

**Descarte Seletivo Sujeito a Processo com
Dependência de Longa Duração**

Marcelo de Jesus Ferreira

Dissertação de Mestrado

Descarte Seletivo Sujeito a Processo com Dependência de Longa Duração

Marcelo de Jesus Ferreira¹

Maio de 1998

Banca Examinadora:

- Nelson Luís Saldanha Fonseca (Orientador)
- José Augusto Suruagy Monteiro
Departamento de Informática - Universidade Federal de Pernambuco
- Edmundo Roberto Mauro Madeira
Instituto de Computação - Universidade Estadual de Campinas
- Ricardo de Oliveira Anido (Suplente)
Instituto de Computação - Universidade Estadual de Campinas

¹O autor é bacharel em Processamento de Dados pela Universidade Federal do Amazonas

Descarte Seletivo Sujeito a Processo com Dependência de Longa Duração

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Marcelo de Jesus Ferreira e aprovada pela
Banca Examinadora.

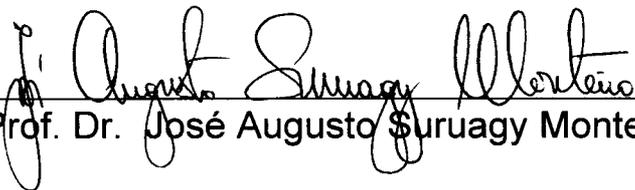
Campinas, 20 de Maio de 1998.

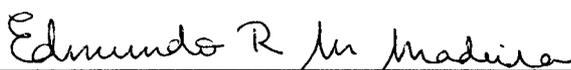


Nelson Luís Saldanha Fonseca (Orientador)

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

Tese de Mestrado defendida e aprovada em 20 de maio de 1998
pela Banca Examinadora composta pelos Professores Doutores


Prof. Dr. José Augusto Suruagy Monteiro


Prof. Dr. Edmundo Roberto Mauro Madeira


Prof. Dr. Nelson Luís Saldanha da Fonseca

© Marcelo de Jesus Ferreira. 1998.
Todos os direitos reservados.

Dedicatória

À mamãe e ao papai.

Agradecimentos

“Os três tesouros de um homem sábio devem ser Deus, a família e os amigos.”

Daniel de J. Ferreira.

Ao Dr. Nelson Fonseca pelos ensinamentos proporcionados ao longo dessa jornada, agradeço a oportunidade e o privilégio de tê-lo tido como orientador.

Ao Professor Altigran pela sua dedicação e competência que inspirou a tantos.

A todos os amigos que encontrei ao longo dessa caminhada, os momentos compartilhados tornaram essa jornada inesquecível. Comentar os podres do reino da Dinamarca e rir da fogueira das vaidades, são atitudes que nos fizeram mais integrados, questionadores e pensadores da realidade que nos rodeia, obrigado a todos. *“Sejamos como os loucos, vamos fazer uma grande fogueira à beira da praia e rir das vaidades do mundo”*.

À Sandra a quem sempre serei grato, pelo seu amor, pela sua paciência, pela sua compreensão. Seus telefonemas me iluminaram e entusiasmaram tantas vezes, parece que estava o tempo todo ao meu lado.

À minha família que é meu grande arcabouço emocional e espiritual.

Ao apoio financeiro do CNPq, CNPq/PROTEM-III ALMADEM, CAPES e ao apoio computacional do CENAPAD/SP.

Resumo

Esse trabalho investiga parcialmente a eficácia de descarte seletivo a nível de pacote e a nível de célula em um comutador ATM sob a influência de um processo com dependência de longa-duração. Estuda-se o comportamento de diversos mecanismos de descarte seletivo frente a diversos parâmetros de tráfego, como por exemplo, carga da rede, parâmetro de auto-semelhança, variância e diferentes tamanhos de pacotes, e também frente a diversos cenários de configurações, como por exemplo, diferentes tamanhos de *buffer*. Resultados apontam que é claramente vantajoso adotar a disciplina Compartilhamento Total (*Complete Sharing*), enquanto que a disciplina Compartilhamento Total com Garantia de Tamanho Mínimo de Fila (*Complete Sharing with Guaranteed Queue Minimum*) não é necessariamente vantajosa. Além disso, mostra-se que a escolha da política de descarte pode impactar significativamente a Qualidade de Serviço de uma aplicação. Resultados de uma fila com quatro níveis de prioridade mostram que é possível transportar a mesma carga de uma fila com dois níveis de prioridade utilizando *buffers* com tamanho menor. Introduce-se uma nova política de descarte de pacotes chamada Aceita-Maior-Pacote (AMP) que maximiza o *goodput* de célula e mostra-se que a política AMP fornece não só o maior *goodput* de célula mas também produz o maior *goodput* de pacote sob um processo com dependências de curta-duração composto de pacotes grandes.

Abstract

In this work we investigate the effectiveness of selective discard of cells and selective discard of packets in a multiplexer subject to a long-range dependent process. In this way, we evaluate diverse selective discard mechanism under different traffic parameters, such as the network load, the Hurst parameter, variance and different packet sizes, and also under distinct configurations scenarios, such as different buffer sizes. We found out that the Complete Sharing disciplines is clearly worth adopting, whereas Complete Sharing with Guaranteed Queue Minimum may be not. Furthermore, we show that the choice of push out policy may significantly impact the perceived QoS. We show that a four-class mechanism demands less buffer space than a two-class mechanism for the transport of the same load. We introduce a novel packet discarding policy called Longest-Packet-In (LPI) which maximizes the cell goodput and we show that LPI not only provides the highest cell goodput but also produces the highest packet goodput under a short-range dependent process composed of long packets.

Siglas

ABR	27
ALL	57
AMP	57
ARIMA	7
ATC	18
ATM	1
BF	21
B-ISDN	1
CAC	17
CBR	18
CDV	18
CDVT	18
CLR	18
CT	24
CTD	18
CTGTMF	24
DAP	58
DAPH	58
DAPHJ	59
DPP	58
F-ARIMA	7
H	5
IBT	18
JD	22
JS	21
JSG	21
LRD	6
mBf	8
MCR	18

MMPE	22
PCPD	26
PCPD-M	26
PCR	18
QoS	1
rGf	8
RAND	26
REM	19
SRC	18
SRD	6
STM	1
UBR	27
UCPD	26
VBR	4

Conteúdo

Dedicatória	v
Agradecimentos	vi
Resumo	vii
Abstract	viii
Siglas	ix
1 Introdução	1
2 Tráfego Auto-Semelhante	4
2.1 Definição	5
2.2 Propriedades	5
2.3 Movimento Browniano Fracional	7
2.4 Modelos de fonte <i>on/off</i> com caudas longas	9
2.5 Métodos para Estimativa do Parâmetro H	10
2.6 Gerador de Ruído Gaussiano	12
2.7 Fenômeno de Ineficácia de <i>Buffer</i>	14
3 Gerenciamento de Tráfego em Redes ATM	17
3.1 Controle de Admissão de Conexões	17
3.1.1 Descritores de Conexões	18
3.1.2 Abordagens de CAC	18
3.2 Policiamento de Tráfego	21
3.3 Mecanismos de Descarte Seletivo	23
3.4 Controle de Fluxo ABR	27
3.4.1 Alocação <i>Max-Min</i>	28

4	Descarte Seletivo de Células Sujeito a Processo LRD	30
4.1	Modelo de Simulação	31
4.2	Taxa de Perda por Classe de Prioridade	32
4.3	Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa	43
4.4	Múltiplas Classes	48
4.5	Resumo	52
5	Descarte Seletivo de Pacotes Sujeito a Processo LRD	57
5.1	Políticas de Descarte de Pacote	58
5.2	Política Aceita-Maior-Pacote	59
5.3	Resultados Numéricos	61
5.3.1	Fontes <i>on-off</i> com períodos geometricamente distribuídos	64
5.3.2	Fontes <i>on-off</i> com cauda longa	70
5.3.3	Fontes com distribuição Pareto nos períodos <i>on</i> e distribuição geométrica nos períodos <i>off</i>	78
5.4	Resumo	80
6	Conclusão	82
6.1	Principais Contribuições	83
6.2	Trabalhos Futuros	84
	Bibliografia	86

Lista de Tabelas

2.1	Estimativa do parâmetro H da amostra gerada.	14
4.1	Valores Máximos de Tamanho de Intervalos para fila CT, $H = 0.75$, $P(alta alta) = 0.7$, $\gamma = 0.0$, $\sigma^2 = 1$, $\rho = 0.8$	46
4.2	Valores Máximos de Tamanho de Intervalos para fila CT, $H = 0.8$, $P(baixa baixa) = 0.5$, Tamanho do <i>buffer</i> = 250, $\sigma^2 = 1$, $\rho = 0.8$	46
4.3	Valores Máximos de Tamanho de Intervalo de perdas para fila CT, $H=0.8$, $P(alta alta) = 0.7$, $\gamma = 0.0$, Tamanho do <i>Buffer</i> = 250, $\sigma^2 = 1$	48
4.4	Valores Máximos de Tamanho de Intervalo de perdas para fila CT, $H = 0.8$, $P(alta alta) = 0.7$, $\gamma = 0.0$, Tamanho do <i>buffer</i> = 400, $\rho = 0.8$	48

Lista de Figuras

2.1	Função de auto-correlação de um processo exatamente auto-semelhante de segunda-ordem com parâmetro Hurst $H = 1 - \beta/2$	5
2.2	Exemplos de estruturas de auto-correlação a) processo de dependência de curta-duração b) processo de dependência de longa duração.	6
2.3	Autocorrelação do Gerador de Ruído Gaussiano Fracional e dados de Bellcore.	13
2.4	O fenômeno de ineficácia de <i>buffer</i>	16
3.1	Exemplos de políticas de organização de <i>buffer</i>	25
3.2	Um exemplo de perda de célula devido a diferentes distribuições de tamanho de fila produzidas pelas políticas de descarte UCPD e PCPD.	27
4.1	Comutador ATM alimentado por um processo LRD.	31
4.2	Taxa de Perda x Tamanho do <i>buffer</i> para diferentes valores de Parâmetro de Hurst e Fila CT, $\rho = 0.8$, $\sigma^2 = 1$, $P(alta alta) = 0.7$, $\gamma = 0.0$	32
4.3	Taxa de Perda da Classe com Prioridade Alta x Tamanho do <i>buffer</i> para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção, $\rho = 0.8$, $\sigma^2 = 1$, $P(alta alta) = 0.7$, $\gamma = 0.0$	33
4.4	Taxa de Perda da Classe com Prioridade Baixa x Tamanho do <i>buffer</i> para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção, $\rho = 0.8$, $\sigma^2 = 1$, $P(alta alta) = 0.7$, $\gamma = 0.0$	34
4.5	Taxa de Perda x Variância para diferentes valores de Parâmetro de Hurst e Fila CT, Tamanho do <i>buffer</i> = 400, $\rho = 0.8$, $P(alta alta) = 0.7$, $\gamma = 0.0$	35
4.6	Taxa de Perda da Classe com Prioridade Alta x Variância para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção, Tamanho do <i>buffer</i> = 400, $\rho = 0.8$, $P(alta alta) = 0.7$, $\gamma = 0.0$	36
4.7	Taxa de Perda da Classe com Prioridade Baixa x Variância para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção, Tamanho do <i>buffer</i> = 400, $\rho = 0.8$, $P(alta alta) = 0.7$, $\gamma = 0.0$	37

4.8	Taxa de Perda da Classe com Prioridade Alta x Parâmetro de Hurst para diferentes valores de $P(alta alta)$, $Buffer = 250$, $\rho = 0.8$, $\sigma^2 = 1$, $P(baixa baixa) = 0.5$	38
4.9	Taxa de Perda da Classe com Prioridade Alta x Parâmetro de Hurst para diferentes valores de $P(baixa—baixa)$. $Buffer = 250$, $\rho = 0.8$, $\sigma^2 = 1$, $P(alta alta) = 0.8$	39
4.10	Taxa de Perda da Classe com Prioridade Alta x Tamanho do $buffer$ para diferentes Políticas de Descarte. $H = 0.75$, $\rho = 0.8$, $\sigma^2 = 1$, $P(alta alta) = 0.7$, $\gamma = 0.0$	41
4.11	Taxa de Perda da Classe com Prioridade Alta x Variância para diferentes Políticas de Descarte. $H = 0.75$, $\rho = 0.8$, Tamanho do $buffer=400$, $P(alta alta) = 0.7$, $\gamma = 0.0$	42
4.12	Taxa de Perda da Classe com Prioridade Alta x $P(alta—ata)$ para diferentes Políticas de Descarte. Tamanho do $buffer = 400$, $H = 0.8$, $\rho = 0.8$, $\sigma^2 = 1$, $P(baixa baixa) = 0.5$	42
4.13	Taxa de Perda da Classe com Prioridade Alta x Carga Fornecida para diferentes Políticas de Descarte. Tamanho do $buffer = 250$, $H = 0.75$, $\sigma^2 = 1$, $P(alta alta) = 0.7$, $\gamma = 0.0$	43
4.14	Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa x i-ésimo Intervalo para diferentes Políticas de Descarte, Fila CT. Tamanho do $buffer= 100$, $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.75$, $P(alta alta) = 0.7$, $\gamma = 0.0$	44
4.15	Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa x i-ésimo Intervalo para diferentes Políticas de Descarte, Tamanho do $buffer= 100$, $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.75$, $P(alta alta) = 0.7$, $\gamma = 0.0$	45
4.16	Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa x Carga Fornecida, Tamanho do $buffer = 400$, $\sigma^2 = 1$, $\rho = 0.8$, $P(alta alta) = 0.7$, $\gamma = 0.0$	47
4.17	Taxa de Perda por classe x Tamanho do $buffer$ para $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.85$, $P_1 = 0.4$, $P_2 = 0.1$, $P_3 = 0.2$ e $P_4 = 0.3$	49
4.18	Taxa de Perda por classe x Tamanho do $buffer$ para $\rho = 0.8$, $\sigma^2 = 1$, $P_1 = 0.60$, $P_2 = 0.07$, $P_3 = 0.13$ e $P_4 = 0.20$	50
4.19	Taxa de Perda por classe x Variância para $\rho = 0.8$, $H = 0.75$, Tamanho do $Buffer = 400$, $P_1 = 0.7$, $P_2 = 0.05$, $P_3 = 0.1$ e $P_4 = 0.15$	51
4.20	Taxa de Perda por classe x Tamanho do $buffer$ para diferentes políticas de descarte, $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.8$, $P_1 = 0.60$, $P_2 = 0.07$, $P_3 = 0.13$ e $P_4 = 0.20$	51

4.21	Tamanho do Intervalo de Perda x i-ésimo Intervalo para política PCPD-M, classes de prioridade 2, 3 e 4. $\rho = 0.8$, $\sigma^2 = 1$, Tamanho do <i>Buffer</i> = 400. $P_1 = 0.70$, $P_2 = 0.05$, $P_3 = 0.1$ e $P_4 = 0.15$	53
4.22	Tamanho do Intervalo de Perda x i-ésimo Intervalo para política PCPD-M, classes de prioridade 2, 3 e 4. $\rho = 0.8$, $\sigma^2 = 1$, Tamanho do <i>Buffer</i> = 400. $P_1 = 0.70$, $P_2 = 0.05$, $P_3 = 0.1$ e $P_4 = 0.15$	54
5.1	Exemplo da política AMP.	62
5.2	Modelo de Simulação.	63
5.3	<i>Goodput</i> de célula e pacote x Carga fornecida para tamanho médio de pacote de 42 células ATM, e Tamanho de <i>buffer</i> = 100.	65
5.4	<i>Goodput</i> de célula e pacote x Carga fornecida, para tamanho médio de pacote de 42 células ATM, e Tamanho de <i>buffer</i> = 400.	65
5.5	<i>Goodput</i> de célula e pacote x Tamanho médio do pacote, Tamanho do <i>buffer</i> = 100, Carga Fornecida = 0.8.	66
5.6	<i>Goodput</i> de célula e pacote x Tamanho médio do pacote, Tamanho do <i>buffer</i> = 400, Carga Fornecida = 0.8.	66
5.7	<i>Goodput</i> de célula e pacote x Proporção constante do Tamanho do <i>buffer</i> pelo Tamanho médio do pacote, Carga Fornecida = 0.8.	68
5.8	<i>Goodput</i> de célula e pacote x Proporção de células com tamanho médio de 170 células ATM em um rede com 32 fontes, pacote podem ter tamanho médio de 5, 21, 85 e 170 células ATM, tamanho do <i>buffer</i> = 400.	69
5.9	<i>Goodput</i> de célula e pacote x Proporção de células com tamanho médio de 5 células ATM em um rede com 32 fontes, pacote podem ter tamanho médio de 5, 21, 85 e 170 células ATM, tamanho do <i>buffer</i> = 400.	70
5.10	Análise Gráfica <i>variance time plot</i>	71
5.11	<i>Goodput</i> de célula x Carga fornecida, Tamanho médio de pacote de 42 células ATM, Tamanho do <i>buffer</i> = 400 e para diferentes valores de parâmetro de Hurst.	72
5.12	<i>Goodput</i> de pacote x Carga fornecida, Tamanho médio de pacote de 42 células ATM, Tamanho do <i>buffer</i> = 400 e para diferentes valores de parâmetro de Hurst.	74
5.13	<i>Goodput</i> de célula x Tamanho médio do pacote, Carga fornecida = 0.8, Tamanho do <i>buffer</i> = 400 e para diferentes valores de parâmetro de Hurst.	75
5.14	<i>Goodput</i> de pacote x Tamanho médio do pacote, Carga fornecida = 0.8, Tamanho do <i>buffer</i> = 400 e para diferentes valores de parâmetro de Hurst.	77

5.15	<i>Goodput</i> de célula x Proporção de células com tamanho médio de pacote de 170 células ATM para diferentes valores de parâmetro de Hurst em uma rede com 32 fontes, pacotes com tamanho médio de 5, 21, 85 e 170 células ATM.	79
5.16	<i>Goodput</i> de pacote x Proporção de células com tamanho médio de pacote de 170 células ATM para diferentes valores de parâmetro de Hurst em uma rede com 32 fontes, pacotes com tamanho médio de 5, 21, 85 e 170 células ATM.	80

Capítulo 1

Introdução

A integração de tráfego de vídeo, áudio e dados nas Redes Digitais de Serviços Integrados de Faixa Larga trouxe diversos desafios ao gerenciamento e dimensionamento destas redes. Se por um lado aplicações multimídia exigem das redes de comunicação, diferentes garantias de Qualidade de Serviço (QoS - *Quality of Service*), por outro lado diferentes tipos de tráfego possuem diferentes características estatísticas.

O Modo Assíncrono de Transferência (ATM - *Asynchronous Transfer Mode*), que foi adotado como padrão para a Rede Digital de Serviços Integrados de Banda-Larga (B-ISDN - *Broadband Integrated Digital Service Network*), foi escolhido ao invés de um sistema de reserva como o Modo Síncrono de Transferência (STM - *Synchronous Transfer Mode*) devido à sua eficiente utilização de banda-passante quando multiplexa fontes em rajada. Mecanismos de controle de congestionamento para redes ATM estão sendo investigados com o objetivo de garantir requisições de QoS do usuário. Entre esses mecanismos menciona-se: mecanismo de controle de admissão, mecanismo de policiamento e mecanismo de descarte seletivo.

O mecanismo de descarte seletivo descarta células em uma situação de transbordo de acordo com o nível de prioridade da célula. O nível de prioridade expressa a relevância da célula. Por exemplo, em uma codificação de vídeo, células com prioridade baixa são menos significativas para a qualidade da imagem. Além do descarte seletivo a nível de células, várias políticas de descarte de pacotes vêm sendo definidas a fim de preservar a integridade de pacotes a nível de camada de transporte. Para algumas aplicações se uma das células de um pacote é perdida, o pacote inteiro tem que ser retransmitido. Portanto, transmitir células de um pacote corrompido ocupa banda-passante, e pode conseqüentemente aumentar o congestionamento. Assim sendo, o mecanismo de descarte seletivo a nível de célula se preocupa com a utilização a nível de *link* de rede sem levar em consideração o pacote a nível de camada de transporte, enquanto que as políticas de descarte de pacotes levam em consideração o desempenho a nível de camada de transporte.

O entendimento do tráfego transportado em uma rede é de vital importância para a elaboração de modelos realistas. A negligência de características estatísticas importantes pode levar a conclusões errôneas e conseqüentemente a soluções inapropriadas. A adoção de uma representação específica para um certo tipo de tráfego deve ser suportada pela verificação experimental desta representação. Com o modelo de tráfego definido, pode-se prever o desempenho da rede. Como por exemplo, pode-se usar modelos de tráfego no dimensionamento de *buffers* e verificação de desempenho de mecanismos de controle da rede.

Em 1993, Leland et. al [LTWW93] demonstraram que o tráfego em redes *Ethernet* seguia um modelo com dependência de longa-duração, contrariando o que se pensava até então. Seus achados foram baseados em dados coletados ao longo de quatro anos nos laboratórios da Bellcore. Leland et. al mostraram que o tráfego *Ethernet* poderia ser representado pelos chamados modelos auto-semelhantes. Processos auto-semelhantes exibem o fenômeno de dependência de longa-duração. Isto é, esses processos possuem correlações que decaem muito lentamente (hiperbolicamente) ao contrário dos modelos comumente usados na literatura cujas correlações decaem rapidamente (exponencialmente). Em outras palavras, em processos com dependência de longa-duração, correlações distantes no tempo influenciam o padrão de chegada de pacotes. Uma outra característica dos processos auto-semelhantes é a semelhança do padrão de chegada de pacotes em diferentes escalas do tempo, o que se opõe à tradicional noção de *burstiness* de um processo. As conclusões deste estudo apontaram uma nova direção para as pesquisas na área de teletráfego.

Outras investigações posteriores demonstraram que o fluxo de vídeo [BSTW95] bem como o fluxo de certas aplicações em redes geograficamente distribuídas tais como *telnet* [PF95] podem também ser modelados através de processos auto-semelhantes com um alto grau de precisão. Se por um lado a constatação de padrões auto-semelhantes em diversos tipos de tráfego reforçam o uso destes modelos, por outro ainda não está claro qual é o impacto de se considerar dependências de longa-duração no controle de tráfego de redes de comunicação [MdV96]. Em outras palavras, dependências de longa-duração certamente produzem um padrão de ocupação de filas diferentes dos processos com dependência de curta-duração, porém não está óbvio nos dias de hoje qual é a influência destas correlações distantes em uma rede com multiplexadores que possuem espaço finito de *buffer* e onde aplicações possuem restrições de tempo real.

O objetivo desse trabalho é investigar a eficácia de descarte seletivo tanto a nível de pacote quanto a nível de célula sob a influência de processo com dependências de longa-duração. Assim sendo, investiga-se o comportamento de diversos mecanismos de descarte seletivo frente a diversos parâmetros de tráfego, como por exemplo, carga da rede, parâmetro de auto-semelhança, variância e diferentes tamanhos médios de pacote, e

também frente a diversos cenários de configurações, como por exemplo, diferentes tamanhos de *buffer*.

No estudo de descarte seletivo a nível de célula considera-se as disciplinas Compartilhamento Total (CT - *Complete Sharing*) e Compartilhamento Total com Garantia de Tamanho Mínimo de Fila (CTGTMF - *Complete Sharing with Guaranteed Queue Minimum*) e as principais políticas de descarte descritas na literatura. Analisa-se através de simulação o impacto das características do processo de entrada, tais como: carga do sistema, parâmetro de Hurst e variância, nas taxas de perda por classe e no tamanho do intervalo de perdas da classe de célula com prioridade baixa. Investiga-se inicialmente descarte seletivo com apenas duas classes de prioridade (baixa e alta) e em seguida investiga-se a influência do uso de múltiplas classes de prioridade.

No estudo de descarte seletivo a nível de pacote introduz-se uma nova política de descarte de pacote, Aceita-Maior-Pacote, e compara-se essa política com as principais políticas de descarte existentes na literatura. Analisa-se o compromisso entre o *goodput* de célula (a proporção de células boas transmitidos pelo número total de células que chegam na fila) e o *goodput* de pacote (a proporção de pacotes bons transmitidos pelo número total de pacotes que chegam na fila). Além disso mostra-se que a nova política de descarte de pacote pode aumentar significativamente o *goodput* de célula e também é capaz de produzir o maior *goodput* de pacote quando sujeita a um processo com dependência de curta-duração composto de pacotes grandes. Estende-se a análise também para redes com fontes que geram pacotes de tamanhos médios distintos.

Essa dissertação está organizada da seguinte maneira: no capítulo 2 são apresentados os principais conceitos de processos auto-semelhantes. No capítulo 3 são introduzidos os mecanismos de gerenciamento de tráfego em redes ATM. O capítulo 4 traz os principais resultados do estudo de descarte seletivo a nível de célula. No capítulo 5 apresenta-se resultados da avaliação das principais políticas de descarte de pacotes. Finalmente no capítulo 6 as conclusões são derivadas.

Capítulo 2

Tráfego Auto-Semelhante

Os processos estocásticos auto-semelhantes são invariantes em sua estrutura de correlação em diferentes escalas de tempo. Intuitivamente, olhando-se vários gráficos de um processo auto-semelhante em diferentes escalas de tempo eles irão parecer muito “semelhantes”. Esses processos são de grande interesse na física, economia, geofísica, turbulência e na modelagem de fenômenos aleatórios com dependência de longa-duração. Taqqu [TL86] fornece um guia bibliográfico comentado para muitas aplicações.

Atualmente os processos auto-semelhantes têm despertado especial interesse na modelagem de tráfego de redes. Isso porque diversos estudos evidenciam que esse tráfego tem o comportamento estatisticamente auto-semelhante (“fractal”). Entre esses estudos incluem-se uma análise de centenas de milhões de pacotes observados em várias redes locais *ethernet* nos laboratórios da Bellcore (*Bellcore Morristown Research and Engineering Center*) [LTWW93, WTSW95], uma análise de milhões de dados de serviços de vídeo VBR (*Variable-Bit-Rate*) [BSTW95], o estudo de certas aplicações de redes geograficamente distribuídas tais como *telnet*, *ftp* [PF95] e *www* [C'B95] e uma investigação detalhada de tráfego de dados coletados a nível de pacote de múltiplos núcleos de comutadores distribuídos pelo E.U.A. [KM94]. Esse comportamento “fractal” é muito diferente dos tráfegos de telefone convencional e dos modelos convencionais até então utilizados para tráfego de pacotes (*e.g.* Poisson puro ou modelos relacionados com Poisson tais como *batch* Poisson, Processos de Poisson Modulados por Markov [HL86], modelos de série de pacote [JR86], modelos de fluxo de fluido [AMS82]).

Os principais conceitos relacionados com processos auto-semelhantes são apresentados neste capítulo. A seção 2.1 define processos auto-semelhantes. Na seção 2.2 comentam-se algumas propriedades dos processos auto-semelhantes. Em seguida na seção 2.3 o movimento Browniano fracional (mBf) é introduzido. O fenômeno de auto-semelhança a nível de fonte de dados é abordado na seção 2.4. Na seção 2.5 discute-se vários métodos para se estimar o grau de auto-semelhança de um processo. E finalmente mostra-se ainda

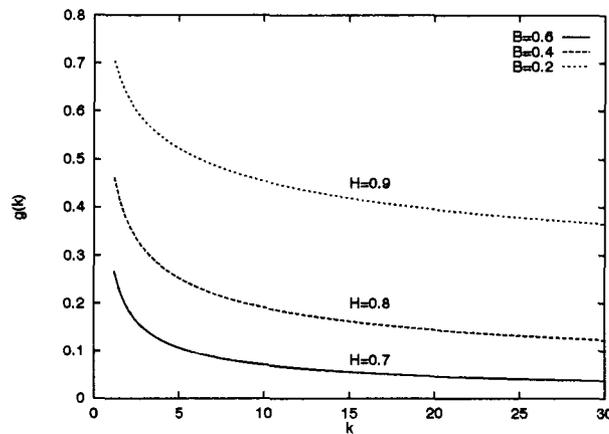


Figura 2.1: Função de auto-correlação de um processo exatamente auto-semelhante de segunda-ordem com parâmetro Hurst $H = 1 - \beta/2$.

um método para geração de dados auto-semelhantes baseado no mBf, na seção 2.6

2.1 Definição

Um processo X é chamado *exatamente auto-semelhante de segunda-ordem* com parâmetro $H = 1 - \beta/2$, $0 < \beta < 1$, se sua função de auto-correlação é dada por [TG97] [Cox84]:

$$r^{(m)}(k) = \frac{1}{2}[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}] \triangleq g(k), \quad 0 < \beta < 1, \quad k = 1, 2, 3, \dots \quad (2.1)$$

e X é chamado *assintoticamente auto-semelhante de segunda-ordem* com parâmetro $H = 1 - \beta/2$, $0 < \beta < 1$, se para todo $K = 1, 2, 3, \dots$

$$\lim_{m \rightarrow \infty} r^{(m)}(k) = \frac{1}{2}[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}] \triangleq g(k) \quad (2.2)$$

A Figura 2.1 mostra a função de auto-correlação $g(k)$ (equação (2.1)) como função de k para três valores de β . Os valores de $\beta = 0.2, 0.4$ e 0.6 representam conforme a equação $H = 1 - \beta/2$, respectivamente $H = 0.7, 0.8$ e 0.9 .

2.2 Propriedades

Matematicamente, auto-semelhança manifesta-se em um número de modos equivalentes:

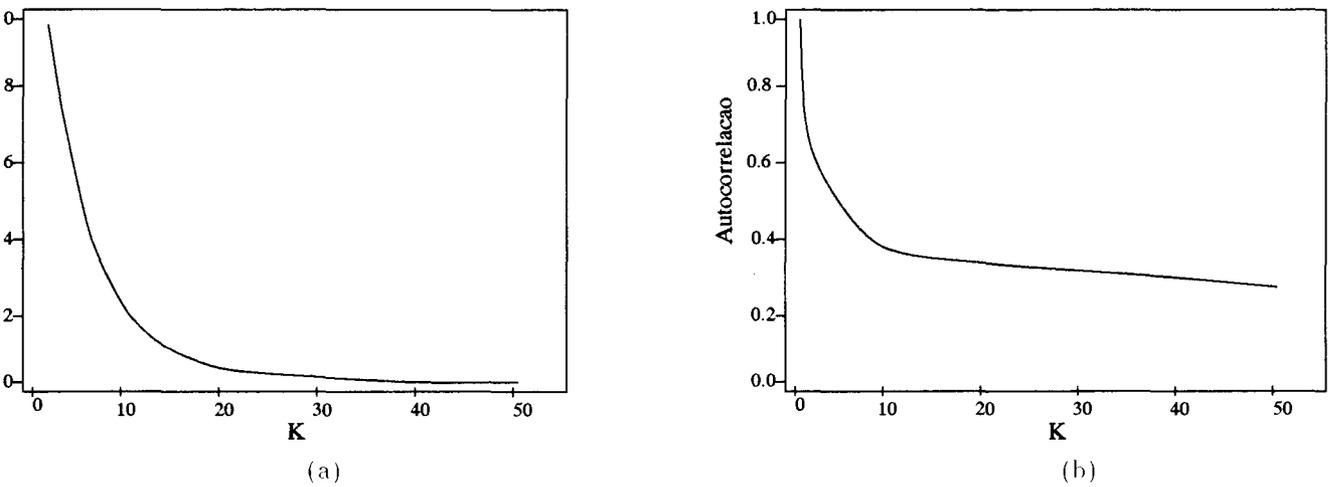


Figura 2.2: Exemplos de estruturas de auto-correlação a) processo de dependência de curta-duração b) processo de dependência de longa duração.

- (i) a variância da média amostral decresce mais lentamente do que a variância do tamanho da amostra (*variância decaindo lentamente*), *i.e.*, $var(X^{(m)}) \sim am^{-\beta}$, com $m \rightarrow \infty$, com $0 < \beta < 1$;
- (ii) as auto-correlações decrescem hiperbolicamente ao invés de exponencialmente, implicando em uma função de auto-correlação não totalizável $\sum_k r(k) = \infty$, dependência de longa-duração (LRD - *Long Range Dependence*), veja Figura 2.2b. e
- (iii) a densidade espectral (Transformada de Fourier) $f(\cdot)$ perto da origem comporta-se como $f(\lambda) \sim a\lambda^{-\gamma}$, para $\lambda \rightarrow 0$, $0 < \gamma < 1$ e $\gamma = 1 - \beta$.

Os processos auto-semelhantes se contrapõem a modelos de tráfego atualmente vigentes na literatura devido a estes últimos serem caracterizados por :

- (i) a variância da média amostral decresce com a variância do tamanho da amostra, *i.e.*, $var(X^{(m)}) \sim am^{-1}$, com $m \rightarrow \infty$;
- (ii) uma função de auto-correlação que decresce exponencialmente (*i.e.*, $r(k) \sim \rho^k$, $0 < \rho < 1$), implicando uma função de auto-correlação totalizável $\sum_k r(k) < \infty$, dependência de curta-duração (SRD - *Short Range Dependence*), veja Figura 2.2a. e
- (iii) a densidade espectral que é limitada na origem.

A Figura 2.2 compara o comportamento da auto-correlação de um processo com dependência de curta-duração (Figura 2.2a) com o comportamento de um processo com dependência de longa-duração (Figura 2.2b). Note que no caso de dependência de curta-duração tem-se valores expressivos de auto-correlação apenas para pequenos k ($k < 10$), enquanto para dependência de longa-duração continua-se a ter valores expressivos de auto-correlação mesmo para valores maiores de k (veja também Figura 2.1).

A característica mais notável dos processos auto-semelhantes de segunda-ordem é que seus processos agregados $X^{(m)}$ possuem uma estrutura de correlação não degenerativa, com $m \rightarrow \infty$. A existência de uma estrutura de correlação não degenerativa para os processos $X^{(m)}$, com $m \rightarrow \infty$, está em completo contraste com os modelos típicos de tráfego de pacotes até então considerados na literatura, os quais tem processos agregados $X^{(m)}$ que tendem para ruído puro de segunda-ordem, *i.e.*; para todo $k \geq 1$.

$$r^{(m)}(k) \rightarrow 0, \quad \text{com } m \rightarrow \infty, \quad \text{para } k \geq 1. \quad (2.3)$$

Uma das grandes vantagens dos processos auto-semelhantes é a parcimonidade do número de parâmetros para representar uma série temporal de centenas de milhares de observações. No caso de processos sem características de processos auto-semelhantes, existe a necessidade de se ter um grande número de parâmetros para modelar longas séries. Em contraste, a componente de longa-duração pode ser modelada com um único parâmetro quando se usa processos auto-semelhantes.

Dois processos estocásticos que produzem representações elegantes de fenômenos auto-semelhantes são: o ruído *Gaussiano fracional* (*fractional Gaussian noise*) e os *processos integrado fracional auto-regressivo de média móvel* (*fractional autoregressive integrated moving-average processes* - (F-ARIMA)). O ruído Gaussiano fracional $X = (X_k : k \geq 0)$ com parâmetro $H \in (0, 1)$ foi introduzido em [MN68], é um processo Gaussiano estacionário com média μ , variância σ^2 , e função de auto-correlação $r(k) = 1/2(|k+1|^{2H} - |k|^{2H} + |k-1|^{2H})$, $k > 0$. É possível mostrar que o ruído Gaussiano fracional é exatamente auto-semelhante de segunda-ordem com parâmetro de auto-semelhança H entre $1/2 < H < 1$. A série de ruídos Gaussianos fracionais $X_i, i > 1$ é o incremento do *movimento Browniano fracional* (mBf), isto é $X_i = B_H(i+1) - B_H(i), i \geq 1$ (mais detalhes sobre mBf na seção 2.3).

Os *processos fracionais ARIMA*(p, d, q) são uma generalização da classe de modelos Box-Jenkins [BJ76] onde permite-se que o parâmetro d assumam valores não inteiros. São processos assintoticamente auto-semelhante de segunda-ordem com parâmetro auto-semelhante $d + 1/2$, onde $0 < d < 1/2$. O F-ARIMA foi usado para modelar tráfego de vídeo VBR [BSTW95, HDLK95b] e o desempenho de fila do F-ARIMA (1, d , 0) foi analisado através de simulação para um servidor (*First Come First Service* - FCFS)[AM95].

2.3 Movimento Browniano Fracional

O movimento Browniano ordinário, $B(t)$, descreve o movimento de uma partícula em um líquido sujeito a colisões e outras forças [Pap78]. É uma função aleatória real com incrementos Gaussianos independentes tal que

$$\begin{aligned} E[B(t+s) - B(t)] &= 0 \\ \text{Var}[B(t+s) - B(t)] &= \sigma^2|s| \end{aligned}$$

Mandelbrot define um movimento Browniano fracional (mBf) como sendo o movimento médio de $dB(t)$ no qual incrementos que ultrapassam de $B(t)$ são ponderados por $(t-s)^{h-1/2}$ [MN68].

Definição: Seja H tal que $0 < H < 1$. O mBf é definido com a integral fracional de Weyl de $B(t)$.

$$B_H(t) = \frac{1}{\Gamma(H + 1/2)} \int_{-\infty}^{\infty} ((t-s)^{H-1/2} - (-s)^{H-1/2}) dB(s) + \int_0^t (t-s)^{H-1/2} dB(s).$$

Essa equação tende para o movimento Browniano ordinário se $H = 1/2$. Sua propriedade auto-semelhante é baseada no fato de que $B_H(\rho s)$ é idêntica na distribuição para $\rho^h * B_H(s)$. Os incrementos do mBf, Y_j , formam uma seqüência estacionária chamada *ruído Gaussiano fracional* (rGif):

$$Y_j = B_H(j+1) - B_H(j), j = \dots, -1, 0, 1, \dots$$

Esses incrementos não são independentes a não ser que se tenha movimento Browniano puro, *i.e.* $H = 1/2$. Além disso, a lei de Hurst estabelece que $\text{Var}[B_H(t+s) - B_H(t)] = \sigma s^{2H}$, *i.e.*, um modelo de chegada mBf é também capaz de capturar a alta *variabilidade* inerente exibida pelo tráfego de rede real.

O mBf normalizado com parâmetro $H \in [\frac{1}{2}, 1)$ é um processo estocástico $B(t), t \in (-\infty, +\infty)$, caracterizado pelas seguintes propriedades [Nor95]:

1. $B(t)$ tem incrementos estacionários;
2. $B(0) = 0$, e $EB(t) = 0$ para todo t ;
3. $EB(t)^2 = |t|^{2H}$ para todo t ;
4. $B(t)$ tem caminhos contínuos; e
5. $B(t)$ é Gaussiano, *i.e.*, todas suas distribuições marginais são Gaussianas.

Esse processo foi descoberto por Kolmogorov [Kol40], mas relativamente pouca atenção foi lhe dispensada até o artigo pioneiro de Mandelbrot e Van Ness [MN68] (onde o mBf ganhou seu nome atual).

2.4 Modelos de fonte *on/off* com caudas longas

Diversos estudos de medidas de tráfego de uma variedade de tipos de redes demonstraram que o tráfego atual de rede tem natureza auto-semelhante. Apesar dessas descobertas faltava compreender qual a explicação física para que o tráfego agregado observado tivesse esse comportamento. Willinger et. al [WTSW95, WTSW97] fornecem uma explicação simples e plausível em termos de tráfego de fontes individuais que formam o tráfego agregado. Desenvolvendo uma abordagem proposta originalmente por Mandelbrot [Man69], eles mostram que a superposição de muitas fontes *on/off* independentes e identicamente distribuídas (*i.i.d.*), cada uma das quais exibindo um fenômeno chamado de efeito Noé (*Noah effect*), resulta em um mBf, que é um processo exatamente auto-semelhante de segunda-ordem.

Uma fonte *on/off* idealizada é uma fonte onde períodos de geração de pacotes (períodos *on*) podem ser sucedidos por outros períodos *on*. Da mesma forma períodos sem geração de pacotes (períodos *off*) podem ser sucedidos por outros períodos *off*. Em um modelo *on/off* de alternância estrita, um período *on* não pode ser seguido de outro período *on*, e um período *off* por outro período *off*. Os modelos de fontes *on/off* de alternância estrita são comumente considerados na literatura de comunicações. Em [WTSW95] foram demonstradas as condições para que fontes *on/off* idealizadas gerassem um processo auto-semelhante, e em [WTSW97] os mesmos resultados foram obtidos usando fontes de alternância estrita. Apesar de os resultados de [WTSW95] serem essencialmente os mesmos de [WTSW97], a prova da convergência com fontes *on/off* com alternância estrita é muito mais complexo, e uma prova rigorosa é fornecida em [WTS97]. Heath et al. [HRS96] obtiveram independentemente resultados que são essencialmente idênticos aos apresentados em [WTS97].

Intuitivamente o efeito Noé para um modelo de fonte *on/off* individual significa períodos *on* e períodos *off* muito grandes com probabilidade não negligenciável. Em outras palavras, o efeito Noé implica que cada fonte *on/off* exibe características que encerram uma grande extensão de escalas de tempo. O efeito Noé é sinônimo de síndrome de variância infinita – a observação empírica de que muitos fenômenos naturais podem ser bem descritos usando distribuições com variância infinita (para referências veja [ST94, TL86, WTE96]). As distribuições de cauda-longa, ou seja, distribuições que decaem hiperbolicamente,

$$P(U > u) \sim cu^{-\alpha} \quad \text{com } u \rightarrow \infty, 1 < \alpha < 2, \quad (2.4)$$

onde c é uma constante finita positiva, independente de u , apresentam a síndrome da variância infinita. Uma distribuição que satisfaz as condições (2.4) é a distribuição discreta Pareto (e.g., veja [ST94]). A distribuição Pareto com parâmetro de forma α e parâmetro

de localização a tem a função de distribuição cumulativa dada por:

$$F(x) = P[X \leq x] = 1 - (a/x)^\alpha, a, \alpha \geq 0, x \geq a.$$

e função densidade de probabilidade correspondente dada por:

$$f(x) = \alpha a^\alpha x^{-\alpha-1}.$$

Se $\alpha \leq 2$, então a distribuição tem variância infinita, e se $\alpha \leq 1$, então a distribuição tem média infinita.

A distribuição Pareto têm sido usada para modelar distribuições de rendimentos, tamanhos de asteróides, ilhas, cidades e eventos de extinção [Kau93, Man63]. Foi constatado também que uma distribuição Pareto com $1.05 < \alpha < 1.25$ é um bom modelo para quantidade de tempo de CPU consumida por um processo arbitrário [LO86]. Em telecomunicações, distribuições com cauda longa tem sido usadas para modelar tempo de duração de chamadas telefônicas e tamanhos de quadros em vídeo com taxa de bit variável.

Provou-se também em [WTSW97] uma relação simples entre o parâmetro α e o parâmetro H dado por $H = 3 - \alpha/2$, onde H mede o grau de auto-semelhança do fluxo de tráfego agregado.

2.5 Métodos para Estimativa do Parâmetro H

O grau de auto-semelhança (parâmetro H) de uma série temporal pode ser estimado através das seguintes ferramentas gráficas: (1) análise do domínio de tempo baseado em estatística R/S, (2) análise de variância de processos agregados $X^{(m)}$, e (3) análise baseada em periodograma no domínio da frequência. A seguir são descritos os métodos para estimativa do grau de auto-semelhança de um processo.

1. **estatística R/S** — Originalmente descrita em [MW69] (veja também [MT79]) consiste em plotar $\log(R(n)/S(n))$ versus $\log(n)$, onde $R(n)/S(n) = 1/S(n)[\max(0, W_1, W_2, \dots, W_n) - \min(0, W_1, W_2, \dots, W_n)]$, com $W_k = (X_1 + X_2 + \dots + X_k) - k\bar{X}(n)$ ($k \geq 1$). Quando H é bem definido, a plotagem típica de extensão de re-escala ajustada inicia com uma zona transiente de natureza de dependência de curta-duração na amostra, mas eventualmente decresce e oscila em torno de uma reta com certo declive. Análise gráfica R/S é usada para determinar se tais comportamentos assintóticos parecem sustentados pelos dados. No caso afirmativo, um valor estimado de H , \hat{H} , é dado pelo declive assintótico, podendo variar entre $1/2$ e 1 .

2. **análise de variância de processos agregados** $X^{(m)}$ — Observa-se que para processos auto-semelhantes de segunda-ordem, a variância dos processos agregados $X^{(m)}$, $m \geq 1$, decresce linearmente (para grandes m) em uma escala log-log. A análise de variância de processos agregados é obtida pela plotagem $\log(\text{var}(X^{(m)}))$ contra $\log(m)$ (“tempo”) e pelo ajuste de uma simples linha quadrática mínima, ignorando-se os valores menores de m . Valores estimados do declive assintótico, $-\hat{\beta}$, entre -1 e 0 sugerem auto-semelhança, e uma estimativa de grau de auto-semelhança é dada por $\hat{H} = 1 - \hat{\beta}/2$.
3. **análise baseada em periodograma** — Consiste em plotar $\log(g(w))$ versus w , onde $g(w) = \frac{1}{2\pi n} |\sum_{j=1}^n X_j e^{ijw}|^2$, $0 \leq w \leq \pi$ de $X = (X_1, X_2, \dots, X_n)$, e aplicar um ajuste linear de mínimo-quadrado em uma seção do gráfico correspondente a w pequeno. O declive desse ajuste é igualado a $-(1 - \hat{\beta})$, de onde \hat{H} é computado com $1 - \hat{\beta}/2$.

A estatística R/S tem propriedades robustas, particularmente com respeito a distribuição de cauda longa. Contudo, ela tem a desvantagem de poder conduzir a estimativas imprecisas [Ber92]. A análise de variância de processos agregados é também um ponto de início útil na análise de dados.

Análises de dados mais refinadas baseadas em métodos MLE (*Maximum Likelihood Estimate*) são usadas para estimar H quando X_t é um processo Gaussiano ou aproximadamente Gaussiano. Esses métodos e suas aproximações são baseados na densidade espectral de X_t e propriedades dos métodos MLE são discutidas por vários autores [Ber92, Dah89, Yaj85, FT86]. Um método específico que tem sido usado amplamente é aproximação de *Whittle* MLE. Em [TT] discute-se as qualidades dos estimadores *Whittle* “padrão”, *Whittle* “agregado” e *Whittle* “local”.

O estimador *Whittle* padrão é um método paramétrico que assume como conhecida a densidade da série espectral, com exceção de poucos parâmetros que são estimados. Essa suposição de parâmetros permite estimativas muito precisas quando a série sendo examinada ajusta-se exatamente ao modelo assumido. Se, por outro lado, a série não é especificada de forma correta no modelo o estimador paramétrico pode gerar valores incorretos. Um técnica adicional para obtenção de um valor estimado de H é fornecida pelo estimador *Whittle* agregado. Esse estimador pode ser usado se a série temporal é muito longa. A idéia é agregar os dados, o que cria uma nova série mais curta.

$$X_i^{(m)} := \frac{1}{m} \sum_{j=m(i-1)+1}^{mi} X_j.$$

A nova série aumenta o desvio padrão do estimador. Por outro lado, $m^{1-H} X_i^{(m)}$ converge fracamente para um ruído Gaussiano fracional. Se o nível de agregação m é bastante alto

e a dependência de longa-duração está presente então a nova série será perto de um ruído Gaussiano fracional. O estimador *Whittle* padrão é então usado com um modelo de rGf na série agregada.

Alternativamente, pode se utilizar um estimador semi-paramétrico, *Whittle* local. Para estimar a dependência de longa-duração, o estimador *Whittle* local geralmente assume que a auto-correlação para a série comporta-se como h^{2H-2} , onde h é o *lag*, e H é o parâmetro de dependência de longa-duração. De forma equivalente ele pode assumir que a densidade espectral é proporcional a $|v|^{1-2H}$ para frequências pequenas de v . Esse estimador foi recentemente desenvolvido por [Rob94]. Ele é obtido através de um método não gráfico pela minimização de uma função similar aquela do *Whittle* padrão.

2.6 Gerador de Ruído Gaussiano

A solução para a distribuição de uma fila alimentada por um mBf não é conhecida até a presente data. Limites para resultados analíticos são derivados através da Teoria dos Valores Extremos para o cálculo desta distribuição. Uma outra maneira para derivação dos resultados de uma fila com mBf é através de simulação. Existem vários algoritmos para geração de um tráfego mBf [Hos84].[MH78]. Recentemente novos métodos foram desenvolvidos. Huang [HDLK95a] propôs um método de simulação baseado em importância de amostragem. Pruthi [PE95] usa mapas caóticos e Lau [LEWW95] usa um algoritmo de deslocamento de ponto médio aleatório. Neste trabalho utilizou-se um algoritmo proposto originalmente por Mandelbrot e aperfeiçoado posteriormente por Chi [CNY73]. Esse algoritmo gera amostras de rGf de tempo discreto. Escolheu-se esse algoritmo levando em consideração os seguintes fatores: i) o algoritmo é baseado em uma soma de processos de Markov, portanto trabalhos futuros poderão investigar possíveis soluções numéricas para esse sistema, ii) as amostras mBf são compostas de uma soma de processos de alta e a baixa frequência, logo esse algoritmo é adequado para investigar as contribuições de cada componente da frequência para o desempenho do sistema e iii) é um algoritmo relativamente rápido (em aproximadamente 4 minutos gera uma amostra de um milhão de pontos em uma máquina Pentium 90).

Processos de ruído Gaussiano fracional discreto normalizados com média zero e variância unitária são processos aleatórios Gaussianos com incrementos Gaussianos e que tem auto-covariância [ST94]:

$$C(s; H) = E[B_H(t+s+1) - B_H(t+s)][B_H(t+1) - B_H(t)] = 2^{-1}[|s+1|^{2H} - 2|s|^{2H} + |s-1|^{2H}]$$

O método proposto por Chi aproxima o rGf por uma soma de processos Markovianos de baixa e alta frequência. O processo agregado resultante tem uma função de auto-covariância muito similar a [LTWW94]. Pode-se dizer que a auto-correlação de um

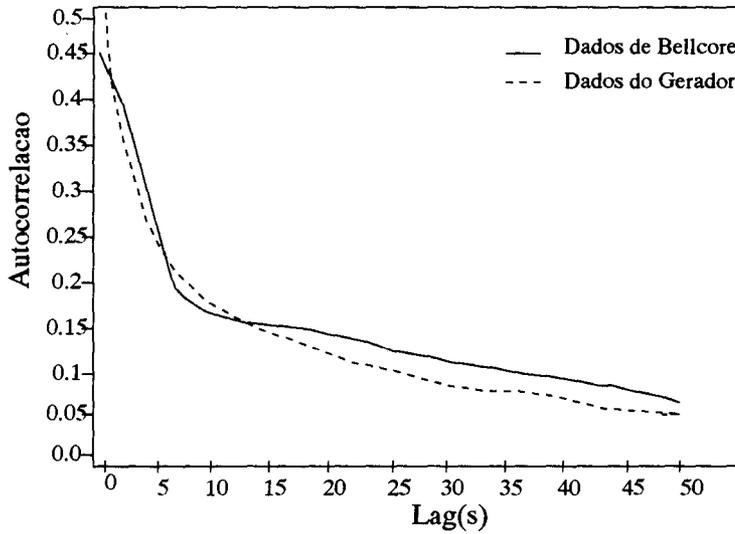


Figura 2.3: Autocorrelação do Gerador de Ruído Gaussiano Fracional e dados de Bellcore.

processo de Markov simples decai exponencialmente enquanto que a auto-correlação de rGf decai linearmente. Portanto, é necessário somar vários processos de Markov independentes a fim de se obter essa mesma estrutura de correlação. Um valor arbitrário de auto-correlação, aproximadamente $1/3$, é escolhido como valor de limite para separar as componentes de alta e baixa frequência. Considere:

$$X_f(t) = X_L(t) + X_H(t)$$

onde X_L e X_H denotam os termos de baixa e alta frequência respectivamente. X_f é o processo agregado. Para grandes lags (s) a covariância de $X_L(t)$, $C_L(s; H)$, é aproximadamente:

$$C_L(s; H) = H(2H - 1)s^{2H-2}$$

Ao invés de aproximar o termo de baixa frequência por um processo de Markov simples, uma soma ponderada de N processos de Markov-Gauss padronizada é usada, isto é:

$$X_L(t) = \sum_{n=0}^N (W_n)^{1/2} M^{(n)}(t)$$

Os processos Markov-Gauss, $M^{(n)}(t)$, são assumidos dois-a-dois não correlacionados. O objetivo é calcular W_n tal que a covariância de $X_L(t)$ seja igual a $C_L(s; H)$. Dessa forma, após alguns cálculos obtém-se:

$$W_n = \frac{H(2H - 1)}{\Gamma(3 - 2H)} (B^{1-H} - B^{H-1}) B^{2(H-1)n}$$

para $0 \leq n \leq N$, onde $\Gamma(3 - 2H)$ é a função gama.

A discrepância devida à aproximação é: $D(s; H) = C(s; H) - C_L(s; H)$.

Para gerar a discrepância de alta frequência utiliza-se outro processo Markov-Gauss com variância $D(0; H)$ e atraso de covariância $D(1; H)$, tal que a covariância da soma do processo de baixa e alta frequência terá a estrutura auto-semelhante [LTWW94]. A aproximação do rGf é dada por:

$$X_f(t) = X_L(t) + X_H(t)$$

$$C_f(s) = C_L(s) + C_H(s)$$

A título de exemplo gerou-se dados com valores de H entre [0.7 - 0.85] tal que tem-se os mesmos valores coletados em Bellcore [LTWW94]. Mayor [May97] estimou o parâmetro *Hurst* (H) pelo uso da análise R/S e nesta dissertação acrescentou-se a análise da variância do processo agregado. A (Tabela 2.1) mostra os dois resultados.

H	Análise R/S	Análise de Variância
0.7	0.69	0.71
0.75	0.73	0.76
0.8	0.78	0.81
0.85	0.83	0.84

Tabela 2.1: Estimativa do parâmetro H da amostra gerada.

Pelos dados mostrados na Tabela 2.1 verifica-se que esse algoritmo de geração tem uma boa precisão. Em [May97] computou-se também o coeficiente de correlação de 1.000.000 pontos da amostra de dados de Bellcore e comparou-se com as amostras do gerador (Figura 2.3). Verifica-se que i) os dados gerados tem uma estrutura com correlação de longa-duração e ii) ambas as estruturas de correlação são muito semelhantes. É uma indicação clara de que esse gerador é uma boa caracterização de tráfego real de rede.

2.7 Fenômeno de Ineficácia de Buffer

Conforme visto na seção 2.2, um processo auto-semelhante é capaz de capturar o fenômeno de dependência de longa-duração (LRD - *Long Range Dependence*). Além disso, uma série de estudos analíticos e de simulação [MS96b, MS96a, FFM97a, FFM97b] indicam que esse fenômeno pode ter um impacto significativo na formação de fila, *i.e.*, existe clara evidência que ele pode potencialmente causar perdas maciças de células em redes ATM. Na realidade, a probabilidade de transbordo de *buffer* para um sistema de

fila ATM com chegadas mBf segue uma distribuição de Weibull. Além disso, esse sistema sofre do fenômeno de ineficácia do *buffer* [MSA97].

O fenômeno de ineficácia de *buffer* é o fenômeno de enfileiramento em que apenas aumentando-se o tamanho do *buffer*, não é possível diminuir consideravelmente a probabilidade de transbordo. Esse fenômeno foi descrito anteriormente por vários outros estudos [MS96a, SL94, LH93]. Nesta seção, apresenta-se uma explicação intuitiva para esse fenômeno e mostra-se sua particular importância quando a fonte de tráfego exibe dependência de longa-duração. Os resultados apresentados a seguir são uma reprodução do trabalho original de Mayor [MS96a].

Modela-se um nó ATM com um sistema de fila determinístico com taxa de partida constante dada por c e tamanho de *buffer* finito por b . O tráfego de entrada é dado pelo processo estocástico $A(t)$ com taxa média de entrada $\bar{a} < c$. Esse processo define o número agregado de chegadas de células até o tempo $t, t \geq 0$. Assume-se que o transbordo do *buffer* ocorre no tempo t tal que se possa escrever $A(t) = ct + b$. Além disso, $A(t)/t \geq c + b/t$.

Pela Lei dos Grandes Números, a taxa média de chegada $A(t)/t$ converge para sua média \bar{a} . Portanto, a probabilidade que ela exceda o termo $(c + b/t)$ decresce com t

$$P(A(t)/t \geq c + b/t) = \Psi(t)$$

Em outras palavras, $\Psi(t)$ é uma função decrescente com o tempo. O fenômeno de ineficácia de *buffer* ocorre se a probabilidade de transbordo, dada por $\Psi(t)$, decai lentamente com t , *i.e.* se $\Psi(t)$ é não negligenciável para grandes t . Nesse caso, uma vez que t é grande, o termo (b/t) é negligenciável. Portanto, mesmo incrementando-se o tamanho do *buffer*, não é possível aumentar significativamente o termo (b/t) com o intuito de diminuir a probabilidade de perda de célula. Intuitivamente, esse fenômeno ocorre se o processo de chegada é capaz de transmitir em altas taxas para qualquer período longo de tempo, isto é, se ele converge lentamente para sua média. Mostra-se que a fonte LRD pode transmitir em altas taxas para qualquer período longo de tempo. De acordo com Norros [Nor94], assume-se que o processo de chegada $A_H(t)$ é um processo de movimento Browniano fracional (mBf) dado por $A_H(t) = \bar{a}t + \sigma Z(t)$ onde $\bar{a} > 0$ é a taxa média de entrada, $\sigma > 0$ é o desvio padrão, $H \in [1/2, 1)$ é o parâmetro (Hurst) auto-semelhante e $Z(t)$ é um movimento Browniano fracional normalizado. Quando $H = 1/2$, tem-se o caso especial de movimento Browniano ordinário. A probabilidade de, através de um intervalo de tempo de tamanho t , a fonte $A_H(t)$ poder superar o serviço potencial ct e exceder um nível de *buffer* b é dada por:

$$P(A_H(t) \geq ct + b) = P(\bar{a}t + \sigma Z(t) > ct + b) = P(Z(t) > \frac{t(c - \bar{a}) + b}{\sigma}).$$

Pela propriedade auto-semelhante $Z(t) = t^H Z(1)$, tem-se:

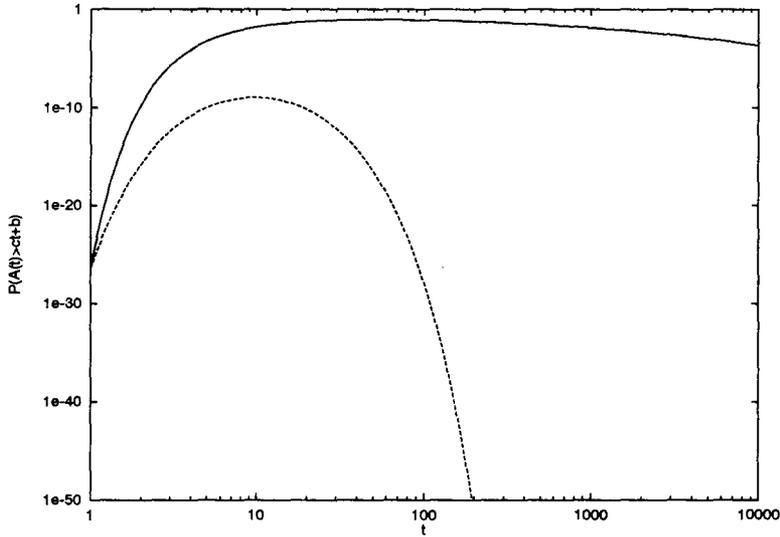


Figura 2.4: O fenômeno de ineficácia de *buffer*.

$$P(Z(1) > \frac{t(c - \bar{a}) + b}{\sigma t^H}) = \Phi\left(\frac{t(c - \bar{a}) + b}{\sigma t^H}\right)$$

onde $\Phi(y) = P(Z(1) > y)$ é a função de distribuição residual da distribuição Gaussiana padrão. De fato, usando a aproximação:

$$\Phi(y) \approx (2\pi)^{-1/2}(1+y)^{-1} \exp(-y^2/2) \approx \exp(-y^2/2)$$

obtem-se:

$$P(A_H(t) > ct + b) = \Phi\left(\frac{t(c - \bar{a}) + b}{\sigma t^H}\right) \approx \exp\left(-\frac{1}{2}g(t)^2\right) = \exp\left(-\frac{1}{2}\left(\frac{t(c - \bar{a}) + b}{\sigma t^H}\right)^2\right) \quad (2.5)$$

A Figura 2.4 mostra a probabilidade de perda em função do tempo para duas fontes com mesma média, desvio padrão e parâmetro $H = 0.5$ e $H = 0.85$ respectivamente e largura de banda tal que a utilização da linha fosse 50%. Pode-se ver que a probabilidade de transbordo do *buffer* para a fonte LRD decai muito lentamente com o tempo. Portanto, aumentando-se o tamanho do *buffer* não é suficiente para acomodar a forte componente de baixa-freqüência dessa fonte no sentido de evitar perdas de células. Por outro lado, Φ decai muito rápido no caso de chegadas não correlacionadas (movimento Browniano).

Calculando-se $t' \in \arg \min_{t > 0} g(t)$ tal que a probabilidade de transbordo é maximizada, encontra-se uma escala de tempo apropriada na qual o transbordo ocorre no sistema. Portanto, t' é dado por $t' = \frac{bH}{(c-\bar{a})(1-H)}$.

Este mesmo resultado foi derivado independentemente por Addie em [AEN95] e Ryu em [RE96]. Nota-se que, uma vez que a probabilidade de transbordo do *buffer* para as fontes LRD decresce lentamente com o tempo, a probabilidade de transbordo de *buffer* no tempo t' não é precisamente estimada pela probabilidade de transbordo de *buffer* no estado fixo. Em outras palavras, estimar a probabilidade de transbordo em um sistema de fila dirigido por uma fonte LRD pela probabilidade de transbordo no tempo t' , pode não capturar a cauda longa exibida pela Equação (2.5) conduzindo a resultados imprecisos.

Capítulo 3

Gerenciamento de Tráfego em Redes ATM

Nas redes ATM enfrenta-se a difícil tarefa de se satisfazer as necessidades de conexões que requerem diferentes Qualidades de Serviço (QoS - *Quality of Service*). Por exemplo, uma conversação telefônica pode tolerar uma taxa de perda alta (até 10^{-3}) sem prejudicar a conversação enquanto que um alto retardo pode ser intolerável, já aplicações de vídeo podem se contentar com um certo retardo, mas não toleram taxas de perda alta (maior que 10^{-9}).

Conforme [CFZ98], o conjunto de funções de gerenciamento pode ser divididas em dois grupos: (1) aquelas que são tomadas durante o estabelecimento da conexão, e (2) aquelas que são tomadas durante a conexão. Como exemplo de mecanismos de gerenciamento de tráfego que se encaixam no grupo (1) tem-se o controle de admissão e roteamento. Policiamento de tráfego, descarte seletivo e fluxo de controle ABR são exemplos de mecanismos de gerenciamento que se encaixam no grupo (2).

Este capítulo está dividido da seguinte maneira: na seção 3.1 apresenta-se o mecanismo de controle de admissão. Na seção 3.2 mostra-se o mecanismo de policiamento de tráfego. A seção 3.3 mostra os mecanismos de descarte seletivo e a seção 3.4 aborda o fluxo de controle ABR. Será dado neste capítulo maior ênfase aos mecanismos de descarte seletivo, pois estes constituem o tema principal desta dissertação.

3.1 Controle de Admissão de Conexões

O Controle de Admissão de Conexões (CAC - *Connection Admission Control*) é o conjunto de ações tomadas pela rede durante a fase do estabelecimento da chamada (ou durante a fase de renegociação dos parâmetros de conexão) para determinar se uma conexão requisitada pode ser aceita ou rejeitada.

Alguns tipos de chamadas podem envolver mais de uma conexão como é o caso de conexões multimídia ou que envolvam mais de um parceiro (por exemplo, videoconferência). Neste caso, cada conexão deve ser aceita ou não independentemente pelo CAC. Baseado no esquema CAC uma requisição de conexão é admitida somente quando recursos suficientes estão disponíveis para garantir os requisitos de QoS fim-a-fim desta conexão e para todas as outras conexões existentes. Uma das dificuldades enfrentadas pelo CAC é que nem todos os parâmetros de uma determinada fonte de tráfego são conhecidos e, portanto não é possível fazer uma alocação precisa, e certamente haverá desperdício de capacidades.

Nesta seção discute-se alguns princípios gerais relacionados com CAC, como por exemplo, quais os critérios para admissão de uma conexão e quais os diferentes princípios e abordagens conceituais no projeto de CAC.

3.1.1 Descritores de Conexões

No estabelecimento da chamada, o usuário e a rede negociam os parâmetros de tráfego tais como: Taxa de Pico de Célula (PCR - *Peak Cell Rate*), Taxa de Célula Sustentada (SCR - *Sustainable Cell Rate*), Taxa de Célula Mínima (MCR - *Minimum Cell Rate*), Tolerância de Rajada Intrínseca (IBT - *Intrinsic Burst Transfer*), Tolerância da Variação do Atraso de Célula (CDVT - *Cell Delay Variation Tolerance*) e os requisitos de QoS solicitados (Taxa de Perda de Célula (CLR - *Cell Loss Rate*), Atraso de Transferência de Célula (CTD - *Cell Tolerance Delay*), Variação de Atraso de Célula (CDV - *Cell Delay Variation*)) e a Capacidade de Transferência ATM (ATC - *ATM Transfer Capability*). O resultado dessa negociação é o *contrato de tráfego*. A função do contrato de tráfego é permitir uma operação da rede eficiente onde os requisitos de QoS de cada conexão são satisfeitos. ATC diferentes requerem parâmetros de tráfego diferentes, conseqüentemente o conjunto de descritores de conexão pode variar de serviço para serviço. Por exemplo, um serviço que requer capacidade de transferência constante (CBR - *Constant Bit Rate*) não precisa especificar a SCR como um serviço de transferência variável (VBR - *Variable Bit Rate*). Também somente uma conexão ABR (*Available Bit Rate*) pode especificar taxa de célula mínima. O CAC pode ser entendido como um sistema no qual as entradas são os descritores da conexão, o CDVT, o QoS e o ATC da conexão proposta, e a saída é a decisão se a conexão será ou não aceita.

3.1.2 Abordagens de CAC

Apesar de muitas propostas para CAC terem sido analisadas e estudadas na literatura, não há clara concordância de qual é a melhor para um dado conjunto de serviços. Nesta seção são discutidas três abordagens diferentes para implementação de CAC. A primeira é

a mais simples e a mais conservativa. Ela se baseia na alocação da taxa de pico. A segunda é baseada na aproximação de *buffer* zero ou Multiplexação de Taxa de Envelope (REM - *Rate Envelope Multiplexing*). A terceira é baseada em modelos de tráfego e enfileiramento e leva em consideração a capacidade de *buffer* da rede para compartilhar a carga excessiva durante o período de tráfego pesado.

CAC baseada em Alocação de Taxa de Pico - Esta abordagem considera a taxa de pico de cada nova conexão. Se para cada *link* da rede no roteamento da nova conexão a soma das taxas de pico de todas conexões já em progresso, mais a taxa de pico da nova conexão, for menor do que a capacidade total do *link*, admite-se a nova conexão, caso contrário rejeita-se. Este esquema pode levar a um péssimo aproveitamento da capacidade da rede (no caso de fluxos VBR), além de não levar em consideração o efeito do CDV.

Enquanto CAC baseado na alocação de taxa de pico pode ser eficiente para fluxo CBR, ele é geralmente ineficiente para fluxos VBR. Em uma conexão VBR típica a taxa de pico é várias vezes maior que a taxa média, logo é possível aceitar um total de conexões cuja soma de taxa de pico seja maior que a capacidade disponível.

O efeito CDV é significativo principalmente no caso de alocação de taxa de pico para fontes CBR. Devido ao CDV, quando se aceita chamadas tais que a soma de diferentes taxas de pico é igual ou bem perto da capacidade disponível, a CLR pode exceder a CLR requerida conforme especificado no contrato de tráfego. Portanto, é importante considerar se o CDV das diferentes conexões assegura que suas CLR's estão de acordo com o contrato de tráfego.

CAC baseado em Multiplexação de Taxa de Envelope (REM - *Rate Envelope Multiplexing*) - O REM é baseado na hipótese de que não existe *buffer* ou que o tamanho do *buffer* é muito pequeno. Essa hipótese é chamada de "Aproximação de *buffer* zero" (*The Zero Buffer Approximation*). Essa hipótese é interessante pelas seguintes razões. Primeiro, ela é prática em muitas aplicações de tempo real onde os *buffers* são muito pequenos para absorver flutuações a nível de rajada. Portanto, é conveniente assumir que os *buffers* evitaram todas as perdas de células que possam ser causadas devido a flutuações na escala do tempo de célula, e não conseguem evitar perdas de célula devido a flutuações a nível de rajada. Assim sendo pode-se usar modelos de tráfego a nível de rajada tais como modelos *on/off* em conjunto com a aproximação de *buffer* zero para obter uma estimativa precisa de perda de célula. Segundo, com a aproximação de *buffer* zero não há necessidade de análise de fila. Terceiro, sobre a aproximação de *buffer* zero não há necessidade de se considerar a correlação no processo de chegada, logo pode-se considerar processos com chegadas independentes. Isso torna o processo de modelagem de tráfego e caracterização mais simples.

Seja S uma variável aleatória representando a quantidade total de trabalho que chega

em um intervalo de tempo pequeno (que inclui a quantidade de trabalho que chega das conexões existentes mais a quantidade da nova conexão). Seja C a capacidade de *link* disponível. A CLR é dada por:

$$P_{loss} = \frac{E(S - C)^+}{E(S)} \quad (3.1)$$

Nesta equação, a CLR é simplesmente a razão entre a quantidade de trabalho perdida e a quantidade de trabalho que chega.

Para se implementar um CAC baseado em REM, precisa-se ter a distribuição de S em todo ponto do tempo que houver uma nova chegada de requisição de conexão. Isso pode ser estimado pelas medidas de tráfego. Tendo-se o comportamento de S , então para uma nova requisição de conexão, computa-se a CLR usando a Eq.(3.1), e se a CLR prevista é mais baixa que a CLR requerida (como especificado no contrato de tráfego), a conexão é aceita, de outro modo ela é rejeitada.

CAC baseado em Compartilhamento de Taxa - CAC baseado em REM não é muito eficiente no caso onde comutadores usam grandes *buffers* capazes de absorver grandes rajadas. Há portanto a necessidade de um CAC que leve em consideração as capacidades de *buffer* dos comutadores. Uma das funções importantes de tal CAC é estimar a CLR. Para toda nova chegada de requisição de conexão, tal estimativa irá computar qual será a CLR. Se a conexão for admitida, e irá compará-la com a CLR requerida. Uma forma de prever a CLR é usar medições e estimativas da distribuição da quantidade de chegadas de células durante diferentes intervalos de tempo. O método é baseado no uso da Eq.(3.1) para o caso de *buffer* infinito.

Seja $S(t)$ uma variável aleatória que representa a quantidade de trabalho que chega em um intervalo de tempo t . Sabe-se que se durante o tempo t , a quantidade de trabalho que chega é maior do que pode ser servido e ou armazenado nos *buffers* então algum trabalho deve ser perdido. Sabe-se também que a perda deve ser pelo menos o excesso de trabalho acima do que pode ser servido e armazenado. Seja $CLR(t)$ o limite inferior para a proporção de perda de célula baseada na variável aleatória $S(t)$, assim obtém-se:

$$CLR(t) \geq \frac{E(S(t) - C(t) - b)^+}{E(S(t))} \quad (3.2)$$

onde $C(t)$ é a quantidade de trabalho que pode ser servida durante o tempo t e b é o tamanho do *buffer*. Uma vez que $CLR(t)$ para todo t é um limite inferior da CLR, tem-se que:

$$CLR \geq \max CLR(t). \quad (3.3)$$

Considerando-se a Eq.(3.3) como uma igualdade, obtém-se uma estimativa para CLR. Em [ZT97] demonstra-se que o lado direito da Eq.(3.3) é um estimador preciso para a CLR.

3.2 Policiamento de Tráfego

Um mecanismo de policiamento é responsável por avaliar se as fontes geradoras de tráfego estão respeitando os parâmetros descritos no contrato de tráfego negociado quando da aceitação da conexão pela rede. A violação do contrato de tráfego pode ser causada por vários fatores, desde a estimativa imprecisa dos parâmetros de tráfego até comportamento malicioso por parte do usuário. Uma célula que viola um contrato pode ser descartada no momento em que se identifica a ocorrência da violação, ou receber uma marca que caracterize sua situação sendo posteriormente descartada caso venha visitar um nó congestionado. Descreve-se a seguir os principais algoritmos conhecidos na literatura para policiamento. Outros algoritmos podem ser encontrados em [Mon94, SM96, Vec94].

O mais conhecido dentre os vários mecanismos de policiamento para redes ATM, é o **Balde Furado (BF - Leaky Bucket)**. No mecanismo de BF, uma célula precisa consumir uma ficha antes de entrar na rede. Fichas são geradas a taxa constante e existe um número máximo de fichas (tamanho do balde) permitidas em um reservatório de fichas (balde). Se uma célula chega em uma rede e não existe ficha no reservatório, ela será descartada ou marcada. Um mecanismo de BF pode também ser implementado por um contador, uma taxa de vazamento e um limite. O contador é incrementado a cada chegada de célula até um limite e é decrementado à taxa de vazamento. Toda vez que o contador alcança o valor limite, o mecanismo de BF marca (descarta) células. Foram propostos na literatura diversas variantes deste algoritmo básico. Uma destas variantes é o mecanismo de **balde furado com buffer (buffered leaky bucket)**, que tem um *buffer* para controlar a taxa do tráfego ao invés de descartar ou apenas marcar a célula. O mecanismo adotado pelo ATM Forum, conhecido por *The Generic Cell Rate Algorithm* é equivalente ao mecanismo de balde furado.

Além do mecanismo de BF existem outros mecanismos na literatura baseados em janelas. O mecanismo de **Janela Saltitante (JS - Jumping Window)** limita o número máximo de células que uma fonte pode emitir, dentro de um intervalo T (tamanho da janela), a um valor N . Um contador é incrementado sempre que chega uma célula da fonte monitorada, e as células só passam a ser marcadas ou descartadas quando o valor ultrapassar N . Um novo intervalo começa imediatamente após o final do precedente. O contador é zerado ao final do intervalo. Para um mecanismo de JS ter uma taxa equivalente a um mecanismo de BF a proporção entre o limite superior do número de células em um intervalo e a duração do intervalo deve ser igual à taxa de vazamento. Uma variação do mecanismo de JS é o mecanismo de **Janela Saltitante com Gatilho (JSG - Triggered Jumping Window)**. Neste mecanismo o início da janela é disparado (*triggered*) pela chegada de uma célula.

A principal desvantagem dos mecanismos JS/JSG é que, com o início de uma nova

3.3 Mecanismos de Descarte Seletivo

Multiplexação estatística foi adotada no padrão ATM a fim de se maximizar os recursos da rede frente à natureza da taxa variável de fontes multimídia. Contudo, o ganho obtido através da multiplexação estatística vem com um custo: células concorrem por recursos compartilhados, e portanto, algumas destas podem ser perdidas. Isso ocorre porque nas redes multimídia enfrenta-se a difícil tarefa de satisfazer às necessidades de conexões que requerem diferentes QoS. Os requisitos de QoS de perda de uma aplicação são usualmente traduzidos em duas métricas: a taxa de perda (razão entre as células perdidas e o número total de células transmitidas) e o tamanho do intervalo de perdas (o número de células perdidas consecutivamente). O descarte seletivo é um mecanismo que visa habilitar a rede a lidar com diversos requisitos de perda. Em um mecanismo de descarte seletivo células são descartadas de acordo com seus níveis de prioridade.

É importante entender o significado de taxa de perda e tamanho do intervalo de perdas. Não é incomum o uso do termo probabilidade de perda quando o correto é taxa de perda. Quando se fala em probabilidade de perda, está implícito que a probabilidade de perda de células é a mesma para todas as células. Contudo, em um fluxo de células correlacionado, o processo de perda de células também é correlacionado. Assim algumas células são mais prováveis de serem perdidas do que outras de acordo com suas posições no processo. Em outras palavras, uma perda de uma célula particular pode ser influenciada pela perda de células anteriores. Dessa forma, não é correto aplicar o termo probabilidade de perda para processos não renováveis.

Apesar da taxa de perda ser um parâmetro significativo, ela é um valor médio e não descreve inteiramente o processo de perda. O número de células perdidas consecutivamente (intervalo de perdas) dá uma descrição mais detalhada do processo de perda. Para um certo valor de taxa de perda, células podem ser perdidas de vários modos diferentes. Por exemplo, para uma taxa de perda de 0.25, pode-se perder uma célula a cada quatro células transmitidas ou pode-se perder um quarto de todas as células de uma única vez. Dependendo do procedimento de recuperação de sinal do lado do receptor, o intervalo de perdas pode ter um impacto diferente na percepção da QoS pelo usuário da rede.

Descarte seletivo está relacionado somente com o gerenciamento do espaço de *buffer* e não com a ordem de transmissão. Um mecanismo de descarte seletivo é completamente especificado por uma política de organização de *buffer* e pela política de descarte. Uma política de organização de *buffer* define qual parte do *buffer* pode ser ocupado por qual célula, uma política de descarte escolhe uma célula para ser descartada entre as células da classe com prioridade mais baixa. As políticas de organização de *buffer* mais comuns são [KK80]:

- (i) **Particionamento Completo** - cada classe de clientes tem sua própria fila indi-

vidual. Não existe compartilhamento de *buffer*. Cada espaço de *buffer* pode ser otimizado para manipular seu fluxo de tráfego esperado. Apesar de ser uma política atrativa por não ter *overhead* de controle, perdas ocorrem mesmo se existem espaço livre de *buffer*:

- (ii) **Compartilhamento Total** - todos os espaços de *buffer* são acessíveis para todos os clientes. Células são descartadas se e somente se o *buffer* está cheio. Ela minimiza a perda total de célula;
- (iii) **Compartilhamento Total com Descarte** - funciona da mesma maneira que a política de Compartilhamento Total, mas se uma célula encontra o *buffer* cheio e existe uma célula da classe com prioridade baixa na fila então esta célula é descartada (Compartilhamento Total é algumas vezes usado para designar Compartilhamento Total com Descarte);
- (iv) **Compartilhamento Parcial** - cada classe tem uma posição limite até a qual células podem ser aceitas dentro do espaço de *buffer*. Se uma célula chega e existe mais células na fila do que a posição limite da sua classe então a célula é perdida;
- (v) **Compartilhamento com Tamanho de Fila Máximo** - existe um limite máximo para o número de clientes de cada classe;
- (vi) **Compartilhamento com Alocação Mínima** - existe uma quantidade mínima de espaço de *buffer* alocado para cada classe de clientes. O resto do espaço de *buffer* é compartilhado por todas as classes de clientes.
- (vii) **Compartilhamento Total com Garantia de Tamanho Mínimo de Fila** - difere do compartilhamento total por garantir um número mínimo de *buffers* para cada classe em situação de transbordo [LZ92]. Nestas situações células das classes que excedem este mínimo são descartadas de acordo com seu nível de prioridade.

A Figura 3.1 ilustra os cenários de perda para diferentes políticas de *buffer*.

Uma política de organização de *buffer* é chamada de política de conservação de taxa de perda, se células (pacotes de tamanho fixo) são perdidas somente em situações de transbordo. Em outras palavras, uma disciplina que conserva a taxa de perda sempre admite uma célula no *buffer* se existir espaço disponível. Das políticas de organização citadas acima, Compartilhamento Total (com e sem descarte) e Compartilhamento Total com Garantia de Tamanho Mínimo de Fila são exemplos de disciplinas que conservam a taxa de perda. Um exemplo de disciplina que não conserva a taxa de perda é a Compartilhamento Parcial, na qual células da classe com prioridade baixa podem ocupar até uma certa posição no *buffer* e conseqüentemente perdas podem ocorrer mesmo se o *buffer* não

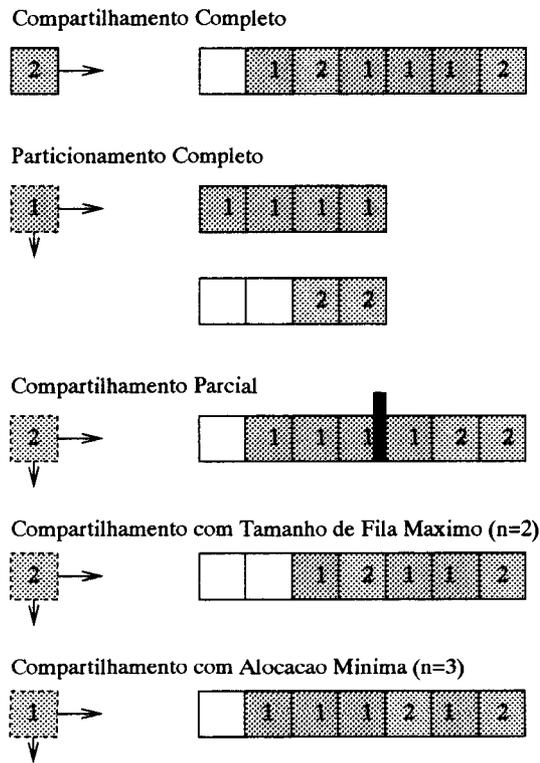


Figura 3.1: Exemplos de políticas de organização de *buffer*.

estiver cheio. Disciplinas que conservam a taxa de perda são de interesse especial porque elas minimizam a perda total de células e conseqüentemente maximizam a vazão.

O nível de prioridade de uma célula pode ser definido estatisticamente ou dinamicamente. Se atribuído estatisticamente, o nível de prioridade indica a importância de uma célula. Por exemplo, células que carregam informações importantes em um fluxo, tais como células MPEG do quadro I, devem ter prioridade alta. Além disso, se todas as células de uma aplicação tem o mesmo nível de prioridade, eles indicam os requisitos de perda dessa aplicação. Prioridades atribuídas dinamicamente são relacionadas à discrepância entre os parâmetros negociados e os observados da transmissão de uma conexão. Como por exemplo, um mecanismo de policiamento marca com prioridade baixa as células correspondentes a períodos de violação, períodos nos quais as características de transmissão de uma conexão não estão de acordo com os valores negociados em tempo de admissão da conexão.

Uma política de descarte seleciona uma célula para ser descartada entre as células da classe com prioridade baixa. As políticas mais comuns são Último-a-Chegar-Primeiro-Descartado (UCPD - *Last-In-First-Drop*), Primeiro-a-Chegar-Primeiro-Descartado (PCPD - *First-In-First-Drop*), Seleção Aleatória (RAND - *Random*) e Primeiro-a-Chegar-Primeiro a ser-Descartado Modificada (PCPD-M - *Modified First-In-First-Drop*). A política PCPD Modificada sempre descarta uma célula da classe com prioridade mais baixa para liberar espaço de *buffer* para uma célula que chegue, independente do nível de prioridade da célula que chegue. Descartar células da classe com prioridade baixa em diferentes posições define diferentes distribuições de fila e, conseqüentemente, pode ter impacto diferenciado no parâmetro de QoS. Como por exemplo, suponha que existem duas células da classe com prioridade baixa na fila, uma no final e outra no início (Figura 3.2). Se uma célula da classe com prioridade alta encontra o *buffer* cheio, conforme a política UCPD, a célula da classe com prioridade baixa no final da fila é descartada. De acordo com a política PCPD a célula do início da fila é descartada. Após várias chegadas de células da classe com prioridade alta, o *buffer* estará cheio novamente. Na fila com UCPD, a célula no início da fila antes das chegadas de células da classe com prioridade alta (que não foi descartada) já foi transmitida e, conseqüentemente, a célula da classe com prioridade alta é perdida. Na fila com PCPD, contudo, a célula da classe com prioridade baixa no final da fila antes das chegadas de células da classe com prioridade alta não foi descartada e pode ser agora descartada para liberar espaço para a nova célula da classe com prioridade alta. A política PCPD-M é a expressão máxima da diferença entre PCPD e UCPD pois a política PCPD-M tenta concentrar as células da classe com prioridade baixa no final da fila e, portanto, aumenta a probabilidade de uma célula da classe com prioridade baixa ser descartada. Dessa forma, a política PCPD-M minimiza a taxa de perda da classe com prioridade alta [THP94].

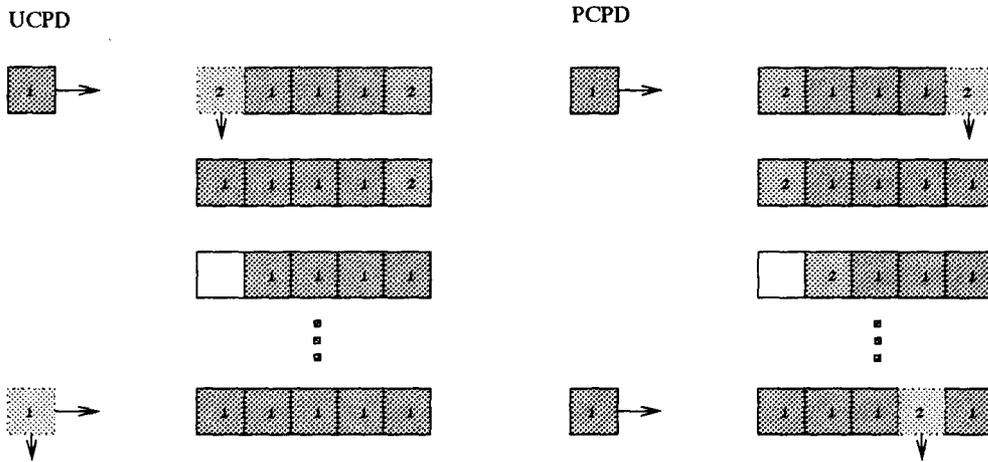


Figura 3.2: Um exemplo de perda de célula devido a diferentes distribuições de tamanho de fila produzidas pelas políticas de descarte UCPD e PCPD.

3.4 Controle de Fluxo ABR

O serviço *Available Bit Rate* (ABR) é definido para o suporte de aplicações que podem requerer banda passante mínima. Ele garante uma proporção baixa ou zero de perda de célula e uma distribuição justa da banda passante disponível ao se moldar o tráfego de acordo com a informação de retorno recebida da rede. Tal serviço é geralmente solicitado por aplicações de dados que não podem prever seus próprios requisitos de banda passante, e esperam conseguir algum compartilhamento da banda passante disponível. Como contraste, a categoria de serviço *Unspecified Bit Rate* (UBR) oferece um serviço de melhor-esforço sem garantias de QoS. Este serviço é projetado para aplicações de dados que querem usar banda passante disponível e não são sensíveis a perda e retardo de célula. Uma conexão UBR não é rejeitada baseada na falta de banda passante e não é policiada pela rede.

O serviço ABR permite que aplicações utilizem banda passante não usada pelos serviços VBR e CBR, pelo ajuste contínuo de suas taxas de transmissão instantânea à capacidade disponível no momento para ABR. Um esquema de controle de congestionamento é essencial para o suporte de tráfego ABR utilizar a banda passante disponível sem causar congestionamento, e para dividir essa banda passante entre as conexões ABR de maneira justa. Computar a divisão justa de uma conexão ABR de um *link* de saída de um comutador ATM requer um algoritmo de alocação justa. Outro algoritmo é necessário no final do sistema para ajustar a sua taxa ao retorno recebido da rede. O comportamento da fonte é importante para garantir a taxa de perda com prioridade baixa. Se uma fonte não se comporta como esperado, isso pode aumentar a taxa de perda de todos os usuários

ABR.

3.4.1 Alocação *Max-Min*

O serviço ABR deve supostamente dividir a banda passante deixada pelas conexões VBR e CBR entre conexões ABR dando a cada fonte ABR uma parte justa. Existem muitas definições para justiça. O Forum ATM decidiu pela definição particular, chamada de justiça *max-min* [Jaf81], que maximiza a vazão total. Alocação *max-min* permite a todas as conexões VC (*Virtual Channel*) usando um *link* em comum compartilhar igualmente a banda passante disponível tanto quanto elas não sejam engarrafadas em outro lugar da rede. As conexões VC competindo por banda passante em um *link* l são divididas em duas categorias: conexões restritas (*constrained*) e conexões não restritas (*unconstrained*). Conexões restritas são aquelas que não podem usar sua divisão justa em l devido à banda passante limitada em outro *link*. Conexões não restritas são aquelas que não podem usar sua divisão justa em outro *link* devido à banda passante limitada em l . Para essas conexões, o *link* l é referido como o “*link* de engarrafamento” (*bottleneck link*). A idéia básica por trás da justiça *max-min* é alocar as conexões não restritas de l e compartilhar igualmente a banda passante deixada pelas conexões restritas.

Um algoritmo centralizado para computar alocações *max-min* trabalha da seguinte maneira. Suponha que a rede tenha L *links*, (designados) por $1 \dots L$. Para cada *link* l , indicado por b_l a capacidade (banda passante) de l , por C_l o conjunto de conexões VC ABR atravessando l , e pelo $|C_l|$ o número de conexões. Uma seqüência de alocação de taxas podem ser representadas por um vetor β no *link* l . Tal alocação é dita *max-min* justa se as seguintes condições são satisfeitas:

(c1) - A taxa $\beta(c)$ alocada por β para uma conexão, c , VC ABR não é negativa.

(c2) - Para cada *link* l , $1 \leq l \leq L$, $\beta[l] \geq b_l$.

(c3) - $\beta(c)$ não pode ser aumentado sem diminuir a taxa alocada para outra conexão c' para qual $\beta(c) \geq \beta(c')$.

Provou-se que as condições (c1)-(c3) são preenchidas por β se e somente se cada conexão c tem um *link* de engarrafamento, isto é, um *link* l sobre o qual a taxa atribuída para c não pode ser aumentada porque toda a capacidade l foi alocada (*i.e.* $B[l] = b_l$) e porque nenhuma das VCs atravessando l foi atribuída mais banda passante do que c (*i.e.* para cada $c' \in C_l$, $\beta(c) \geq \beta(c')$).

O algoritmo para achar um vetor *max-min* justo trabalha em iterações até achar o *link* engarrafado para cada conexão VC. Ele inicia com um vetor β^0 de taxa zero, o que obviamente preenche as condições (c1) e (c2). Ele então aumenta algumas das

componentes de β^0 tal que (c2) continue valendo e um *link* engarrafado seja encontrado, neste caso β^1 é definido. Para garantir que (c2) não é violado, o algoritmo olha para o *link* l , para o qual a banda passante disponível dividida pelo número de VC's não engarrafadas ainda é "mínimo". Ele então aloca uma distribuição igual a banda passante do *link* selecionado para toda VC atravessando esse *link*. Essa alocação afeta a banda passante disponível nos outros, ainda não engarrafados, *links*. Então, o algoritmo desconsidera o *link* engarrafado e todas as VC's atravessando esse *link*, e continua para a próxima iteração onde um novo *link* é selecionado. Esse processo continua até todas as VC's serem engarrafadas.

Esse algoritmo não é útil em uma rede ATM dinâmica real, porque ele assume um controle centralizado. Contudo, seus conceitos principais são úteis para o projeto de algoritmos distribuídos, para ser executado pelos comutadores ATM e nós de fontes.

Capítulo 4

Descarte Seletivo de Células Sujeito a Processo LRD

O mecanismo de descarte seletivo visa permitir à rede lidar com diversos requisitos de perda. Este mecanismo tem sido estudado nos últimos anos, contudo, a maioria destes estudos consideram processos com dependência de curta duração (SRD - *Short Range Dependence*). Conforme descrito no Capítulo 2 vários tipos de tráfego de rede podem ser modelados com maior precisão pelos processos auto-semelhantes, e estes processos exibem o fenômeno de dependência de longa duração (LRD - *Longe Range Dependence*). Além disso, existe clara evidência que esses processos podem potencialmente causar perdas maciças de células em redes ATM. Demonstrou-se também que uma fila alimentada por um processo LRD sofre do fenômeno de ineficácia de *buffer*, *i.e.* pelo simples incremento do tamanho do *buffer* não é possível diminuir a probabilidade de transbordo consideravelmente.

Neste capítulo investiga-se a eficácia dos mecanismos de descarte seletivo em um multiplexador ATM sujeito a um processo com dependência de longa duração. Considera-se disciplinas que conservam a taxa de perda total, Compartilhamento Total (CT - *Complete Sharing*) e Compartilhamento Total com Garantia de Tamanho Mínimo de Fila (CTGTMF - *Complete Sharing with Guaranteed Queue Minimum*) e diversas políticas de descarte, PCPD, UCPD, RAND e M-PCPD. Analisa-se através de simulação o impacto das características do processo de entrada, tais como: carga do sistema, parâmetro de Hurst e variância, nas taxas de perda por classe e no tamanho do intervalo de perdas da classe de célula com prioridade baixa. Trabalha-se inicialmente com apenas duas classes de prioridade (baixa e alta) e em seguida investiga-se a influência do uso de múltiplas classes de prioridade.

Este capítulo está organizado da seguinte forma: a seção 4.1 descreve o modelo de simulação adotado. Na seção 4.2 mostra-se os resultados de taxa de perda das classes

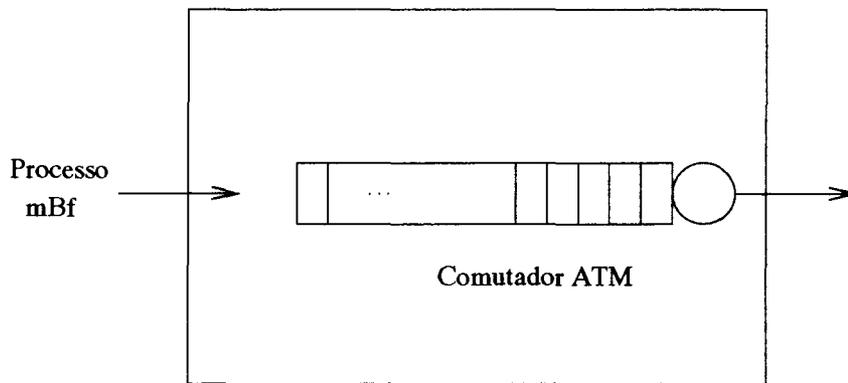


Figura 4.1: Comutador ATM alimentado por um processo LRD.

com baixa e com alta prioridade frente a vários parâmetros de entrada. A seção 4.3 mostra os resultados do tamanho do intervalo de perdas da classe com prioridade baixa. A seção 4.4 analisa o descarte seletivo com múltiplas classes de prioridade.

4.1 Modelo de Simulação

Nas próximas seções mostrar-se-á resultados de simulação de um comutador ATM com mecanismos de descarte seletivo alimentado por um processo de entrada mBf. O Processo de entrada mBf é gerado pelo procedimento descrito na seção 2.6. A Figura 4.1 ilustra o modelo de simulação.

Considera-se neste estudo apenas disciplinas que conservam a taxa de perda total (CT e CTGTMF). As duas métricas principais de desempenho em termos de perda, nas quais os requisitos de QoS de uma aplicação são normalmente traduzidos, serão utilizadas, a saber, a taxa de perda por classe e o tamanho do intervalo de perdas. Investiga-se a influência de parâmetros tais como: carga do sistema, parâmetro de Hurst, variância, tamanho do *buffer*, nível de proteção (nível de garantia de tamanho mínimo de fila) e políticas de descarte nas métricas mencionadas.

Com o objetivo de modelar prioridades atribuídas dinamicamente, considera-se que a probabilidade de uma célula pertencer a uma certa classe de prioridade (probabilidade de prioridade) depende da prioridade da célula anterior no fluxo. Depois de um período sem chegadas, a probabilidade de prioridade da primeira célula a chegar é independente de qualquer outra célula no fluxo. Logo, a “memória” da classificação de prioridade existe entre dois períodos sem chegadas. Assim sendo, define-se:

$P(alta|sem)$ - é a probabilidade de que a primeira célula depois de um período sem chegadas ter prioridade alta.

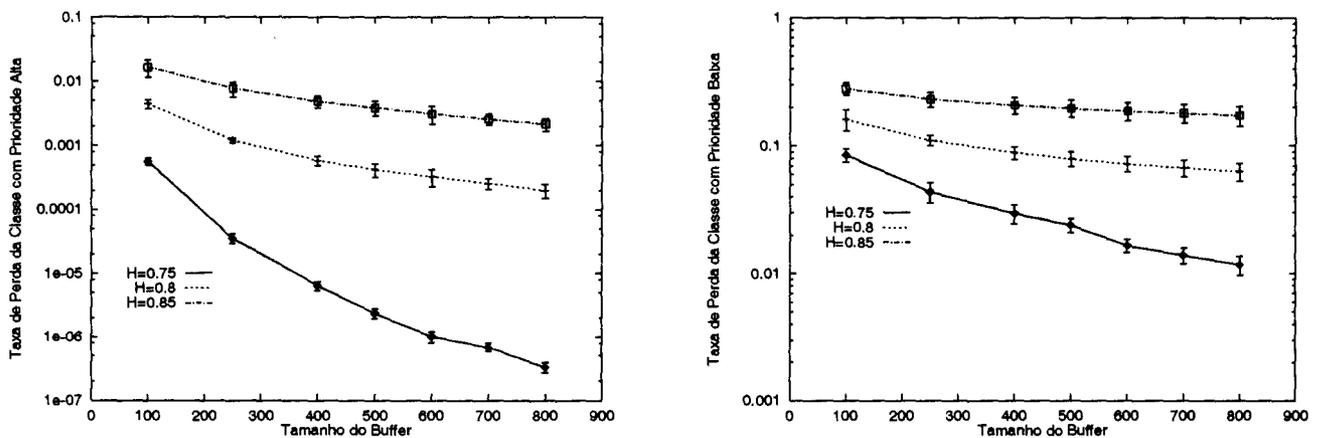


Figura 4.2: Taxa de Perda \times Tamanho do *buffer* para diferentes valores de Parâmetro de Hurst e Fila CT. $\rho = 0.8$, $\sigma^2 = 1$, $P(\text{alta}|\text{alta}) = 0.7$, $\gamma = 0.0$.

$P(\text{alta}|\text{alta})$ - é a probabilidade de uma célula ter prioridade alta, dado que a anterior tem prioridade alta.

$P(\text{baixa}|\text{baixa})$ - é a probabilidade de uma célula ter prioridade baixa, dado que a anterior tem prioridade baixa.

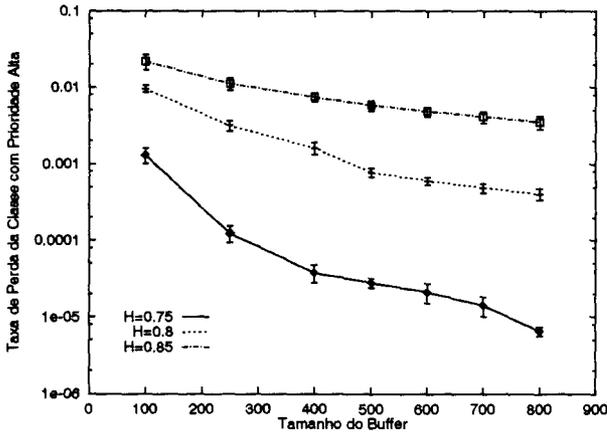
Define-se a medida de correlação γ como $P(\text{alta}|\text{alta}) + P(\text{baixa}|\text{baixa}) - 1$. Se $P(\text{alta}|\text{sem}) = P(\text{alta}|\text{alta}) = 1 - P(\text{baixa}|\text{baixa})$ (e conseqüentemente $\gamma = 0$) tem-se prioridades independentes. Um valor positivo de γ indica que células de pelo menos uma das classes tende a ser aglutinada em rajadas, enquanto um valor negativo de γ mostra que, para pelo menos uma classe, não existe tendência de formação de rajada. O valor máximo de γ ($= 1$) acontece quando $P(\text{alta}|\text{alta}) = P(\text{baixa}|\text{baixa}) = 1$ e corresponde à situação na qual tem-se rajadas inteiras de apenas um nível de prioridade.

Os resultados apresentados a seguir foram obtidos através de simulação. O método de replicação independente foi usado para gerar intervalos com 95% de nível de confiança. O tamanho da amostra usada para calcular cada ponto das curvas foi tal que a largura dos intervalos de confiança é menor do que 5% da média. Em todas as figuras são mostrados os respectivos intervalos de confiança.

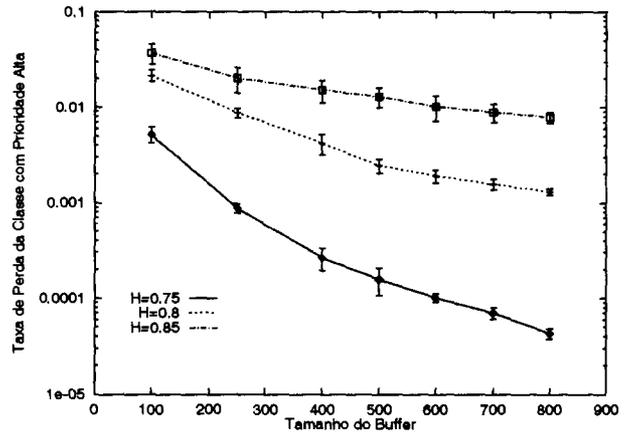
4.2 Taxa de Perda por Classe de Prioridade

▷ Influência do tamanho do *buffer*

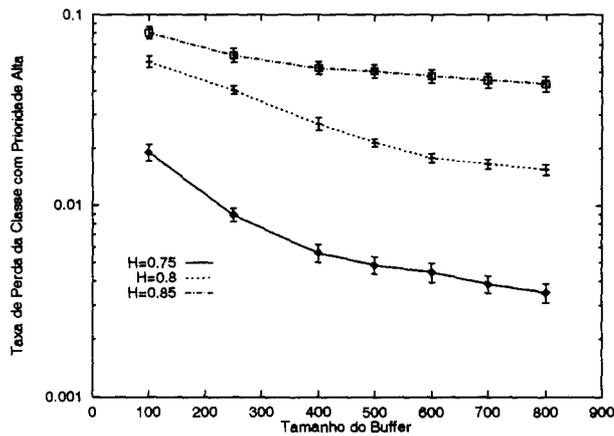
A Figura 4.2 mostra a taxa de perda como uma função do tamanho do *buffer* para valores diferentes de parâmetro de Hurst (parâmetro H) considerando a fila CT com



(CTGTMF - 10%)

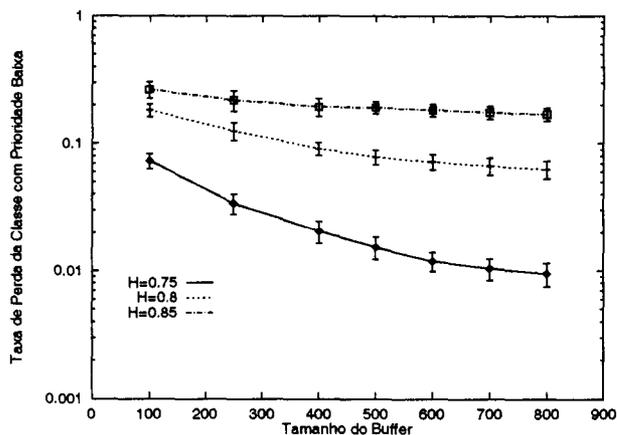


(CTGTMF - 20%)

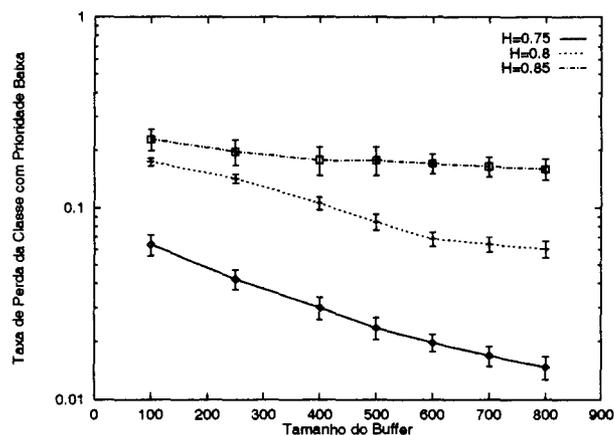


(CTGTMF - 30%)

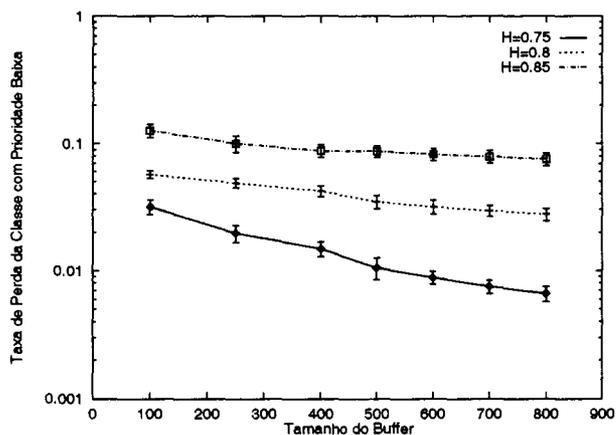
Figura 4.3: Taxa de Perda da Classe com Prioridade Alta x Tamanho do *buffer* para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção. $\rho = 0.8$, $\sigma^2 = 1$, $P(alta|alta) = 0.7$, $\gamma = 0.0$.



(CTGTMF - 10%)



(CTGTMF - 20%)



(CTGTMF - 30%)

Figura 4.4: Taxa de Perda da Classe com Prioridade Baixa x Tamanho do *buffer* para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção. $\rho = 0.8$, $\sigma^2 = 1$, $P(alta|alta) = 0.7$, $\gamma = 0.0$.

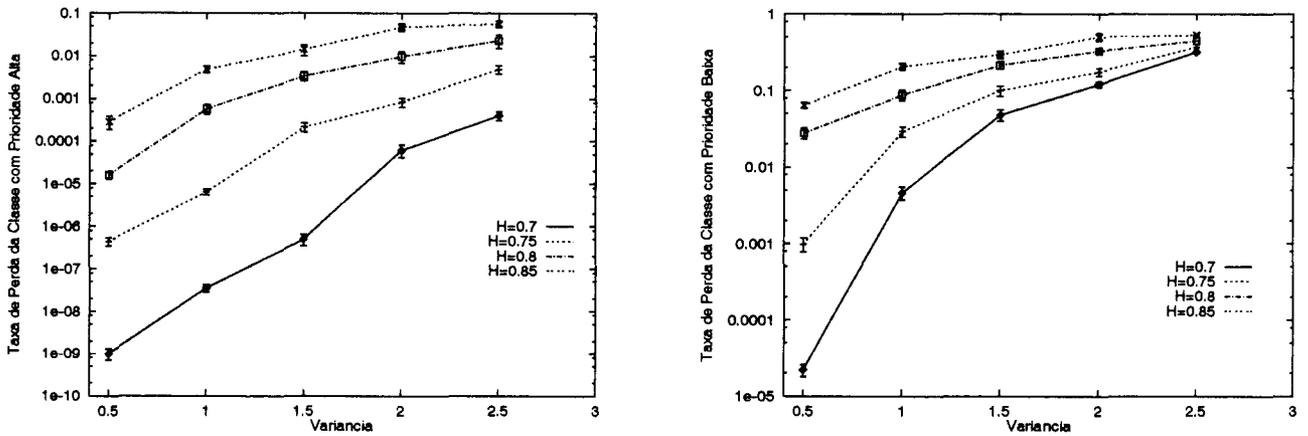


Figura 4.5: Taxa de Perda x Variância para diferentes valores de Parâmetro de Hurst e Fila C'T. Tamanho do *buffer* = 400. $\rho = 0.8$. $P(alta|alta) = 0.7$. $\gamma = 0.0$.

política de descarte PCPD (assume-se PCPD como a política de descarte para todos os exemplos desta seção, exceto quando explicitamente mencionado ao contrário). A taxa de chegada média (ρ), a variância (σ^2), $P(alta|alta)$ e γ para essa figura são 0.8, 1.0, 0.7 e 0.0, respectivamente (note que quando $\gamma = 0.0$ tem-se $P(alta|alta) = P(alta|scm) = 1 - P(baixa|baixa)$). Nota-se que para valores altos do parâmetro de Hurst ($H > 0.8$) a taxa de perda da classe com prioridade alta é quase insensível ao tamanho do *buffer*. Portanto, introduzir descarte seletivo tem um impacto mínimo para fluxos com parâmetro de Hurst 0.8. Para valores baixos de H , a taxa de perda da classe com prioridade alta decresce quando se aumenta o tamanho do *buffer*. Por exemplo, pode-se reduzir a taxa de perdas nesse exemplo específico, de uma ordem de grandeza a cada 200 unidades de espaço de *buffer*. A taxa de perda da classe com prioridade baixa está na ordem da taxa de perda total.

As Figuras 4.3 e 4.4 mostram a taxa de perda da classe com prioridade alta e a taxa de perda da classe com prioridade baixa para a disciplina C'TGTMF. Nota-se que, para um nível de proteção de 10% (do tamanho total do *buffer*), os resultados para uma disciplina C'TGTMF são praticamente os mesmos mostrados pela C'T. Para um nível de proteção de 20%, ainda pode-se oferecer serviços diferenciados. Contudo, para um nível de proteção de 30%, classes com prioridade alta e baixa têm quase a mesma taxa de perda, ou seja, elimina-se completamente a vantagem de descarte seletivo. Além disso, ao se garantir um espaço de *buffer* mínimo para a classe com prioridade baixa, sua taxa de perda diminui claramente. Em outras palavras, sob um processo LRD, uma fila C'TGTMF aumenta a taxa de perda da classe com prioridade alta e não diminui significativamente a taxa de perda da classe com prioridade baixa.

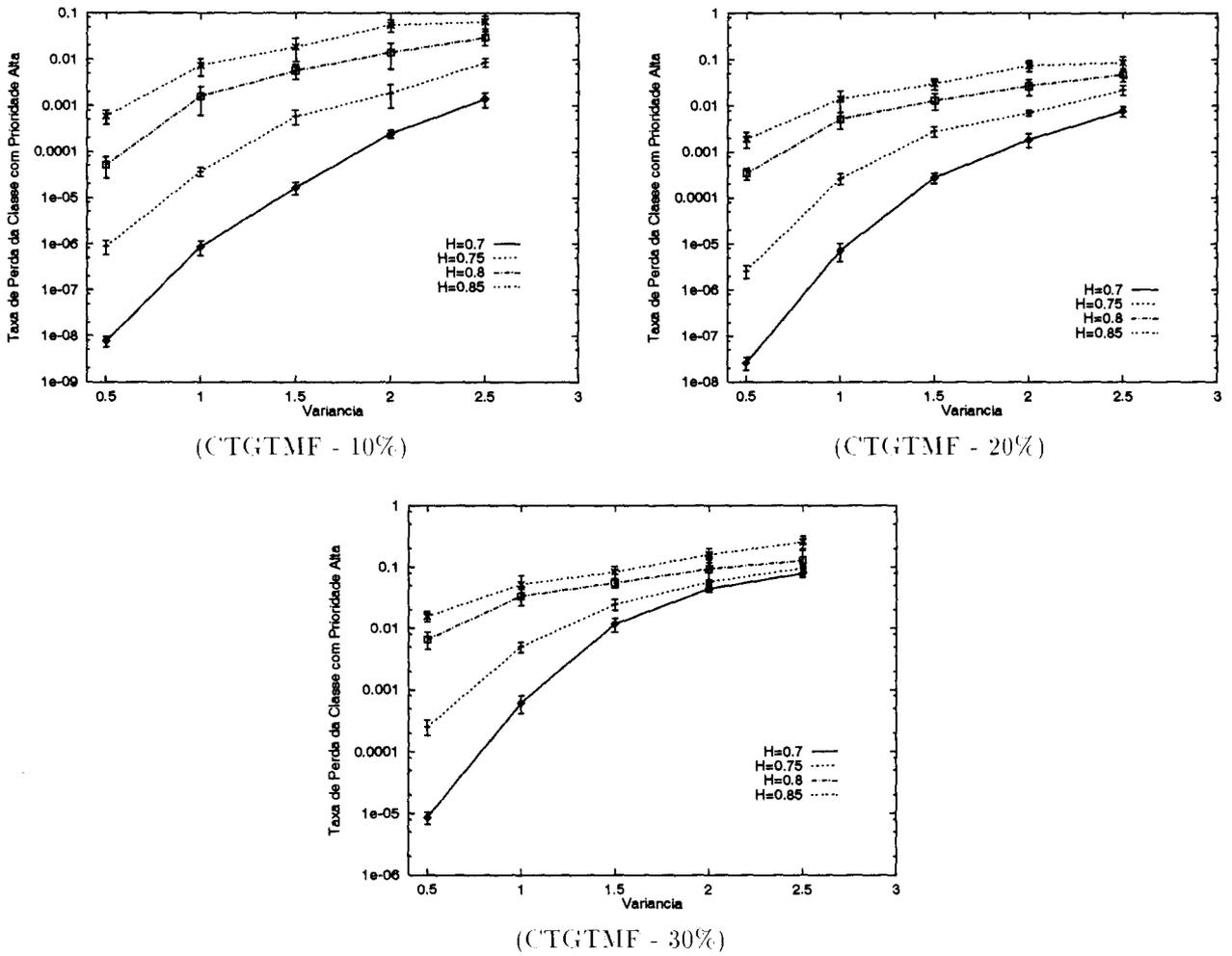
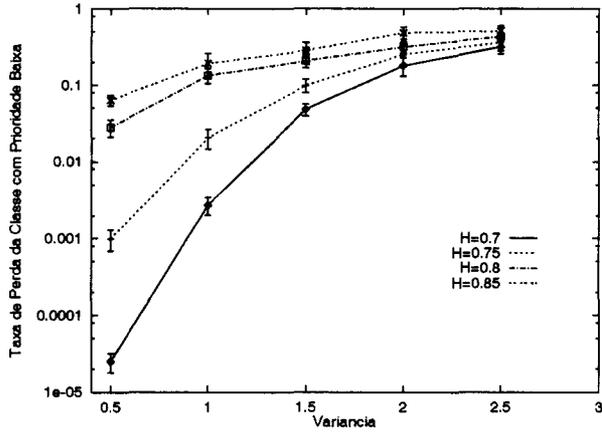


Figura 4.6: Taxa de Perda da Classe com Prioridade Alta x Variância para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção. Tamanho do *buffer* = 400. $\rho = 0.8, P(alta|alta) = 0.7, \gamma = 0.0$.

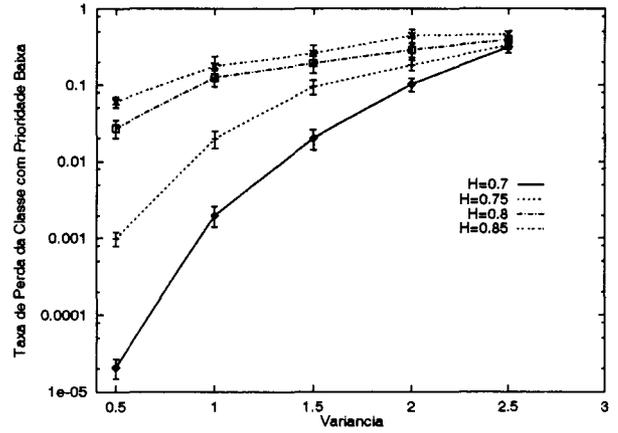
▷ **Influência da variância**

Na Figura 4.5 analisa-se a taxa de perda em uma fila CT como função da variância para diferentes valores do parâmetro de Hurst. Observa-se que ao se aumentar a variância, aumenta-se também a taxa de perda de células da classe com prioridade alta independentemente do valor de H , um vez que a probabilidade de transbordo do *buffer* (dada pela equação 2.5) aumenta com a variância. Nota-se que é possível oferecer um serviço seletivo para valores altos de variância somente se estes tiverem valores baixos de parâmetro de Hurst.

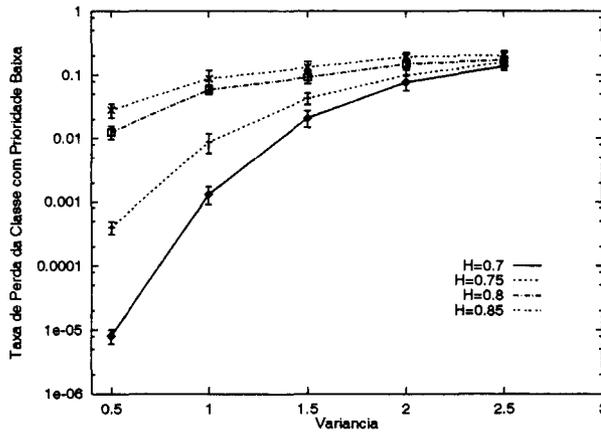
As Figuras 4.6 e 4.7 mostram respectivamente a taxa de perda da classe com prioridade alta e a taxa de perda da classe com prioridade baixa para disciplina CTGTMF. Observa-



(CTGTMF - 10%)



(CTGTMF - 20%)



(CTGTMF - 30%)

Figura 4.7: Taxa de Perda da Classe com Prioridade Baixa x Variância para diferentes valores de Parâmetro de Hurst e Fila CTGTMF com diferentes níveis de proteção. Tamanho do *buffer* = 400, $\rho = 0.8$, $P(alta|alta) = 0.7$, $\gamma = 0.0$.

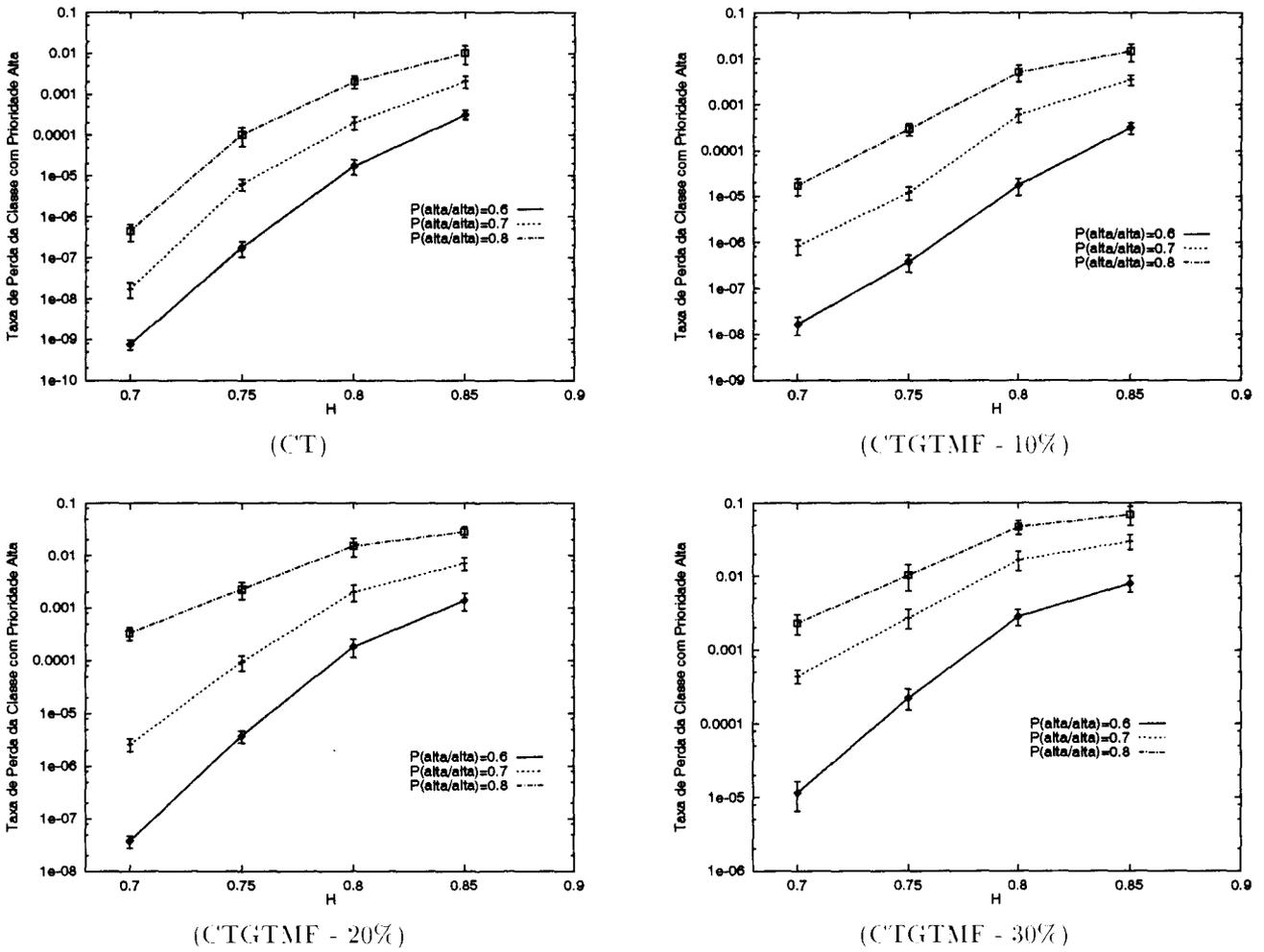
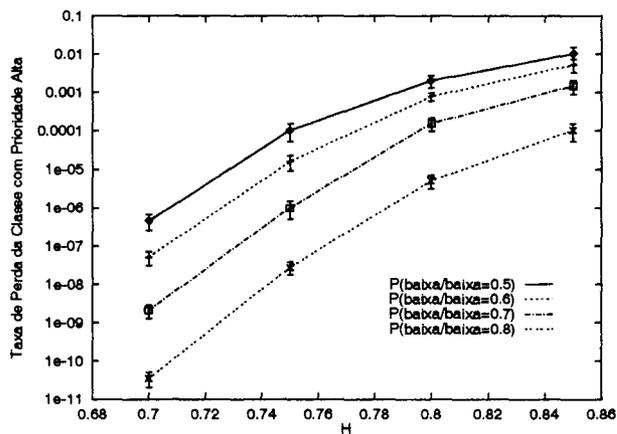
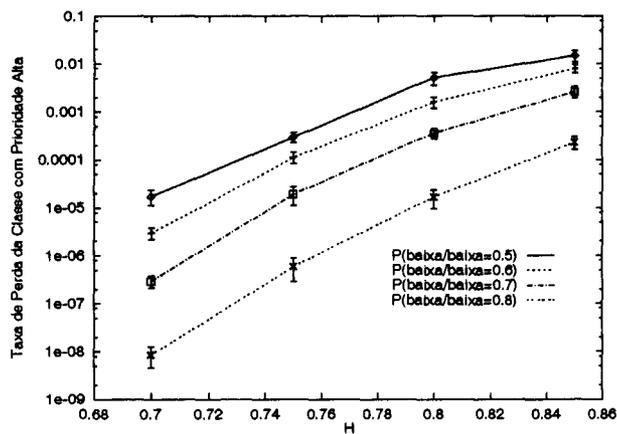


Figura 4.8: Taxa de Perda da Classe com Prioridade Alta x Parâmetro de Hurst para diferentes valores de $P(\text{alta}|\text{alta})$. $Buffer = 250$, $\rho = 0.8$, $\sigma^2 = 1$, $P(\text{baixa}|\text{baixa}) = 0.5$.

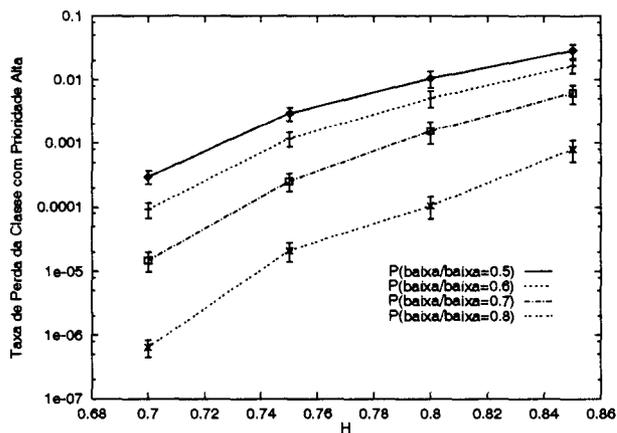
se que, para um nível de proteção de 10%, a CTGTMF produz resultados similares à CT. Ao se aumentar o nível de proteção para 20%, para fluxos com valores altos de variância, quase não há diferença entre a taxa de perda da classe com prioridade alta e a taxa de perda da classe com prioridade baixa, especialmente para valores altos do parâmetro de Hurst. Para valores baixos de H , a diferença é de duas ordens de grandeza. Para um nível de proteção de 30%, as duas taxas de perda têm quase o mesmo valor. Além disso, para a classe com prioridade baixa, o nível de proteção não tem influência na taxa de perda para valores altos de variância. A falta de sensibilidade para o nível de proteção pode ser explicado pelo fato de a maioria das perdas da classe com prioridade baixa poderem ocorrer em longas rajadas.



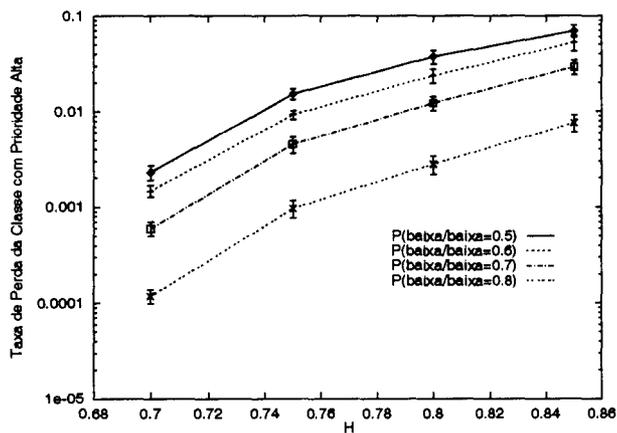
(CT)



(CTGTMF - 10%)



(CTGTMF - 20%)



(CTGTMF - 30%)

Figura 4.9: Taxa de Perda da Classe com Prioridade Alta x Parâmetro de Hurst para diferentes valores de $P(\text{baixa}-\text{baixa})$. $\text{Buffer} = 250$, $\rho = 0.8$, $\sigma^2 = 1$, $P(\text{alta}|\text{alta}) = 0.8$.

▷ Influência do parâmetro H

Nas Figuras 4.8 e 4.9, a relação entre o parâmetro de Hurst e a correlação de prioridade de um fluxo é mostrada. Na Figura 4.8 considera-se diferentes valores de γ pela variação de $P(\text{alta}|\text{alta})$. Observa-se que a taxa de perda da classe com prioridade alta para valores altos de $P(\text{alta}|\text{alta})$ é sempre maior do que para valores baixos de $P(\text{alta}|\text{alta})$ porque para valores altos de $P(\text{alta}|\text{alta})$ tem-se longas rajadas de células da classe com prioridade alta e, conseqüentemente, tem-se longas rajadas com prioridade alta em situações de transbordo.

Ao se aumentar o parâmetro H , aumenta-se também a taxa de perda total e a diferença entre a taxa de perda dada por um valor alto de $P(\text{alta}|\text{alta})$ e por um valor baixo de $P(\text{alta}|\text{alta})$ diminui. Para valores altos do parâmetro de Hurst ($H = 0.85$), a diferença na taxa de perda da classe com prioridade alta para valores de $P(\text{alta}|\text{alta})$ de 0.8 e 0.6 pode ser de até duas ordens de grandeza. Com aumento do nível de proteção da política CTGTMF essa diferença diminui, chegando a uma ordem de grandeza para um nível de proteção de 30%. Observa-se, por exemplo, que para $P(\text{alta}|\text{alta}) = 0.6$ e $H = 0.7$ os valores de taxa de perda da classe com prioridade alta para os níveis de proteção de 10%, 20% e 30% aumentam respectivamente de $9\epsilon-09$ para $1\epsilon-07$, $7\epsilon-07$ e $1\epsilon-04$.

Na Figura 4.9, varia-se γ alterando-se $P(\text{baixa}|\text{baixa})$. Novamente, ao se aumentar H aumenta-se a taxa de perda total e, conseqüentemente, a taxa de perda da classe com prioridade alta. Para valores altos de $P(\text{baixa}|\text{baixa})$ tem-se valores baixos de taxa de perda da classe com prioridade baixa uma vez que se aumenta a probabilidade de uma célula com prioridade alta encontrem longas rajadas de células com prioridade baixa na fila em situações de transbordo. A diferença na taxa de perda da classe com prioridade alta dada por um valor $P(\text{baixa}|\text{baixa})$ de 0.8 e por um valor de $P(\text{baixa}|\text{baixa})$ de 0.5 pode ser de quatro ordens de grandeza. Comportamento semelhante foi observado para CTGTMF, exceto que ao se aumentar o nível de proteção, aumenta-se também a taxa de perda da classe com prioridade alta. Por exemplo, para $P(\text{baixa}|\text{baixa}) = 0.8$ e $H = 0.7$, os valores de taxa de perda de alta prioridade para os níveis de proteção 10%, 20% e 30% são respectivamente $9\epsilon-08$, $8\epsilon-06$ e $1\epsilon-03$.

▷ Influência das políticas de descartes

Para se avaliar o impacto da introdução de diferentes políticas de descarte no fenômeno de perda compara-se as disciplinas UCPD, PCPD, RAND e PCPD-M. Conforme mencionado na seção 3.3, distintas distribuições de fila produzidas por diferentes políticas de descarte podem influenciar principalmente a taxa de perda da classe com prioridade alta e o tamanho do intervalo de perdas da classe com prioridade baixa. Inicialmente discute-se a relação entre as políticas de descarte e as taxas de perda da classe com prioridade alta.

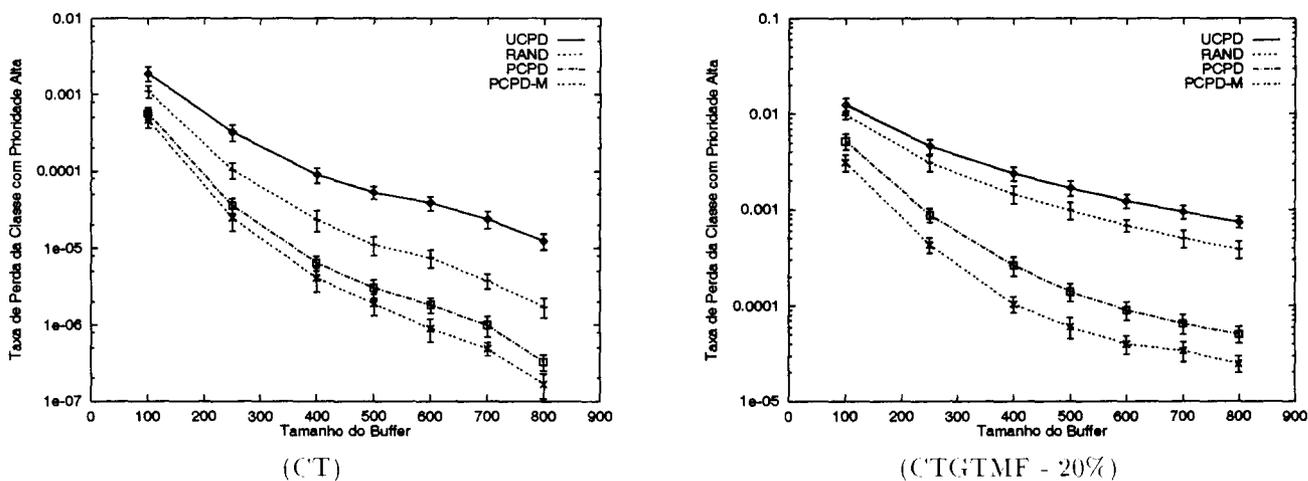


Figura 4.10: Taxa de Perda da Classe com Prioridade Alta x Tamanho do *buffer* para diferentes Políticas de Descarte. $H = 0.75$, $\rho = 0.8$, $\sigma^2 = 1$, $P(alta|alta) = 0.7$, $\gamma = 0.0$.

O gráfico da Figura 4.10 retrata as taxas de perda da classe com prioridade alta como função do tamanho do *buffer*. Nota-se a evidente diferença na taxa de perda da classe com prioridade alta produzida por políticas de descarte distintas. A diferença entre os resultados gerados pela UCPD e os gerados pela PCPD-M pode alcançar até três ordens de grandeza para tamanhos de *buffer* grandes e pode ser de até uma ordem de grandeza para tamanhos de *buffer* pequenos. Observa-se, também, que os resultados gerados pelas políticas PCPD e PCPD-M são semelhantes. Mostra-se ainda na Figura 4.10 resultados para CTGTMF com um nível de proteção de 20%. É possível reduzir a taxa de perda de alta prioridade de uma ordem de grandeza ao se usar a política PCPD-M. Essa vantagem desaparece com altos graus de proteção.

Ao se investigar a dependência de taxa de perda da classe com prioridade alta em relação a variância, verifica-se que, para valores baixos de variância, é possível obter uma diferença de uma ordem de grandeza. Contudo, isso quase não ocorre para valores altos de variância. Observa-se, por exemplo, que para um nível de proteção de 20% da política CTGTMF, tamanho do *buffer* = 400 e parâmetro de Hurst = 0.75 (ver Figura 4.11) tem-se um aumento da taxa de perda da classe com prioridade alta em todas as políticas de descarte de no máximo uma ordem de grandeza em relação à política CT.

A Figura 4.12 indica que para rajadas curtas de células de alta prioridade ($P(alta|alta) = 0.6$) é possível obter uma diferença de duas ordens de grandeza, enquanto para longas rajadas ($P(alta|alta) = 0.8$), não há diferença significativa nos resultados produzidos por essas políticas. Isso é devido ao fato de que para rajadas longas de células com prioridade alta perde-se uma alta porcentagem de células com prioridade alta em situações de transbordo. Observa-se, também, que a diferença entre as taxas de perda da classe com

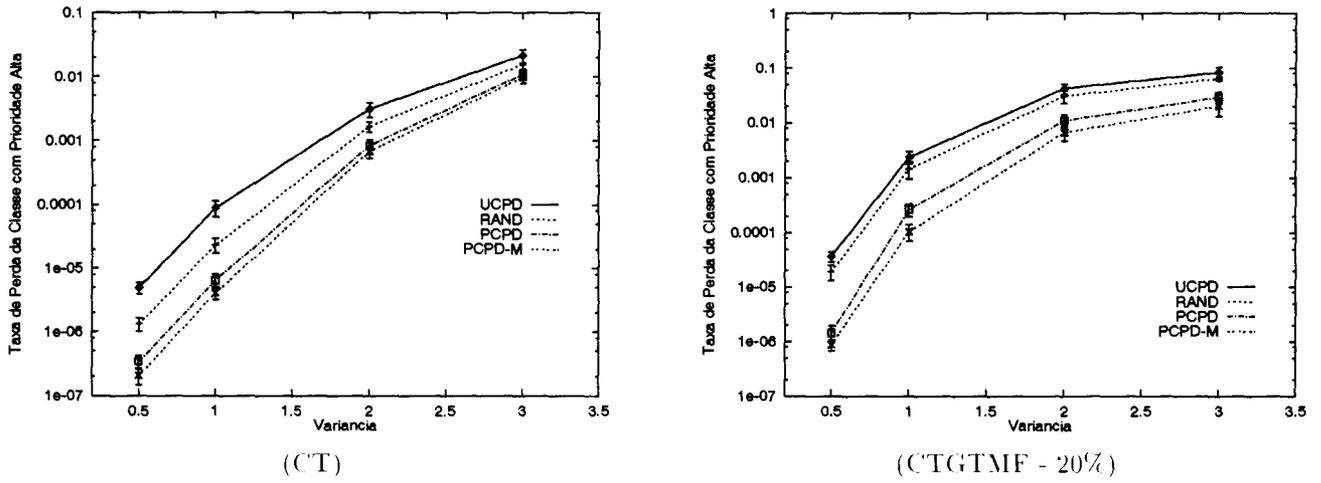


Figura 4.11: Taxa de Perda da Classe com Prioridade Alta x Variância para diferentes Políticas de Descarte. $H = 0.75$, $\rho = 0.8$. Tamanho do *buffer*=400. $P(alta|alta) = 0.7$. $\gamma = 0.0$.

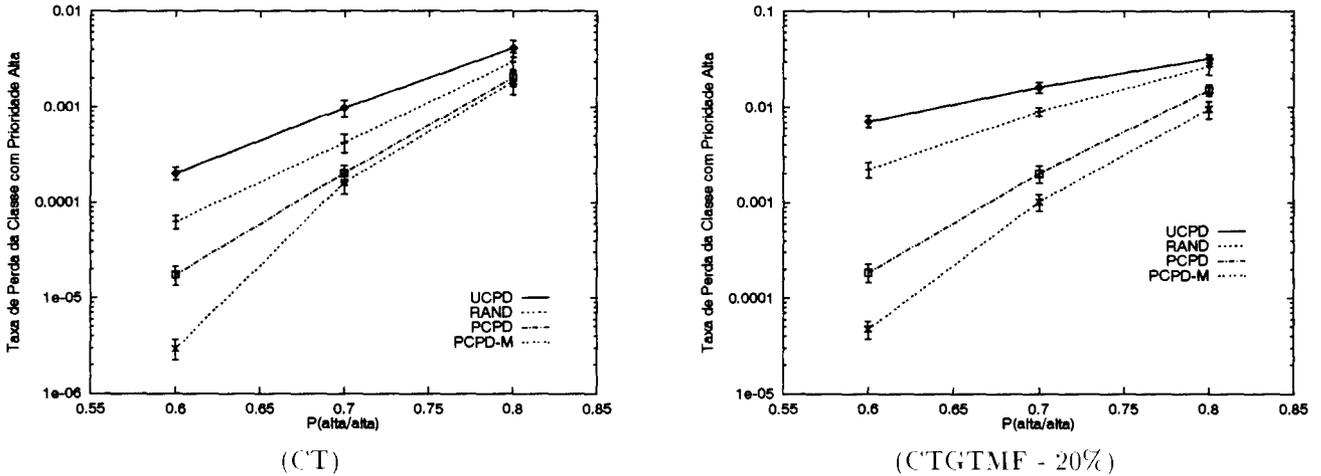


Figura 4.12: Taxa de Perda da Classe com Prioridade Alta x $P(alta|alta)$ para diferentes Políticas de Descarte. Tamanho do *buffer* = 400, $H = 0.8$, $\rho = 0.8$, $\sigma^2 = 1$. $P(baixa|baixa) = 0.5$.

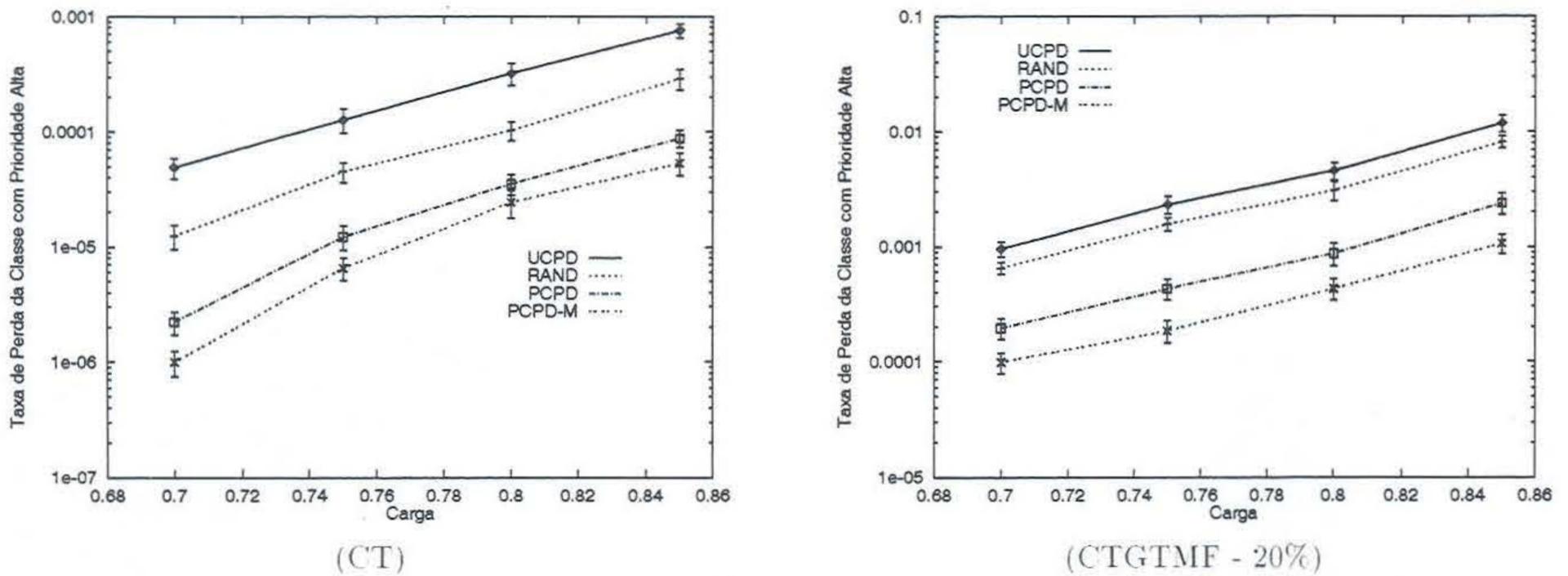


Figura 4.13: Taxa de Perda da Classe com Prioridade Alta x Carga Fornecida para diferentes Políticas de Descarte. Tamanho do *buffer* = 250. $H = 0.75$. $\sigma^2 = 1$. $P(\text{alta}|\text{alta}) = 0.7$. $\gamma = 0.0$.

prioridade alta produzidas por essas políticas é quase constante ao se variar a carga (*offered load*), mantendo constante os demais parâmetros de tráfego (Figura 4.13). Em resumo, pode-se dizer que $P_{\text{alta}}(\text{UCPD}) > P_{\text{alta}}(\text{RAND}) > P_{\text{alta}}(\text{PCPD}) > P_{\text{alta}}(\text{PCPD} - M)$ onde P_{alta} é a taxa de perda da classe com prioridade alta e que PCPD e PCPD-M produzem resultados semelhantes.

4.3 Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa

Apesar de útil, a taxa de perda é um valor médio que não descreve por completo o processo de perda. Por outro lado, o tamanho do intervalo de perdas fornece informações mais detalhadas sobre o processo de perda. A utilização do tamanho médio do intervalo de perdas para comparar diferentes políticas não é o índice mais apropriado dado que a maioria dos intervalos de perda são de tamanho pequeno, o que faz com que o valor médio esconda informações relevantes. Portanto, a comparação entre as políticas é baseada nos *traces* produzidos pelos experimentos de simulação. Nas figuras a seguir, mostram-se os *traces* associados com uma replicação do experimento de simulação. Escolheu-se experimentos que melhor representassem o comportamento médio observado entre todas as replicações.

A Figura 4.14 mostra as distribuições UCPD, RAND, PCPD e PCPD-M para um tamanho de *buffer* de 100. Nota-se que o tamanho máximo do intervalo de perda da

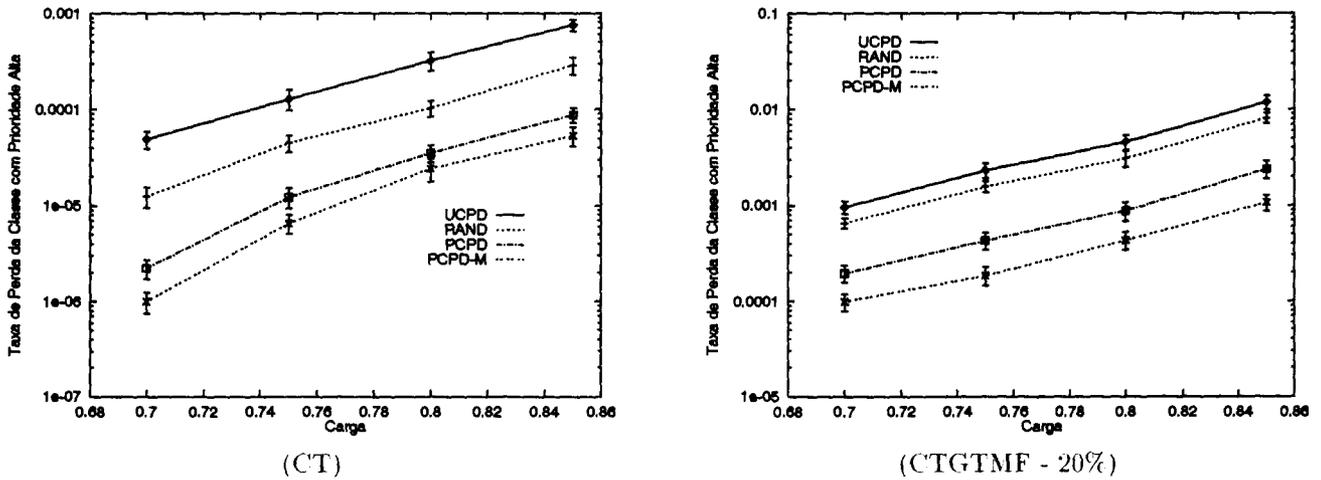


Figura 4.13: Taxa de Perda da Classe com Prioridade Alta x Carga Fornecida para diferentes Políticas de Descarte. Tamanho do *buffer* = 250. $H = 0.75$. $\sigma^2 = 1$. $P(\text{alta}|\text{alta}) = 0.7$. $\gamma = 0.0$.

prioridade alta produzidas por essas políticas é quase constante ao se variar a carga (*offered load*), mantendo constante os demais parâmetros de tráfego (Figura 4.13). Em resumo, pode-se dizer que $P_{\text{alta}}(\text{UCPD}) > P_{\text{alta}}(\text{RAND}) > P_{\text{alta}}(\text{PCPD}) > P_{\text{alta}}(\text{PCPD} - M)$ onde P_{alta} é a taxa de perda da classe com prioridade alta e que PCPD e PCPD-M produzem resultados semelhantes.

4.3 Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa

Apesar de útil, a taxa de perda é um valor médio que não descreve por completo o processo de perda. Por outro lado, o tamanho do intervalo de perdas fornece informações mais detalhadas sobre o processo de perda. A utilização do tamanho médio do intervalo de perdas para comparar diferentes políticas não é o índice mais apropriado dado que a maioria dos intervalos de perda são de tamanho pequeno, o que faz com que o valor médio esconda informações relevantes. Portanto, a comparação entre as políticas é baseada nos *traces* produzidos pelos experimentos de simulação. Nas figuras a seguir, mostram-se os *traces* associados com uma replicação do experimento de simulação. Escolheu-se experimentos que melhor representassem o comportamento médio observado entre todas as replicações.

A Figura 4.14 mostra as distribuições UCPD, RAND, PCPD e PCPD-M para um tamanho de *buffer* de 100. Nota-se que o tamanho máximo do intervalo de perda da

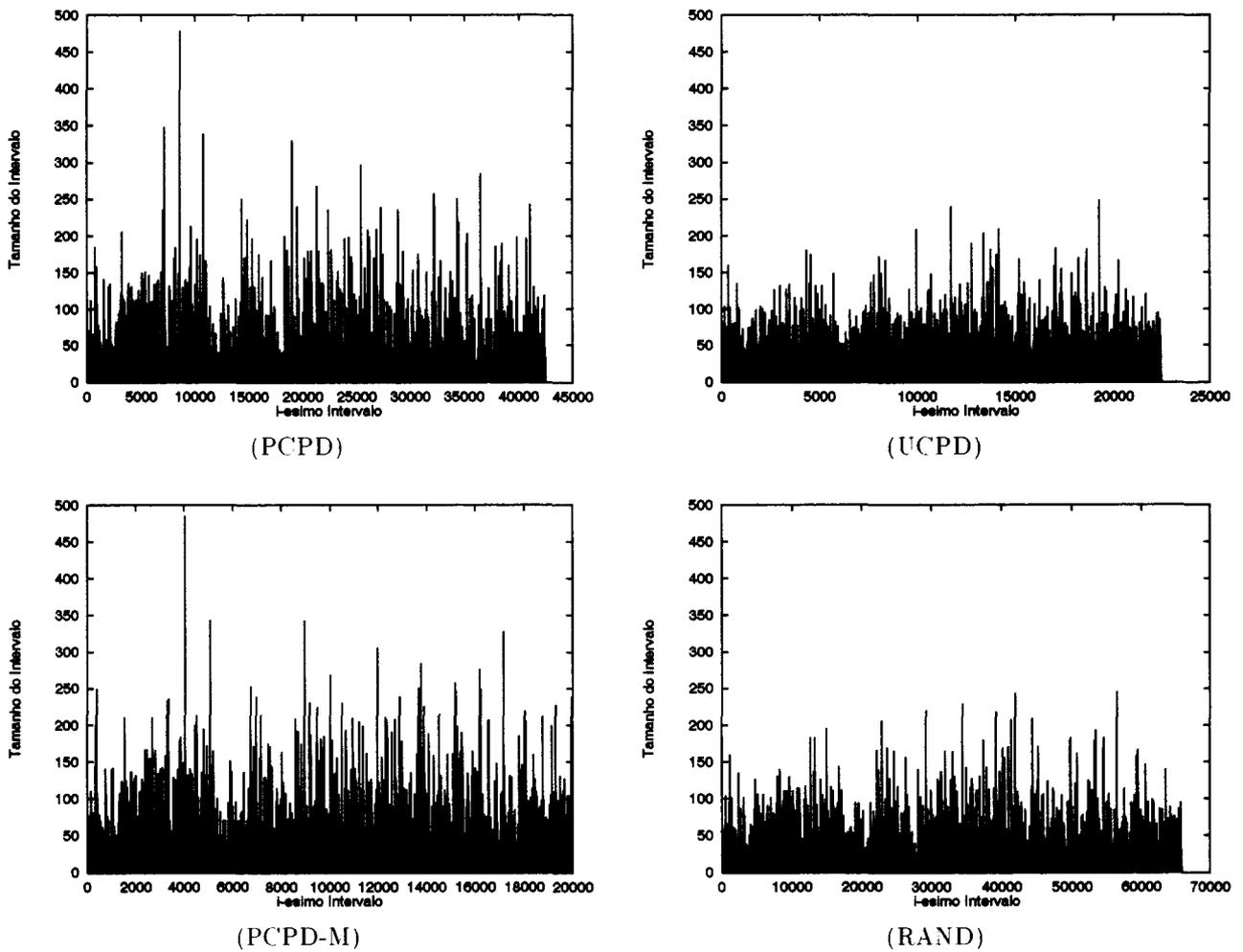


Figura 4.14: Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa \times i -ésimo Intervalo para diferentes Políticas de Descarte, Fila CT, Tamanho do $buffer=100$, $\rho=0.8$, $\sigma^2=1$, $H=0.75$, $P(alta|alta)=0.7$, $\gamma=0.0$.

política PCPD e da política PCPD-M são duas vezes o valor máximo da política UCPD. O valor máximo da política RAND aproxima-se do valor máximo da política UCPD. Observa-se que pelo aumento do tamanho do $buffer$ a diferença entre os valores de tamanho máximo diminui (ver tabela 4.1). A política PCPD-M sempre produz o valor máximo mais alto de intervalo de perdas seguido de perto pela política PCPD e depois pela política RAND e finalmente pela política UCPD. Por exemplo, para um tamanho de $buffer$ de 800 o tamanho máximo de intervalo de perda para política UCPD, RAND, PCPD e PCPD-M são 730, 750, 870 e 990, respectivamente.

A política UCPD sempre produz o valor máximo mais baixo de intervalo de perda porque uma eventual transmissão de uma célula perto do início da fila quebra um in-

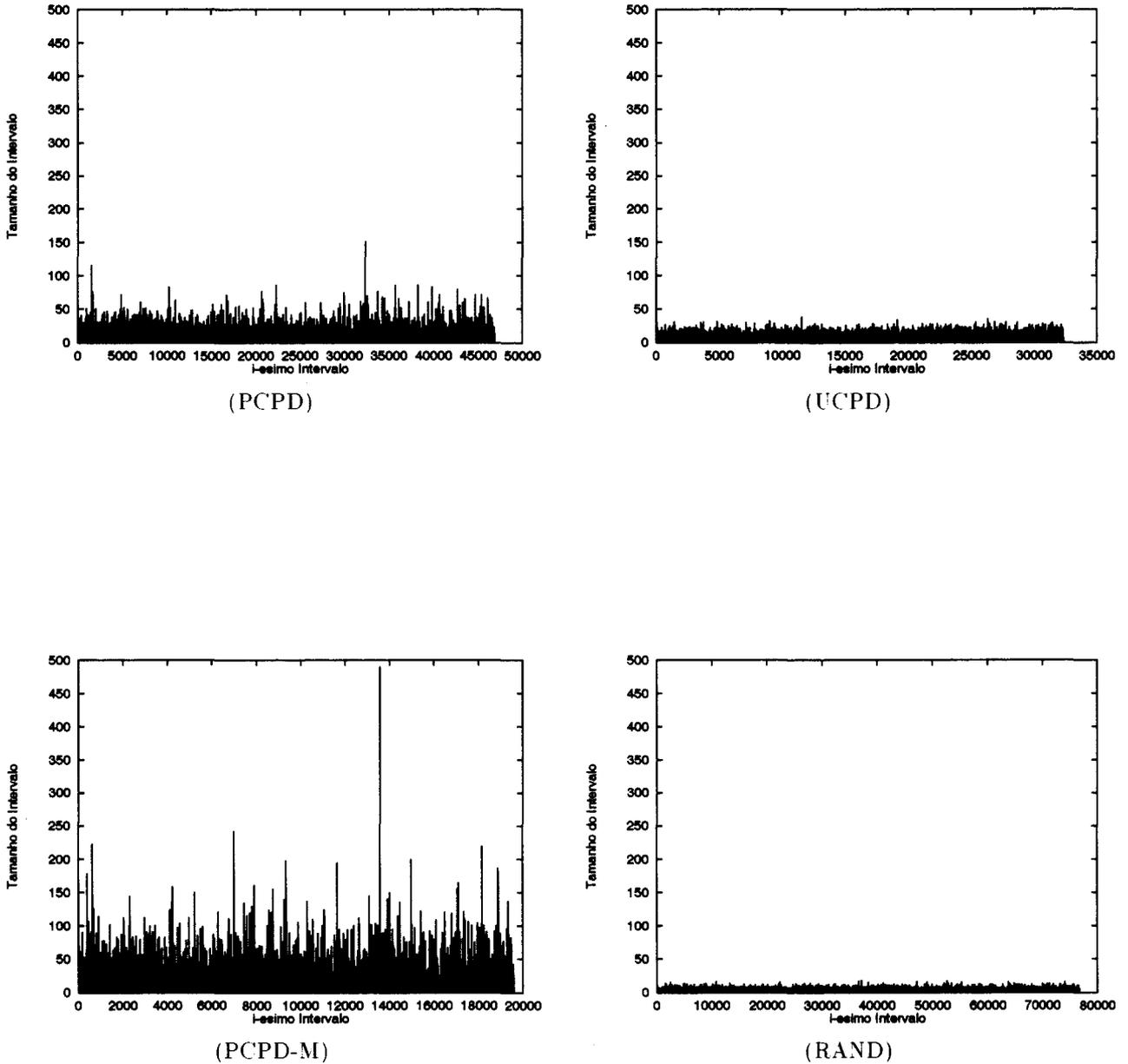


Figura 4.15: Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa x i -ésimo Intervalo para diferentes Políticas de Descarte, Tamanho do $buffer = 100$, $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.75$, $P(alta|alta) = 0.7$, $\gamma = 0.0$.

Buffer	PCPD	UCPD	RAND	PCPD-M
100	478	245	249	485
250	532	430	450	545
400	556	450	494	562
500	562	459	501	575
600	581	468	510	574
700	824	687	736	950
800	870	730	750	990

Tabela 4.1: Valores Máximos de Tamanho de Intervalos para fila CT. $H = 0.75$, $P(alta|alta) = 0.7$, $\gamma = 0.0$, $\sigma^2 = 1$, $\rho = 0.8$.

tervalo longo de perda. Por outro lado, a política PCPD-M sempre tenta concentrar as células de baixa prioridade no final da fila aumentando sua chance de ser descartada e, conseqüentemente, produz o valor máximo mais alto de intervalo de perda. A Figura 4.15 mostra o mesmo cenário para CTGTMF com 20% de proteção de *buffer*. Nota-se que garantindo uma certa percentagem do espaço de *buffer* para células da classe com prioridade baixa, o tamanho máximo do intervalo de perdas diminui para todas as políticas exceto para PCPD-M. Com o aumento do tamanho do *buffer*, o tamanho máximo do intervalo de perdas também aumenta, contudo, para valores mais baixos quando comparados com os resultados da política CT. Por exemplo, o tamanho máximo com um tamanho de *buffer* de 800 para UCPD, RAND, PCPD e PCPD-M são 130, 150, 400 e 500, respectivamente.

$P(alta alta)$	PCPD	UCPD	RAND	PCPD-M
0.6	1381	731	1071	1402
0.7	1569	957	1232	1748
0.8	1639	970	1428	2003

Tabela 4.2: Valores Máximos de Tamanho de Intervalos para fila CT. $H = 0.8$, $P(baixa|baixa) = 0.5$, Tamanho do *buffer* = 250, $\sigma^2 = 1$, $\rho = 0.8$.

Na tabela 4.2 mostra-se o comprimento do maior intervalo de perdas das diversas políticas de descarte pelo aumento de $P(alta|alta)$. Ou seja, analisa-se diferentes valores de γ (correlação de prioridade) pelo aumento da variação de $P(alta|alta)$. Com aumento de $P(alta|alta)$ tem-se um aumento do tamanho do intervalo de perdas da classe com prioridade baixa. Esses resultados justificam-se pelo fato de ao aumentar-se $P(alta|alta)$ tem-se longas rajadas de células da classe com prioridade alta em situações de transbordo,

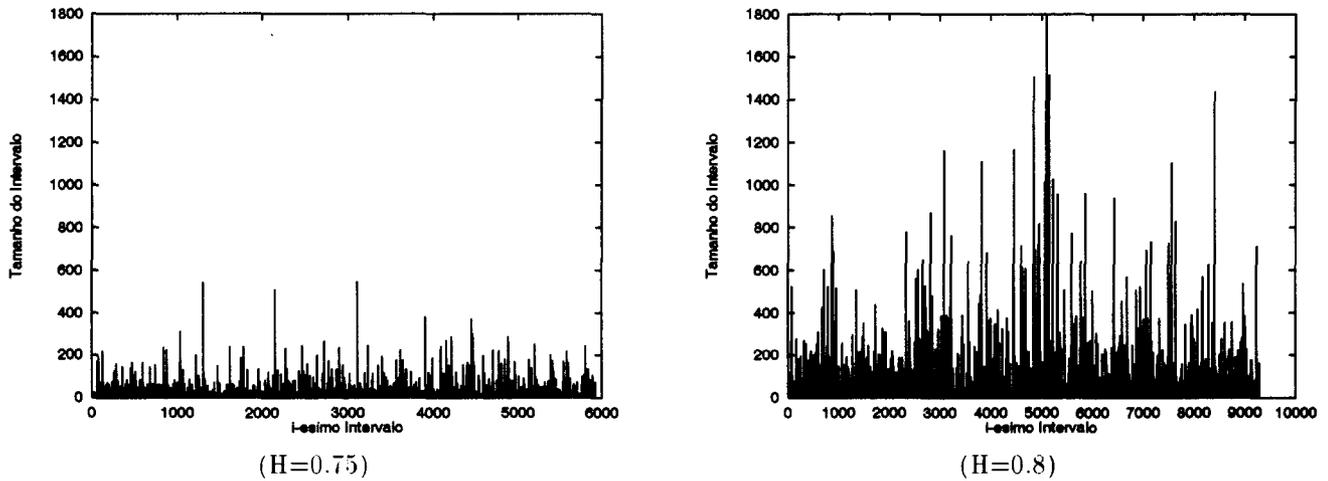


Figura 4.16: Tamanho do Intervalo de Perdas da Classe com Prioridade Baixa x Carga Fornecida. Tamanho do *buffer* = 400, $\sigma^2 = 1$, $\rho = 0.8$, $P(\text{alta}|\text{alta}) = 0.7$, $\gamma = 0.0$.

e assim mais células da classe com prioridade baixa são descartadas.

Observa-se que o tamanho do intervalo de perdas da classe com prioridade baixa é altamente sensível ao parâmetro de Hurst. A Figura 4.16 mostra, a distribuição associada à política PCPD-M para $H = 0.75$ e $H = 0.8$. Observa-se que o valor máximo fornecido pela política PCPD (~ 1800) e pela política PCPD-M (1800) para $H = 0.8$ são quase três vezes o valor fornecido para um parâmetro de Hurst igual a $H = 0.75$ (~ 600). Para as políticas UCPD e RAND, o tamanho máximo do intervalo e perdas para $H = 0.8$ (~ 500) é quase duas vezes o valor máximo fornecido com parâmetro de Hurst igual a $H = 0.75$ (~ 250).

A carga obviamente afeta o tamanho do intervalo de perdas da classe com baixa prioridade. No entanto, é possível reduzir o valor máximo do tamanho do intervalo de perdas ao se aumentar o nível de proteção em filas CTGTMF. Como por exemplo, em um dos experimentos o tamanho máximo para as políticas UCPD, RAND, PCPD, PCPD-M com uma carga de 0.75 são respectivamente 50, 70, 140 e 250 e para uma carga de 0.85 estes valores são 70, 90, 400 e 500. A tabela 4.3 mostra outros exemplos de valores máximos de tamanho de rajada considerando a política CT e $H = 0.8$.

O tamanho do intervalo de perdas também é bastante sensível à variância do processo de entrada. Como por exemplo, em um dos experimentos encontrou-se para um fila CT e um valor de variância de 1.0 o tamanho máximo gerado pelas políticas UCPD, RAND, PCPD e PCPD-M são respectivamente iguais a 100, 120, 500 e 1000, enquanto que para um valor de variância de 2.5 estes valores são, respectivamente, 4500, 4700, 5300 e 9000. Em uma fila CTGTMF, e um valor de variância de 1.0 esses valores são 100, 120, 470 e 950, enquanto que para um valor de variância de 2.5 estes valores são 150, 190, 14004

Carga	PCPD	UCPD	RAND	PCPD-M
0.7	1077	627	715	1130
0.75	1808	632	732	1916
0.8	2134	1123	1457	2179
0.85	2484	1614	2160	2445

Tabela 4.3: Valores Máximos de Tamanho de Intervalo de perdas para fila CT, $H=0.8$, $P(alta|alta) = 0.7$, $\gamma = 0.0$, Tamanho do *Buffer* = 250, $\sigma^2 = 1$.

e 5000. Esses resultados indicam claramente os benefícios de proteção de *buffer* para a classe de baixa prioridade. A tabela 4.4 exibe outro conjunto de valores para política CT e $H = 0.8$.

Variância	PCPD	UCPD	RAND	PCPD-M
1.0	1770	1061	1572	1799
2.0	4108	2782	2696	5282
3.0	5407	4840	4845	9059

Tabela 4.4: Valores Máximos de Tamanho de Intervalo de perdas para fila CT, $H = 0.8$, $P(alta|alta) = 0.7$, $\gamma = 0.0$, Tamanho do *buffer* = 400, $\rho = 0.8$.

4.4 Múltiplas Classes

Um mecanismo de descarte seletivo com múltiplas classes de prioridade fornece mais do que dois níveis de prioridade. Estes mecanismos possibilitam o atendimento de um número diverso de requisições de QoS, tornando possível a utilização de menos espaço de *buffer* para suportar diversas QoS do que um mecanismo com duas classes de prioridade. Em outras palavras, com um número maior de classes de prioridade não precisa-se garantir taxas de perdas inferiores do que as requeridas pela aplicação. Além disso, pode-se carregar cargas mais altas do que em um mecanismo com somente duas classes de prioridade. Descarte seletivo com múltiplas classes de prioridade sob um processo com dependência de curta-duração não é tão atrativo quanto sobre um processo com dependência de longa-duração. De fato, sob um processo com dependência de curta-duração com o aumento do tamanho do *buffer*, diminui-se a taxa de perda por classe consideravelmente[FS93].

Para avaliar a efetividade do descarte seletivo com múltiplas classes de prioridade

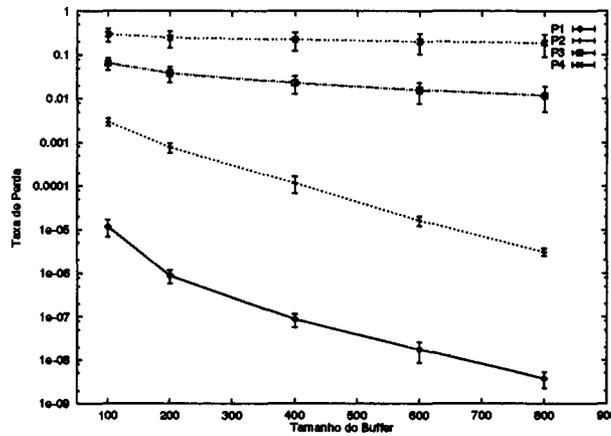


Figura 4.17: Taxa de Perda por classe x Tamanho do *buffer* para $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.85$, $P_1 = 0.4$, $P_2 = 0.1$, $P_3 = 0.2$ e $P_4 = 0.3$.

sob processos com dependência de longa-duração, simulou-se uma fila com a política de organização de *buffer* compartilhamento total com quatro níveis de prioridade. Utilizou-se como processo de entrada um processo de movimento Browniano fracional (descrito na seção 2.6). Os resultados mostram o comportamento de quatro níveis de prioridade frente a diversos parâmetros de entrada e a utilização das políticas de descarte PCPD, PCPD-M e UCPD. Avaliou-se impacto do aumento do *buffer*, parâmetro de Hurst e variância na taxa de perda por classe e no tamanho do intervalo de perdas.

Na Figura 4.17, mostra-se a taxa de perda por classe como função do tamanho do *buffer*. Observa-se que em uma fila com quatro níveis de prioridade pode-se oferecer taxa de perdas distintas para os quatro níveis. Além disso, com o aumento do tamanho do *buffer* diferencia-se ainda mais as taxas de perda. Se ao invés de quatro níveis existisse apenas dois níveis de prioridade, teria-se que transportar as classes 2 e 3 com uma taxa de perda maior ou deveria se ter um tamanho de *buffer* maior para se obter as mesmas taxas de perda. Por exemplo, necessita-se de um tamanho de *buffer* maior que 25000 para se fornecer uma taxa de perda de 10^{-6} enquanto que é necessário um tamanho de *buffer* de 700 em uma fila com quatro níveis de prioridade.

Na Figura 4.18 mostra-se que com o aumento do parâmetro de Hurst de 0.75 (Figura 4.18a) para 0.8 (Figura 4.18b) a taxa de variação da taxa de perdas das classes 2 e 3 diminui. Em outras palavras, com o aumento do parâmetro de Hurst, aumenta-se a taxa de perda total e, conseqüentemente, pode-se diminuir somente a taxa de perda da classe de prioridade mais alta.

Fornecer taxas de perdas distintas implica que quando comparado com um sistema de dois níveis de prioridade pode-se satisfazer requisitos de perda (QoS) com menos espaços

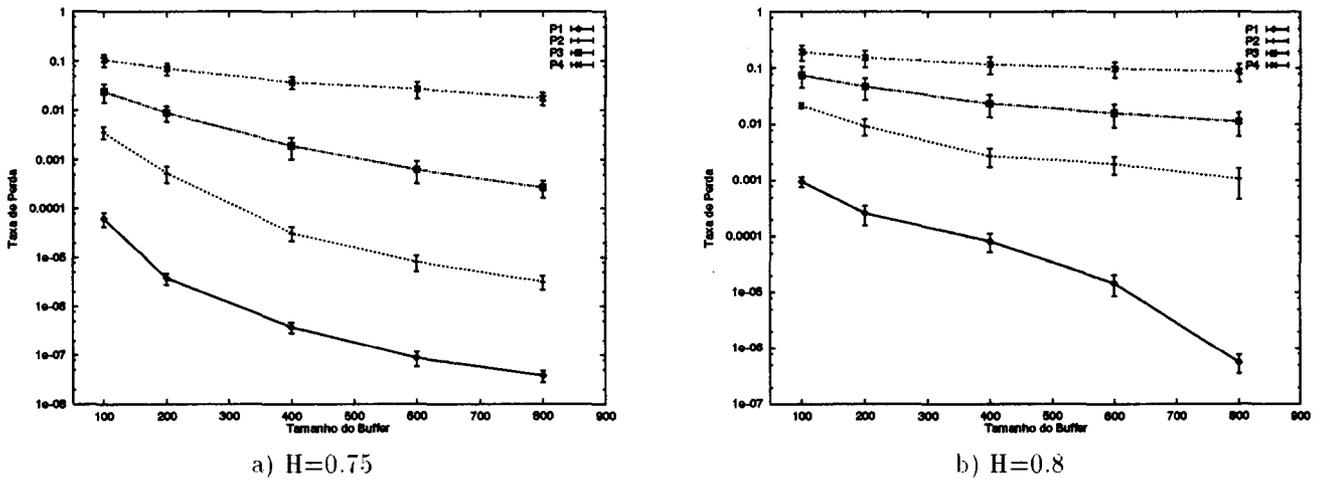


Figura 4.18: Taxa de Perda por classe x Tamanho do *buffer* para $\rho = 0.8$, $\sigma^2 = 1$. $P_1 = 0.60$, $P_2 = 0.07$, $P_3 = 0.13$ e $P_4 = 0.20$.

de *buffers*. Em um mecanismo com duas classes de prioridade, deve-se agregar aplicações com diversos requisitos de QoS e transportar a carga agregada de acordo com os requisitos de QoS da aplicação agregada mais exigente. Por exemplo, considere uma fila com quatro níveis de prioridade com tamanho de *buffer* 100, com uma carga de 0.8, onde a proporção de células em cada classe de prioridade da classe mais alta para mais baixa é respectivamente: 0.55, 0.05, 0.15 e 0.25. Nesse cenário, é possível fornecer taxas de perda de 10^{-7} e 10^{-5} para duas classes de maior prioridade. Em uma fila com apenas dois níveis de prioridade, sob as mesmas condições de tráfego poderia-se precisar de um tamanho de *buffer* de 1200.

A demanda de *buffer* é fortemente afetada pelo parâmetro de Hurst e pela variância do processo de entrada. No exemplo anterior, se a variância fosse 1.2 ao invés de 0.5 é possível fornecer taxas de perda de 10^{-5} e 10^{-3} para as duas classes de maior prioridade em uma fila com quatro níveis de prioridade, enquanto necessita-se de 2500 espaços de *buffer* em uma fila com dois níveis de prioridade.

Na Figura 4.19, mostra-se a taxa de perda por classe como função da variância do processo de entrada. Observa-se que mesmo para valores altos de variância, um mecanismo de descarte seletivo com múltiplas classes de prioridade fornece taxas de perda distintas por classe. Este comportamento difere de um mecanismo com duas classes de prioridade. Em filas com dois níveis de prioridade, com o aumento da variância tanto a taxa de perda da classe com alta e da classe com baixa prioridade convergem para o mesmo valor.

Outra vantagem de um mecanismo com múltiplas classes de prioridade sobre um mecanismo com duas classes de prioridade é que se pode ter cargas mais altas em um sistema com múltiplas classes de prioridade. De fato, pode-se aumentar a carga oferecida

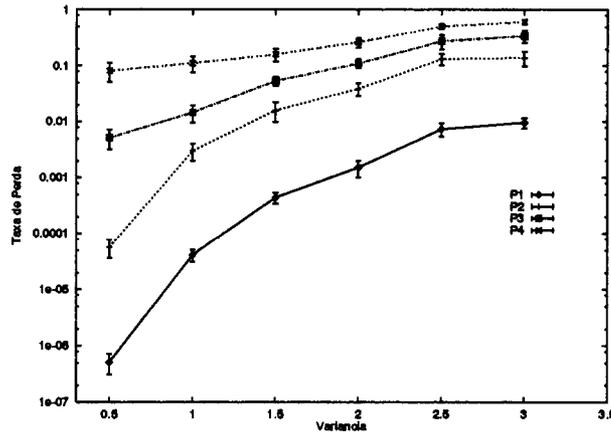
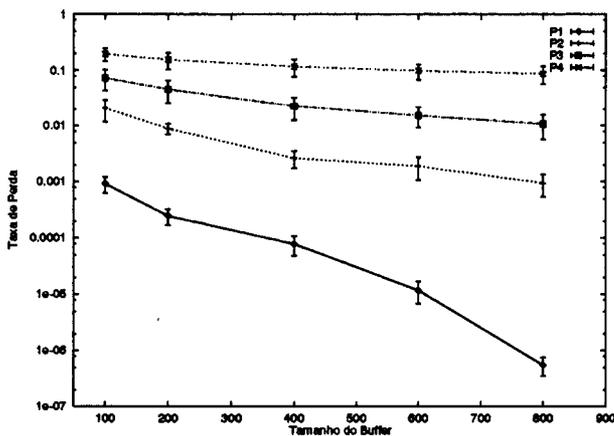
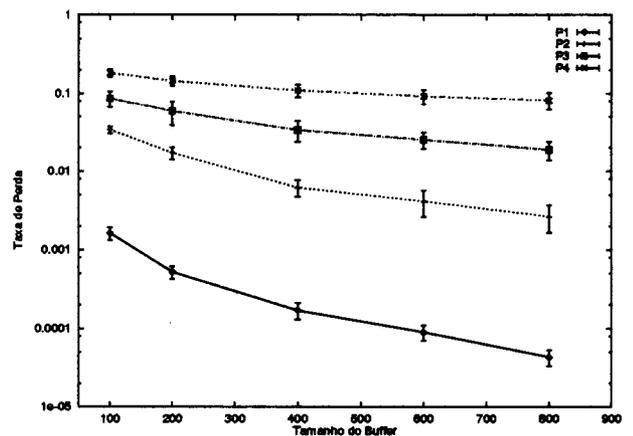


Figura 4.19: Taxa de Perda por classe x Variância para $\rho = 0.8$, $H = 0.75$. Tamanho do Buffer = 400, $P_1 = 0.7$, $P_2 = 0.05$, $P_3 = 0.1$ e $P_4 = 0.15$.



a) PCPD-M



b) UCPD

Figura 4.20: Taxa de Perda por classe x Tamanho do buffer para diferentes políticas de descarte. $\rho = 0.8$, $\sigma^2 = 1$, $H = 0.8$, $P_1 = 0.60$, $P_2 = 0.07$, $P_3 = 0.13$ e $P_4 = 0.20$.

aumentando-se a carga das aplicações com menor requisito de QoS. Por exemplo, em uma fila com dois níveis de prioridade com tamanho de *buffer* 500 e carga de 0.65 ($H = 0.75$, $P_1 = 0.7$ e $P_2 = 0.3$), ao se introduzir mais dois níveis de prioridade, pode-se aumentar a carga até 0.9 distribuindo-se a carga adicional entre as duas novas classes com prioridade mais baixas.

Na Figura 4.20 mostra-se a taxa de perda por classe fornecida pelas políticas PCPD-M e UCPD. Nota-se que a mesma tendência da taxa de perda da classe com prioridade alta em uma fila com duas classes de prioridade acontece para taxa de perda da classe com maior prioridade em uma fila com quatro níveis de prioridade. A política de descarte impacta mais significativamente a taxa de perda da classe de mais alta prioridade em uma fila com quatro níveis de prioridade do que impacta a taxa de perda das outras classes.

Nas Figuras 4.21 e 4.22 mostra-se o tamanho do intervalo de perda para as classes 2, 3 e 4 produzidos pelas políticas PCPD-M e UCPD, respectivamente. Em uma fila com quatro níveis de prioridade, o intervalo de perda é distribuído entre as várias classes de prioridade. Portanto, o intervalo de perda máximo da classe de prioridade mais baixa em uma fila com quatro níveis de prioridade é menor do que o intervalo de perda máximo da classe com prioridade baixa em uma fila com dois níveis de prioridade. Nesse exemplo específico, o intervalo de perda máximo produzido pelas políticas PCPD-M e UCPD em uma fila com dois níveis de prioridade são 1800 e 1200, respectivamente. O tamanho do intervalo de perda máximo produzido pela política PCPD-M e pela política UCPD em uma fila com quatro níveis de prioridade não diferem tanto quanto diferem em uma fila com dois níveis de prioridade. Contudo, a frequência de intervalos de perda grandes sob a política PCPD-M é muito maior do que a frequência sob a política UCPD.

Considerando-se o mesmo cenário de tráfego da Figura 4.21, mas com $H = 0.75$ o tamanho máximo de intervalo de perda pode ser a metade do valor encontrado na Figura 4.21. O intervalo de perda da classe de prioridade mais baixa em uma fila com quatro níveis de prioridade é menos influenciada do que o intervalo de perda da classe de baixa prioridade em uma fila com dois níveis de prioridade uma vez que as células perdidas são distribuídas entre outras classes de prioridade. Por exemplo, para o mesmo cenário de tráfego e para os mesmos parâmetros de Hurst ($H = 0.75$ e $H = 0.8$) o intervalo de perda da classe com baixa prioridade em uma fila com duas classes de prioridade cresce três vezes mais do valor com $H = 0.75$ para $H = 0.8$ (ver Figura 4.16).

4.5 Resumo

- Resultados baseados na hipótese de se ter uma política de descarte PCPD indicaram que o parâmetro de Hurst tem uma influência grande na eficácia dos mecanismos de descarte seletivo baseado na disciplina C.T. Para valores altos de parâmetro de

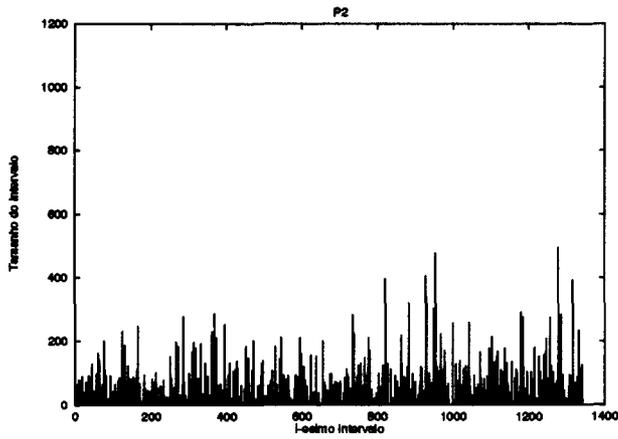
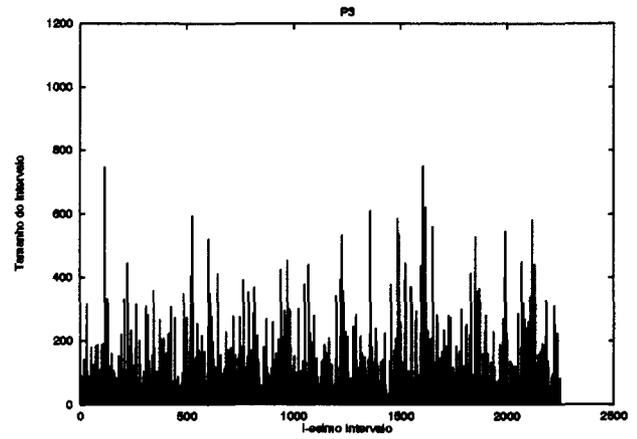
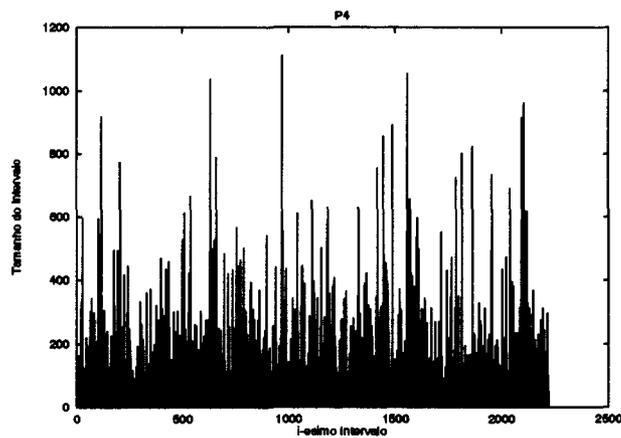
a) P_2 b) P_3 c) P_4

Figura 4.21: Tamanho do Intervalo de Perda x i -ésimo Intervalo para política PCPD-M, classes de prioridade 2, 3 e 4, $\rho = 0.8$, $\sigma^2 = 1$, Tamanho do *Buffer* = 400. $P_1 = 0.70$, $P_2 = 0.05$, $P_3 = 0.1$ e $P_4 = 0.15$.

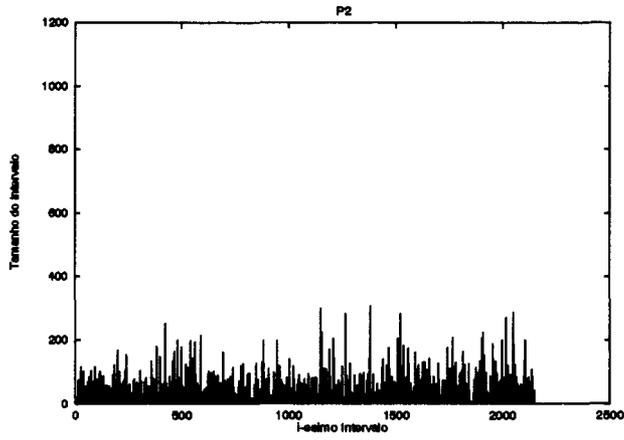
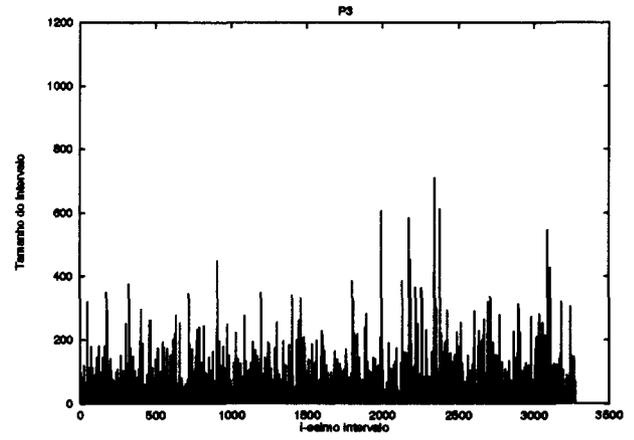
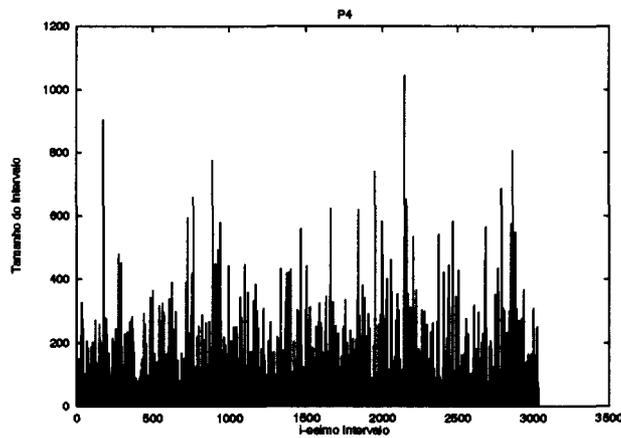
a) P_2 b) P_3 c) P_4

Figura 4.22: Tamanho do Intervalo de Perda x i -ésimo Intervalo para política PCPD-M, classes de prioridade 2, 3 e 4, $\rho = 0.8$, $\sigma^2 = 1$, Tamanho do *Buffer* = 400, $P_1 = 0.70$, $P_2 = 0.05$, $P_3 = 0.1$ e $P_4 = 0.15$.

Hurst ($H > 0.8$) não existe diminuição significativa na taxa de perda da classe com prioridade alta mesmo para grandes *buffers* (> 800). Contrariamente, para valores baixos do parâmetro de Hurst, a taxa de perda da classe com prioridade alta diminui quando aumenta-se o tamanho do *buffer*.

- Para filas baseadas em CTGTMF, a taxa de perda da classe com prioridade alta aumenta significativamente com um nível de proteção de 20%. Adicionalmente, nenhum impacto significativo na taxa de perda da classe com prioridade baixa foi observado. Esses achados indicam que o descarte seletivo baseado em CT é claramente vantajoso para valores baixos de parâmetro de Hurst (< 0.8) enquanto não é necessariamente vantajoso para a disciplina de CTGTMF.
- Com o aumento da variância, aumenta-se também a taxa de perda de células da classe com prioridade alta independente do valor de H , uma vez que a probabilidade de transbordo do *buffer* aumenta com a variância.
- Observa-se que a taxa de perda da classe com prioridade alta para valores altos de $P(\text{alta}|\text{alta})$ é sempre maior do que para valores baixos de $P(\text{alta}|\text{alta})$ porque para valores altos de $P(\text{alta}|\text{alta})$ tem-se longas rajadas de células da classe com prioridade alta em situações de transbordo. Já para valores altos de $P(\text{baixa}|\text{baixa})$ tem-se valores baixos de taxa de perda de células da classe com prioridade alta uma vez que aumenta-se a probabilidade de uma célula de alta prioridade encontrar longas rajadas de células de baixa prioridade na fila em situações de transbordo.
- A política de descarte tem um grande impacto nos resultados. A disciplina PCPD-M minimiza a taxa de perda da classe com prioridade alta e pode gerar uma taxa de perda três ordens de grandeza mais baixa do que UCPD. Resultados baseados na PCPD são muito próximos daqueles mostrados pela PCPD-M. A disciplina PCPD tem a vantagem de ser menos complexa de implementar uma vez que requer menos deslocamento de *buffer* do que a PCPD-M. Logo, PCPD é uma opção atrativa para implementação em um comutador ATM.
- A política PCPD-M sempre produz o valor máximo de comprimento de intervalo de perdas da classe com prioridade baixa seguido de perto pela política PCPD e depois pela política RAND e finalmente pela política UCPD.
- Verifica-se que a política PCPD-M produz um comprimento máximo de intervalo de perdas que pode ser três vezes maior do que o comprimento máximo de intervalo de perdas da política UCPD.
- Observa-se que com o aumento do tamanho do *buffer* a diferença entre o comprimento máximo de intervalo das perdas das diversas políticas de descarte diminui.

Além disso, o comprimento de rajada de perda da classe com prioridade baixa é muito sensível ao parâmetro H .

- Observa-se também que o mecanismo de proteção de *buffer* (CTGTMF) pode reduzir significativamente o tamanho máximo do intervalo de perdas da classe com prioridade baixa. Conclui-se também que aumentar a carga, $P(alta|alta)$, ou a variância aumenta também o comprimento do intervalo de perdas de todas as políticas de descarte.
- Utilizando-se múltiplas classes de prioridade pode-se oferecer taxas de perdas bem distintas para os diversos níveis de prioridade. Fornecer taxas de perdas distintas implica que quando comparado com um sistema com dois níveis de prioridade pode-se satisfazer requisitos de perda (QoS) com menos espaços de *buffers*.
- Em uma fila com múltiplos níveis de prioridade, o intervalo de perda é distribuído entre as várias classes de prioridade.

Capítulo 5

Descarte Seletivo de Pacotes Sujeito a Processo LRD

Em redes ATM pacotes a nível de camada de transporte são transmitidos em células de tamanho fixo consistindo de *48-bytes* de dados (*payload*) e *5-bytes* de cabeçalho. Como dados de usuários são geralmente maiores que *48-bytes*, um pacote a nível de camada de transporte é segmentado em várias células ATM. A camada de adaptação ATM (*AAL* - *ATM Adaptation Layer*) fornece do lado do transmissor segmentação de pacotes de tamanho variável em células e função de reconstrução do pacote do lado do receptor. O padrão ATM especifica controle de erro para o cabeçalho da célula, mas não para o campo de dados (*payload*). Controle de erro a nível do campo de dados é realizado de forma fim-a-fim pela camada de transporte acima da camada ATM. Para algumas aplicações se uma das células de um pacote é perdida, o pacote inteiro tem que ser retransmitido. Portanto, transmitir células de um pacote corrompido ocupa banda-passante, e pode conseqüentemente aumentar o congestionamento. Assim sendo, várias políticas de descarte de pacotes vêm sendo definidas a fim de preservar a integridade de pacotes [RF95, TKT+97, Tur96, LRS97].

Neste capítulo, introduz-se um nova política de descarte de pacote chamada "Aceita-Maior-Pacote" (AMP) que maximiza o *goodput* de células, isto é, a proporção de células boas (células de pacotes não corrompidos) pelo número total de células que chegam na fila. Em outras palavras, ao invés de tentar maximizar o número de pacotes transmitidos com sucesso, maximiza-se a banda-passante usada para transportar pacotes não corrompidos. Compara-se essa nova política com as políticas Descarte Parcial de Pacote, Descarte Antecipado de Pacote, Descarte Antecipado de Pacote com *Histerese* e Descarte Antecipado de Pacote com *Histerese* Justa, investigando-se o compromisso entre o *goodput* de célula e o *goodput* de pacote (a proporção de pacotes bons transmitidos pelo número total de pacotes que chegam na fila). Mostra-se que essa política pode aumentar significativamente

o *goodput* de célula e também é capaz de produzir o maior *goodput* de pacote quando sujeita a um processo com dependência de curta-duração composto de pacotes grandes. Avalia-se o impacto da carga da rede, tamanho médio do pacote, distribuição do pacote e parâmetro de Hurst no *goodput* de célula e no *goodput* de pacote. Além disso, estende-se a análise para redes com fontes que geram pacotes de tamanhos médios distintos.

Este capítulo está organizado da seguinte maneira: na seção 5.1 faz-se uma breve revisão das políticas de descarte de pacotes existentes. Na seção 5.2 introduz-se a política Aceita-Maior-Pacote. Na seção 5.3 os principais resultados são discutidos.

5.1 Políticas de Descarte de Pacote

Para algumas aplicações, se uma célula de um pacote é perdida, o pacote inteiro tem que ser retransmitido. Portanto, transmitir células de um pacote corrompido desperdiça banda-passante, e conseqüentemente pode aumentar o congestionamento em nós da rede já congestionados. Dessa forma, várias políticas de descarte de pacotes foram recentemente definidas. Na política Descarte Parcial de Pacote (DPP - *Partial-Packet-Discard* ou *Tail Drop*), se uma célula é perdida, células subseqüentes do mesmo pacote são descartas. Na política Descarte Antecipado de Pacote (DAP - *Early-Packet-Discard*), uma posição limite da fila (*threshold*) é definida e um pacote que esteja chegando é descartado caso encontre o *buffer* ocupado acima desse *threshold* [RF95]. Tanto a política DPP quanto a política DAP podem aumentar significativamente o *goodput* quando comparadas a sistemas sem controle a nível de pacote.

Vários estudos têm investigado essas políticas. Ramanow e Floyd [RF95] concluíram que para pacotes de tamanho fixo a política DAP fornece maior *goodput* do que a política DPP. Lapid et. al [LRS97] analisaram um sistema com entrada Poisson e tamanho de pacote distribuído geometricamente. Este trabalho verificou que a política DPP tem um desempenho melhor do que política DAP para sistemas com carga moderada, enquanto a política DAP é preferível para sistemas com carga elevada. Tsukumatani et. al [TKT+97] consideraram um sistema carregado com várias fontes *on-off* com períodos distribuídos geometricamente. Este trabalho aponta que o *threshold* ótimo da política DAP torna-se menor a medida que se aumenta o tamanho médio dos pacotes.

Com o objetivo de aumentar o desempenho da política DAP para *buffers* pequenos, Turner definiu a política Descarte Antecipado de Pacote com *Histerese* (DAPH - *Early-Packet-Discard with Histerese*) [Tur96]. Na política DAPH, uma fonte (conexão) pode estar ativa ou inativa. Quando uma fonte é considerada inativa todos seus pacotes são descartados. A política DAPH define um nível de fila adicional chamado nível *floor* que deve ser atribuído um número pequeno de células. No final da transmissão de um pacote, se o nível de ocupação da fila exceder o *threshold* e o nível da fila desde a última vez que

se ultrapassou o *threshold*, a fonte é considerada inativa. Além disso, uma fonte é também considerada inativa se ela perder uma célula de um pacote, retornando ao estado ativo no próximo pacote. Por outro lado, se o nível da fila estiver abaixo do *threshold* e também abaixo do nível mínimo desde a última vez que ultrapassou o *threshold* ou abaixo do nível de *floor*, a fonte é considerada ativa. Observando que fontes com taxa de transmissão alta tendem a iniciar a transmissão antes do nível de fila ultrapassar o *threshold* do que fontes com taxa de transmissão baixa, Turner definiu outra variante da política DAP chamada Descarte Antecipado de Pacote com *Histerese* Justa (DAPHJ- *Fair Early-Packet-Discard with Histerese*). Além das regras da política DAPH, a política DAPHJ define um *threshold* alto e um *threshold* baixo. Se o nível atual da fila está entre os dois *thresholds*, e o nível da fila está decrescendo, uma fonte é marcada como ativa. Por outro lado, se o nível da fila está crescendo, a fonte é marcada como inativa.

5.2 Política Aceita-Maior-Pacote

As políticas de descarte existentes tem por objetivo maximizar o número de pacotes transmitidos com sucesso sem levar em consideração o tamanho desses pacotes. Por exemplo, na política DPP um pacote é descartado se uma das suas células for perdida, contudo, o tamanho do pacote descartado pode ser maior do que os pacotes enfileirados. Na política DAP um pacote pode ser descartado mesmo se existir espaço disponível de *buffer* para acomodá-lo. Para melhorar essa situação adversa, define-se uma nova política de descarte que maximiza o número de células em pacotes transmitidos com sucesso. Uma vez que células são de tamanho fixo, maximizar o número de células em pacotes transmitidos com sucesso é equivalente a maximizar a banda-passante utilizada para transmitir pacotes inteiros.

A política Aceita-Maior-Pacote (AMP) sempre admite um pacote que chega ao *buffer* caso exista espaço disponível (no *buffer*). Caso o espaço disponível seja menor do que o tamanho do pacote que chega, descarta-se um ou mais pacotes enfileirados. Assume-se que células consecutivas de um pacote chegam em fatias de tempo consecutivos (*time slots*) e que a informação do tamanho do pacote é obtida na primeira célula do pacote. Assim sendo, a decisão de aceitar o pacote que chega é feita no instante de chegada da sua primeira célula. Qualquer pacote com sua primeira célula ainda na fila é eleito para ser descartado, isto é, qualquer pacote cuja primeira célula não tenha deixado a fila é passivo de descarte. Se um pacote é escolhido para ser descartado, todas suas células enfileiradas são simultaneamente descartadas e qualquer outra célula desse pacote não é mais admitida no buffer. A política AMP é descrita abaixo:

se $(D = L + \sum_{i \in S} u_i - B + b - \max(L, \max_{i \in S}(u_i))) \leq 0$
 enfileirar o pacote que chega

senão Solucionar o Problema I:

$$\text{Min } \sum_{i \in Q} l_i x_i$$

Sujeito a:

$$L > \sum_{i \in Q} l_i x_i \geq D$$

$$x_i \in \{0, 1\}, i = 1..|Q|$$

se (Problema I admite uma solução \hat{x})

descarta pacote i tal que $\hat{x}_i = 1$

senão

descarta o pacote que chega

onde:

D - Demanda de *buffer* necessária para transmitir o pacote que chega;

S - Conjunto de fontes transmitindo pacotes não corrompidos no instante em que chega um pacote, excluindo a fonte do pacote que chega;

Q - Conjunto de fontes com sua primeira célula ainda na fila;

l_i - Tamanho do i -ésimo pacote;

u_i - Número de células do i -ésimo pacote, $i \in S$, que ainda estão para chegar no início da fatia de tempo na qual o novo pacote chega;

B - Tamanho do *buffer*;

b - Tamanho da fila no instante em que chega o novo pacote;

L - Tamanho do pacote que chega;

\hat{x} - Solução do Problema I.

Note que o algoritmo ao verificar se existe espaço disponível no *buffer* para enfileirar o pacote que chega (se $D \leq 0$), considera que $\max(L, \max_{i \in S}(u_i))$ espaços de *buffer* serão liberados durante a transmissão das fontes ativas. Se não existir espaço de *buffer* suficiente, procura-se por um conjunto de pacotes com tamanho total mínimo ($\text{Min } \sum_{i \in Q} l_i x_i$) que caso descartados liberem $\sum_{i \in Q} l_i x_i \geq D$ espaços de *buffer* para admitir o pacote que chega. Obviamente, o tamanho total de pacotes descartados deve ser menor do que o pacote que chega ($L > \sum_{i \in Q} l_i x_i$), caso contrário diminuir-se-ia o *goodput* de célula.

A Figura 5.1 ilustra um exemplo da política AMP. Neste exemplo mostra-se quatro fontes transmitindo pacotes de tamanho de 5, 4, 3 e 8 células ATM em um determinado instante de tempo $t = T$. A fonte transmitindo o pacote de tamanho cinco ($l = 5$) está (no tempo $t = T$) colocando na fila sua terceira célula, as fontes transmitindo os pacotes de tamanho quatro e três ($l = 4$ e $l = 3$) estão colocando na fila suas segundas células, e a fonte com o pacote de tamanho oito ($l = 8$) está iniciando sua primeira transmissão. Nesse instante, a política AMP detecta que não haverá espaço suficiente para transmitir todos os pacotes ($(D = 1) > 0$). Solucionando-se o Problema I (conjunto de pacotes com tamanho total mínimo) escolheu-se o pacote de tamanho três ($l = 3$) para ser descartado. Observa-se que no tempo $t = T + 1$, as células do pacote de tamanho

três foram descartadas da fila, e as células que estão chegando desse pacote estão sendo descartadas também.

O Problema I é um problema do tipo da mochila (*knapsack problem*). Pode-se resolver o problema da mochila em $O(c * n)$ onde c é a capacidade da mochila e n é o número de objetos. No Problema I, a capacidade da mochila é o número de espaços de *buffer* que devem ser liberados com objetivo de admitir o pacote que chega (D), e os objetos são os pacotes a serem descartados. Note que D leva em consideração o tamanho do crescimento da fila até o tempo de transmissão da última célula das fontes ativas, isto é, leva em consideração o número de células das fontes ativas que ainda vão chegar e o número de espaços de *buffer* que serão liberados na janela de tempo que inicia na chegada do novo pacote e termina no tempo de transmissão da última célula do pacote mais longo a ser transmitido.

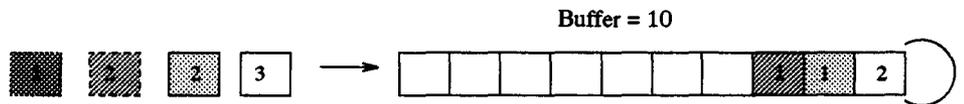
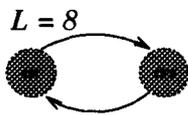
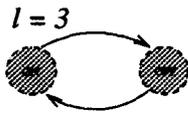
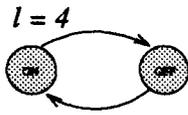
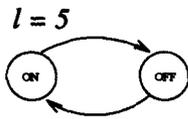
Descartar pacotes cujas primeiras células ainda estão na fila é uma abordagem conservadora, uma vez que um *goodput* de célula alto pode ser atingido em um multiplexador isolado ao se escolher pacotes para serem descartados entre todos os pacotes na fila. De fato, selecionar um pacote entre todos enfileirados ou selecionar um pacote entre pacotes cujas primeiras células ainda estão na fila é um compromisso entre maximizar o *goodput* de célula em um multiplexador e evitar o envio para os próximos multiplexadores de células de pacotes descartados. Através dos exemplos numéricos mostrados na seção 5.3, nota-se que enfileirar células de pacotes corrompidos pode ter um efeito negativo no *goodput* de célula. Uma vez que o fluxo que chega em um multiplexador é composto pela saída de diversos outros multiplexadores, o número de células inúteis armazenadas em um multiplexador pode aumentar consideravelmente. Prevenir que células de pacotes corrompidos fluam através da rede pode ser obtido a um custo alto de sinalização. Contudo, esse *overhead* adicional não é desejável em redes de alta velocidade.

5.3 Resultados Numéricos

A fim de se avaliar a eficácia da política AMP, compara-se essa com as políticas: Descarte Parcial de Pacote (DPP), Descarte Antecipado de Pacote (DAP), Descarte Antecipado de Pacote com *Histerese* (DAPH) e Descarte Antecipado de Pacote com *Histerese* Justa (DAPHJ). Analisa-se o compromisso entre maximizar o número de células transmitidas em pacotes com sucesso e maximizar o número de pacotes transmitidos com sucesso. Além disso, avalia-se numericamente as políticas DAPH e DAPHJ. Pelo que se tem conhecimento essas duas políticas nunca foram analisadas antes.

Estudos anteriores adotaram diferentes objetivos de desempenho para comparar políticas de descarte. Turner [Tur96] e Romanow e Floyd [RF95] consideraram a vazão efetiva (*effective throughput*) que é a proporção de células boas a nível de pacote no fluxo de saída

$t = T$



$t = T + 1$

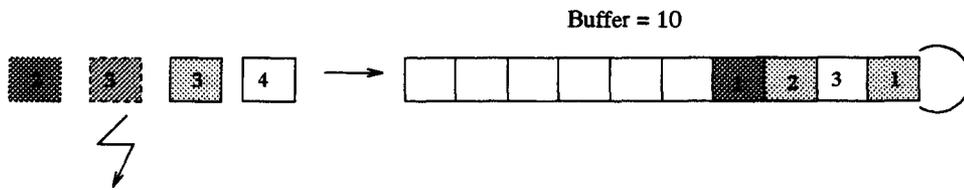
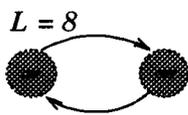
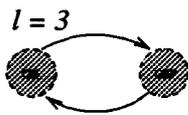
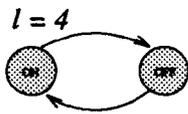
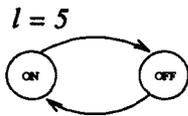


Figura 5.1: Exemplo da política AMP.

pelo total de células no fluxo de saída. Lapid et. al definiram o *goodput* como a proporção de células boas no fluxo de saída pelo número total de células que chegam na fila. No presente trabalho, define-se o *goodput* de célula como em Lapid et. al [LRS97]. O *goodput* de célula mostra quanto do tráfego não é desperdiçado. Ele difere da vazão efetiva no sentido de que a vazão efetiva não leva em consideração células descartadas. A vazão efetiva considera apenas o fluxo de saída. Pela definição, a política AMP produz vazão efetiva 1.0, uma vez que a política AMP permite somente células boas a nível de pacote no fluxo de saída. Além disso, apresenta-se o *goodput* de pacote que é a proporção de pacotes bons no fluxo de saída pelo número total de pacotes que chegam na fila. O *goodput* de pacote fornece a fração de pacotes de usuário transmitidos com sucesso. Em resumo, o *goodput* de célula fornece a eficácia de uma política a nível do *link* de rede, enquanto o *goodput* de pacote indica a eficácia de uma política a nível da camada de transporte.

Dado que não é possível resolver analiticamente uma fila com a política AMP devido à explosão de estados, faz-se uso de simulações. O método de replicação é usado para derivar resultados com 95% de intervalo de confiança. Nos experimentos de simulação, considera-se um conjunto de filas. Cada fila está sob uma política diferente de descarte de pacote e sua entrada é o processo agregado de varias fontes *on-off* (Figura 5.2). Enquanto em estado *on*, uma fonte produz uma célula de um pacote a cada fatia de tempo, isto é, uma fonte gera uma célula diferente a cada vez que visita o estado *on* e o tamanho do pacote é a duração do período *on*. Os períodos *off* representam o tempo entre geração de pacotes. Para avaliar como diferentes políticas desempenham sob diversos modelos de tráfego, gera-se processos com dependência de curta-duração e de longa-duração usando-se distribuição exponencial e Pareto no período *on/off*, respectivamente. Além disso, investiga-se os resultados para processos com dependência de longa-duração quando o tempo de geração entre chegadas de pacotes (períodos *off*) é distribuído exponencialmente.

A carga do sistema é dada por $\rho = N * \varepsilon$, onde N é o número de fontes, e $\varepsilon = T_{on} / (T_{on} + T_{off})$ é a taxa de atividade, isto é, a probabilidade estacionária de uma fonte estar em estado *on*. O T_{on} e T_{off} são a média de duração dos períodos *on* e *off*, respectivamente. Varia-se a carga do sistema pela mudança de N ou de ε . Comenta-se as discrepâncias de resultados caso venham a ocorrer.

Nos experimentos de simulação, investiga-se o impacto do tamanho do *buffer*, tamanho médio do pacote, valor de *threshold* e carga da rede no *goodput* de célula e no *goodput* de pacote. Mostra-se resultados para tamanhos de *buffer* 100 e 400, e um valor de *threshold* de 70% do tamanho do *buffer*. Observa-se que para valores de *threshold* acima de 80%, resultados gerados pela política DAP tendem para valores produzidos pela política DPP. Os resultados para valores de *threshold* entre 50% e 70% são muito parecidos. Esses resultados estão de acordo com os achados em [LRS97]. Para política DAPH mostra-se resultados para *threshold* baixo de 60% do tamanho do *buffer* e para *threshold* alto de 80%

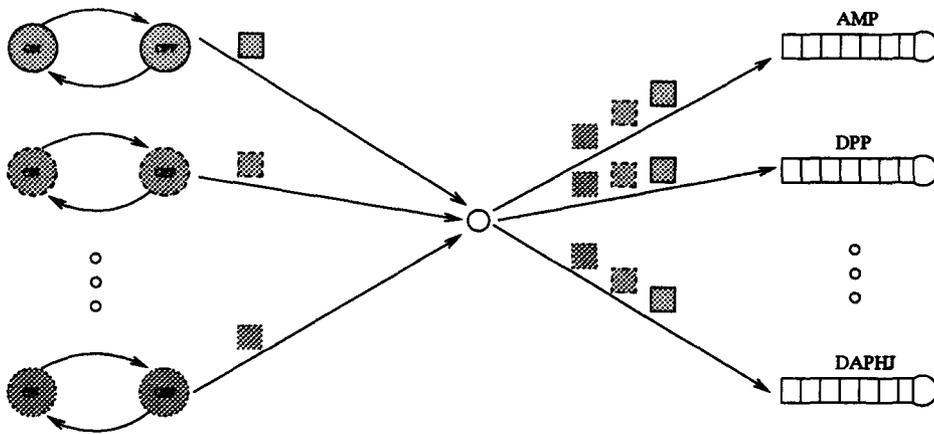


Figura 5.2: Modelo de Simulação.

do tamanho do *buffer*. Mostra-se resultados para fluxos com tamanho médio de pacote de 42 células ATM. Finalmente, fixa-se a taxa de atividade de cada fonte em 0.025.

5.3.1 Fontes *on-off* com períodos geometricamente distribuídos

▷ Influência da carga

Primeiramente considera-se períodos *on-off* geometricamente distribuídos. Nas Figuras 5.3 e 5.4, mostra-se o *goodput* de célula e o *goodput* de pacote como uma função da carga da rede, para tamanho médio do pacote de 42 células ATM e tamanho de *buffer* 100 e 400, respectivamente. Varia-se a carga pelo aumento do número de fontes. Cada fonte contribui com 0.025 da carga do sistema o que fornece um tempo entre chegadas de 1638 fatias de tempo. Com o aumento da intensidade da carga, o *goodput*, obviamente, diminui. Contudo, a política AMP fornece o maior *goodput* de células. Para tamanho pequeno de *buffer* (100) e para carga da rede de 1.0, enquanto o *goodput* de célula fornecido pela política AMP está acima de 0.78, o *goodput* de célula fornecido pelas políticas DPP e DAPH(J) estão abaixo de 0.65 (Observa-se que o *goodput* fornecido pelas políticas DAPH e DAPHJ são indistinguíveis). A diferença entre o *goodput* de célula produzido pela política AMP e o *goodput* de célula produzido pela política DAPH (ou pela DAPHJ) pode chegar a 0.2. Com o aumento do tamanho do *buffer* (para 400), a diferença entre o *goodput* de célula produzido pela política AMP e o *goodput* de célula produzido pelas outras políticas diminui. Contudo, essa diferença pode chegar ainda a 0.1. Além disso, o *goodput* de célula produzido pela política AMP tem a menor taxa de diminuição. Por exemplo, para tamanho de *buffer* 400, o *goodput* de célula produzido pela política AMP diminui 0.06 enquanto a política DAPH(J) diminuiu 0.15.

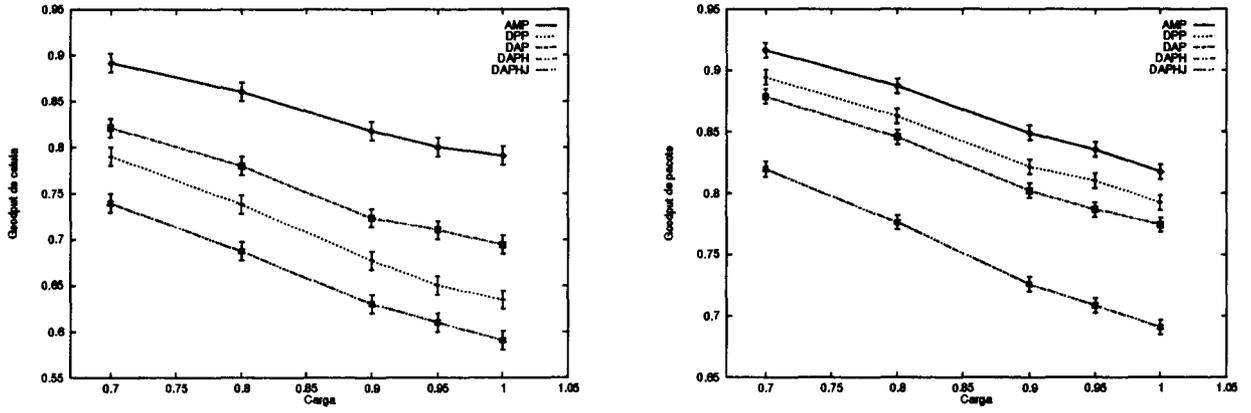


Figura 5.3: *Goodput* de célula e pacote x Carga fornecida para tamanho médio de pacote de 42 células ATM, e Tamanho de *buffer* = 100.

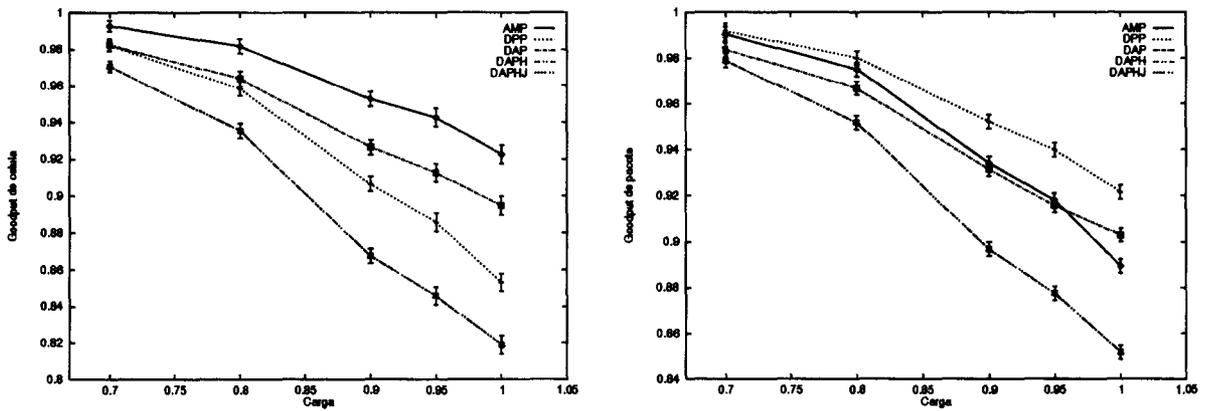


Figura 5.4: *Goodput* de célula e pacote x Carga fornecida, para tamanho médio de pacote de 42 células ATM, e Tamanho de *buffer* = 400.

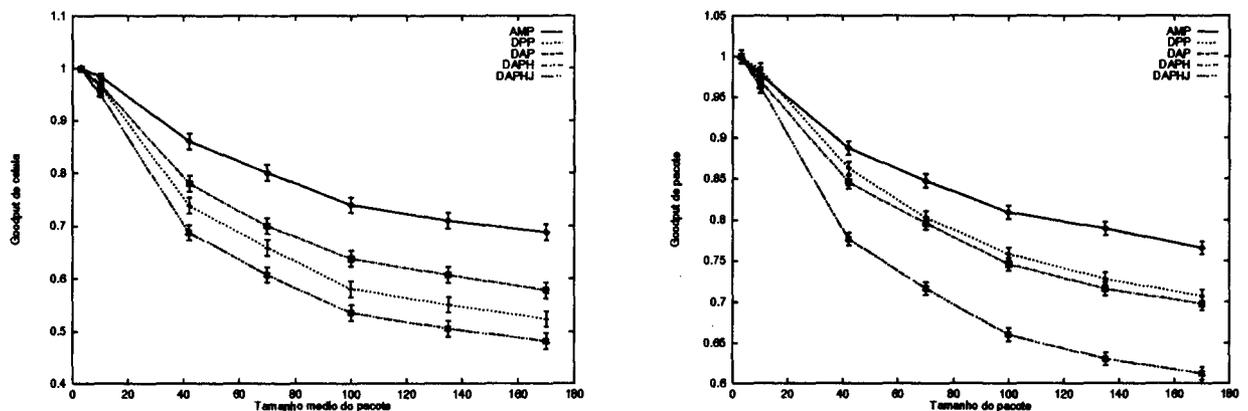


Figura 5.5: *Goodput* de célula e pacote x Tamanho médio do pacote, Tamanho do *buffer* = 100, Carga Fornecida = 0.8.

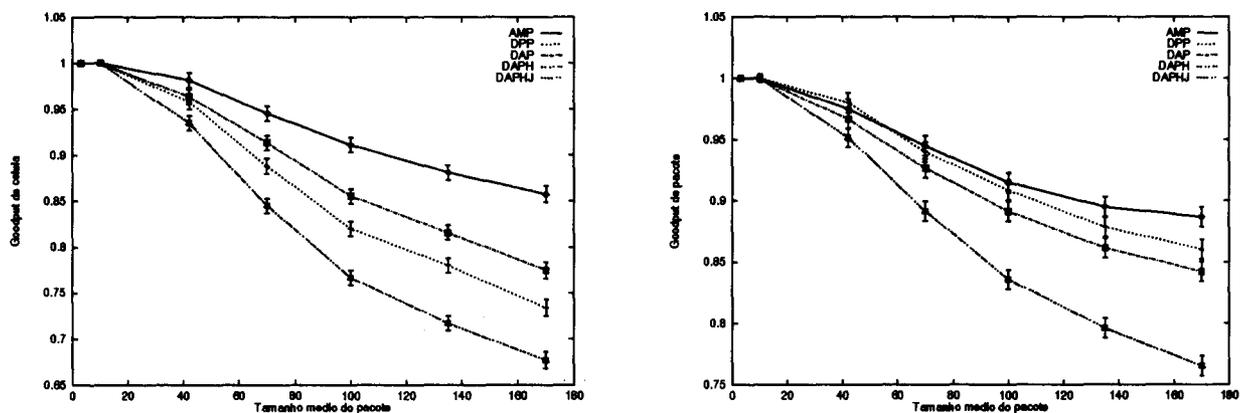


Figura 5.6: *Goodput* de célula e pacote x Tamanho médio do pacote, Tamanho do *buffer* = 400, Carga Fornecida = 0.8.

Para tamanho de *buffer* 100 e tamanho médio de pacote de 42 células ATM, isto é, para uma proporção alta de tamanho médio do pacote em relação ao tamanho do *buffer*, a política AMP fornece o maior *goodput* de pacote. Isso acontece porque sob a política AMP espaços de *buffer* (*buffer slots*) são somente ocupados por células de pacotes não corrompidos. Em outras políticas, espaços de *buffer* podem ser ocupados por células de pacotes corrompidos, e conseqüentemente, pode não existir espaço suficiente para acomodar o pacote que chega. Além disso, a política AMP tipicamente precisa descartar somente um pacote para acomodar um pacote que chega. Para tamanho de *buffer* 400, a política DPP fornece o maior *goodput* de pacote. Contudo, a diferença do *goodput* de pacote entre a política DPP e a política AMP é no máximo 0.04.

▷ Influência do tamanho médio do pacote

Nas Figuras 5.5 e 5.6, mostra-se a relação entre o *goodput* e o tamanho médio do pacote para tamanho de *buffer* 100 e 400, respectivamente. Fixa-se o número de fontes em 32, o que fornece um carga de 0.8. Logo, para manter constante a taxa de atividade (ε) com a variação do tamanho médio do pacote (duração do período *on*), varia-se também a duração média dos períodos *off*. Para tamanhos de pacotes menores do que 10 células ATM, o *goodput* produzido por diferentes políticas são quase indistinguíveis. Com o aumento do tamanho do pacote, o *goodput* de célula diminui, porque o espaço de *buffer* disponível é normalmente menor do que pacotes que estão chegando. Para tamanhos pequenos de *buffer* (100), enquanto o *goodput* de célula da política AMP está acima de 0.7, o *goodput* de célula fornecido pelas políticas DAPH e DAPHJ fica abaixo de 0.57, o que de maneira geral é inaceitável. Isso ilustra o benefício de ocupar o espaço de *buffer* somente com células de pacotes não corrompidos. Para tamanhos grandes de *buffer* (> 400), a diferença entre o *goodput* de célula produzido pela política AMP e o *goodput* produzido pelas outras políticas decresce. Contudo, para tamanhos grandes de pacotes (> 40 células ATM), a política AMP ainda é capaz de manter o *goodput* pelo menos 0.1 mais alto do que qualquer outra política. A diferença entre o *goodput* de célula da política AMP e da política DAPH(J) pode ser maior do que 0.2.

O *goodput* de pacote decresce também com o aumento do tamanho do pacote. Para *buffers* pequenos, a diferença entre o *goodput* de pacote produzido pela política AMP e o *goodput* produzido pelas outras políticas pode ser 0.05 e a diferença de *goodput* da política AMP e da política DAPHJ pode ser 0.15. Para tamanhos grandes de *buffers* (> 400) e tamanho de pacote menor que 50 células ATM, a política DPP produz o maior *goodput* de pacote, enquanto que para pacotes maiores do que 50 células ATM, a política AMP fornece o maior *goodput* de pacote. Para pacotes grandes, a diferença entre o *goodput* de pacote da política AMP e o *goodput* de pacote da política DAPHJ pode ser 0.17. Para pacotes menores que 50 células ATM, a política AMP pode descartar mais do que um pacote para enfileirar um pacote grande que esteja chegando, enquanto a política DPP perde somente um pacote em situações de transbordo. Com o aumento do tamanho do pacote, aumenta-se a probabilidade de haver pacotes grandes na fila, e, conseqüentemente, a política AMP tipicamente descarta somente um pacote para aceitar um pacote que está chegando. Além disso, sob outras políticas com o aumento do tamanho do pacote o *buffer* fica "entupido" com um número alto de células de pacotes corrompidos. É notório que o *goodput* de pacote produzido pela políticas DAPH e DAPHJ decresce abruptamente para pacotes grandes, enquanto o *goodput* de pacote produzido pelas políticas DPP e DAP decresce suavemente. Em resumo, a política AMP fornece o maior *goodput* de células sem levar em consideração o tamanho do pacote, e produz o maior *goodput* de pacotes para pacotes grandes.

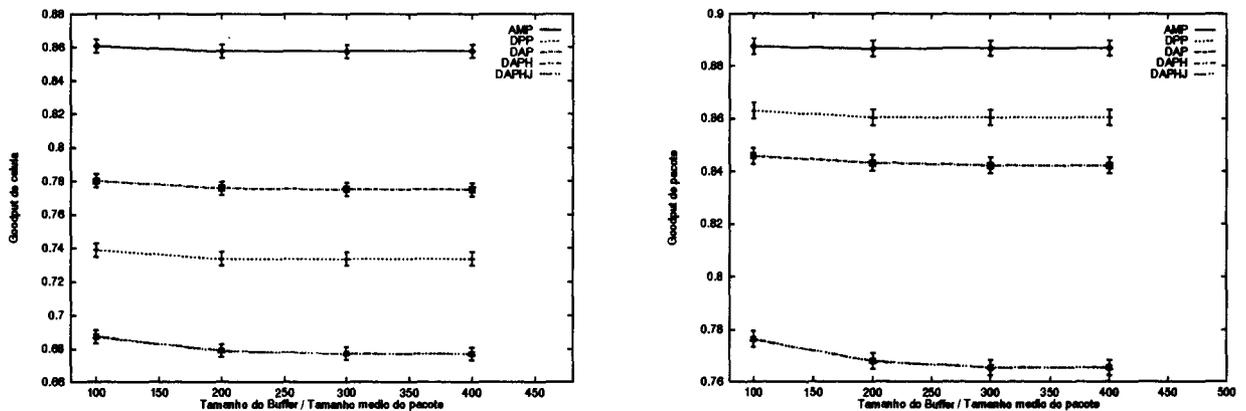


Figura 5.7: *Goodput* de célula e pacote x Proporção constante do Tamanho do *buffer* pelo Tamanho médio do pacote. Carga Fornecida = 0.8.

Nos experimentos de simulação, nota-se que se ao invés de se aumentar o número de fontes, aumenta-se a taxa de atividade, isto é, mantém-se o número de fontes e a duração média de períodos *on* constante e decresce-se a duração média dos períodos *off*, o *goodput* de célula e o *goodput* de pacote da política AMP permanecem no mesmo valor, enquanto o *goodput* de célula e o *goodput* de pacote produzido pelas outras políticas aumentam 0.05. Em outras palavras, para um número fixo de fontes com taxa de atividade alta, os resultados da política AMP são claramente menos vantajosos do que são para números altos de fontes com taxa de atividade baixa.

Para entender o quanto os resultados do exemplo anterior são dependentes da razão entre o tamanho do pacote e o tamanho do *buffer*, avalia-se o *goodput* com função desta razão, isto é, aumenta-se simultaneamente o tamanho médio do pacote e o tamanho do *buffer* para uma carga fixa. Pela Figura 5.7, conclui-se que tanto o *goodput* de célula e o de pacote são quase os mesmos para um proporção fixa do tamanho médio do pacote pelo tamanho do *buffer*.

▷ Redes com fontes heterogêneas

Para avaliar políticas de descarte em redes com fontes heterogêneas, considera-se quatro tipos diferentes de fontes com tamanhos médios de pacotes distintos. Fixa-se o número total de fontes em 32, o que fornece uma carga de 0.8, e varia-se a proporção de cada tipo de fonte. Considera-se fontes com tamanho médio de pacote de 5, 21, 85 e 170 células ATM, e distribui-se inicialmente a carga igualmente entre os quatro conjunto de fontes. Varia-se, então, o número de fontes de um conjunto específico, e distribui-se o restante da carga entre os outros três conjuntos de fontes. Na Figura 5.8, para tamanho de *buffer*

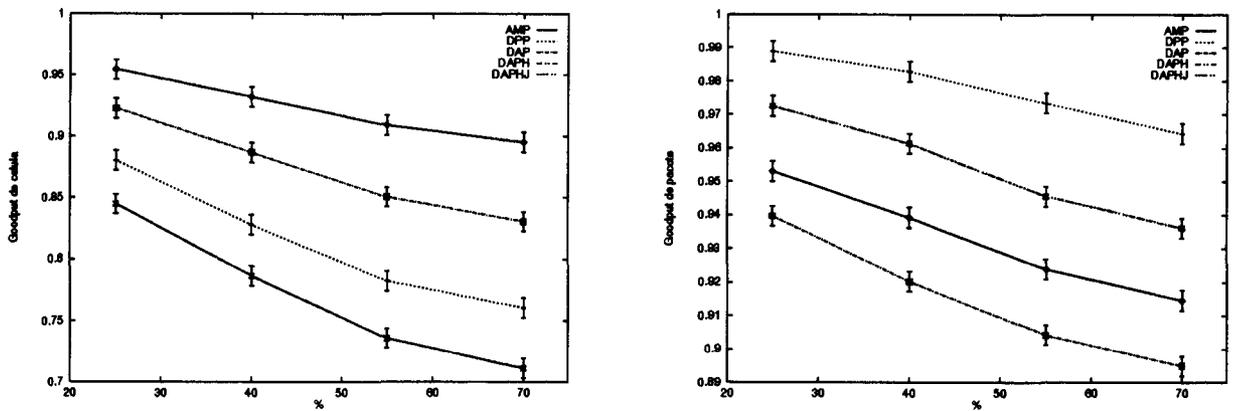


Figura 5.8: *Goodput* de célula e pacote x Proporção de células com tamanho médio de 170 células ATM em um rede com 32 fontes, pacote podem ter tamanho médio de 5, 21, 85 e 170 células ATM, tamanho do *buffer* = 400.

400, varia-se o número de fontes com tamanho médio de pacote de 170 células ATM. No eixo horizontal, mostra-se a proporção de fontes desse tipo. Com o aumento do número de pacotes grandes o *goodput* de célula diminui uma vez que se aumenta a chance de um pacote que chega encontrar um número baixo de espaços de *buffer* disponíveis. Contudo, a política AMP fornece o maior *goodput* de célula. Enquanto o *goodput* da política AMP está acima de 0.9, o *goodput* de célula da política DAPHJ é 0.71. A política AMP também tem a menor taxa de mudança do *goodput* da célula. Nesse exemplo específico, enquanto o *goodput* da política AMP decresce 0.05, o *goodput* de célula produzido pelas outras políticas decresce pelo menos 0.1. Para tamanho de *buffer* pequeno (100), a diferença entre o *goodput* de célula da política AMP e o *goodput* produzido pelas outras políticas pode ser tão baixo quanto 0.1 (DAP) e tão alto quanto 0.3 (DAPH e DAPHJ).

Apesar do *goodput* de pacote das políticas DPP e DAP serem maior do que o *goodput* de pacote da política AMP, a diferença entre o maior *goodput* de pacote (*goodput* de pacote da política DPP) e o *goodput* de pacote da política AMP é menor do que 0.05, sem levar em consideração o tamanho do *buffer*. Em resumo, a política AMP fornece o maior *goodput* de célula, e seu *goodput* de pacote não difere significativamente do *goodput* de pacote dado pelas outras políticas.

Contudo, para redes com uma proporção alta de pacotes pequenos, tem-se um quadro consideravelmente diferente. Na Figura 5.9, aumenta-se a proporção de fontes com tamanho médio de pacote de 5 células ATM e distribui-se o restante da carga entre as outras políticas. Com o aumento da proporção de pacotes pequenos, o *goodput* de célula também aumenta, e a diferença entre o *goodput* de célula produzido pelas diferentes políticas diminui. Mais uma vez, o *goodput* de pacote da política AMP é menor do que o *goodput* de

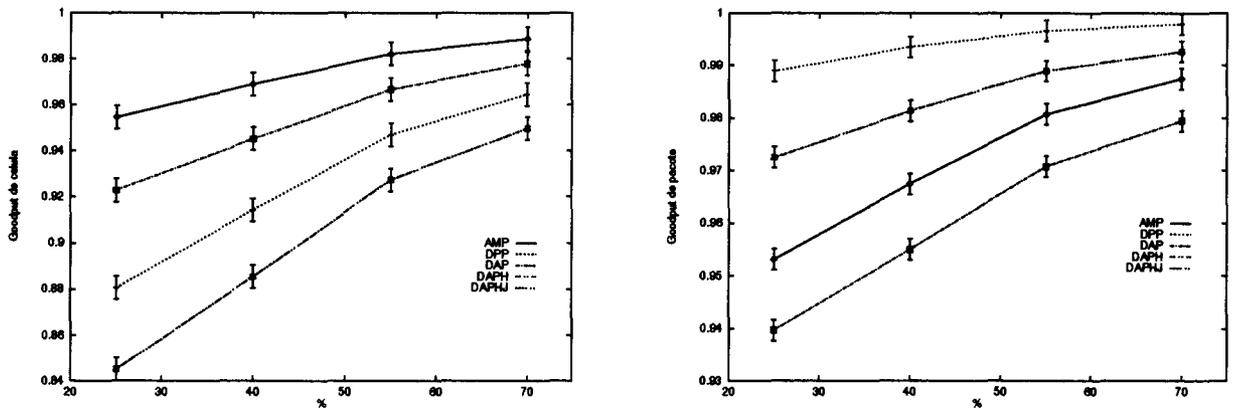


Figura 5.9: *Goodput* de célula e pacote x Proporção de células com tamanho médio de 5 células ATM em um rede com 32 fontes, pacote podem ter tamanho médio de 5, 21, 85 e 170 células ATM, tamanho do *buffer* = 400.

pacote das políticas DPP e DAP, mas a diferença entre o *goodput* de pacote da política DPP e o *goodput* de pacote da política AMP é novamente menor do que 0.05. Para tamanhos grandes de *buffer* (> 400) o *goodput* de célula dado pelas diferentes políticas é indistinguível.

5.3.2 Fontes *on-off* com cauda longa

Um processo auto-semelhante pode ter um efeito significativo no desempenho de enfileiramento, isto é, pelo simples aumento do tamanho do *buffer*, não é possível diminuir a probabilidade de transbordo consideravelmente. Portanto, é de grande importância avaliar a efetividade de políticas de descarte de pacotes sobre processos auto-semelhantes. De acordo com esse raciocínio, gera-se um processo movimento Browniano fracional através da agregação de fontes *on-off* com cauda longa. Usa-se a distribuição Pareto tanto para duração dos períodos *on* quanto para a duração dos períodos *off*. Nos experimentos de simulação, verificou-se que 20 fontes é geralmente suficiente para gerar um processo agregado auto-semelhante. Para se ter certeza que o processo agregado era realmente um processo com dependência de longa-duração, coletou-se para cada experimento de simulação o número de células (*bytes*) gerados a cada intervalo de tempo. Usou-se então a análise gráfica *variance time* para verificar se o parâmetro de Hurst do fluxo agregado era o valor especificado. Em outras palavras, verificou-se se o parâmetro de Hurst do fluxo agregado era igual a $(3 - \alpha)/2$ onde α é a forma (*shape*) da distribuição Pareto.

A Figura 5.10 exibe uma análise gráfica *variance time* feita sobre um fluxo gerado com tamanho de pacote 05 e parâmetro $\alpha = 1.6$, o que implica em um valor de $H = 0.8$. Neste

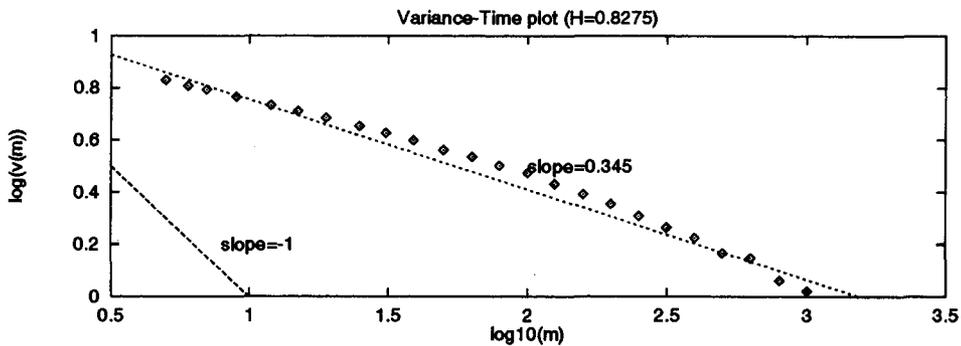


Figura 5.10: Análise Gráfica *variance time plot*.

caso a análise gráfica mostra para esse fluxo um valor de $H = 0.8275$.

▷ Influência da carga

Na Figura 5.11, mostra-se o *goodput* de célula como função da carga da rede para tamanho de *buffer* 400, tamanho médio de pacote de 42 células ATM, e para diferentes valores de parâmetro de Hurst. Com o aumento do parâmetro de Hurst, períodos de transbordo tornam-se mais longos, e conseqüentemente o *goodput* de célula diminui. A política AMP fornece o maior *goodput* de célula, e o *goodput* de célula da política DAP tem valor próximo do *goodput* de célula da política AMP. A política AMP é capaz de manter o *goodput* de célula acima de 0.74, mesmo sob um fluxo com $H = 0.9$. O *goodput* produzido pela política DPP e pela política DAPHJ fica abaixo de 0.6. Nota-se que dependências de longa-duração impactam consideravelmente o *goodput* de célula produzido por todas as políticas. Por exemplo, para uma carga de 0.7, o *goodput* de célula da política AMP é 0.95 para $H = 0.7$, e 0.8 para $H = 0.9$. Além disso, as políticas DPP, DAPH e DAPHJ são mais sensíveis ao parâmetro de Hurst do que a política AMP. Por exemplo, o *goodput* de célula da política DPP para uma carga de 0.7 e para $H = 0.7$ é 0.88 enquanto para $H = 0.9$ é 0.64. Com o aumento do parâmetro de Hurst, períodos de transbordo são mais demorados do que o tempo médio de residência nos períodos *on* e *off*. Uma vez que a política DPP usa qualquer espaço de *buffer* disponível para tentar acomodar um novo pacote, ela “entope” a fila com um número maior de células de pacotes corrompidos do que sob um processo com dependência de curta-duração. Portanto, um pacote que esteja chegando pode encontrar menos espaços de *buffer* disponíveis do que precisa, enquanto espaços de *buffer* são ocupados por células inúteis. As políticas DAPH e DAPHJ são mais sensíveis ao parâmetro de Hurst porque durante períodos longos de transbordo, existe uma chance maior que o nível de ocupação da fila no fim da transmissão de um pacote esteja acima da última vez que ultrapassou o nível de ocupação. Em outras palavras, as políticas

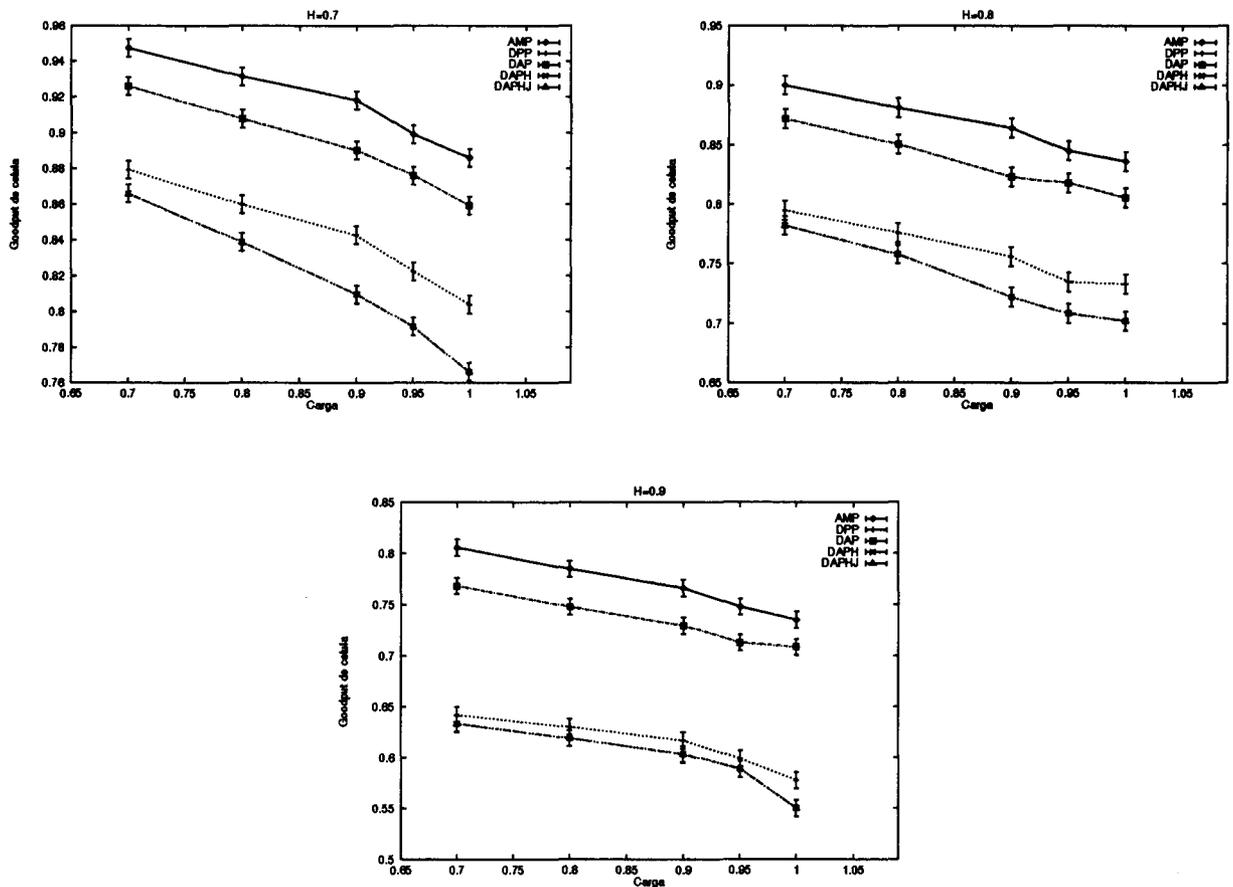


Figura 5.11: *Goodput* de célula x *Carga* fornecida, Tamanho médio de pacote de 42 células ATM. Tamanho do *buffer* = 400 e para diferentes valores de parâmetro de Hurst.

DAPH e DAPHJ tendem a considerar uma fonte como inativa desnecessariamente. Além de produzir o maior *goodput* de célula, para um valor fixo de H , a política AMP tem a menor taxa de diminuição do *goodput* de célula.

Para tamanho de *buffer* 100, e tamanho médio de pacote de 42 células ATM (isto é, para uma razão alta entre o tamanho médio do pacote e o tamanho do *buffer*) a política AMP também tem um bom desempenho. Para um fluxo com parâmetro de Hurst de 0.9, o *goodput* de célula da política AMP está acima de 0.8, enquanto o *goodput* fornecido pelas políticas DPP, DAPH e DAPHJ estão abaixo de 0.6. Em outras palavras, mesmo para valores altos de parâmetro de Hurst, a política AMP é capaz de manter uma utilização de banda-passante razoável, enquanto as políticas DPP, DAPH e DAPHJ utilizam menos de 60% da banda-passante para carregar informação útil. Para tamanho pequeno de *buffer* (100), o *goodput* de célula da política DAP não é mais semelhante ao *goodput* da política AMP, porque nesse cenário o mecanismo *threshold* descarta pacotes desnecessariamente.

De fato, o *goodput* da política DAP é 0.1 menor do que o *goodput* de célula da política AMP. Com o aumento do H , o *goodput* da política DPP fica mais perto do *goodput* das políticas DAPH e DAPHJ.

Comparando-se o *goodput* de célula sob um processo com dependência de curta-duração (períodos *on-off* com distribuição exponencial) com o *goodput* de célula sob um processo com dependência de longa-duração, observa-se que o *goodput* de célula é mais baixo sob um processo com dependência de longa-duração do que sob um processo com dependência de curta-duração. Por exemplo, para tamanho de *buffer* 400, e carga de 0.8, o *goodput* da política DPP é 0.96 sob um processo com dependência de curta-duração, enquanto é 0.62 sob um fluxo com $H = 0.9$. Contudo, o *goodput* da política AMP sob um fluxo com dependência de longa-duração decresce menos do que qualquer outra política. Por exemplo, para uma carga de 0.7 e para $H = 0.9$, o *goodput* de célula da política AMP decresce 0.18 quando comparado a um processo com dependência de curta-duração enquanto que o *goodput* de célula da política DAPH(J) decresce 0.33.

Por outro lado, o *goodput* de pacote da política AMP é altamente sensível ao parâmetro de Hurst e fornece o *goodput* de pacote mais baixo entre todas as políticas independente do tamanho do *buffer* (Figura 5.12). Isso acontece porque períodos de transbordo tornam-se mais longos e conseqüentemente, a política AMP descarta um número alto de pequenos pacotes para enfileirar pacotes grandes que chegam. Apesar de a política AMP produzir o menor *goodput* de pacote, ele está acima de 0.775, mesmo para $H = 0.9$. A diferença entre o maior *goodput* de pacote, o *goodput* de pacote da política DPP, e o *goodput* de pacote da política AMP é quase 0.2.

Nota-se que independente do comportamento de queda da correlação do processo de entrada (dependência de curta ou de longa duração), o *goodput* de célula da política DPP é sempre menor do que *goodput* de célula da política DAP, enquanto o *goodput* de pacote da política DPP é sempre maior do que o *goodput* da política DAP. Essa tendência pode ser facilmente entendida considerando-se o fato de que durante períodos de transbordo a política DPP tende a “entupir” a fila com um número alto de células inúteis. Portanto, em um certo momento, um pacote grande que esteja chegando pode achar o tamanho da fila da política DAP abaixo do valor de *threshold*, enquanto pode não encontrar espaço suficiente de *buffer* na fila da política DPP. Por outro lado, a política DPP pode admitir um número alto de pacotes de tamanho pequeno quando o tamanho da fila da política DAP está acima do valor de *threshold*. Contudo, o tamanho total de um número alto de pequenos pacotes admitidos sob a política DPP pode ser menor do que o tamanho total de um número pequeno de pacotes grandes admitidos na política DAP.

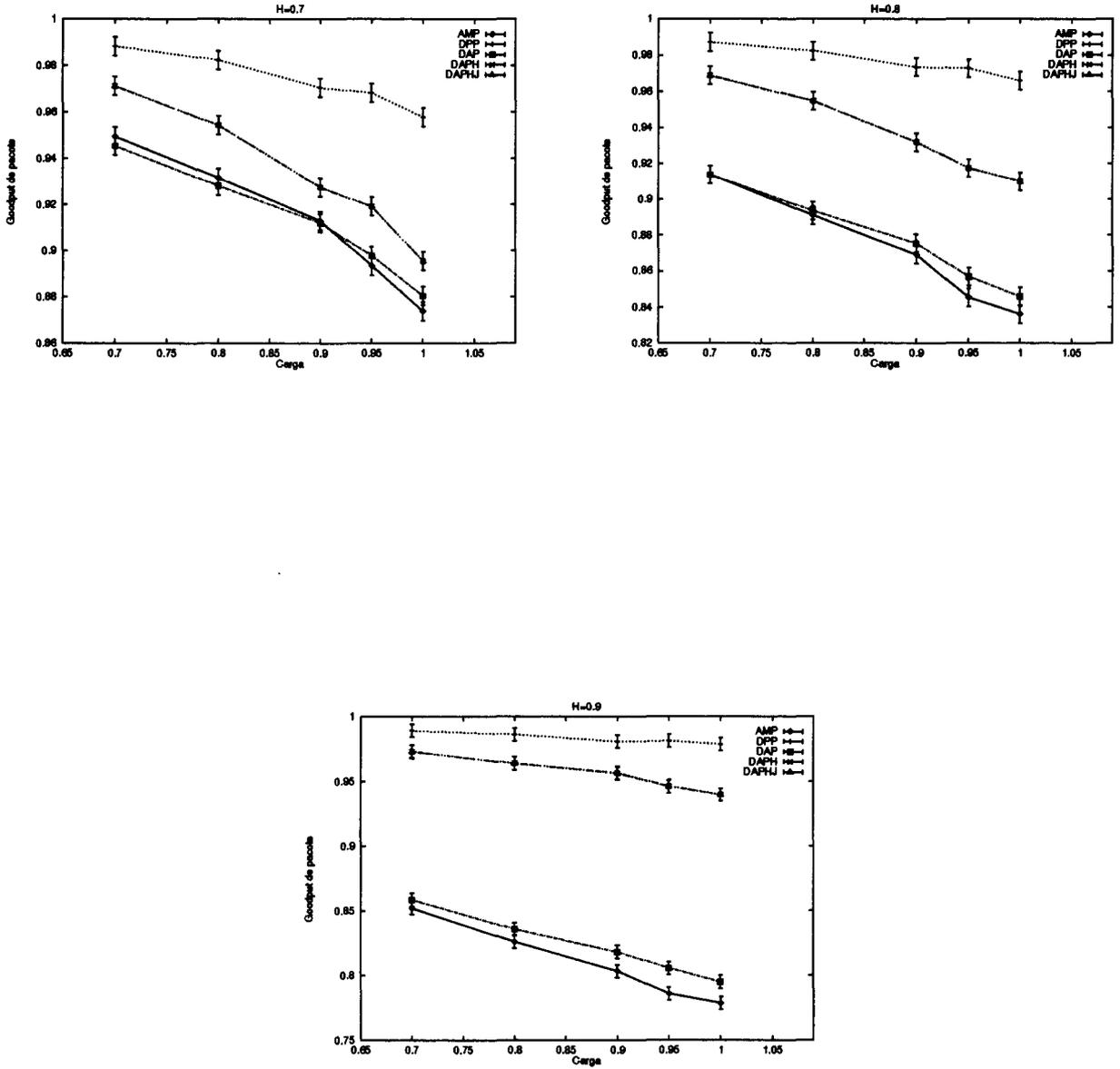


Figura 5.12: *Goodput* de pacote x *Carga* fornecida, Tamanho médio de pacote de 42 células ATM, Tamanho do *buffer* = 400 e para diferentes valores de parâmetro de Hurst.

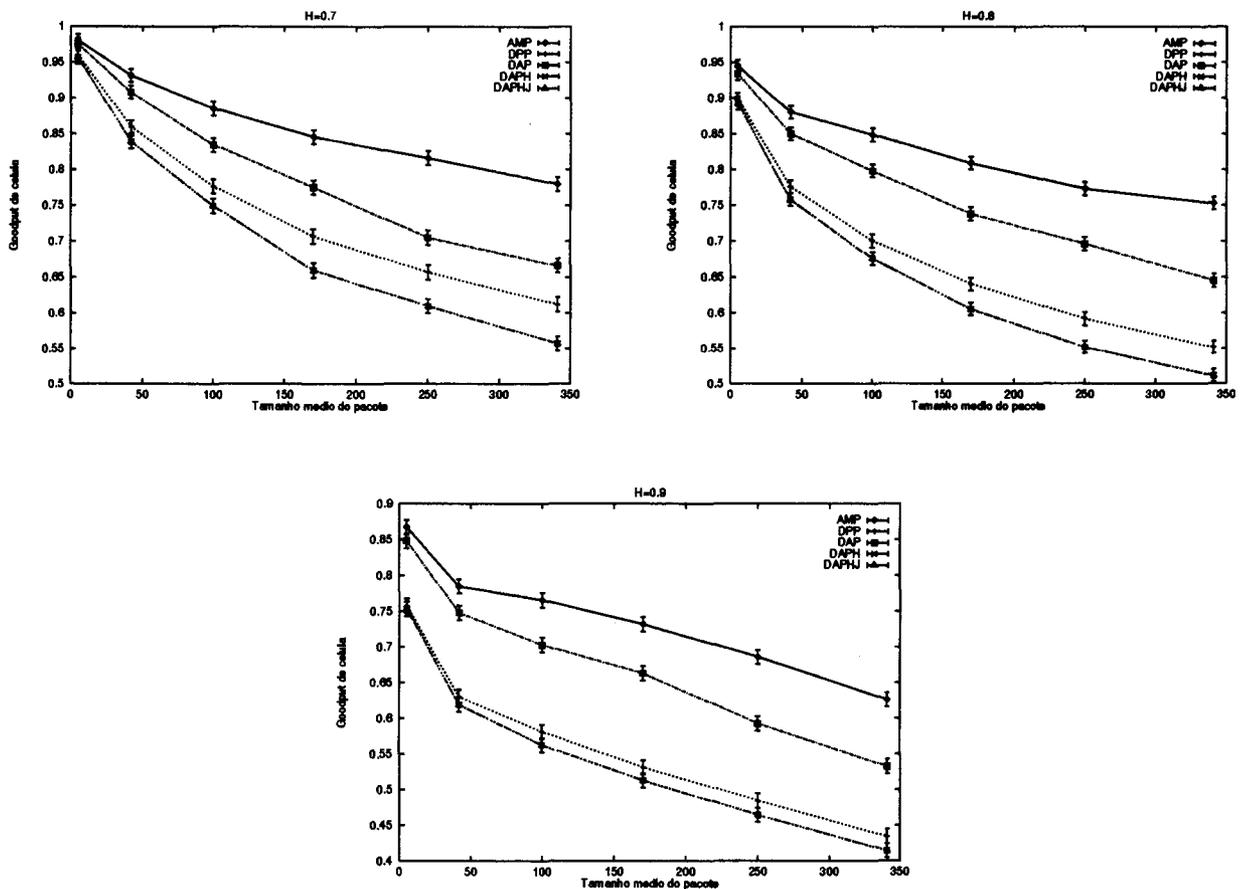


Figura 5.13: *Goodput* de célula x Tamanho médio do pacote, Carga fornecida = 0.8, Tamanho do *buffer* = 400 e para diferentes valores de parâmetro de Hurst.

▷ Influência do tamanho médio do pacote

Na Figura 5.13, mostra-se o *goodput* de célula com função do tamanho médio do pacote para tamanho de *buffer* 400, e para diferentes valores de H . Com o aumento do tamanho do pacote, o *goodput* de célula obviamente diminui. Contudo, a política AMP produz o maior *goodput* de célula. O parâmetro de Hurst impacta consideravelmente o *goodput* de célula.

Fluxos com um tamanho médio de pacote alto são mais afetados do que fluxos com tamanho médio de pacote baixo, isto é, pacotes longos têm alta probabilidade de serem descartados durante períodos de transbordo, e essa probabilidade aumenta quando períodos de transbordo tornam-se maiores. Por exemplo, o *goodput* de célula da política AMP para tamanho de pacote menor que 10 células ATM e para $H = 0.7$ é 0.98, enquanto para $H = 0.9$ é 0.875. Adicionalmente, para $H = 0.9$ e para um fluxo com tamanho médio

de pacote menor que 10 células ATM, o *goodput* de célula da política AMP é 0.875 enquanto é 0.625 para um tamanho médio de pacote de 350 células ATM. Além disso, a política AMP é menos afetada pelo parâmetro de Hurst do que as outras políticas. Por exemplo, para $H = 0.9$ e tamanho médio de pacote menor que 10 células ATM o *goodput* de célula da política DPP é 0.75, enquanto para tamanho médio de pacote de 350 células ATM esse valor é 0.45. Além de fornecer o maior *goodput* de célula, a política AMP tem também o menor declive.

Para tamanhos pequenos de *buffer*, isto é, para uma razão alta entre tamanho médio do pacote e tamanho do *buffer*, o *goodput* de célula é menor do que para uma razão baixa. Contudo, a política AMP é ainda capaz de manter um *goodput* razoável enquanto o *goodput* de célula da política DAPH(J) decai para valores inaceitáveis. Por exemplo, para um tamanho de pacote de 350, o *goodput* de célula da política AMP é 0.65 enquanto o *goodput* da política DAPH(J) é 0.35.

Nota-se que o *goodput* de célula é tipicamente mais alto sob um processo com dependência de curta-duração do que sob um processo com dependência longa-duração. Por exemplo, para um tamanho de pacote de 170 células ATM, tamanho de *buffer* 400 e sob um processo com dependência de curta-duração, o *goodput* de célula da política AMP é 0.87, enquanto o *goodput* de células é 0.725 sob um fluxo com $H = 0.9$. Além disso, sob um processo com dependência de curta-duração os *goodput* de célula para tamanho pequeno de pacote são quase indistinguíveis o que não acontece sob um processo com dependência de longa-duração, especialmente para valores altos de H .

Por outro lado, tanto o parâmetro de Hurst quanto o tamanho médio do pacote influenciam o *goodput* de pacote (Figura 5.14). Apesar da política AMP fornecer o *goodput* de pacote mais baixo para valores alto de parâmetro de Hurst, esse valor é sempre maior do que 0.725. Enquanto sob um processo com dependência de curta-duração e para pacotes grandes, a política AMP produz o *goodput* de pacote mais alto, sob um processo com dependência de longa-duração, ela produz o *goodput* de pacote mais baixo. Para fontes que geram pacotes longos, com os períodos de transbordo ficando maiores, a probabilidade de uma fonte produzir pacotes pequenos aumenta, assim sendo, aumenta também a probabilidade de um pacote longo que chega encontrar pacotes pequenos na fila. Logo, para períodos de transbordo longos a política AMP pode descartar mais do que um pacote para acomodar um pacote que chega.

▷ Redes com fontes heterogêneas

Para investigar políticas de descarte em redes heterogêneas, considera-se fontes com tamanho médio de pacote de 5, 21, 85 e 170 células ATM. Inicialmente distribui-se a carga igualmente entre todos os conjuntos de fontes. Aumenta-se o número de fontes para um conjunto específico de fontes e distribui-se o restante da carga igualmente entre os outros

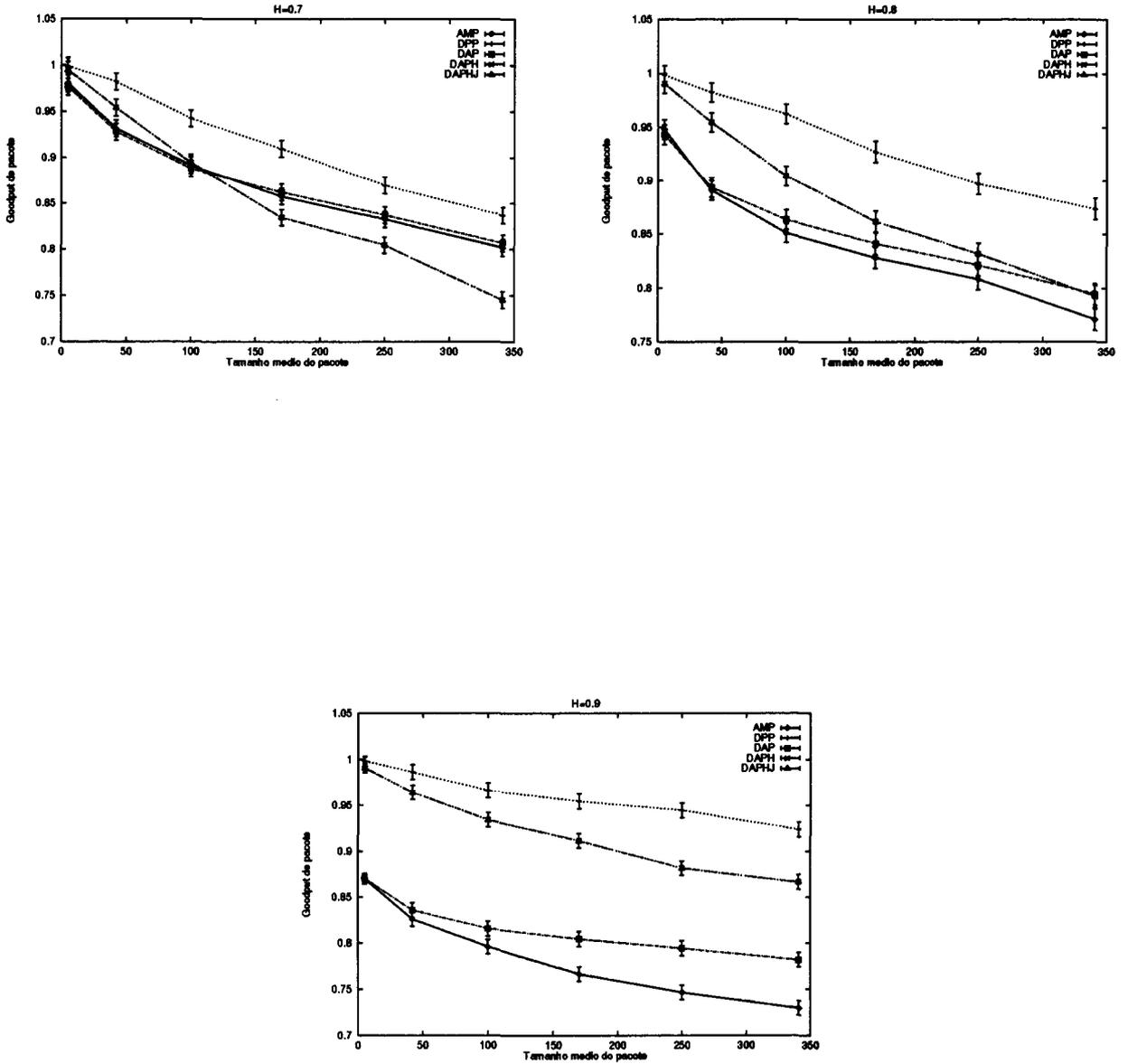


Figura 5.14: *Goodput* de pacote x Tamanho médio do pacote, Carga fornecida = 0.8, Tamanho do *buffer* = 400 e para diferentes valores de parâmetro de Hurst.

conjuntos. Na Figura 5.15, varia-se o número de fontes com tamanho médio de pacote de 170 células ATM para uma carga fixa de 0.8 e tamanho de *buffer* 400. A política AMP produz o *goodput* de célula mais alto. O parâmetro de Hurst tem uma grande influência no *goodput* de célula. Contudo, o *goodput* de célula da política AMP está acima de 0.67 mesmo para $H = 0.9$. Novamente, manter o *goodput* de célula alto quando o *goodput* das outras políticas decai consideravelmente ilustra a vantagem de maximizar a ocupação do *buffer* somente com células boas. Por outro lado, em redes heterogêneas com uma porcentagem alta de pacotes grandes, a política AMP tipicamente descarta mais que um pacote para admitir um pacote que chega, e, portanto produz o *goodput* de pacote mais baixo. Com o aumento do parâmetro de Hurst, o *goodput* de pacote decai consideravelmente. Enquanto as políticas DPP e DAP são capazes de manter um *goodput* de pacote alto, o *goodput* de pacote da política AMP para $H = 0.9$ decai para 0.67 (Figura 5.16).

Esses resultados diferem dos resultados fornecidos para um processo com dependências de curta-duração. Sob um processo com dependência de curta-duração, a política AMP produz o *goodput* de célula mais alto e também preserva o valor do *goodput* de pacote perto do *goodput* de pacote das políticas DPP e DAP. Ao se variar a proporção de pacotes com média de 5 células ATM, o *goodput* de célula da política AMP permanece acima de 0.85 mesmo para $H = 0.9$ e *goodput* de pacote da política AMP difere de no máximo 0.1 do *goodput* de pacote mais alto (política DPP).

5.3.3 Fontes com distribuição Pareto nos períodos *on* e distribuição geométrica nos períodos *off*

A convergência da agregação de um número grande de fontes *on-off* com cauda-longa para um movimento Browniano fracional não requer que a distribuição da duração de ambos os períodos *on* e os períodos *off* tenham cauda-longa. Para investigar se os resultados podem ser afetados pela hipótese de se ter períodos *off* com distribuição geométrica, repetiu-se os mesmos experimentos descritos na última seção. Nenhuma diferença significativa nos resultados foi observada.

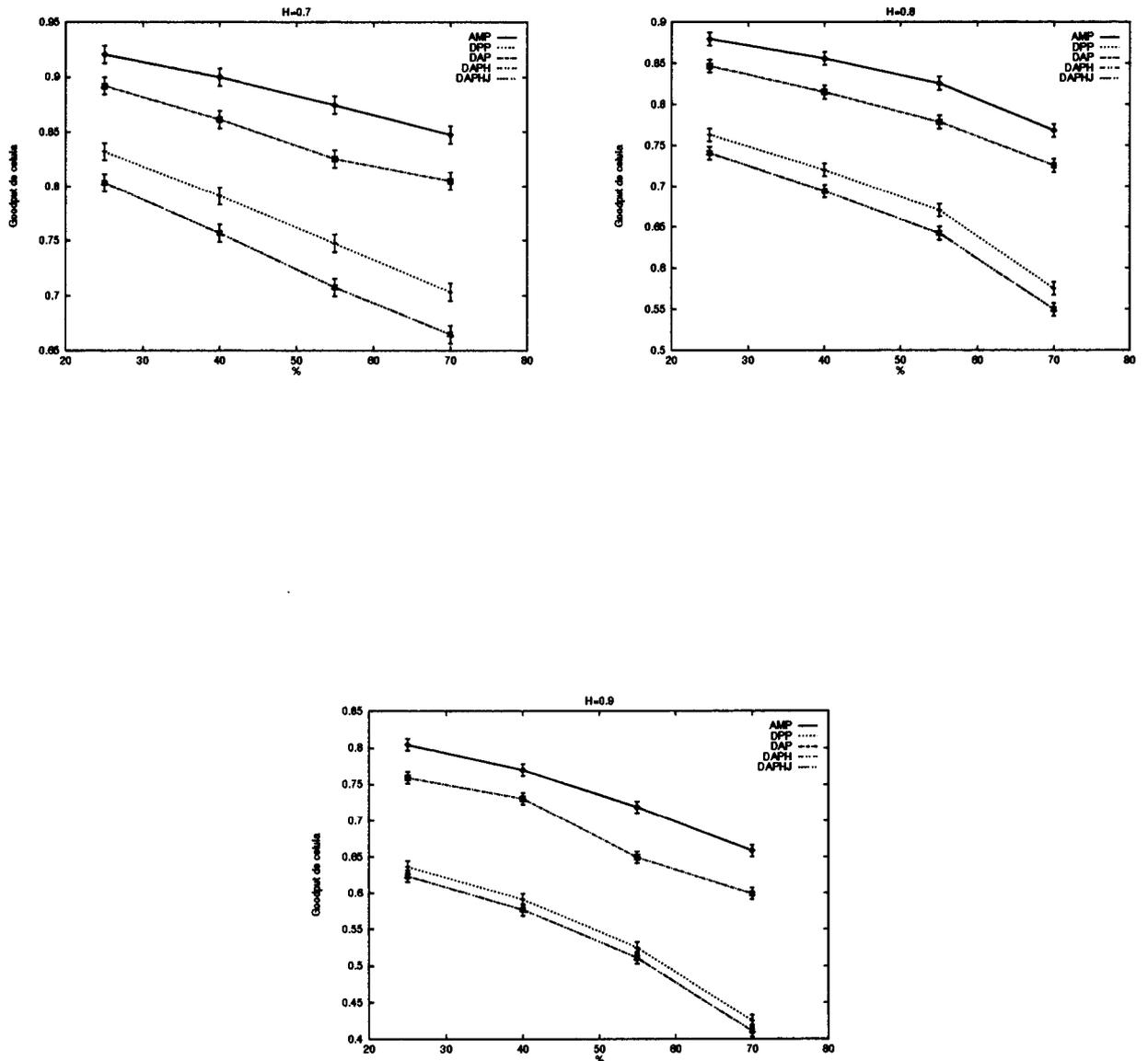


Figura 5.15: *Goodput* de célula x Proporção de células com tamanho médio de pacote de 170 células ATM para diferentes valores de parâmetro de Hurst em uma rede com 32 fontes, pacotes com tamanho médio de 5, 21, 85 e 170 células ATM.

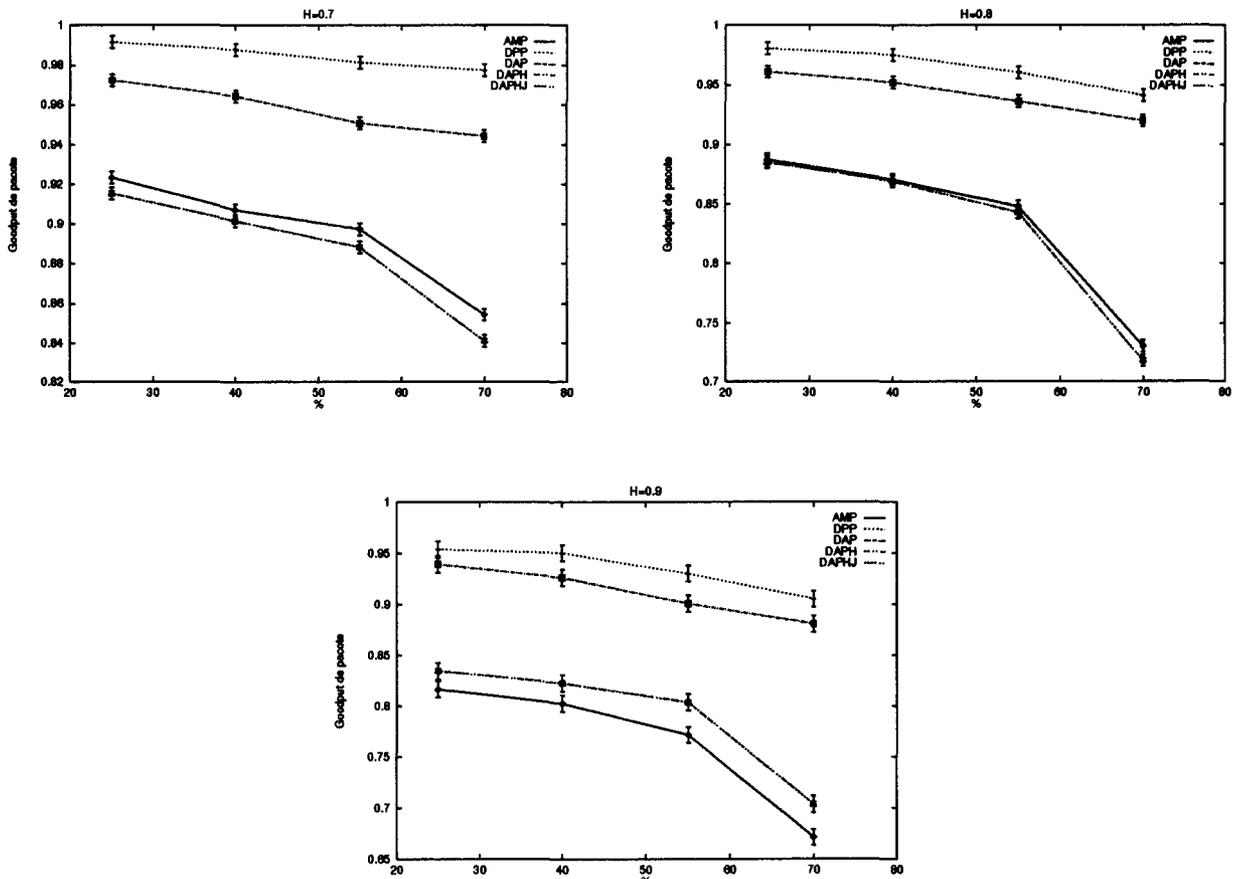


Figura 5.16: *Goodput* de pacote x Proporção de células com tamanho médio de pacote de 170 células ATM para diferentes valores de parâmetro de Hurst em uma rede com 32 fontes, pacotes com tamanho médio de 5, 21, 85 e 170 células ATM.

5.4 Resumo

- Mostrou-se através de resultados de simulação que a política AMP produz o *goodput* de célula mais alto entre todas as políticas. E ainda, para uma razão alta de tamanho médio de pacote pelo tamanho do *buffer* e sob um processo com dependência de curta-duração, a política AMP fornece o *goodput* de pacote mais alto. Contudo, sob um processo com dependência de longa-duração o *goodput* de pacote da política AMP é o menor entre todas as políticas. Sob um processo com dependência de longa-duração a política DPP fornece o *goodput* de pacote mais alto.
- As políticas DAPH e DAPHJ podem fornecer um desempenho muito ruim independente da característica de tráfego.

- A política AMP requer que o tamanho do pacote seja transmitido na primeira célula do pacote. Sabe-se que metade dos dados do usuário (*payload*) de uma célula é estatisticamente perdido ao se dividir um pacote de tamanho variável em um certo número de células de tamanhos fixos. Portanto, o *overhead* pode ser menor que uma célula por pacote. Contudo, durante períodos de congestionamento, células resultantes de *overhead* podem impactar o desempenho da rede.
- A implementação da política AMP em tempo real demanda que se solucione o problema da mochila, que pode ser feito em $O(c * n)$ onde c é o número de espaços de *buffer* a serem colocados à disposição para o pacote que chega e n é o número de pacotes possíveis para serem descartados. Uma vez que a política AMP seleciona pacotes de serem descartados entre pacotes com a primeira célula ainda na fila, a complexidade de *hardware* pode ser reduzida.

Capítulo 6

Conclusão

Desde a publicação do primeiro trabalho acerca de modelagem de tráfego através de processos auto-semelhantes, estes vêm ocupando uma crescente importância na literatura. Apesar da constatação de padrões auto-semelhantes em diversos tipos de tráfego reforçarem o uso destes modelos, é necessário estudar o impacto de se considerar dependências de longa-duração no dimensionamento de redes de comunicação. Ou seja, é preciso esclarecer qual é a influência de correlações distantes no tempo em uma rede com multiplexadores que possuem espaço finito de *buffer* e onde aplicações possuem restrições de tempo real.

Esse trabalho investigou a eficácia de descarte seletivo tanto a nível de pacote quanto a nível de célula sob a influência de processo com dependências de longa-duração. Considerou-se no estudo de descarte seletivo a nível de célula as disciplinas Compartilhamento Total (CT - *Complete Sharing*) e Compartilhamento Total com Garantia de Tamanho Mínimo de Fila (CTGTMF - *Complete Sharing with Guaranteed Queue Minimum*) e as principais políticas de descarte descritas na literatura. Analisou-se através de simulação o impacto das características do processo de entrada, tais como: carga do sistema, parâmetro de Hurst e variância, nas taxas de perda por classe e no tamanho do intervalo de perdas da classe de célula com prioridade baixa. Investigou-se inicialmente descarte seletivo com apenas duas classes de prioridade (baixa e alta) e em seguida investigou-se a influência do uso de múltiplas classes de prioridade. No estudo de descarte seletivo a nível de pacote introduziu-se uma nova política de descarte de pacote, Aceita-Maior-Pacote, e comparou-se essa política com as principais políticas de descarte existentes na literatura. Analisou-se o compromisso entre o *goodput* de célula e o *goodput* de pacote. Estendeu-se a análise também para redes com fontes que geram pacotes de tamanhos médios distintos.

6.1 Principais Contribuições

Pode-se resumir as contribuições da presente dissertação nos seguintes itens:

- Verificou-se que o parâmetro de Hurst tem uma influência significativa na eficácia dos mecanismos de descarte seletivo de células baseado na disciplina Compartilhamento Total e na política de descarte PCPD. Para valores altos de parâmetro de Hurst ($H > 0.8$) não existe uma diminuição significativa na taxa de perda da classe com prioridade alta mesmo para grandes *buffer*s (> 800). Contrariamente, para valores baixos do parâmetro de Hurst, a taxa de perda da classe com prioridade alta diminui ao se aumentar o tamanho do *buffer*. Para filas baseadas em CTGTMF, a taxa de perda da classe com prioridade alta aumenta significativamente com um nível de proteção de 20%. Adicionalmente, não se observou impacto significativo na taxa de perda da classe com prioridade baixa. Essas evidências indicam que o descarte seletivo baseado em CT é claramente vantajoso para valores baixos de parâmetro de Hurst (< 0.8) enquanto não é necessariamente vantajoso para a disciplina de CTGTMF;
- A política de descarte de célula tem um grande impacto nos resultados. A disciplina PCPD-M minimiza a taxa de perda da classe com prioridade alta e pode gerar uma taxa de perda três ordens de grandeza mais baixa do que UCPD. Resultados baseados na PCPD são muito próximos daqueles mostrados pela PCPD-M. A disciplina PCPD tem a vantagem de ser menos complexa de implementar uma vez que requer menos deslocamento de *buffer* do que a PCPD-M. Logo, PCPD é uma opção atrativa para implementação em um comutador ATM. A política PCPD-M sempre produz o valor máximo de comprimento de intervalo de perdas da classe com prioridade baixa seguido de perto pela política PCPD e depois pela política RAND e finalmente pela política UCPD. Verificou-se que a política PCPD-M produz um comprimento máximo de intervalo de perdas que pode ser três vezes maior do que o comprimento máximo de intervalo de perdas da política UCPD;
- Observa-se que com o aumento do tamanho do *buffer* a diferença entre o comprimento máximo de intervalo das perdas das diversas políticas de descarte diminui. Além disso, o comprimento de rajada de perda a classe com prioridade baixa é muito sensível ao parâmetro H . Nota-se também que o mecanismo de proteção de *buffer* (CTGTMF) pode reduzir significativamente o tamanho máximo do intervalo de perdas da classe com prioridade baixa, e que ao se aumentar a carga, $P(\text{alta}/\text{alta})$, ou a variância aumenta-se o comprimento do intervalo de perdas de todas as políticas de descarte;

- Mostrou-se que em uma fila com múltiplos níveis de prioridade pode-se transportar a mesma carga de uma fila com dois níveis de prioridade utilizando-se um tamanho de *buffer* menor. Verificou-se também que em uma fila com quatro níveis de prioridade tem uma flexibilidade maior para lidar com um fluxo com valor alto de variância;
- Os resultados de simulação de políticas de descarte de pacotes mostraram que a política Aceita-Maior-Pacote produz o *goodput* de célula mais alto entre todas as políticas. E ainda, para uma razão alta de tamanho médio de pacote pelo tamanho do *buffer* e sob um processo com dependência de curta-duração, a política Aceita-Maior-Pacote fornece o *goodput* de pacote mais alto. Contudo, sob um processo com dependência de longa-duração o *goodput* de pacote da política Aceita-Maior-Pacote é o menor entre todas as políticas. Sob um processo com dependência de longa-duração a política Descarte Parcial de Pacote fornece o *goodput* de pacote mais alto.

6.2 Trabalhos Futuros

A presente dissertação pode ser estendida em diversas direções:

- A política Aceita-Maior-Pacote requer que o tamanho do pacote seja transmitido na primeira célula do pacote. Sabe-se que metade dos dados do usuário (*payload*) de uma célula é estatisticamente perdido ao se dividir um pacote de tamanho variável em um certo número de células de tamanhos fixos. Portanto, o *overhead* de transmissão do tamanho do pacote pode ser menor que uma célula por pacote. Contudo, durante períodos de congestionamento, células resultantes de *overhead* podem diminuir o desempenho da rede. Pesquisas futuras devem investigar o efeito de células geradas devido ao *overhead*.
- Selecionar um pacote para ser descartado entre os pacotes cujas primeiras células ainda estão na fila ou selecionar entre todos os pacotes enfileirados é o compromisso entre maximizar o *goodput* de célula em um multiplexador isoladamente e “entupir” os próximos multiplexadores com células inúteis. Investigações futuras devem abordar esta questão.
- Esse trabalho traz contribuições específicas para o entendimento do comportamento de mecanismos de controle de congestionamento (mecanismos de descarte seletivo e políticas de descarte de pacotes) sujeitos a processos auto-selhantes. Pesquisas futuras devem investigar a derivação de resultados analíticos destes mecanismos.

- Investigar o relacionamento entre descarte seletivo a nível de pacote e a nível de célula sobre influência de processo com Dependências de Longa-Duração é um tarefa interessante a ser abordada.
- Pesquisar o comportamento de mecanismos de controle de tráfego (mecanismos de controle de admissão e controle de policiamento) relacionado com descarte seletivo sobre influência de processo com Dependências de Longa-Duração.

Bibliografia

- [AEN95] R. G. Addie, A. Elwalid, and T. Neame. Fractal Traffic: Measurements, Modelling and Performance Evaluation. *in Proc. of IEEE INFOCOM*, 1995. Pg. 977-984.
- [AM95] A. Adas and A. Mukherjee. On resource management and QoS guarantees for long range dependent traffic. *Proc. IEEE INFOCOM*, 1995. Pg. 779-787.
- [AMSS82] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Tech. J.*, 61:1871-1894, 1982.
- [BCT91] M. Butto, E. Cavallero, and A. Tonietti. Effectiveness of the leaky bucket policing mechanism in ATM networks. *IEEE Journal on Selected Areas in Communications*, 9:335-342, April 1991.
- [Ber92] J. Beran. Statistical methods for data with long-range dependence. *Statistical Science*, 7:404-427, 1992.
- [BJ76] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. 1976.
- [BSTW95] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Transaction on Communications*, 43(2/3/4):1566-1579, 1995.
- [CB95] Mark E. Crovella and Azer Bestavros. Explaining World Wide Web Traffic Self-Similarity. *Technical Report TR-95-015*, October 1995.
- [CFZ98] R. Choen, N.L.S. Fonseca, and M. Zukerman. Traffic Management and Network Dimensioning. In M. Tatipamula and B. Kasnabish, editors. *Multimedia Communications Networks: Technologies and Services*. Artech House, 1998.
- [CNY73] M. Chi, E. Neal, and G. Young. Practical Applications of Fractional Brownian Motion and Noise to Synthetic Hydrology. *Water Resources Research*, 9:1523-1533, December 1973.

- [Cox84] D. R. Cox. *Long-range dependence: A review*. Ames, IA: Iowa State Univ. Press., 1984. pp. 55-74.
- [Dah89] R. Dahlhaus. Efficient parameter estimation for self-similar processes. *Annals of Statistics*, 17:1749–1766, 1989.
- [FFM97a] N. L. Fonseca, M. J. Ferreira, and G. S. Mayor. On the effectiveness of loss-conserving disciplines under a long-range dependent process. In *Conferência IEEE Global Telecommunications Conference (IEEE Globecom)*, November 1997. Pg. 1102-1107.
- [FFM97b] N. L. Fonseca, M. J. Ferreira, and G. S. Mayor. Um estudo comparativo de políticas de expulsão em multiplexadores ATM sujeitos a processos com dependência de longa duração. In *Segundo Workshop Franco-Brasileiro de Sistema de Informática Distribuído*, November 1997. Pg. 245-256.
- [FS93] N. L. S. Fonseca and J. A. Silvester. Estimating the Loss Probability in a Multiplexer Loaded with Multi-priority MMPP Streams. In *Proc. of IEEE ICC*, May 1993. Pg. 1037-1041.
- [FT86] R. Fox and M. S. Taqqu. Large sample properties of parameter estimates for strongly stationary gaussian time series. *Ann. Statist.*, 14:517–532, 1986.
- [HDLK95a] C. Huang, M. Devetikiotis, I. Lambadaris, and A. R. Kaye. Fast simulation for self-similar traffic in ATM networks. *IEEE ICC*, June 1995. Pg. 438-444.
- [HDLK95b] C. Huang, M. Devetsikiotis, I. Lambadaris, and A. Kaye. Modeling and Simulation of Self-similar Modeling of Variable Bit Rate Compressed Video: A Unified Approach. *Proc. ACM SIGCOM*, 1995. Pg. 114-125.
- [HL86] H. Heffes and D. M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Select. Areas Commun.*, 4:858–868, 1986.
- [Hos84] J. R. Hosking. Modeling Persistence in Hydrological Time Series Using Fractional Differencing. *Water Resources Research*, 20(12):1898–1908, 1984.
- [HRS96] D. Heath, S. Resnick, and G. Samorodnitsky. Heavy tails and long-range dependence in on/off processes and associated fluid models. *Preprint*, 1996. School of ORIE, Cornell University, Ithaca, NY, 1996.
- [Jaf81] J. Jaffe. A decentralized optimal multiple-user flow control algorithm. *IEEE Transaction on Communications*, (29):954–962, 1981.

- [JR86] R. Jain and S. A. Routhier. Packet trains: Measurements and a new model for computer network traffic. *IEEE J. Select. Areas Commun.*, 4:986–995, 1986.
- [Kau93] S. Kauffman. *The origins of Order: Self-Organization and Selection in Evolution*. Oxford University, London, England, 1993.
- [KK80] F. Kamoun and L. Kleinrock. Analysis of shared finite storage in a computer network node environment under general traffic conditions. *IEEE Trans. Commun.*, 28:992–1003, 1980.
- [KM94] S. M. Klivansky and A. Mukherjee. On Long-Range Dependence in NSFNET Traffic. *Tech. Rep. GIT-CC-94-61*, 1994. Georgia Tech.
- [Kol40] A. N. Kolmogorov. Wienersche spiralen and einige andere interessante kurven im hilbertschen raum. *C.R.(Doklady Acad. Sci. USSR (N.S.))*, 26:115–118, 1940.
- [LEWW95] W. Lau, A. Erramilli, J. L. Wang, and W. Willinger. Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and Its Properties. in *Proc. of IEEE ICC*, 1995. Pg. 466-470.
- [LH93] S. Q. Li and C. L. Hwang. Queue Response to Input Correlation Functions: discrete spectral analysis. *IEEE/ACM Transactions on Networking*, 1(15):522–533, October 1993.
- [LO86] W. Leland and T. Ott. Load-balancing heuristics and process behavior. in *PERFORM'86 and ACM SIGM. 1986 Joint Conf. Comput. Perform. Model., Meas., Eval...*, May 1986. Pg. 54-69.
- [LRS97] Y. Lapid, R. Rom, and M. Sidi. Analysis of packet discarding policies in high-speed networks. In *Proc. of INFOCOM*, 1997. Pg. 1191-1199.
- [LTWW93] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On The Self-Similar Nature of Ethernet Traffic. *ACM/SIGCOMM Computer Communications Review*, 23:183–193, September 1993.
- [LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

- [LZ92] P. Landsberg and C. Zukowski. A novel buffer sharing method: Complete sharing subject to guaranteed queue minimums. *in Proc. of The First Conf. Comp. Commun. and Networks*, 1992. Pg. 43-48.
- [Man63] B. Mandelbrot. New methods in statistical economics. *J. Political Economy*, 71(5):421-440, October 1963.
- [Man69] B. B. Mandelbrot. Long-run linearity, locally gaussian processes, h-spectra and infinite variances. *International Economic Review*, 10:82-113, 1969.
- [May97] Gilberto Mayor. *Performance Modeling and Network Management for Self-Similar Traffic*. Ph. D. Dissertattion, University of Southern California, Faculty of the Graduate School, May 1997.
- [MdV96] M. Montgomery and G. de Veciana. On the Relevance of Time Scales in Performance Oriented Traffic Characterizations. *IEEE Proc. of INFOCOM*, 2:513-520, 1996.
- [MH78] A. I. McLeod and K. W. Hipel. Preservation of the rescaled adjusted range: 1. a reassessment of the Hurst phenomenon. *Water Resources Research*, 14(3):491-508, 1978.
- [MN68] B. B. Mandelbrot and J. W. V. Ness. Fractional brownian motions, fractional noises and aplications. *SIAM Review*, 10:422-437, October 1968.
- [Mon94] J. A. Suruagy Monteiro. *Rede Digital de Servicos Integrados de Faixa Larga (RDSI-FL)*. IX Escola de Computacao, July 1994.
- [MS95] G. Mayor and J. A. Silvester. The multi-level leaky bucket mechanism. *In Proc of the 4th IEEE International Conference on Computer Communications and Networks*, 1995.
- [MS96a] G. Mayor and J. A. Silvester. Simulation of an ATM Queueing System with Fractional Brownian Noise Arrival Process. *Proc. of IEEE ICC*, 1996. Pg. 1607-1611.
- [MS96b] G. Mayor and J. A. Silvester. A Trace-Driven Simulation of an ATM Queueing System with Real Network Traffic. *in Proc. of IEEE ICCCN*, 1996.
- [MSA97] G. Mayor, J. A. Silvester, and K. Aleronder. Time Scale Analysis of on ATM Queueing System with Long Range Dependence Traffic. *Proc. of IEEE INFOCON*, 1997. Pg. 205-212.

- [MT79] B. B. Mandelbrot and M. S. Taqqu. Robust R/S analysis of long run serial correlation. *Proc. 42nd Session ISI*, 1979. Pg. 69-99. Book 2.
- [MW69] B. B. Mandelbrot and J. R. Wallis. Computer experiments with fractional gaussian noises. *Walter Resources Research*, 5:228-267, 1969.
- [Nor94] I. Norros. A Storage Model with Self-Similar Input. *Queueing Systems 16*, 16:387-396, 1994.
- [Nor95] I. Norros. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE Journal On Selected Areas in Communications*, 13(6):953-962, August 1995.
- [Pap78] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 3 edition, 1978.
- [PE95] P. Pruthi and A. Erramilli. Heavy-Tailed on/off Source Behavior and Self-Similar Traffic. in *Proc. of IEEE ICC*, 1995. Pg. 445-450.
- [PF95] V. Paxson and S. Floyd. Wide Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transaction on Networking*, 3(3):226-244, June 1995.
- [RE96] B. Ryu and A. Elwalid. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: Myths and realities. in *Proc. of ACM SIGCOMM*, 1996.
- [RF95] Allyn Romanow and Sally Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal of Selected Areas in Communications*, 13(4):633-641, May 1995.
- [Rob94] P. M. Robinson. Semiparametric analysis of long-memory time series. *Ann. Statistic*, (22):515-539, 1994.
- [SFMS96] J. A. Silvester, N.L.S. Fonseca, G. S. Mayor, and S. P. S. Sobral. The effectiveness of multi-level policing mechanisms in ATM traffic control. In *IEEE International Telecommunications Symposium '96*, 1996. Pg. 98-102.
- [SL94] H. D. Sheng and S. Q. Li. Spectral Analysis of Packet Loss Rate at a Statistical Multiplexer of Multimedia Services. *IEEE/ACM Transactions on Networking*, 2(1):53-65, January 1994.
- [SM96] P. I. Sanchez and R. R. Mazumdar. Source policing for ATM systems based on peakedness rate. In *Proc of IEEE INFOCOM*, 1996. Pg. 369-377.

- [ST94] G. Samorodnitsky and M. S. Taqqu. *Stable Non-Gaussian Processes: Stochastic Models with Infinite Variance*. Chapman and Hall, 1994.
- [TG97] Boris Tsybakov and Nicolas D. Georgana. On Self-Similar Traffic in ATM Queues: Definitions, Overflow Probability Bound, and Cell Delay Distribution. *IEEE/ACM Transactions on Networking*, 5(3):397–409, June 1997.
- [THP94] L. Tassiulas, Y. C. Hung, and S. S. Panwar. Optimal Buffer Control during Congestion in a ATM Network Node. *IEEE/ACM Transactions on Networking*, 4(2):374–386, August 1994.
- [TKT+97] N. Tsukutani, K. Kawahara, T. Takine, H. Sunahara, and Y. Oie. Throughput analysis of selective cell discard schemes in transport layer over ATM networks. In *In Proc. of IEEE Globecom*, 1997. Pg. 1067-1074.
- [TL86] M. S. Taqqu and J. B. Levy. *Using renewal processes to generate long-range dependence and high variability*, volume 11. in *Dependence in Probability and Statistics*. E. Eberlein and M. S. Taqqu, eds. Boston, MA:Birkhauser, 1986.
- [TT] Murad S. Taqqu and Vadim Teverovsky. Robustness of Whittle-type Estimators for Time Series with Long-Range Dependence. *to appear in Stochastic Models*.
- [Tur96] Jonathan S. Turner. Maintaining high throughput during overload in ATM switches. In *Proc. INFOCOM*, pages 287–295, April 1996.
- [Vec94] G. Veciana. Leakey buckets and self-tuning rate control. In *In Proc. of IEEE GLOBECOM*, 1994. Pg. 1207-1211.
- [WTE96] W. Willinger, M.S. Taqqu, and A. Erramilli. *A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks*. Stochastic Networks: Theory and Applications, Oxford, U.K.: Oxford Univ., 1996.
- [WTS97] W. Willinger, M. S. Taqqu, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. *Computer Commun. Rev.*, 27:5–23, 1997.
- [WTSW95] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *SIGCOMM*, 1995. Pg. 100-113.

- [WTSW97] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. 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.
- [Yaj85] Y. Yajima. On estimation of long-memory time series models. *Australian Journal of Statistics*, 27:303–320, 1985.
- [YSS95] N. Yamanaka, Y. Sato, and K.I. Sato. Performance limitation of the leaky bucket algorithm for ATM networks. *IEEE Transacion on Communication*, (43):2298–2230, 1995.
- [ZT97] M. Zukerman and P. W. Tse. An adaptive connection admission control scheme for ATM networks. In *Proceeding of IEEE ICC*, Montreal, Canada, June 1997. Pg. 1153-1157.