxa de perda causada pelo *trace* real agregado. Neste caso, as diferenças entre os dados reais e os dados agregados foram de 18,78% para $B = 50$; 16,22% para $B = 100$; 6,22% para $B = 200$; 3,36% para $B = 400$; 2,02% para $B = 800$; 1,08% para $B = 1600$ e tornou-se desprezível para $B = 3200$. Esses resultados mostram que os dados gerados sinteticamente se adaptam muito bem aos dados Ethernet agregados, entretanto, percebe-se que há ainda uma diferença significativa na taxa de perda de células quando se compara os dados como eles foram coletados em relação aos dados agregados. O processo de agregação torna o tráfego bem menos prejudicial à rede.

O terceiro ponto analisado foi a influência do parâmetro H no atraso médio e na taxa de perda de células em uma rede ATM. A rede simulada e os parâmetros utilizados para a geração do tráfego (taxa média de bits e variância) são os mesmos descritos anteriormente.

As figuras 6.2 e 6.3 mostram as taxas de perda de células ocasionadas pelos diferentes valores de H para os dois algoritmos de geração implementados. As figuras mostram

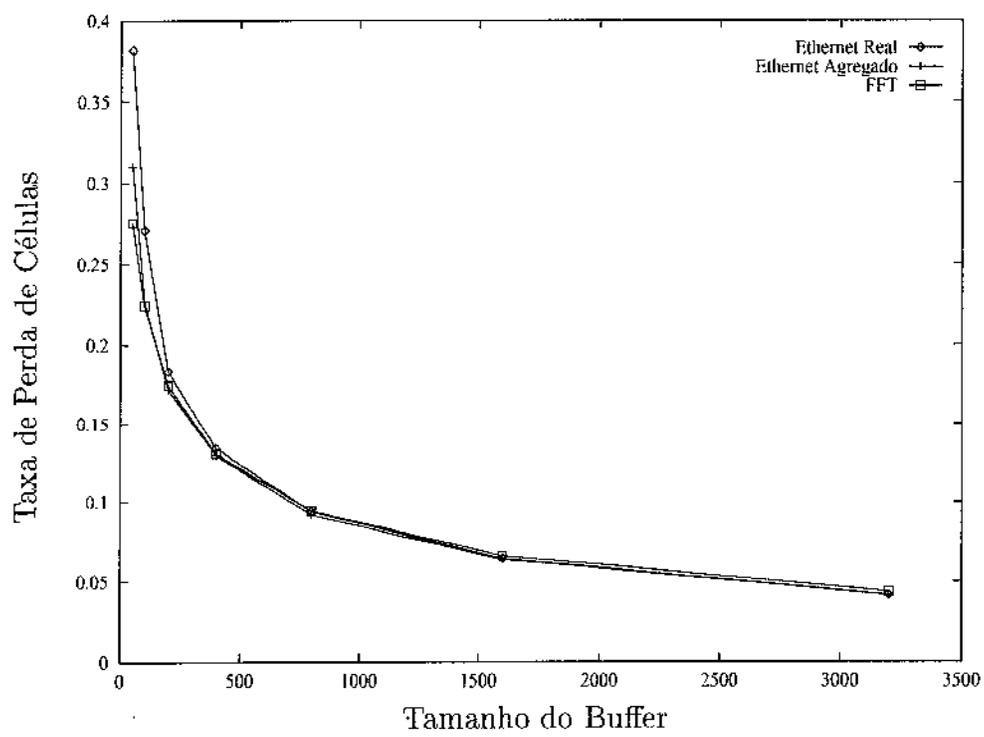


Figura 6.1: Taxa de perda de células ocasionada pelo tráfego Ethernet e pelo tráfego gerado sinteticamente pelo método da FFT.

que a adição de mais espaço no buffer do comutador diminui a taxa de perda de células. Entretanto, o aumento do buffer é bem menos efetivo quando a fonte de tráfego possui um alto grau de auto-semelhança. Por exemplo, para $H = 0,5$ o aumento do buffer de 50 para 800 reduziu a taxa de perda de 15,46% para 0%, ao passo que para $H = 0,95$ o aumento do buffer de 50 para 3200 reduziu a taxa de perda de 26,24% para apenas 11,31%. É importante salientar que a o valor 0 da taxa de perda de células deve ser interpretado como uma taxa inferior a 10^{-5} , pelo fato do número de células geradas estar na ordem de 10^6 . É possível verificar, também, a diferença entre os tráfegos gerados pelos dois algoritmos. O tráfego gerado pelo método da FFT apresentou, sempre, uma taxa de perda de células maior do que a do algoritmo RMD.

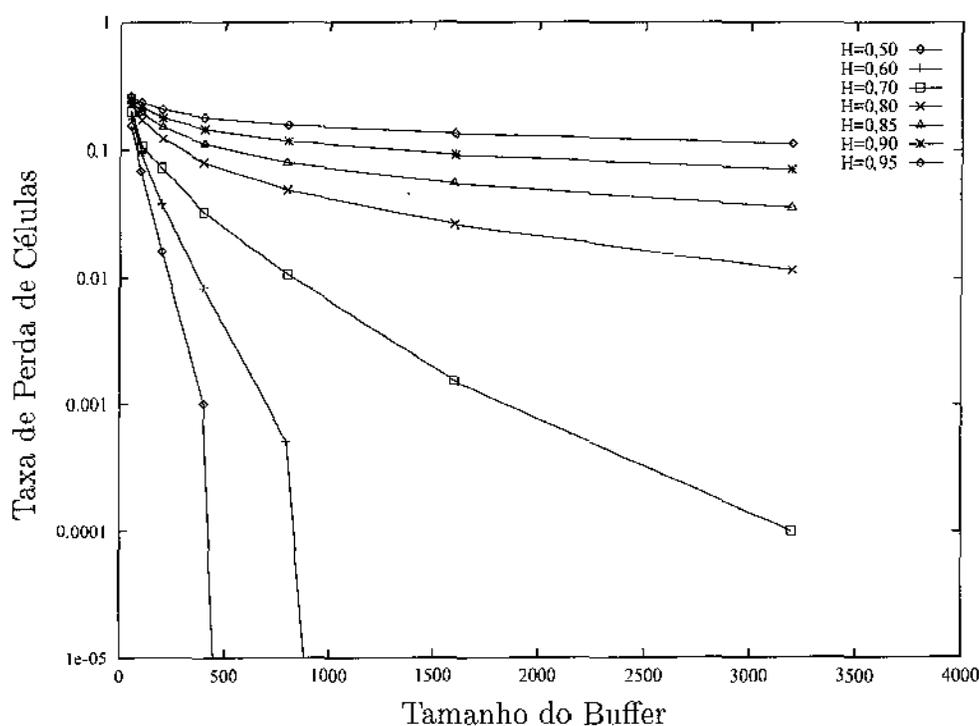


Figura 6.2: Taxa de perda de células utilizando o método da FFT

A figura 6.4 mostra o atraso médio enfrentado pelas células no buffer de saída do comutador ATM. Para este experimento o tamanho do buffer foi mantido fixo e igual a 200 células, e o parâmetro H foi variado de 0,5 a 0,95 com incrementos de 0,05. Percebe-se claramente que ao passo que o grau de auto-semelhança, valor de H , aumenta, o atraso médio também aumenta.

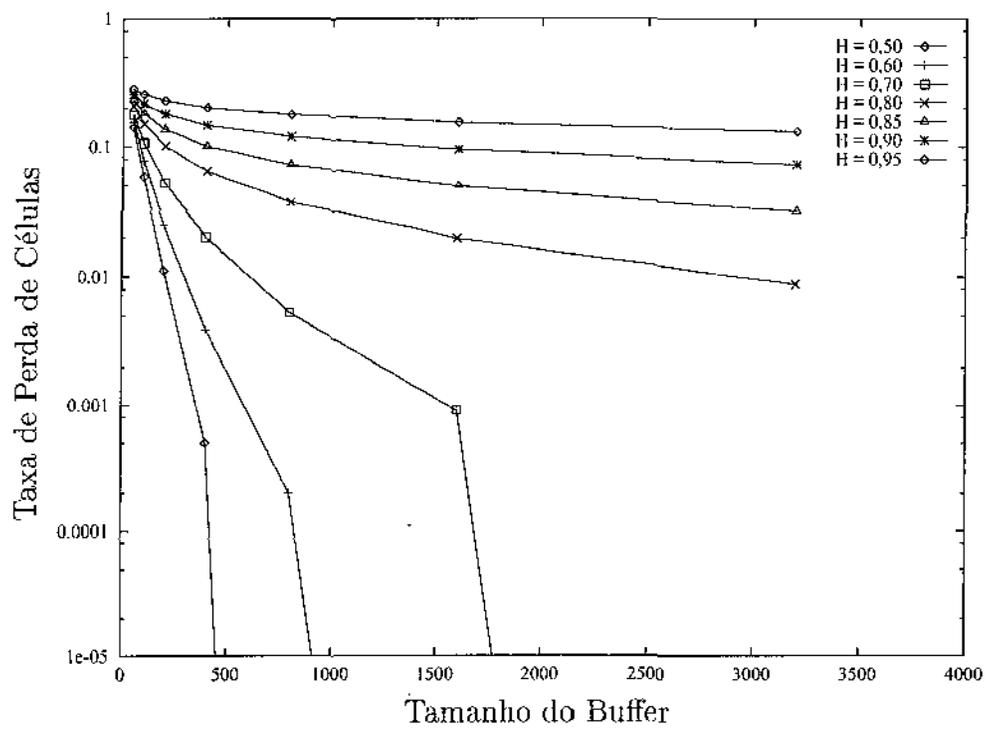


Figura 6.3: Taxa de perda de células utilizando o algoritmo RMD

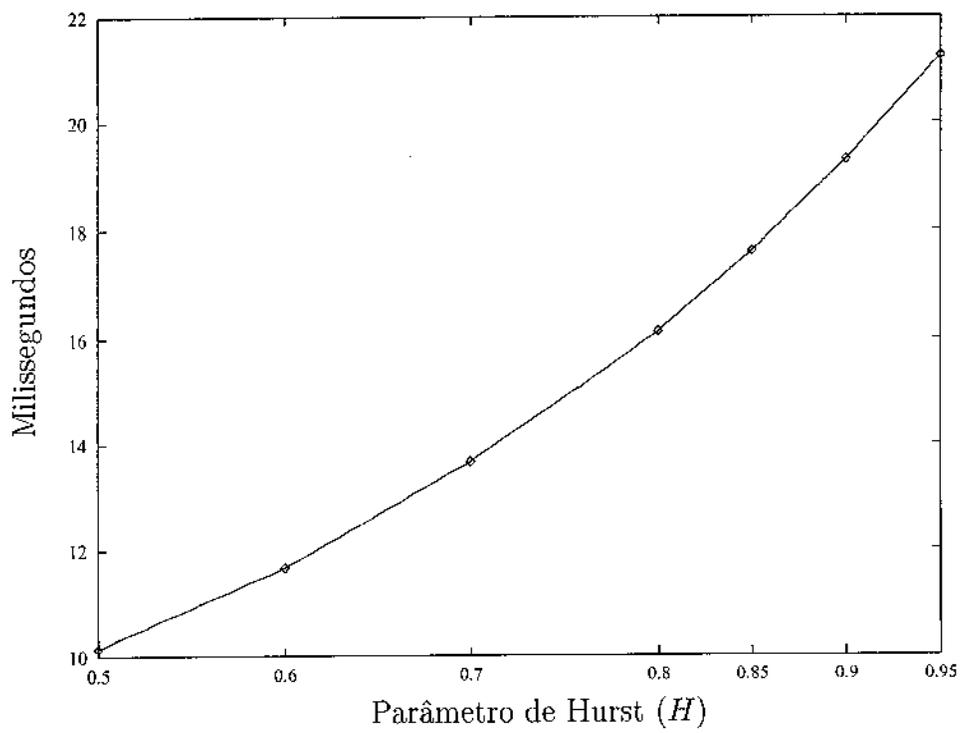


Figura 6.4: Atraso médio enfrentado pelas células no buffer do comutador ATM. O tamanho do buffer foi fixado em 200 células.

6.2 Modelo de Configuração da Rede e Parâmetros Adotados

O objetivo principal deste trabalho é investigar o comportamento de alguns mecanismos de gerenciamento de capacidade para o transporte de tráfego não orientado a conexão. Para se ter uma idéia do contexto em que estarão inseridos esses mecanismos, o sistema a ser simulado está representado na figura 6.5. Assume-se que todos os caminhos virtuais que conectam os diferentes pares de gateways passarão por um mesmo enlace de 150 Mbps, identificado na figura como enlace congestionado. Além dos caminhos virtuais destinados a interligar os gateways, serão considerados outros caminhos virtuais no mesmo enlace transportando dados orientados a conexão. O objetivo desses caminhos é oferecer uma mistura de tráfego a rede para verificar algum possível ganho apresentado pelos mecanismos devido à multiplexação estatística. O tráfego entre os gateways será denominado tráfego principal e o tráfego orientado a conexão será denominado tráfego de retaguarda.

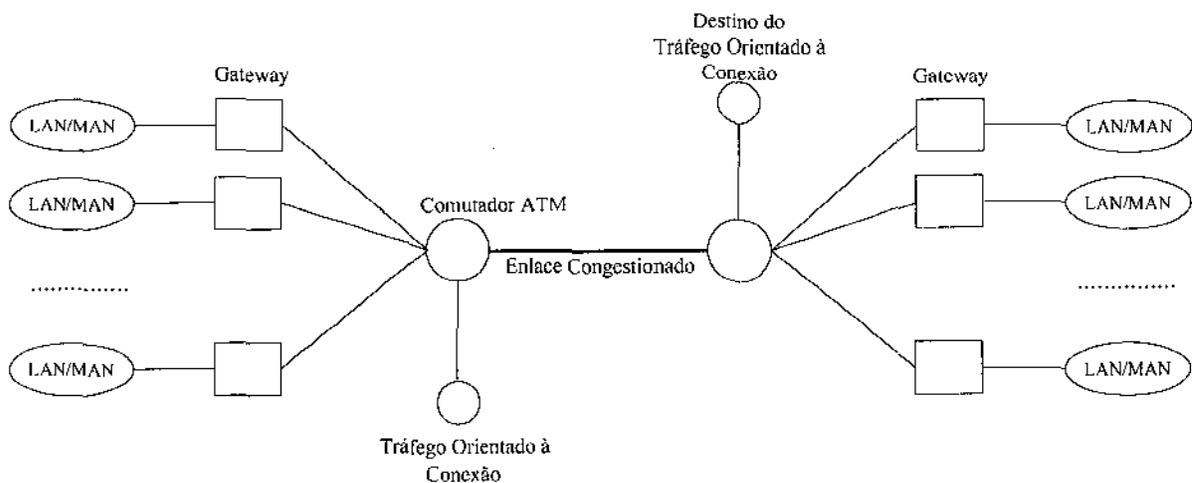


Figura 6.5: Configuração de rede para a simulação

Para os objetivos deste trabalho, o sistema apresentado pode ser simplificado para o modelo mostrado na figura 6.6. O sistema todo pode ser dividido em seis componentes: fontes de tráfego de retaguarda, fontes de tráfego principal, gateways para o tráfego principal, multiplexador para o tráfego de retaguarda, comutador ATM e sorvedouro (*sink*).

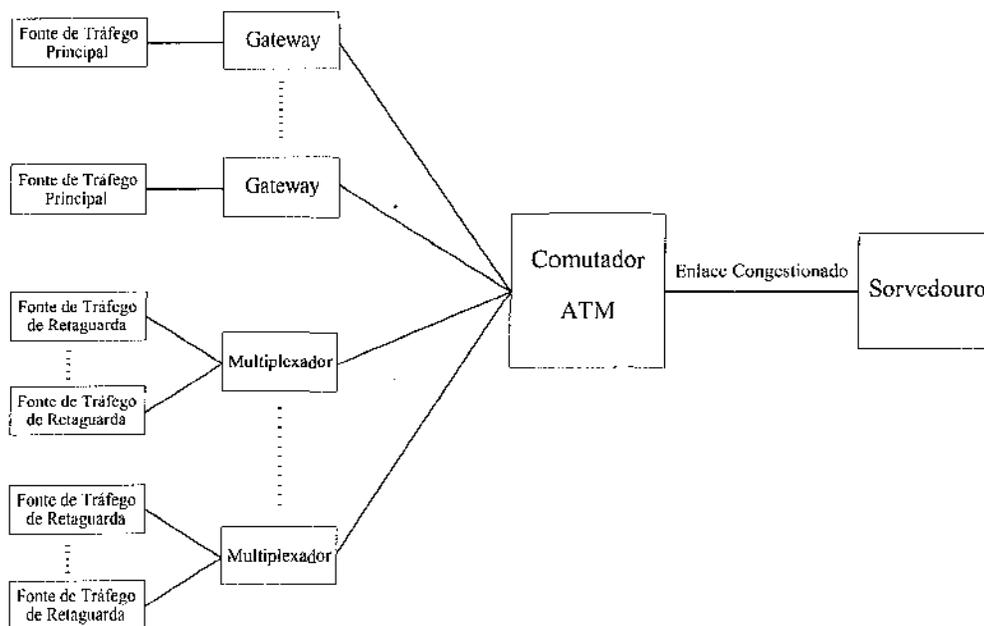


Figura 6.6: Modelo simplificado da rede para a simulação

Fontes de Tráfego

No enlace congestionado, serão considerados, além dos caminhos virtuais destinados ao tráfego principal, três caminhos virtuais destinados ao tráfego de retaguarda, cada um transportando dez fontes de tráfego. Assume-se que todas as fontes são do mesmo tipo, representadas pelo modelo *On-Off*, e caracterizada pelos seguintes parâmetros: $B_p = 5$ Mbps, $b = 10$, e $L = 100$. Sendo que B_p é a taxa de pico, b é o grau de explosividade e L é comprimento médio de rajada em células. A figura 6.7 mostra o relacionamento dos parâmetros no modelo. Como explicado no capítulo 4, o comprimento de rajada e o intervalo de tempo entre as rajadas são variáveis aleatórias com distribuição exponencial.



Figura 6.7: Fonte *On-Off*.

Para cada caminho virtual de retaguarda é alocada uma capacidade fixa. A capacidade alocada é de 26,236 Mbps determinada a partir do modelo UAS (*Uniform Arrival System*) para o caso de buffer finito [Mon90]. Esses dados, bem como as características das fontes de retaguarda e os parâmetros de dimensionamento dos três mecanismos, foram extraídos de [GTG93]. Isso foi feito, principalmente, para validar a implementação comparando os resultados obtidos com os apresentados em [GTG93]. Cabe salientar que em [GTG93] as fontes principais também são representadas pelo modelo *On-Off*, o que será alterado neste trabalho.

Cada par de gateways mantém um caminho virtual permanente com baixa capacidade alocada para suportar o seu tráfego principal. A capacidade inicial alocada para cada caminho virtual é 0,1 Mbps. Os caminhos virtuais irão transportar dados que serão gerados a partir do método da FFT. O que se procurará observar é a influência do parâmetro H na taxa de perda de células e também no atraso enfrentado pelas células do tráfego principal e do tráfego de retaguarda. O parâmetro H será fixado em 0,5 para representar dados sem características de auto-semelhança e em 0,85 para representar dados com características de auto-semelhança. Além do parâmetro H , o gerador de tráfego deve receber como parâmetros a média e a variância do número de bytes por intervalo amostral. A média do número de bytes por intervalo amostral de 10 milissegundos foi fixada em 2250 bytes tanto para $H = 0,5$ quanto para $H = 0,85$, isso corresponde a uma taxa média de bits de aproximadamente 2 Mbps. Para se determinar o valor da variância, é necessário primeiro definir um parâmetro normalmente utilizado em estudos de auto-semelhança. O parâmetro "*peakedness*" representa a razão entre a variância e a média. Para este estudo foi utilizado o valor de 3000 para o parâmetro *peakedness*, conforme extraído dos dados da *Bellcore*, o que resulta em uma variância igual a 6750000.

Comutador ATM

As características do comutador ATM utilizado são as mesmas das descritas no capítulo 5, ou seja, são não bloqueantes e com armazenamento na saída. O tamanho do buffer da porta de saída foi fixado em 50 células e para o gerenciamento do buffer foi utilizada a estratégia *Threshold*, sendo que o limiar para descarte foi fixado em 40 células.

Gateways

Os gateways são os responsáveis pelo controle da capacidade alocada ao caminho virtual que transporta o tráfego principal (não orientado a conexão). Quatro mecanismos diferentes foram implementados nos gateways: o mecanismo de reservas, a variante do mecanismo de reservas proposta em [GTG93], o mecanismo de anúncio de capacidade “*on-the-fly*” e o mecanismo de anúncio de capacidade híbrido. Por questões de simplicidade, apenas os três últimos serão analisados. O mecanismo de reservas foi deixado de fora porque o dimensionamento dos limiares no buffer é um processo muito complicado e pelo fato da variante proposta ter apresentado melhores resultados, como reportado em [GTG93]. A partir de agora será utilizado o termo *mecanismo de reservas* para indicar a variante apresentada em [GTG93] e não mais a proposta original de Mongiovi e outros [MFT91]. Nesta variante, os dados provenientes do ambiente não orientado a conexão são armazenados no gateway e transmitidos apenas na taxa alocada ao canal entre os gateways, como no mecanismo original. A diferença é que as renegociações não são disparadas pela ocupação do buffer e sim pela verificação da taxa de chegada. Se a taxa de chegada se mantém acima da taxa alocada ao canal entre os gateways por um período superior a t_{crit_a} , o gateway dispara uma solicitação de incremento de capacidade. Diferente da proposta original que faz solicitações de incremento em vários passos de capacidade, na variante, a solicitação é feita com base na medição da taxa de chegada. Com isso, a variante consegue compensar mais rapidamente a diferença entre a taxa de chegada e a taxa alocada. Para liberação de capacidade, a variante utiliza um mecanismo semelhante ao de incremento. Se a taxa de chegada se mantém abaixo da taxa alocada ao canal entre os gateways por um período superior a t_{crit_d} , o gateway dispara uma solicitação de liberação de capacidade.

Uma questão importante no esquema de reservas é a determinação dos instantes para renegociação de capacidade. Como mencionado nos capítulos 3 e 5, o mecanismo utiliza dois parâmetros para disparar solicitações de capacidade: **crit_A** e **crit_D**. **crit_A** indica o tempo que a taxa de chegada de dados deve ficar acima da taxa alocada para ser disparada uma solicitação de capacidade adicional. **crit_D** indica o tempo que a taxa de chegada de dados deve ficar abaixo da taxa alocada para ser disparada uma solicitação de liberação de capacidade. Esses parâmetros são especificados em segundos no simulador. Para tornar a apresentação mais simples e clara, ao invés do segundo será utilizada uma unidade de tempo correspondente ao tempo de transmissão de cem células ATM em

um enlace de 150 Mbps denominada TS (*Time Slot*) e que equivale a 282,6 μ s. Os valores utilizados para **crit_A** e **crit_D** foram respectivamente 10 e 5 TS's. Quando o mecanismo detecta a necessidade de renegociar a capacidade alocada, a nova capacidade é determinada a partir da média móvel do tráfego de entrada. Para cada TS, a taxa de chegada é registrada e a média móvel é determinada a partir dos N últimos TS's. O valor de N foi fixado em 10.

Diferente do mecanismo de reservas que sempre transmite na taxa alocada ao caminho virtual, o mecanismo de anúncio de capacidade tenta tirar proveito da capacidade não alocada nos enlaces ao longo da conexão e procura transmitir grandes rajadas de dados diretamente na rede sem se preocupar em renegociar a capacidade alocada. Para isso, ele se baseia em informações de realimentação da rede. No simulador foi implementada a estratégia de sondagem de capacidade (*bandwidth probing*). Nessa estratégia, o gateway destino envia periodicamente uma célula de sondagem em direção ao gateway origem, sendo que os comutadores ao longo da conexão informam as capacidades disponíveis em cada um dos enlaces. A capacidade transportada é sempre a menor entre as capacidades disponíveis. Para esta estratégia, assume-se que existe um canal de controle dedicado e que as células são enviadas regularmente em intervalos de 10 TS's (tempo equivalente ao tempo de transmissão de 1000 células ATM em um enlace de 150 Mbps). A menor capacidade disponível ao longo do caminho entre os gateways origem e destino indica o valor máximo de dados que podem ser transmitidos acima da taxa alocada e é utilizada, também, no mecanismo de policiamento de dois estágios discutido no capítulo 3.

O mecanismo de anúncio de capacidade híbrido tenta combinar as idéias dos dois outros mecanismos, ou seja, ele transmite rajadas diretamente na rede sem renegociar a capacidade alocada, mas se o volume de tráfego se mantiver maior que a capacidade alocada por um intervalo significativo de tempo, ele dispara uma solicitação de incremento de capacidade. Diferente do mecanismo de reservas que renegocia a capacidade alocada depois que a taxa de entrada se mantém acima da taxa alocada por um período superior a **crit_A**, o mecanismo híbrido renegociará a capacidade apenas se a taxa de entrada estiver acima da alocada por um período superior a **crit_A** e, além disso, se a taxa de entrada for superior a um determinado percentual da capacidade anunciada como disponível no caminho da conexão. O percentual da capacidade disponível que deve ser considerado é especificado através do parâmetro **allocation_threshold** e foi fixado em 10%, ou seja, o mecanismo irá permitir que sejam transmitidos dados com taxa de chegada acima da

taxa alocada desde que a diferença entre a taxa de chegada e a taxa alocada seja inferior a dez por cento da capacidade anunciada como disponível.

Como a idéia do mecanismo híbrido é evitar ao máximo solicitações de aumento de capacidade, dado que ele permite a transmissão de dados acima da capacidade alocada, ele renegocia com a rede apenas se a rajada for mantida por um período de tempo que a renegociação traga alguma vantagem. Por isso, o parâmetro **crit_A** é dimensionado de acordo com o tempo necessário para que uma renegociação de capacidade seja completada. Este parâmetro foi fixado em 35 TS's que é comparável ao tempo estipulado para a renegociação de capacidade definido logo abaixo. Visando manter sempre uma baixa capacidade alocada, o parâmetro **crit_D** foi fixado em 2 TS's, bem abaixo do parâmetro para alocação, isso significa que se a taxa de chegada de dados ficar abaixo da taxa alocada por um período de tempo igual a 2 TS's, o mecanismo dispara imediatamente uma solicitação de decremento de capacidade.

Para cada solicitação de renegociação de capacidade, tanto no mecanismo de reservas como no mecanismo de anúncio de capacidade híbrido, existe um tempo para que a rede responda de forma positiva ou negativa à solicitação. Este tempo inclui os atrasos de propagação e processamento e foi fixado em 10 milissegundos.

O tamanho do buffer de entrada dos gateways foi fixado em 1000 células para os três mecanismos. A tabela 6.2 resume os parâmetros utilizados nos três mecanismos.

6.3 Resultados Obtidos

No experimento simulado, o que se procurou verificar foi principalmente a taxa de perda de células provocada pelos três mecanismos de gerenciamento de capacidade estudados em função do número de caminhos virtuais que transportam o chamado tráfego principal. Todos os parâmetros discutidos na seção anterior foram mantidos fixos com exceção do grau de auto-semelhança do tráfego principal e o número de conexões entre gateways.

A figura 6.8 apresenta a taxa de perda de células para o tráfego principal com parâmetro de Hurst igual a 0,5 em função do número de conexões entre gateways. Os resultados são coerentes com os apresentados em [GTG93], isto é, o mecanismo de reservas apresentou sempre uma taxa de perda maior que os demais, o mecanismo de anúncio de capacidades híbrido apresentou taxas inferiores ao de reservas e superiores ao de anúncio de capacidade *on-the-fly*.

Parâmetro	Reservas	Híbrido	On-the-fly
Tamanho do Buffer do Gateway	1000 células	1000 células	1000 células
Capacidade Alocada Inicialmente	0,1 Mbps	0,1 Mbps	0,1 Mbps
Tamanho do Buffer do Comutador ATM	50 células	50 células	50 células
Gerenciamento de Buffer	<i>Threshold</i>	<i>Threshold</i>	<i>Threshold</i>
Limiar para Descarte	40 células	40 células	40 células
t_{crit_a}	10 TS's	35 TS's	
t_{crit_d}	5 TS's	2 TS's	-
Tamanho da Janela do Mecanismo de Média Móvel	1 TS	1 TS	-
Número de Amostras para Média Móvel	10	10	-
Atraso de Renegociação de Capacidade	0,01s	0,01 s	-
Intervalo entre Células de Sondagem	-	10 TS's	10 TS's

Tabela 6.2: Parâmetros utilizados para o dimensionamento dos três mecanismos simulados.

Todas as perdas apresentadas pelo mecanismo de reservas foram verificadas nos gateways, já que o mecanismo só transmite na taxa alocada ao canal. No caso do mecanismo de anúncio de capacidade *on-the-fly* as perdas ocorreram todas dentro da rede, porque o mecanismo transmite a taxas superiores às alocadas aos canais e evita ao máximo o armazenamento de dados nos gateways. O mecanismo híbrido apresentou perdas tanto nos gateways quanto dentro da rede.

No mecanismo híbrido, quando a carga simulada foi baixa, as perdas foram basicamente verificadas dentro da rede, ao passo que quando a carga simulada foi aumentada as perdas passaram a ocorrer principalmente nos gateways. Isso se deve à diminuição da capacidade residual na rede e portanto à necessidade do mecanismo disparar solicitações de aumento da capacidade alocada. Quando a capacidade residual na rede é grande, o mecanismo híbrido transmite a taxas superiores à taxa alocada até um percentual, como discutido anteriormente, e portanto possui um comportamento semelhante ao do mecanismo *on-the-fly*. Entretanto, quando o número de caminhos virtuais na rede é aumentado, a capacidade residual diminui e conseqüentemente o mecanismo não pode transmitir rajadas

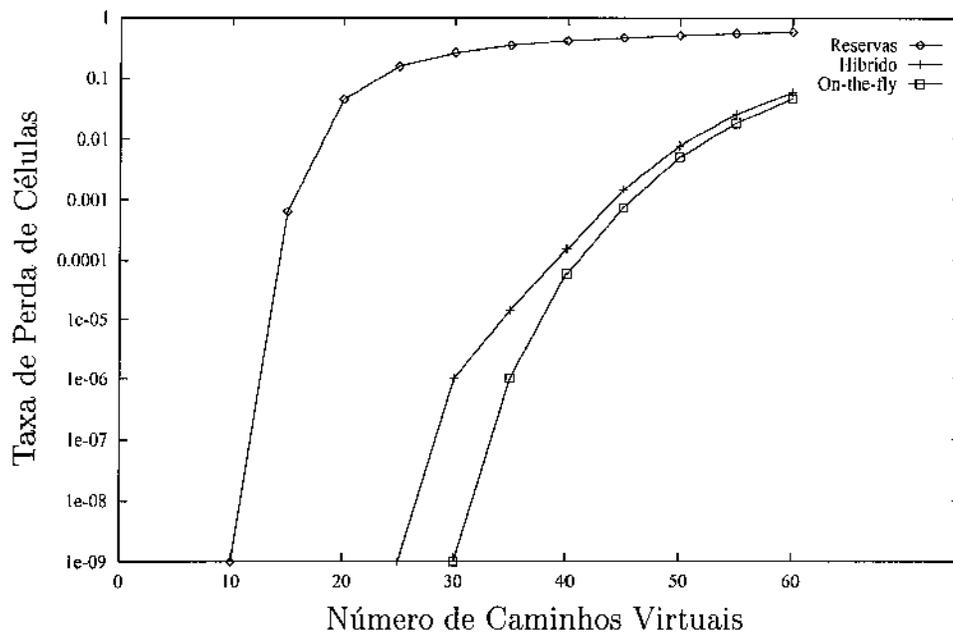


Figura 6.8: Taxa de perda de células em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,5.

excessivas por um período muito longo, devendo renegociar a capacidade alocada.

O comportamento inferior do mecanismo híbrido em relação ao mecanismo *on-the-fly*, como já explicado em [GTG93], deve-se ao seguinte fator: pelo fato do mecanismo híbrido poder transmitir dados excessivos até um determinado percentual da capacidade residual, diferente do mecanismo *on-the-fly* que transmite dados excessivos até a capacidade total disponível, os gateways passam a reter dados na entrada da rede. Quando há liberação de capacidade na rede, todos os gateways percebem a alteração da capacidade disponível, praticamente ao mesmo tempo, e passam a descarregar os dados acumulados na entrada a uma taxa superior, o que ocasiona um aumento considerável de chegada de células no comutador ATM que, conseqüentemente, passa a efetuar descartes.

A figura 6.9 apresenta a taxa de perda de células para o mesmo experimento e com a alteração do parâmetro de Hurst de 0,5 para 0,85. Novamente neste caso, os mecanismos apresentaram desempenhos semelhantes, ou seja, o mecanismo de reservas apresentou a maior taxa de perda de células, o mecanismo híbrido a segunda maior e o mecanismo *on-the-fly* a menor.

As figuras 6.10, 6.11, 6.12 apresentam as perdas para cada um dos mecanismos para os

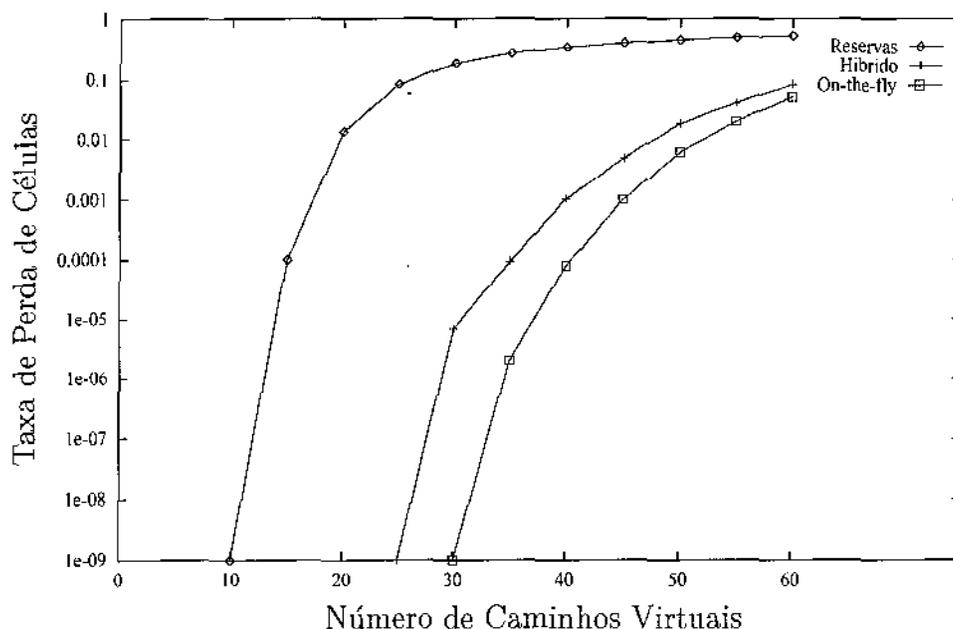


Figura 6.9: Taxa de perda de células em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,85.

valores de H iguais a 0,5 e 0,85. O ponto interessante observado neste experimento foi que, diferente dos mecanismos híbrido e *on-the-fly* que apresentaram taxas de perda maiores para H igual a 0,85 em relação a H igual a 0,50, o mecanismo de reservas apresentou taxa de perda inferior para H igual a 0,85. Isso pode ser explicado, provavelmente, pelo fenômeno de persistência nos dados com parâmetro de Hurst 0,85. O mecanismo de média móvel se adaptou melhor a este tipo de tráfego e provocou um número menor de renegociações com a rede. O número de renegociações de capacidade realizadas para o tráfego com H igual a 0,5 foi, em média, 14,43 % maior que o número de renegociações para o tráfego com H igual a 0,85.

As figuras 6.13 e 6.14 mostram, os atrasos fim a fim, descontados os atrasos de propagação, enfrentados pelas células do tráfego principal e pelas células do tráfego de retaguarda, respectivamente. Como pode ser observado nas figuras, o mecanismo de anúncio de capacidade *On-the-fly* produz os menores atrasos para o tráfego principal, mas em contrapartida produz os maiores atrasos para o tráfego de retaguarda. Os menores atrasos no tráfego principal se deve as baixas filas formadas no gateway origem, pelo fato do mecanismo transmitir a uma taxa superior a taxa alocada. Por este mesmo motivo, o mecanismo

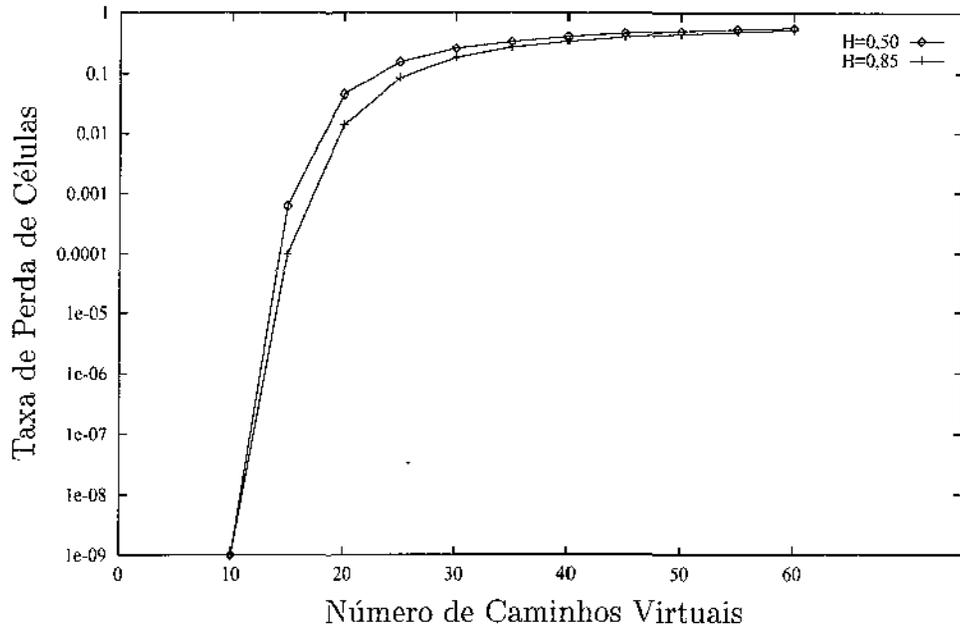


Figura 6.10: Taxa de perda de células do mecanismo de reservas para os valores de H iguais a 0,5 e 0,85.

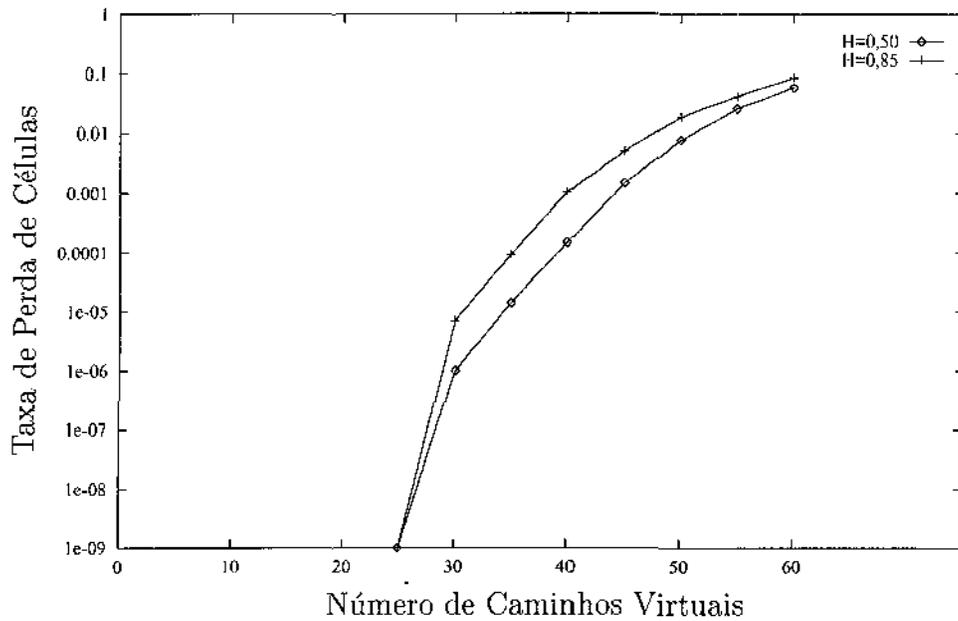


Figura 6.11: Taxa de perda de células do mecanismo de anúncio de capacidades híbrido para os valores de H iguais a 0,5 e 0,85.

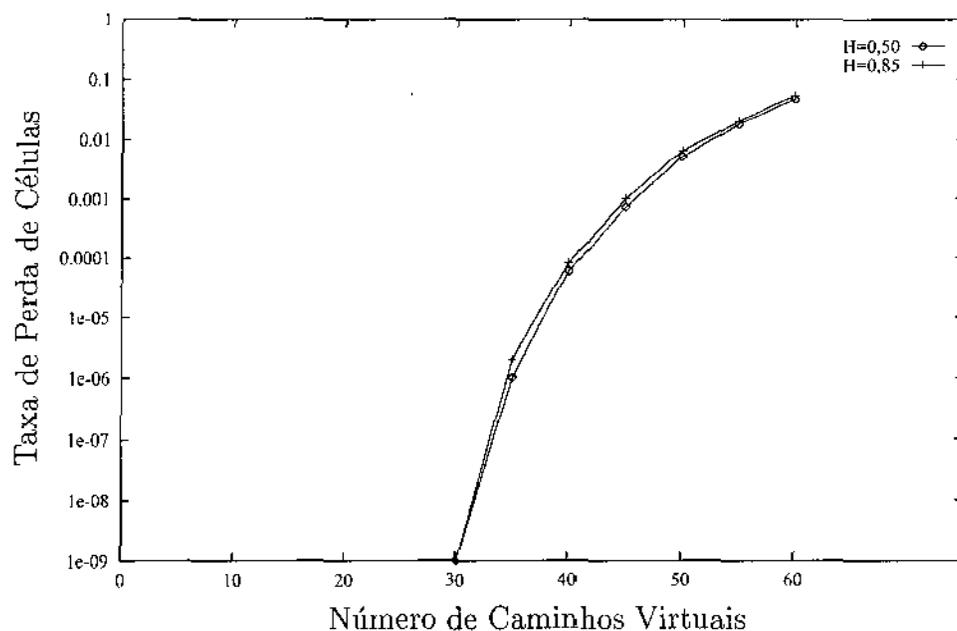


Figura 6.12: Taxa de perda de células do mecanismo de anúncio de capacidades *on-the-fly* para os valores de H iguais a 0,5 e 0,85.

acaba produzindo um grande atraso nos demais tráfegos transmitidos na rede.

No caso do mecanismo de reservas, a situação se inverte, já que o mecanismo transmite o tráfego principal somente na taxa alocada ao canal, o que produz maiores filas no gateway origem. Por outro lado, o mecanismo protege os outros tipos de tráfego transmitidos na rede, provocando atrasos desprezíveis mesmo para um número elevado de caminhos virtuais entre gateways.

O mecanismo de anúncio de capacidade híbrido apresentou atrasos no tráfego principal semelhantes aos provocados pelo mecanismo *on-the-fly* para uma carga baixa na rede. Ao passo em que a carga aumenta, ou seja, aumenta o número de caminhos virtuais, o mecanismo híbrido se distancia do mecanismo *on-the-fly*, mas permanece ainda bem distante do mecanismo de reservas. No caso do tráfego de retaguarda, a situação se inverte, com o aumento da carga, o mecanismo híbrido tende a provocar atrasos maiores no tráfego de retaguarda aproximando-se do mecanismo *on-the-fly*.

As figuras 6.15, 6.16 e 6.17 apresentam os atrasos provocados pelos dois tipos de tráfego ($H = 0,50$ e $H = 0,85$) no tráfego principal em cada um dos mecanismos. Para os mecanismos híbrido e *on-the-fly* os atrasos provocados pelo tráfego auto-semelhante (H

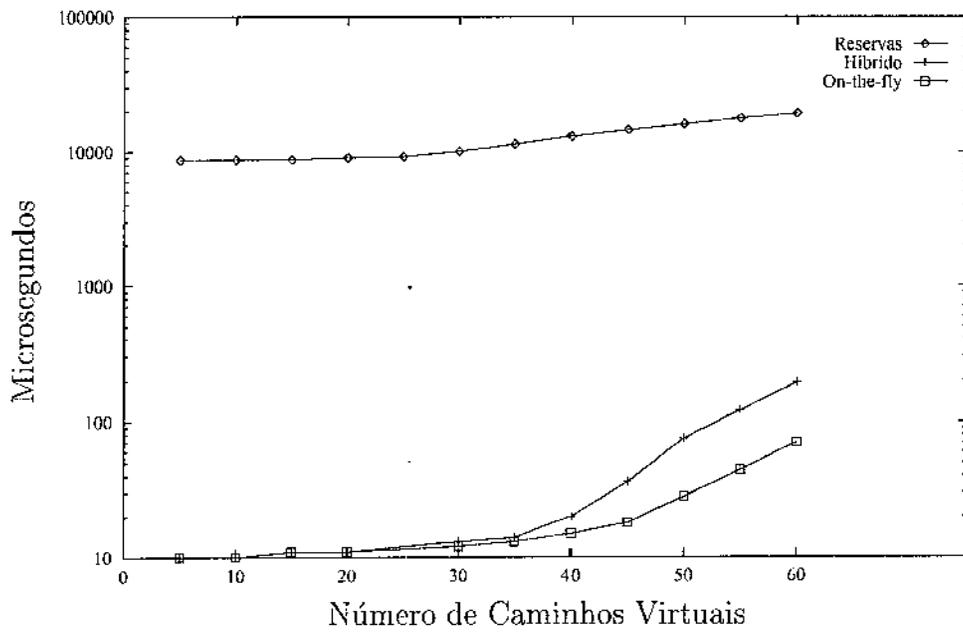


Figura 6.13: Atraso médio enfrentado pelo tráfego principal em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,85.

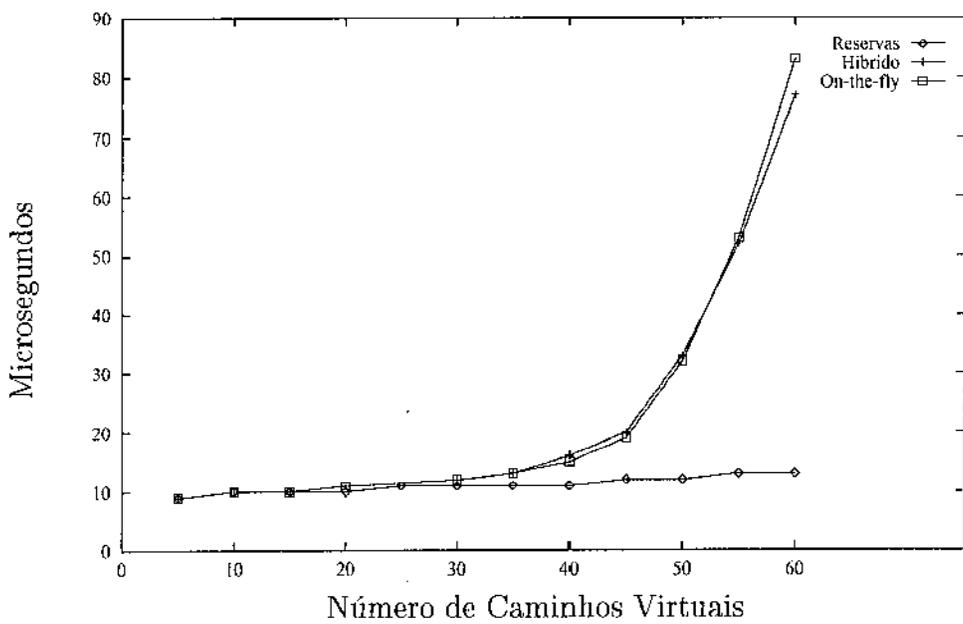


Figura 6.14: Atraso médio enfrentado pelo tráfego de retaguarda em função do número de caminhos virtuais transportando tráfego principal com parâmetro de Hurst igual a 0,85.

= 0,85) foi sempre maior que os atrasos provocados pelo tráfego sem características de auto-similaridade ($H = 0,50$). Já no caso do mecanismo de reservas, a situação se inverteu. Como explicado anteriormente, o mecanismo de média móvel se adaptou melhor ao tráfego auto-similar e conseguiu atualizar de maneira mais eficiente as capacidade alocadas para o tráfego auto-similar.

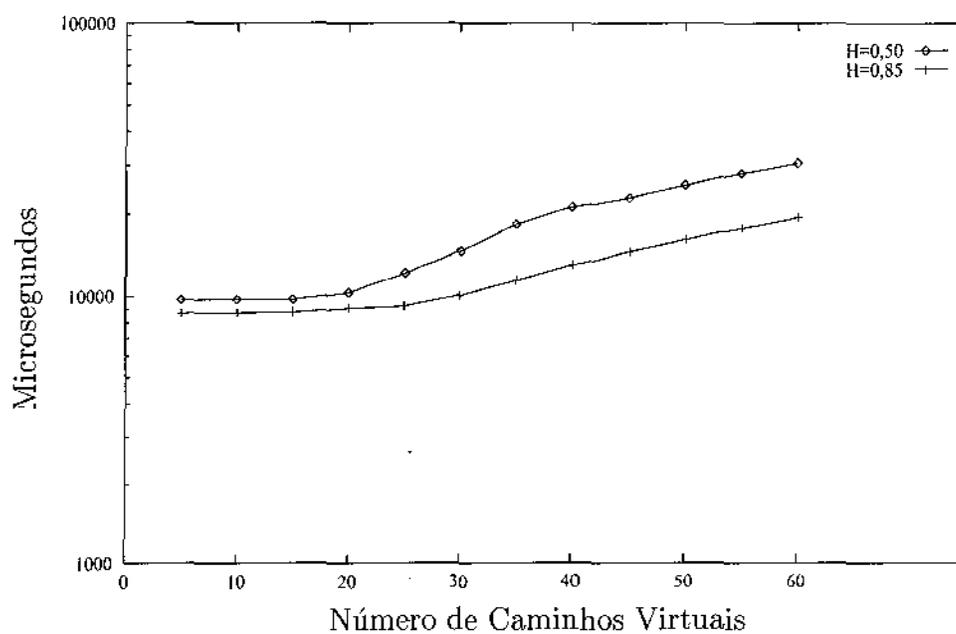


Figura 6.15: Atraso médio enfrentado pelo tráfego principal no mecanismo de reservas para os valores de H iguais a 0,5 e 0,85.

Por último, as figuras 6.18 e 6.19 apresentam os atrasos provocados pelos dois tipos de tráfego ($H = 0,50$ e $H = 0,85$) no tráfego de retaguarda no mecanismos híbrido e *on-the-fly*. Nos dois casos, os atrasos foram ligeiramente maior para o tráfego principal com $H = 0,85$. No caso do mecanismos de reservas, não houve diferença entre os atrasos e portanto os valores podem ser observados diretamente da figura 6.14.

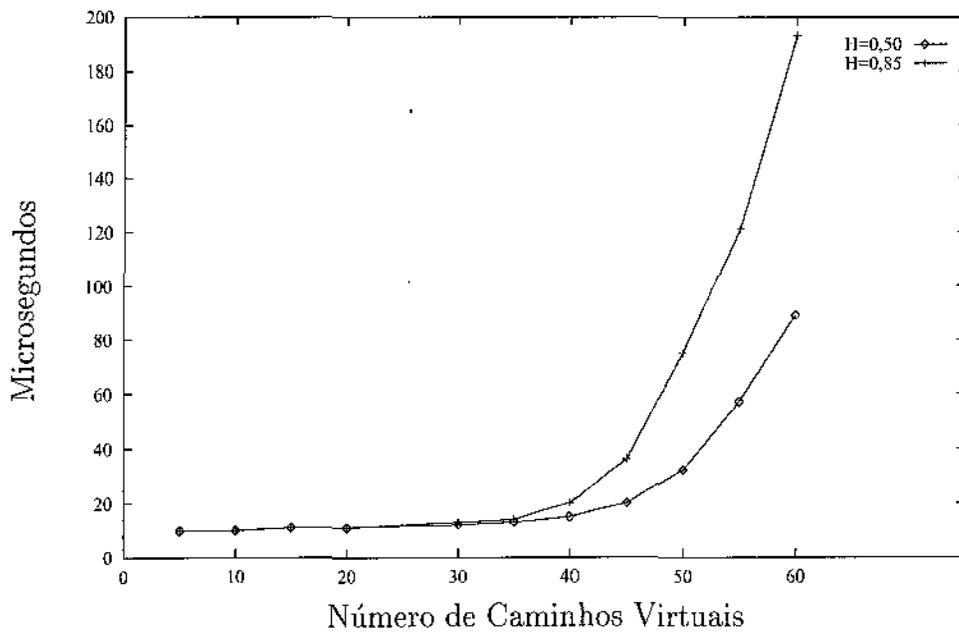


Figura 6.16: Atraso médio enfrentado pelo tráfego principal no mecanismo de anúncio de capacidades híbrido para os valores de H iguais a 0,5 e 0,85.

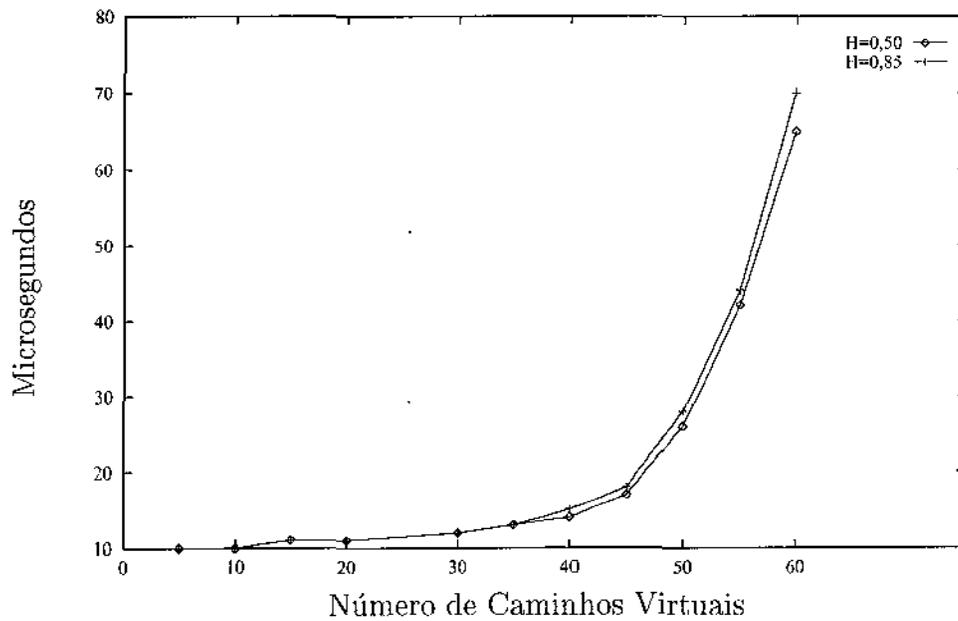


Figura 6.17: Atraso médio enfrentado pelo tráfego principal no mecanismo de anúncio de capacidades *On-the-fly* para os valores de H iguais a 0,5 e 0,85.

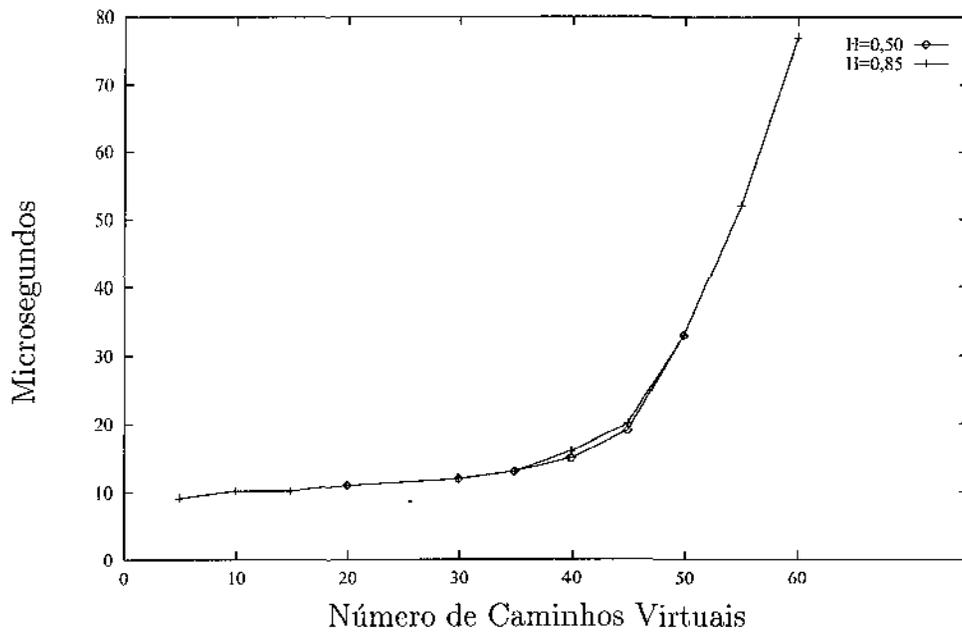


Figura 6.18: Atraso médio enfrentado pelo tráfego de retaguarda no mecanismo de anúncio de capacidades híbrido para os valores de H iguais a 0,5 e 0,85.

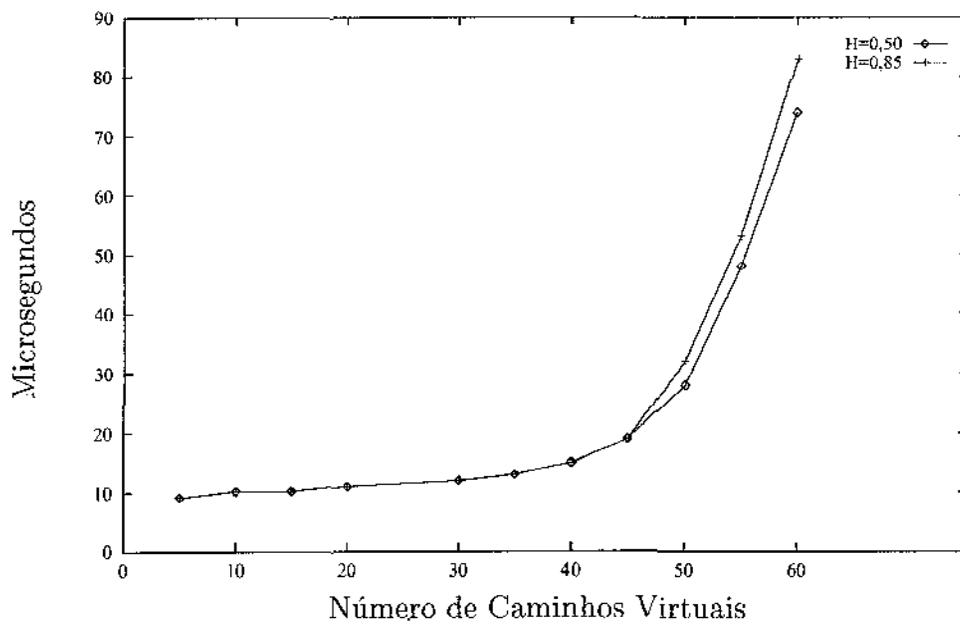


Figura 6.19: Atraso médio enfrentado pelo tráfego de retaguarda no mecanismo de anúncio de capacidades *On-the-fly* para os valores de H iguais a 0,5 e 0,85.

Capítulo 7

Conclusões e Trabalhos Futuros

Neste último capítulo são apresentados um resumo da dissertação, as principais contribuições (Seção 7.1) e também algumas sugestões para trabalhos futuros (Seção 7.2).

7.1 Resumo e Contribuições

Além da análise dos mecanismos de gerenciamento de capacidade, este trabalho apresentou um estudo amplo sobre controle de tráfego e de congestionamento em redes ATM no capítulo 2 e sobre o modelo de tráfego auto-semelhante, sendo que o capítulo 4 representa um bom ponto de partida para aqueles que desejam estudar esse modelo. No capítulo 6 foram apresentados alguns resultados importantes sobre a influência do parâmetro de auto-semelhança na taxa de perda de células e no atraso médio enfrentado por células em um sistema simples com uma fila e um servidor.

No estudo realizado com os mecanismos de gerenciamento de capacidade, o tráfego auto-semelhante se mostrou mais prejudicial à rede em dois dos três mecanismos estudados, mas se mostrou bem menos prejudicial do que se esperava pelos experimentos preliminares. Este comportamento pode ser explicado pela alteração imposta pelos mecanismos de gerenciamento de capacidade ao tráfego que entra na rede. Todos os mecanismos, um em maior escala e os outros em menor escala, alteram a cadeia de dados que é enviada para dentro da rede. Os dados são armazenados temporariamente no gateway e transmitidos em seguida a uma taxa diferente da taxa original. Com isso, não se pode garantir que o tráfego que chega ao comutador ATM dentro da rede ainda possui as mesmas características de auto-semelhança que o tráfego de entrada nos gateways.

Como contribuição adicional, neste trabalho foi desenvolvido um simulador genérico e altamente parametrizável que pode ser utilizado em estudos de outros aspectos de redes ATM diferentes daqueles aqui estudados.

7.2 Extensões e Trabalhos Futuros

Algumas formas possíveis pelas quais o presente trabalho pode ser estendido estão descritas a seguir.

- As simulações aqui realizadas foram todas conduzidas no nível de célula e conseqüentemente exigiram tempos elevados de simulação. Novas simulações podem ser realizadas considerando unidades maiores (rajadas ou pacotes) ou técnicas para redução do tempo de simulação, como por exemplo, *Importance Sampling* [FM94] e Teoria do Valor Extremo [Ber90].
- Os geradores de tráfego auto-semelhante utilizados podem ser substituídos por novas técnicas que simulam os dados por fontes individuais e realizam a agregação de várias fontes. Neste tipo de simulação é possível identificar as perdas em cada uma das fontes e conseqüentemente a identificação de perda de unidades maiores, como por exemplo pacotes ou rajadas. As fontes individuais são normalmente caracterizadas por distribuições *heavy tail* [WTSW97]. Estudos recentes mostram que a agregação de um número elevado de fontes desse tipo resulta em tráfego agregado com características auto-semelhantes [WTSW97, TWS97].
- As simulações podem ser repetidas, dessa vez usando *traces* reais, objetivando com isso não só avaliar o desempenho dos mecanismos estudados com esses *traces*, como também verificar em que medida, os *traces* sintéticos, na forma como são gerados atualmente no simulador, são capazes de substituir satisfatoriamente os *traces* reais.
- Um modelo analítico poderia ser desenvolvido para o sistema simulado a fim de comparar os resultados indicados pelo modelo com os resultados obtidos por simulação.
- O simulador desenvolvido pode ser estendido de várias formas para se tornar uma ferramenta de maior utilidade para simulações de novas características em redes ATM. Características adicionais, como por exemplo, novas classes de tráfego (ABR,

UBR), esquemas para admissão de conexões e políticas para atendimento de buffers, podem ser incluídas no simulador.

Bibliografia

- [BC89] R. Ballart and Y. Ching. SONET: Now It's the Standard Optical Network. *IEEE Communications Magazine*, 27(3):8–15, March 1989.
- [BD91] J. Burgin and D. Dorman. Broadband ISDN Resource Management: The Role of Virtual Paths. *IEEE Communications Magazine*, pp. 44–48, September 1991.
- [Ber90] I. Berberana. Application of Extreme Value Theory to the Analysis of a Network Simulation. In *Proceedings of 23rd Annual Simulation Symposium*, pp. 105–121, Nashville, USA, 1990.
- [Ber94] J. Beran. *Statistic for Long-Memory Processes*. Chapman and Hall, New York, 1994.
- [BG92] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall International Inc., 1992.
- [BM92] R. Beraldi and S. Marano. Limiting Removal Depth in the Pushout Scheme for ATM Networks. In *Proc. ICC '92*, pp. 997–1003, Chicago, USA, 1992.
- [Bou92] J-Y Le Boudec. The Asynchronous Transfer Mode: A Tutorial. *Computer Networks and ISDN Systems*, 24:279–309, 1992.
- [Boy90] P. E. Boyer. A Congestion Control for the ATM. In *Proc. ITC Workshop*, Morristown, USA, 1990.
- [BS91] J. J. Bae and T. Suda. Survey of traffic control schemes and protocols in ATM networks. *Proceedings of the IEEE*, 79(2):170–189, February 1991.

- [BSTW95] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Transactions on Communications*, 43(2/3/4):1566–1579, February/March/April 1995.
- [BT92] P. E. Boyer and D. P. Tranchier. A Reservation Principle with Applications to the ATM Traffic Control. *Computer Networks and ISDN Systems*, 24:321–334, 1992.
- [CFT92] P. Castelli, A. Forcina, and A. Tonietti. Dimensioning Criteria for Policing Functions in ATM Networks. In *Proc. INFOCOM '92*, pp. 759–766, Florence, Italy, 1992.
- [Cox84] D. R. Cox. Long-Range Dependence. in: *Statistics: An Appraisal*, pp. 55–74, 1984.
- [FFC92] A. Fournier, D. Fussel, and L. Carpenter. Computer Rendering of Stochastic Models. *Communications of ACM*, 25(6):371–384, June 1992.
- [FL91] H. J. Fowler and W. E. Leland. Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management. *IEEE Journal on Selected Areas in Communications*, 9(7):1139–1149, September 1991.
- [FM92] V. S. Frost and M. T. Mullen. Dynamic Bandwidth Allocation for B-ISDN Based on End-to-End Delay Estimates. In *Proc. ICC '92*, pp. 225–231, Chicago, USA, 1992.
- [FM94] V. S. Frost and B. Melamed. Traffic Modeling for Telecommunications Networks. *IEEE Communications Magazine*, 32(3):70–81, March 1994.
- [FMGV92] L. Fratta, L. Musumeci, G. Gallassi, and L. Verri. Congestion control strategies in ATM networks. *European Transactions on Telecommunications*, 3(2):183–193, March-April 1992.
- [For94] ATM Forum. *ATM User-Network Interface Specification – Version 3.1*. Prentice-Hall, September 1994.
- [For96] ATM Forum. *ATM Forum Traffic Management Specification – Version 4.0*. Public Available Document, April 1996.

- [Gar93] M. W. Garret. *Contributions Toward Real-Time Services on Packet Switched Networks*. PhD thesis, Columbia University, 1993.
- [GRF89] G. Gallassi, G. Rigolio, and L. Fratta. ATM: Bandwidth Assignment and Bandwidth Endorcement Policies. In *Proceedings of ACM GLOBECOM '89*, pp. 1788-1793, Dallas, TX, 1989.
- [GRV90] G. Gallassi, G. Rigolio, and L. Verri. Resource Management and Dimensioning in ATM Networks. *IEEE Network Magazine*, 11(3):8-17, May 1990.
- [GTG92] M. Gerla, T. C. Tai, and G. Gallassi. LAN/MAN Interconnection to ATM a Simulation Study. In *Proc. INFOCOM '92*, pp. 2270-2279, Florence, Italy, 1992.
- [GTG93] M. Gerla, T. C. Tai, and G. Gallassi. Internetting LAN's and MAN's to B-ISDN's for Connectionless Traffic Support. *IEEE Journal on Selected Areas in Communications*, 11(8):1145-1159, October 1993.
- [GTMG91] M. Gerla, T. C. Tai, J. A. S. Monteiro, and G. Gallassi. Interconnecting LANs and MANs to ATM. In *Proc. 16th Conference on Local Computer Networks*, pp. 259-270, Minneapolis, USA, 1991.
- [GW94] M. W. Garret and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Proceedings of ACM Sigcomm '94*, pp. 269-280, London, UK, 1994.
- [HB92] M. G. Hluchyj and A. Bhargava. Queueing Disciplines for Integrated Fast Packet Networks. In *Proc. ICC '92*, pp. 990-996, Chicago, USA, June 1992.
- [IT92] ITU-T. *Recommendation I.361: Asynchronous Transfer Mode*. Internation Telecommunication Union, 1992.
- [IT93a] ITU-T. *Recommendation I.211: B-ISDN Services Aspects*. Internation Telecommunication Union, March 1993.
- [IT93b] ITU-T. *Recommendation I.363: B-ISDN ATM Adaptation Layer (AAL) specification*. Internation Telecommunication Union, March 1993.

- [IT93c] ITU-T. *Recommendation I.371: Traffic Control and Congestion Control in B-ISDN*. International Telecommunication Union, March 1993.
- [IT95] ITU-T. *Recommendation I.364: Support of Broadband Connectionless Data Service on B-ISDN*. International Telecommunication Union, November 1995.
- [Jai90] R. Jain. Congestion Control in Computer Networks: Issues and Trends. *IEEE Network Magazine*, 11(3):24–30, May 1990.
- [Jai96] R. Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey. *Computer Networks and ISDN Systems*, November 1996.
- [Jr93] J. A. Frazao Jr. Estudo Comparativo da Eficiência de Mecanismos de Controle de Tráfego para a RDSI-FL. Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife, Brasil, 1993.
- [Kav95] N. Kavak. Data Communication in ATM Networks. *IEEE Network*, pp. 28–37, May/June 1995.
- [Kes96] G. Kesidis. *ATM Network Performance*. Kluwer Academic Publishers, 1996.
- [KMU90] H. Kasai, T. Murase, and H. Ueda. Synchronous Digital Transmission Systems Based on CCITT SDH Standard. *IEEE Communications Magazine*, 28(8):50–59, August 1990.
- [Kow83] T. Kowaltowski. *Implementação de Linguagens de Programação*. Editora Guanabara Dois S.A., Rio de Janeiro, RJ, 1983.
- [KVJ+98] S. Kalyanaraman, B. Vandalore, R. Jain, R. Goyal, and S. Fahmy. Performance of TCP over ABR with Long-Range Dependent VBR Background Traffic over Terrestrial and Satellite ATM networks. *Disponível em ftp://netlab.ohio-state.edu/pub/jain/papers/mpeg_lcn.ps*. *Submetido para o LCN 98*, , 1998.
- [Lav83] S. S. Lavenberg. *Computer Performance Modeling Handbook*. Academic Press, 1983.

- [LEWW95] W-C Lau, A. Erramilli, J. L. Wang, and W. Willinger. Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and Its Properties. In *Proceedings of IEEE ICC '95*, pp. 466–472, Seattle, USA, 1995.
- [LRG92] B. Laguë, C. Rosenberg, and F. Guillemin. A Generalization of Some Policing Mechanisms. In *Proc. INFOCOM '92*, pp. 767–775, Florence, Italy, 1992.
- [LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.
- [LW91] W. E. Leland and D. V. Wilson. High Time-Resolution Measurement and Analysis of LAN Traffic: Implications for LAN interconnection. In *Proceedings of the IEEE INFOCOM'91*, pp. 1360–1366, Bal Harbour, USA, 1991.
- [Man83] B. B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman, 1983.
- [MFT91] L. Mongiovi, M. Farrel, and V. Trecordi. A Proposal for Interconnecting FDDI Networks through B-ISDN. In *Proc. INFOCOM '91*, pp. 259–270, Miami, USA, 1991.
- [Min89] S. E. Minzer. Broadband ISDN and Asynchronous Transfer Mode (ATM). *IEEE Communications Magazine*, pp. 17–24, September 1989.
- [MN68] B. B. Mandelbrot and J. W. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437, October 1968.
- [Mon90] J. A. S. Monteiro. *Bandwidth Allocation in Broadband Integrated Services Digital Networks*. PhD thesis, University of California at Los Angeles, Los Angeles, USA, 1990.
- [Mon94] J. A. S. Monteiro. *Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL)*. IX Escola de Computação, Recife, PE, 1994.
- [MW69a] B. B. Mandelbrot and J. R. Wallis. Computer Experiments with Fractional Gaussian Noise. *Water Resources Research*, 5(1):228–267, February 1969.

- [MW69b] B. B. Mandelbrot and J. R. Wallis. Robustness of the Rescaled Range R/S in the Measurement of Noncyclic Long Run Statistical Dependence. *Water Resources Research*, 5(5):967-988, October 1969.
- [MW69c] B. B. Mandelbrot and J. R. Wallis. Some Long-Run Properties of Geophysical Records. *Water Resources Research*, 5(2):321-340, April 1969.
- [Nee87] F. Neelamkavil. *Computer Simulation and Modelling*. John Wiley and Sons, 1987.
- [Nor94] I. Norros. A Storage Model with Self-Similar Input. *Queueing Systems*, 16:387-396, 1994.
- [Par90] G. M. Parulkar. The Next Generation of Internetworking. *ACM SIGCOMM Communication Review*, 20(1):228-267, January 1990.
- [Par94] C. Partridge. *Gigabit Networking*. Addison Wesley Publishing Company, 1994.
- [Pax97] V. Paxson. Fast Approximation of Self-Similar Network Traffic. *Computer Communications Review*, 27(5):5-18, October 1997.
- [PF95] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226-244, June 1995.
- [PPS86] B. B. Pereira, M. B. Z. Pais, and P. R. H. Sales. *Análise Espectral de Séries Temporais*. Arte Final, 1986.
- [Pri81] M. B. Priestley. *Spectral Analysis and Times Series*. Academic Press, 1981.
- [PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 1992.
- [Ros86] H. L. Ross. FDDI - a tutorial. *IEEE Communications Magazine*, 24(5):10-17, May 1986.
- [SJ95] K-Y Siu and R. Jain. A Brief Overview of ATM: Protocol Layers, LAN Emulation, and Traffic Management. *Computer Communications Review*, 25(2):6-20, April 1995.

- [Soa90] L. F. G. Soares. *Modelagem e Simulação Discreta de Eventos*. VII Escola de Computação, São Paulo, 1990.
- [ST94] G. Samorodnitsky and M. S. Taqqu. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. Chapman and Hall, New York, 1994.
- [Tur86] J. S. Turner. New Directions in Communications (or Which Way to The Information Age?). *IEEE Communications Magazine*, 24(10):50–58, October 1986.
- [Tur92] J. S. Turner. Managing Bandwidth in ATM Networks with Bursty Traffic. *IEEE Network*, pp. 50–58, September 1992.
- [TWS97] M. S. Taqqu, W. Willinger, and R. Sherman. Proof of a Fundamental Result in Self-Similar Traffic Modeling. *Computer Communications Review*, 27(2):5–23, 1997.
- [WTE96] W. Willinger, M. S. Taqqu, and A. Erramilli. A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks. In *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary and I. Ziedins editors, Clarendon Press, Oxford, pp. 339–366, 1996.
- [WTSW97] W. Willinger, M. S. Taqqu, R. Sherman, and D. Wilson. Self-Similarity through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.

Apêndice A

Transformações Lineares e o Parâmetro de Hurst

Os dados gerados pelos algoritmos estudados no capítulo 4 possuem, normalmente, valores no intervalo $[-1,1]$. Para ser utilizado em simulações, os dados devem corresponder a medições de redes reais. Os dados obtidos através de medições em redes são sempre positivos e correspondem ao número de bytes ou número de pacotes ou células por intervalo amostral. É necessário, portanto, efetuar algum tipo de transformação nos dados gerados pelos algoritmos para que eles possam representar *traces* do mundo real. Algumas transformações foram discutidas no capítulo 5. A transformação normalmente utilizada é a transformação linear. A seguir será demonstrado que esse tipo de transformação não altera o grau de auto-semelhança, parâmetro H , dos dados gerados.

Considere a seqüência $(x_t : t = 1, 2, \dots, n)$ correspondendo aos dados gerados por um dos algoritmos estudados. A média da seqüência é:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

e a variância amostral é:

$$\sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

A estatística R/S para a seqüência x é dada por:

$$\frac{R_x}{S_x} = \frac{1}{S_x} [\max(0, W_{x1}, W_{x2}, \dots, W_{xn}) - \min(0, W_{x1}, W_{x2}, \dots, W_{xn})]$$

em que

$$W_{xl} = (x_1 + x_2 + \cdots + x_l) - l\bar{x}(n)$$

com $l = 1, 2, \dots, n$.

O efeito de Hurst é descrito por:

$$E \left[\frac{R}{S} \right] \sim cn^H$$

quando $n \rightarrow \infty$, e H é conhecido como parâmetro de Hurst.

Uma transformação linear é descrita pela equação:

$$u_i = kx_i + c$$

com $i = 1, 2, \dots, n$, e k e c são constantes e, para os propósitos aqui esperados, $k > 0$. u_i é a nova seqüência.

A média da nova seqüência é dada por:

$$\begin{aligned} \bar{u} &= \frac{1}{n} \sum_{i=1}^n (kx_i + c) \\ &= k\bar{x} + c \end{aligned}$$

e a variância amostral por:

$$\begin{aligned} \sigma_u^2 &= \frac{1}{n-1} \sum_{i=1}^n (u_i - \bar{u})^2 \\ &= \frac{1}{n-1} \sum_{i=1}^n [(kx_i + c) - (k\bar{x} + c)]^2 \\ &= k^2 \left[\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right] \\ &= k^2 \sigma_x^2 \end{aligned}$$

e

$$\begin{aligned} W_{ul} &= (u_1 + u_2 + \cdots + u_l) - l\bar{u}(n) \\ &= [k(x_1 + x_2 + \cdots + x_l) + lc] - l(k\bar{x}(n) + c) \\ &= k[(x_1 + x_2 + \cdots + x_l) - l\bar{x}(n)] + lc - lc \\ &= kW_{xl} \end{aligned}$$

para $l = 1, 2, \dots, n$.

Seja

$$Max_u = \max(0, W_{u1}, W_{u2}, \dots, W_{un})$$

$$Max_x = \max(0, W_{x1}, W_{x2}, \dots, W_{xn})$$

e

$$Min_u = \min(0, W_{u1}, W_{u2}, \dots, W_{un})$$

$$Min_x = \min(0, W_{x1}, W_{x2}, \dots, W_{xn})$$

Assuma que

$$Max_u = W_{ui}$$

e

$$Min_u = W_{uj}$$

Como $W_{ui} = kW_{xi}$ e $k > 0$, portanto,

$$Max_u = W_{ui} = kW_{xi} = kMax_x$$

$$Min_u = W_{uj} = kW_{xj} = KMin_x$$

Considerando todos os casos acima, tem-se,

$$\begin{aligned} \frac{R_u}{S_u} &= \frac{1}{S_u} [\max(0, W_{u1}, W_{u2}, \dots, W_{un}) - \min(0, W_{u1}, W_{u2}, \dots, W_{un})] \\ &= \frac{1}{kS_x} [k \max(0, W_{x1}, W_{x2}, \dots, W_{xn}) - k \min(0, W_{x1}, W_{x2}, \dots, W_{xn})] \\ &= \frac{R_x}{S_x} \end{aligned}$$

Como transformações lineares não alteram o valor da estatística R/S , portanto, elas não alteram o parâmetro de Hurst.

Apêndice B

Dependência em Longos Intervalos do FGN

Deseja-se mostrar que:

$$\lim_{\tau \rightarrow \infty} \tau^{2-2H} \rho(\tau) = H(2H - 1)$$

com $\rho(\tau) = \frac{1}{2}[(\tau + 1)^{2H} - 2\tau^{2H} + (\tau - 1)^{2H}]$ e $\tau \geq 0$.

Considere

$$t = \frac{\tau - 1}{\tau}$$

Isso significa que $\tau = \frac{1}{1-t}$ e quando $\tau \rightarrow \infty$, $t \rightarrow 1$.

$$\begin{aligned} \lim_{\tau \rightarrow \infty} \tau^{2-2H} \rho(\tau) &= \lim_{t \rightarrow 1} \left(\frac{1}{1-t} \right)^{2-2H} \rho \left(\frac{1}{1-t} \right) \\ &= \lim_{t \rightarrow 1} \left(\frac{1}{1-t} \right)^{2-2H} \frac{1}{2} \left[\left(\frac{2-t}{1-t} \right)^{2H} - 2 \left(\frac{1}{1-t} \right)^{2H} + \left(\frac{t}{1-t} \right)^{2H} \right] \\ &= \lim_{t \rightarrow 1} \frac{1}{2} \left(\frac{1}{1-t} \right)^2 \left[(2-t)^{2H} - 2 + t^{2H} \right] \\ &= \lim_{t \rightarrow 1} \frac{(2-t)^{2H} - 2 + t^{2H}}{2(1-t)^2} \end{aligned}$$

Derivando numerador e denominador, tem-se:

$$\lim_{\tau \rightarrow \infty} \tau^{2-2H} \rho(\tau) = \lim_{t \rightarrow 1} \frac{-2H(2-t)^{2H-1} + 2Ht^{2H-1}}{-4(1-t)}$$

Derivando novamente numerador e denominador, tem-se:

$$\begin{aligned}\lim_{\tau \rightarrow \infty} \tau^{2-2H} \rho(\tau) &= \lim_{t \rightarrow 1} \frac{2H(2H-1)(2-t)^{2H-2} + 2H(2H-1)t^{2H-2}}{4} \\ &= \frac{2H(2H-1) + 2H(2H-1)}{4} \\ &= H(2H-1)\end{aligned}$$

Apêndice C

Linguagem para Entrada de Dados do Simulador

Como mencionado no capítulo 5, foi desenvolvida uma linguagem e um interpretador para a entrada de dados do simulador. Na seção a seguir será dada a especificação formal da gramática da linguagem. A notação adotada segue as convenções adotadas em [Kow83]. A principal diferença em relação à especificação tradicional BNF consiste na inclusão de uma construção iterativa utilizada em substituição a construção recursiva. A ocorrência da construção $\{\alpha\}$ numa produção denotará a repetição da cadeia α zero ou mais vezes, ou seja, um elemento de α^* , em que $*$ denota o fecho de Kleene. Além dessa, a notação $[\alpha]$ será equivalente a $\alpha|\lambda$, em que λ representa a cadeia vazia, ou seja, indicará que a cadeia α é opcional.

C.1 Especificação Formal

```
file ::= [global_section] {node_section | link_section}
```

```
global_section ::= SIMULATION_PARAMETERS LC {global_parameters} RC
```

```
global_parameters ::= LOG_FILE = STRING ;  
                    | END_SIMULATION = NUMBER ;  
                    | PRINT_INTERVAL = NUMBER ;  
                    | PLOT = (TRUE | FALSE) ;
```

```
link_section ::= LINKS LC {default_link | link} RC
```

default_link ::= DEFAULT LC link_parameters RC

link ::= LC INTC -> INTC; {link_parameters} RC

link_parameters ::= BANDWIDTH = NUMBER ;
 | LATENCY = NUMBER ;

node_section ::= NODE INTC LC {node_parameters} RC

node_parameters ::= NAME = STRING ;
 | PLOT = (TRUE | FALSE) ;
 | node_function

node_function ::= FUNCTION = functions

functions ::= cbr_function
 | vbr_function
 | switch_function
 | gateway_function
 | sink_function

cbr_function ::= CBR LC {cbr_parameters} RC

cbr_parameters ::= CONNECTION_TYPE = (PERMANENT | SWITCHED) ;
 | SOURCE_TYPE = (CO | CL) ;
 | DESTINATION_NODE = INTC ;
 | PEAK_RATE = NUMBER ;
 | START_TIME = NUMBER ;
 | NUMBER_OF_BYTES = NUMBER ;
 | DURATION_TIME = NUMBER ;

vbr_function ::= VBR LC {vbr_parameters} RC

```
vbr_parameters ::= CONNECTION_TYPE = (PERMANENT | SWITCHED) ;  
                | SOURCE_TYPE = (CO | CL) ;  
                | DESTINATION_NODE = INTC ;  
                | START_TIME = NUMBER ;  
                | NUMBER_OF_BYTES = NUMBER ;  
                | DURATION_TIME = NUMBER ;  
                | vbr_type ;
```

```
vbr_type ::= poisson_source  
          | agregated_poisson_source  
          | fft_source  
          | rmd_source  
          | trace_source  
          | agregated_source
```

```
poisson_source ::= POISSON LC {poisson_parameters} RC
```

```
poisson_parameters ::= PEAK_RATE = NUMBER ;  
                    | MEAN_BURST_LENGTH = NUMBER ;  
                    | BURSTINESS = NUMBER ;  
                    | ON_SEED = INTC ;  
                    | OFF_SEED = INTC ;
```

```
agregated_poisson_source ::= AGREGATED_POISSON LC {ag_poisson_parameters} RC
```

```
ag_poisson_parameters ::= PEAK_RATE = NUMBER ;  
                       | MEAN_BURST_LENGTH = NUMBER ;  
                       | BURSTINESS = NUMBER ;  
                       | ON_SEED = INTC ;  
                       | OFF_SEED = INTC ;  
                       | TIME_INTERVAL = NUMBER ;  
                       | NUMBER_OF_FONTS = NUMBER ;
```

```
fft_source ::= FFT LC {fft_parameters} RC
```

```
fft_parameters ::= BYTE_AVERAGE = NUMBER ;
                | BYTE_VARIANCE = NUMBER ;
                | TIME_INTERVAL = NUMBER ;
                | HURST = NUMBER ;
                | SEED = INTC ;
                | TRANSFORMATION = (INT | EXPONENTIAL) ;

rind_source ::= RMD LC {fft_parameters} RC

trace_source ::= TRACE LC TRACE_FILE = STRING ; RC

agregated_source ::= AG_TRACE LC {agregated_parameters} RC

agregated_parameters ::= TRACE_FILE = STRING ;
                       | TIME_INTERVAL = NUMBER ;

switch_function ::= ATM_SWITCH LC {switch_parameters} RC

switch_parameters ::= OUTPUT_QUEUE_SIZE = INTC ;
                   | BUFFER_MANAGEMENT = buffer_mgmt_policies ;
                   | QUEUE_THRESHOLD = INTC ;
                   | DELAY = NUMBER ;
                   | POLICING = (YES | NO) ;

buffer_mgmt_policies ::= PUSHOUT
                      | L_PUSHOUT
                      | THRESHOLD

gateway_function ::= GATEWAY LC {gateway_parameters} RC

gateway_parameters ::= BANDWIDTH_MANAGEMENT = bw_mgmt_function
                   | DELAY = NUMBER ;
                   | LEASED_BANDWIDTH = NUMBER ;
                   | BUFFER_MANAGEMENT = buffer_mgmt_policies ;
                   | DESTINATION_NODE = INTC {, INTC};
                   | TOKEN_POOL_SIZE = INTC ;
```

```
    | OUTPUT_QUEUE_SIZE = INTC ;
    | QUEUE_THRESHOLD = INTC ;
    | BUFFER_SIZE = INTC ;

bw_mgmt_function ::= reservation_function
    | ba_on_the_fly_function
    | ba_hybrid_function
    | reservation2_function

reservation_function ::= RESERVATION LC {reservation_parameters} RC

reservation_parameters ::= NUMBER_OF_STEPS = INTC ;
    | INCREMENT_STEP = NUMBER ;
    | MAXIMUM_INCOMING_LOAD = NUMBER ;
    | T1 = NUMBER ;
    | ESTIMATED_TNA ;
    | BANDWIDTH_RELEASE = bw_release_function

bw_release_function ::= HALF_INCREMENT ;
    | moving_average_function

moving_average_function ::= MOVING_AVERAGE LC {moving_average_parameters} RC

moving_average_parameters ::= WINDOW_SIZE = NUMBER ;
    | NUMBER_OF_SAMPLES = INTC ;

ba_on_the_fly_function ::= BA_ON_THE_FLY LC PROBE_INTERVAL = NUMBER ; RC

ba_hybrid_function ::= BA_HYBRID LC {ba_hybrid_parameters} RC

ba_hybrid_parameters ::= PROBE_INTERVAL = NUMBER ;
    | CRIT_A = INTC ;
    | CRIT_D = INTC ;
    | ESTIMATED_TNA = NUMBER ;
    | ALLOCATION_THRESHOLD = NUMBER ;
    | DEALLOCATION_THRESHOLD = NUMBER ;
    | moving_average_function
```

reservation2_function ::= RESERVATION2 LC {reservation2_parameters} RC

reservation2_parameters ::= CRIT_A = INTC ;
 | CRIT_D = INTC ;
 | ESTIMATED_TNA = NUMBER ;
 | moving_average_function

sink_function ::= SINK ;

NUMBER ::= digit {.digit}

INTC ::= digit {digit}

digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

LC ::= '{'

RC ::= '}'

STRING ::= "{character}"

character ::= qualquer caracter do conjunto ASCII exceto caracteres especiais.