

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL**

KLEBER KENDY HORIKAWA NABAS

**PROPOSTA DE UM MODELO PARA ANÁLISE DE DESEMPENHO DO
ESCALONADOR WFQ ALIMENTADO COM TRÁFEGO LRD**

TESE

CURITIBA

2009

KLEBER KENDY HORIKAWA NABAS

**PROPOSTA DE UM MODELO PARA ANÁLISE DE DESEMPENHO DO
ESCALONADOR WFQ ALIMENTADO COM TRÁFEGO LRD**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Doutor em Ciências” – Área de Concentração: Telemática.

Orientador: Prof. Dr. Walter Godoy Junior
Co-orientador: Prof. Dr. Emilio Carlos Gomes Wille

CURITIBA

2009

Dados Internacionais de Catalogação na Publicação

- N113 Nabas, Kleber Kendy Horikawa
Proposta de um modelo para análise de desempenho do escalonador WFQ alimentado com tráfego LRD / Kleber Kendy Horikawa Nabas. — 2009.
93 f. : il. ; 30 cm
- Orientador : Walter Godoy Junior
Co-orientador : Emilio Carlos Gomes Wille
Tese (Doutorado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2009
Bibliografia : 79-85
1. Teoria das filas. 2. TCP/IP (Protocolo de rede de computação). 3. Sistemas de telecomunicação. 4. Telecomunicações – Tráfego. 5. Algoritmos. 6. Engenharia elétrica – Teses. I. Godoy Junior, Walter, orient. II. Wille, Emilio Carlos Gomes, co-orient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD (22. ed.) 621.3

Biblioteca Central da UTFPR, Campus Curitiba



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Campus Curitiba



Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

Título da Tese Nº 44:

“Proposta de um Modelo para Análise de Desempenho do Escalonador WFQ Alimentado com Tráfego LRD”

por

Kleber Kendy Horikawa Nabas

Esta tese foi apresentada, às 09h15min do dia **18 de setembro de 2009**, como requisito parcial para a obtenção do título de DOUTOR EM CIÊNCIAS – Área de Concentração: Telemática, pelo Programa de Pós Graduação em Engenharia Elétrica e Informática Industrial – CPGEI –, na Universidade Tecnológica Federal do Paraná – UTFPR. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

Prof. Dr. Emilio Carlos Gomes Wille
(Co-Orientador - UTFPR)

Prof. Dr. Luiz Nacamura Júnior
(UTFPR)

Prof.ª Dr. Anélise Munaretto Fonseca
(UTFPR)

Prof. Dr. Luiz Carlos Pessoa Albini
(UFPR)

Prof. Dr. Anilton Salles Garcia
(UFES)

Prof. Dr. Carlos Marcelo Pedroso
(PUC-PR)

Visto da coordenação:

Prof. Dr. Humberto Remígio Gamba
(Coordenador do CPGEI)

AGRADECIMENTOS

Ao Professor Doutor Walter Godoy Júnior, pelo voto de confiança dada a minha pessoa como seu aluno de doutorado e pelo incentivo à produção desta tese.

Ao Professor Doutor Emilio Carlos Gomes Wille, que sempre esteve ao meu lado nos momentos mais difíceis deste trabalho com suas orientações e sugestões.

Aos colegas de laboratório que participaram direta e indiretamente na conclusão desta tese, em especial ao Lincoln Herbert Teixeira e ao Murilo Santos Holtman.

À minha família que sempre me incentivou, em especial a minha esposa Claudia, à minha filha Fernandinha e a minha sogra Dona Clementina que durante muitos dias ficou cuidando da minha filha para que eu pudesse vir às aulas do doutorado.

RESUMO

NABAS, Kleber K. Horikawa. Proposta de um modelo para análise de desempenho do escalonador WFQ alimentado com tráfego LRD. 2009. 93 f. Tese — Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2009.

Técnicas de gerenciamento de filas e descarte de pacotes são empregadas pelos escalonadores de pacotes presentes nos roteadores internos de uma rede de telecomunicações. Devido às características auto-similares do tráfego presente nas redes, o desenvolvimento de modelos para análise de desempenho de protocolos e de elementos de rede é enormemente dificultado. Tais características auto-similares são oriundas da ação dos protocolos predominantes nas redes em questão, e dos mecanismos fim-a-fim de controle de congestionamento existentes, determinantes no comportamento do fluxo de informações entre diferentes camadas na hierarquia de protocolos TCP/IP. Esta tese tem como objetivo principal propor modelos analíticos de baixo esforço computacional que permitam fazer uma análise de desempenho de um nó de rede (podendo este nó ser do tipo: *DiffServ*, *IntServ*, *IP*, *Multi Protocol Label Switching*, entre outros) com o escalonador WFQ. O tráfego considerado é do tipo TCP, devido ao fato que este protocolo é o mais utilizado na Internet atualmente, apresentando, portanto, características auto-similares. Foram desenvolvidos modelos para o escalonador WFQ quando o mesmo é alimentado com tráfego Poissoniano e tráfego do tipo auto-similar. O primeiro modelo está baseado em uma cadeia de Markov bidimensional para representar o comportamento do escalonador WFQ alimentado por dois fluxos de prioridades diferentes. O segundo modelo faz uso de duas filas separadas com taxas de serviço ajustadas de modo a simular o comportamento do escalonador WFQ. As filas são do tipo $M_{[X]}/M/1/B$ para levar em conta o comportamento auto-similar do tráfego. Como resultado, mostra-se que o segundo modelo é capaz de estimar as métricas de desempenho, com erro aceitável, dentro de uma faixa adequada de valores dos parâmetros do sistema, conforme demonstra a validação de resultados conduzido com uso do software NS-2.

Palavras-chave: WFQ. Tráfego LRD. Tráfego de Poisson.

ABSTRACT

NABAS, Kleber K. Horikawa. Proposal of a model for performance analysis of the WFQ scheduler with LRD traffic. 2009. 93 p. Thesis — Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Federal University of Technology of Paraná. Curitiba, 2009.

Internal routers of telecommunication networks employ a variety of queue management and packet discarding techniques. The development of models for analyzing performance of different protocols and devices is made difficult by the self-similar nature of network traffic. Traffic self-similarity stems from the action of predominant protocols and end-to-end congestion control systems, which are key factors in determining the behavior of data flow between layers of the TCP/IP protocol stack. This thesis proposes analytical models that allow for the evaluation of a (*DiffServ*, *IntServ*, *IP*, *Multi Protocol Label Switching* etc.) network node's performance at a low computational cost. Only TCP traffic is considered, since it accounts for most of current Internet traffic and therefore exhibits self-similar characteristics. Models were developed for the WFQ scheduler with Poisson and self-similar traffic inputs. The former is based on a bi-dimensional Markov chain representing the behavior of the WFQ scheduler with two input flows with different priorities. The latter consists in two separate queues whose service rates are tuned so as to simulate the behavior of a WFQ scheduler. $M_{[X]}/M/1/B$ queues are used to account for the self-similar nature of traffic. As a result, it is shown that the second model accurately estimates performance metrics for certain ranges of parameter values, as confirmed by NS-2 simulations.

Keywords: WFQ. LRD Traffic. Poisson Traffic.

LISTA DE ACRÔNIMOS

ACK	Acknowledgment
AFQ	Adaptive Fair Queue
CBQ	Class Based Queueing
CDMA	Code Division Multiple Access
CDRD	Combined Delay and Rate Differentiation
CMTC	Cadeia de Markov em Tempo Contínuo
CoS	Class of Service
CQ	Custom Queueing
CSA	Cardwell, Savage e Anderson
DiffServ	Differentiated Service
DSCP	DiffServ Code Point
FIFOQ	First In First Out Queueing
FQ	Fair Queueing
GPS	Generalized Processor Sharing
IETF	Internet Engineering Task Force
IntServ	Integrated Service
IP	Internet Protocol
ISP	Internet Service Provider
LLQ	Low Latency Queueing
LRD	Long Range Dependence
MAC	Media Access Control
MPLS	Multi Protocol Label Switching
NS	Network Simulator
PGPS	Packet Generalized Processor Sharing
PHB	Per Hop Behavior

Ploss	Probabilidade de Descarte de Pacotes
PQ	Priority Queueing
PSLR	Proportional Sharing with Load Reservation
QoS	Quality of Service
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTT	Round Trip Time
SRD	Short Range Dependence
TCP	Transmission Control Protocol
ToS	Type of Service
TTL	Time to Live
T-WFQ	Timestamp-update Weighted Fair Queueing
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
WiMAX	Worldwide Inter-operability for Microwave Access
WFQ	Weighted Fair Queueing
WRR	Weighted Round Robin
WWW	Wide World Web

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivos	15
1.2 Metodologia Utilizada	16
1.3 Organização da Tese	16
2 TRÁFEGO AUTO-SIMILAR EM REDES	17
2.1 Introdução	17
2.2 Processos Estocásticos Auto-Similares	19
2.3 Processos Auto-Similares de Segunda Ordem	22
2.4 Estimação do Parâmetro de Auto-Similaridade	25
2.4.1 Estimador de Whittle	25
2.4.2 Estatística R/S	26
2.4.3 Estimador Baseado em <i>Wavelets</i>	26
2.5 Geração de Tráfego Auto-Similar	27
2.6 Conclusão do Capítulo	32
3 ESCALONADOR WFQ	33
3.1 Introdução	33
3.2 Algoritmos de Escalonamento de Pacotes	34
3.2.1 FIFOQ (<i>First-In First-Out Queueing</i>)	35
3.2.2 FQ (<i>Fair Queueing</i>)	36
3.2.3 CQ (<i>Custom Queueing</i>)	36
3.2.4 GPS (<i>Generalized Processor Sharing</i>)	36
3.2.5 WFQ (<i>Weighted Fair Queueing</i>)	36
3.2.6 WRR (<i>Weighted Round Robin</i>)	37
3.2.7 PQ (<i>Priority Queueing</i>)	37
3.3 O Escalonador WFQ	37
3.4 Aspectos Analíticos do Escalonador WFQ	41
3.5 Exemplo do Funcionamento do Escalonador WFQ	42
3.6 Conclusões do Capítulo	46
4 PROPOSTA DE MODELOS ANALÍTICOS PARA UM NÓ DE REDE COM O ESCALONADOR WFQ	48
4.1 Introdução	48
4.2 Modelo do WFQ com Tráfego de Poisson	49
4.2.1 Resolução da Cadeia de Markov	54
4.2.2 Cálculo das Métricas	55
4.2.3 Resultados Numéricos	55
4.3 Modelo WFQ com Tráfego Auto-Similar	59
4.3.1 Formulação Matemática da Fila $M[X]/M/1/B$	61
4.4 Cenário para Simulação	62
4.5 Comparação de Resultados entre o Modelo Analítico e a Simulação	63
4.5.1 Resultados Obtidos na Camada de Rede	63
4.5.2 Resultados Obtidos Variando o Tamanho do <i>Buffer</i>	70
4.5.3 Resultados Obtidos na Camada de Transporte	72
4.6 Conclusão do Capítulo	74
5 CONCLUSÃO	76
REFERÊNCIAS	79
ANEXO A — PROGRAMA PARA CÁLCULO DA AUTO-CORRELAÇÃO	86

ANEXO B — PROGRAMA PARA CÁLCULO DO PARÂMETRO DE HURST PELO MÉTODO R/S	88
ANEXO C — PSEUDOCÓDIGO PARA RESOLUÇÃO DA CADEIA DE MARKOV DA FILA M/M/1/B	91
ANEXO D — MODELO ANALÍTICO PARA O TCP	92

1 INTRODUÇÃO

O explosivo crescimento numérico e geográfico da Internet requer a necessidade de uma séria mudança para os provedores de serviços e para os fornecedores de equipamentos, diante do grande volume de tráfego de informações que nela trafegam. Há ainda uma insatisfação por parte dos usuários, uma vez que estes utilizam cada vez mais serviços multimídia, que necessitam de diferentes níveis de QoS (*Quality of Service*). Por essa razão, há um desejo, por parte dos usuários, para que se criem diferentes tipos de serviços IP (*Internet Protocol*) na Internet, e, diante desse cenário, a IETF (*Internet Engineering Task Force*) não tem medido esforços em pesquisas sobre a QoS para que estas medidas possam melhorar as redes de transmissão de dados.

Os mecanismos de oferecimento de QoS podem ser definidos como sendo um conjunto de técnicas para gerenciar a largura de banda da rede, o atraso, o *jitter* e o descarte de pacotes. Para adicionar QoS às redes IP existem duas arquiteturas definidas pelo IETF: *IntServ* (*Integrated Service*) e *DiffServ* (*Differentiated Services*).

O *IntServ* mantém a qualidade fim-a-fim para fluxos individuais com o auxílio do protocolo de reserva de recursos RSVP (*Resource Reservation Protocol*), o qual exige muita sinalização e não possui uma escalabilidade para grandes redes (BRADEN; CLARK; SHENKER, 1994).

Na arquitetura *DiffServ*, os pacotes que entram na rede são agrupados em diferentes classes. Cada classe tem uma marca a ela associada, utilizando os bits DSCP (*DiffServ Code Point*). Essa classificação não exige muita sinalização, tornando o *DiffServ* extremamente escalável e ainda assegura aos fluxos agregados largura de banda apropriada e garantias de atraso no interior da rede. Essa arquitetura se fundamenta no princípio de se concentrar na borda da rede a complexidade das operações envolvidas no tratamento do tráfego, de modo que o núcleo da rede permaneça o mais simples possível. Assim, os roteadores de borda dessa arquitetura são responsáveis por monitorar, classificar e marcar os pacotes a serem inseridos na rede. Cada roteador no interior da rede implementa técnicas de gerenciamento de filas e descarte de pacotes (*escalonador de pacotes*), baseados na marcação que o pacote carrega, ou seja, seu PHB (*Per Hop Behavior*) (LI; REKHTER, 1998).

A função básica do escalonador é reordenar a saída das filas visando a diferenciação de serviços e isso pode ser realizado de diversas maneiras: a forma mais simples é utilizar a estratégia de prioridade, onde os pacotes são ordenados pela sua prioridade; outra forma é o

escalonador denominado cíclico (*round-robin*), que atribui acessos distintos aos pacotes com classes diferentes para compartilhar um enlace.

Dentre os escalonadores mais conhecidos destaca-se o escalonador justo ponderado (WFQ - *Weighted Fair Queueing*). Esse escalonador, também denominado PGPS (*Packet Generalized Processor Sharing*), permite que várias filas de espera sejam servidas simultaneamente. Cada fila utiliza, em cada instante, uma percentagem da capacidade do enlace que é proporcional a um peso que lhe é atribuída.

Muitos escalonadores são derivados do WFQ padrão como por exemplo: *WFQ-based CBQ* (MASI; FISCHER; GARBIN, 2007), *Adaptive Fair Queue* (AFQ) (DEKERIS; NARBUTAITE; ADOMKUS, 2007), *Timestamp-update WFQ* (T-WFQ) (WANG; FAN; TAFAZOLLI, 2008), *WFQ Scheduling Discipline with Low Latency Queue* (LLQ) (DEKERIS; ADOMKUS; BUDNIKAS, 2006), *Combined Delay and Rate Differentiation* (CDRD) (comumente usado em comunicações via satélite) (DU; FAN; EVANS, 2007), *Proportional Sharing with Load Reservation* (PSLR) (usado em redes Ethernet ópticas passivas) (PEREIRA; FONSECA; ARANTES, 2007). O algoritmo WFQ está sendo amplamente utilizado com redes sem fio, onde o escalonador faz a priorização dos pacotes através do endereço de MAC (*Media Access Control*) das estações de trabalho (CASONE, 2007) (LU; HUANG; LIN, 2007), e também é usado em redes de telefonia móvel com tecnologia CDMA e além desta com tecnologia 3G (ALQAHTANI, 2007), redes militares (CHANG, 2007) e WiMAX (LIN; CHOU; LIU, 2008) (RATH; KARANDIKAR; SHARMA, 2008) (ALI; HAYAJNEH; HASSANEIN, 2008) e VoIP (ANDREWS; ZHANG, 2008) (FGEE; PHILLIPS; ROBERTSON, 2008) (FGEE et al., 2008).

Do ponto de vista de análise de desempenho do escalonador WFQ pode-se citar os trabalhos de (HORVÁTH; TELEK, 2003), (HEDGE; AVRACHENKOV, 2004) e (SOO; CHUNG, 2002).

O artigo de (HORVÁTH; TELEK, 2003), salienta-se que o escalonador WFQ é aplicado em muitos computadores e sistemas de telecomunicações, sendo esta forma muito popular para compartilhar recursos. Sua solução analítica eficiente é uma questão em aberto há muito tempo. Diferentes soluções foram propostas usando análises complexas ou técnicas numéricas, mas todos esses métodos têm suas limitações. No artigo de (HORVÁTH; TELEK, 2003) é apresentada uma abordagem simples que fornece um método para estimar o comprimento e tempo de espera no sistema WFQ. Ainda que simples, esse método

proporciona uma boa aproximação. O método consiste em separar o sistema de duas filas, que compartilham recursos de um único servidor, em dois sistemas independentes.

O artigo de (HEDGE; AVRACHENKOV, 2004) analisa um sistema WFQ de duas filas onde as mesmas são alimentadas por um tráfego Poissoniano. Esse modelo está baseado em uma cadeia de Markov bidimensional para representar o comportamento do escalonador WFQ alimentado por dois fluxos de prioridades diferentes. O modelo proposto é preciso para se calcular o tempo médio de espera no sistema e a probabilidade de descarte de pacotes.

Em (SOO; CHUNG, 2002) o atraso e a probabilidade de descarte de pacotes na fila de um nó de uma rede MPLS são analisadas através de um modelo de fila finita, não preemptiva, M/M/m/B, com chegada de pacotes em grupos de tamanho fixo (rajadas). Os resultados analíticos revelaram uma relação do tempo de fila do tráfego de dados com as classes prioritárias conforme a classificação do MPLS. Os pacotes de baixa prioridade recebem o serviço melhor esforço sofrendo atrasos significativos. Portanto, as equações obtidas em (SOO; CHUNG, 2002) servem como diretriz e podem ser aplicadas na topologia MPLS (na camada 2 e camada 3 no modelo TCP/IP) para garantir serviços diferenciados reduzindo o atraso dos pacotes de acordo com a prioridade da classe.

Outro fator que tem impacto decisivo no desempenho das redes de transmissão de dados é o perfil do tráfego. O tráfego que chega em um nó de rede não é um tráfego que possa ser modelado pelo processo de Poisson. O tráfego que melhor modela a transferência de dados reais na Internet apresenta um comportamento auto-similar (PAXSON; FLOYD, 1994).

Recentemente os conjuntos fractais passaram a receber atenção especial, pois uma série de trabalhos mostrou que diversos objetos naturais são mais bem descritos quando modelados como fractais. Exemplos típicos são: a superfície das nuvens, superfícies topográficas, turbulência em fluidos, entre outros (MANDELBROT, 1982). As pesquisas sobre o tráfego de redes cruzaram com a teoria dos fractais a partir da publicação do trabalho de Leland (LELAND et al., 1994). Leland et al. constataram experimentalmente que o tráfego coletado na rede Ethernet do *Bellcore Morristown Research and Engineering Center* exibia propriedades fractais tais com auto-similaridade e dependência de longa duração. Estudos posteriores constataram a presença de propriedades fractais também no tráfego gerado por transmissão de vídeo a taxa variável (GARRETT; WILLINGER, 1994) (BERAN, 1995), no tráfego em redes WAN (*Wide Area Network*) (PAXSON; FLOYD, 1994), e no tráfego WWW (*World Wide Web*) (CROVELLA; BESTAVROS, 1996), dentre outros. Foi constatado que tais propriedades, com destaque para a dependência de longa duração, influenciam fortemente

o desempenho das redes (NORROS, 1995), além de trazer implicações relevantes em diversas áreas, tais como teoria de filas (NORROS, 1994) e projeto de redes. A distribuição de cauda pesada, que modela o tamanho das sessões ou conexões que originam o tráfego agregado, é apontada como causa da característica auto-similar observada (CROVELLA; BESTAVROS, 1996) (PARK; WILLINGER, 2000).

Diferentes modelos matemáticos foram propostos na literatura com o objetivo de representar a característica auto-similar constatada no tráfego de redes. O modelo de tráfego Browniano fracionário (fBt) (NORROS, 1995) tornou-se muito utilizado por ser o modelo matemático que mais facilmente incorpora as características auto-similares observadas no tráfego. Entretanto, observou-se que enquanto em escalas de tempo da ordem de centenas de milissegundos e maiores, o comportamento do tráfego era bem representado por modelos auto-similares, em escalas de tempo da ordem de centenas de milissegundos e menores, as características de tais modelos afastavam-se das apresentadas pelo tráfego real. Investigações envolvendo tráfego WAN TCP/IP constataram que as diferentes propriedades do tráfego observadas em pequenas escalas de tempo eram convenientemente descritas utilizando-se análise multifractal (RIEDI; VÉHEL, 1997) (FELDMANN et al., 1998). Tais propriedades são oriundas da ação dos protocolos predominantes nas redes em questão, e dos mecanismos fim-a-fim de controle de congestionamento existentes, determinantes no comportamento do fluxo de informações entre diferentes camadas na hierarquia de protocolos TCP/IP (FELDMANN; GILBERT; WILLINGERT, 1998).

Abordagens alternativas foram propostas para incorporar as características auto-similares do tráfego na análise e dimensionamento de redes.

Em (GARETTO, 2004) foi desenvolvida uma ferramenta analítica para o projeto e configuração de redes IP baseadas na arquitetura da Internet sendo que o tráfego que alimenta este sistema tem um comportamento auto-similar. A disponibilidade dessa ferramenta permite que operadoras de telecomunicações possam melhor projetar suas redes, com vantagens em termos de redução de custos e taxas e de garantia de uma melhor QoS para o usuário. A principal contribuição de (GARETTO, 2004) é a demonstração de que a abordagem tradicional de modelagem, baseada em processos Markovianos e na teoria de filas padrão, pode ser aplicada com sucesso na análise de sistemas complexos de comunicação de dados (tais como a Internet). Extensivos testes realizados em (GARETTO; TOWSLEY, 2003) mostraram que a fila $M_{[X]}/M/1/B$ com chegada de pacotes em grupo, é capaz de fornecer indicadores de desempenho importantes como, por exemplo, o atraso médio e a probabilidade

de descarte de pacotes na fila, quando o tráfego de entrada é caracterizado principalmente pela transmissão TCP.

Observa-se que os artigos (HORVÁTH; TELEK, 2003), (HEDGE; AVRACHENKOV, 2004) e (SOO; CHUNG, 2002) não levam em conta características auto-similares do tráfego. Por outro lado, os trabalhos de Garetto tratam os fluxos de forma igual, independentemente de suas prioridades, buscando-se analisar, por meios analíticos e de simulação, o desempenho de redes sujeitas ao tráfego TCP sem levar em consideração o tratamento diferenciado de fluxos de pacotes (GARETTO, 2004).

1.1 Objetivos

Esta tese tem como objetivo principal propor um modelo analítico de baixo esforço computacional que permita fazer uma análise de desempenho de um nó de rede (podendo este nó ser do tipo: *DiffServ*, *IntServ*, *IP*, *Multi Protocol Label Switching* entre outros) com o escalonador WFQ. O tráfego considerado nessa tese é do tipo TCP, devido ao fato que este protocolo é o mais utilizado na Internet atualmente (LUO; MARIN, 2005) (MEISS; MENCZER; FORTUNATO, 2008) (ERMAN et al., 2007) (BASHER et al., 2008). O intervalo de variação com relação ao fator de utilização de interesse para este modelo está compreendido entre 0,40 a 0,85 visto que estes valores são normalmente encontrados pelas operadoras de telecomunicações (NUCCI et al., 2007) (CISCO, 2009) e o tamanho do *buffer* de interesse concentra-se próximo de 80 pacotes (GARETTO, 2004) (BUSH; MEYER, 2002) (BEHESHTI et al., 2008) (BARMAN; SMARAGDAKIS; MATTA, 2004).

Na proposta o escalonador WFQ é estudado como se fosse composto de duas filas independentes aonde cada fila considera o modelo Markoviano com chegada em grupos $M_{[X]}/M/1/B$, onde $[X]$ representa a distribuição do tamanho do grupo (em pacotes), com configurações de parâmetros de tempo de serviço ajustados para modelar o comportamento das filas de baixa e alta prioridades. Este modelo é validado através da comparação com os valores obtidos utilizando-se o simulador NS-2 (NS, 2007), aonde os valores comparados são: RTT (*Round Trip Time*), Ploss (probabilidade de descarte de pacotes) e Latência para transferência de arquivos. A escolha deste modelo de fila deve-se ao fato de que seus indicadores de desempenho mostram boa concordância com os resultados de simulação (GARETTO; TOWSLEY, 2003) (GARETTO, 2004).

Como objetivo secundário esta tese propõe um modelo analítico do escalonador WFQ alimentado com tráfego de Poisson. Este modelo está baseado na abordagem proposta por

Hedge e Avrachenkov (HEDGE; AVRACHENKOV, 2004), conhecido por modelo do particionamento completo. Com este modelo pode-se provar a importância de se considerar corretamente o perfil do tráfego que alimenta o nó da rede.

No que diz respeito as simulações o processo escolhido para a geração de tráfego sintético, utilizando o software *Network Simulator* (NS-2) (NS, 2007), consiste na agregação do tráfego gerado por diversas conexões TCP que transmitem arquivos cuja dimensão é modelada por uma distribuição de cauda pesada. Essa distribuição foi obtida por medidas realizadas na rede ao longo de uma semana (ROSSI; MUSCARIELLO; MELLIA, 2004). Dessa forma o tráfego simulado apresenta características de dependência de longa duração e auto-similaridade.

1.2 Metodologia Utilizada

Foi realizado um estudo bibliográfico sobre o estado da arte na utilização do escalonador WFQ em redes IP. Foram considerados principalmente os artigos/trabalhos de Garetto, Horváth & Telek e Hedge & Avrachenkov.

A revisão bibliográfica foi realizada continuamente durante o desenvolvimento deste trabalho, devido à atualidade do tema.

Algumas disciplinas cursadas durante o doutorado foram especialmente úteis para a execução deste trabalho, com destaque para Teoria de Filas 1 e Teoria de Filas 2.

1.3 Organização da Tese

Além desta seção introdutória, esta tese está organizada da seguinte maneira. No Capítulo 2 são apresentados os conceitos de auto-similaridade bem como o cálculo do parâmetro de Hurst para o tráfego sintético gerado na simulação usando o NS-2. O Capítulo 3 mostra o funcionamento do escalonador WFQ, visto que este escalonador foi o escolhido para priorização de usuários. No Capítulo 4 são apresentados dois modelos analíticos para um nó de rede com o escalonador WFQ, o primeiro alimentado com tráfego Poissoniano e o segundo alimentado por um tráfego auto-similar. Nesse mesmo capítulo serão comparados os resultados obtidos por meio do simulador NS-2 e do modelo proposto. No Capítulo 5 são apresentadas as conclusões sobre esta tese e as sugestões relativas à continuidade deste trabalho.

2 TRÁFEGO AUTO-SIMILAR EM REDES

2.1 Introdução

Auto-similaridade e fractais são noções primeiramente estudadas por Benoit B. Mandelbrot (MANDELBROT; VAN NESS, 1982) para descrever objetos muito irregulares para se adequarem à geometria euclidiana tradicional. Num sentido determinístico, uma forma geométrica é fractal ou auto-similar, caso preserve a mesma aparência quando observada sobre todas as escalas no espaço ou no tempo. O conjunto de Cantor é um fractal. A Figura 1 ilustra cinco iterações da construção recursiva e infinita do conjunto de Cantor, de acordo com as seguintes regras:

- 1) Seja um segmento no intervalo $[0,1]$;
- 2) Divida o segmento anterior em três partes iguais e remova a parte central;
- 3) A cada nova iteração, remova a parte central dos segmentos resultantes do passo anterior.

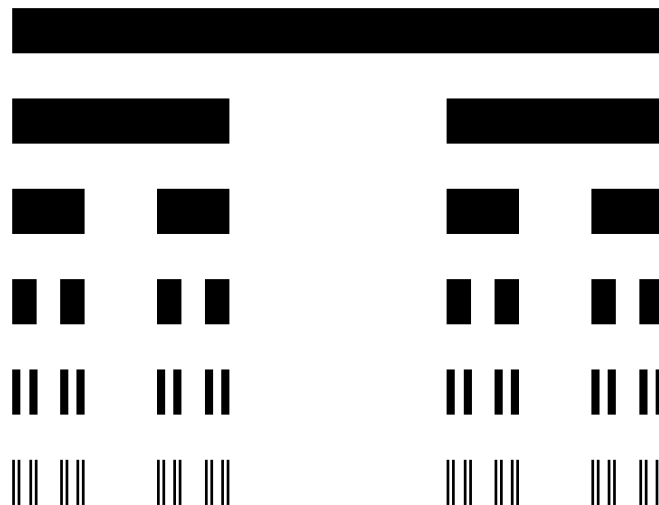


Figura 1 – Conjunto de Cantor com cinco iterações

Segundo Freitas (2009), a auto-similaridade descreve a geometria de objetos no qual uma pequena parte quando expandida parece a parte inteira do objeto. Muitos objetos na natureza são auto-similares. Qualquer que seja a ampliação considerada, obtêm-se sucessivas cópias do objeto inicial (FREITAS, 2009).

Num passado recente, conjuntos não suficientemente suaves ou regulares, tais como os fractais, tendiam a ser ignorados por serem considerados objetos matemáticos pouco merecedores de estudo. Assim como o interesse pela geometria euclidiana foi estimulado por

suas aplicações nas ciências naturais, a geometria dos conjuntos fractais também ganhou importância a partir de trabalhos que mostraram que diversos objetos naturais podem ser descritos quando vistos como fractais. Exemplos típicos são: a superfície das nuvens, superfícies topográficas, turbulências em fluidos, etc. (MANDELBROT, 1982). Ao se perceber que conjuntos irregulares provêm uma representação melhorada de diversos fenômenos naturais, houve uma mudança de atitude por parte da comunidade científica e compreendeu-se a importância da matemática dos conjuntos fractais. A geometria fractal provê, portanto, um sistema para o estudo de conjuntos irregulares.

No contexto das redes de comunicações, a auto-similaridade ganhou importância após o estudo de Leland et al., que constataram a natureza auto-similar do tráfego em redes Ethernet. Este estudo, baseado na análise do tráfego do *Bellcore Morristown Research and Engineering Center*, mostra que o tráfego Ethernet é composto pela alternância de períodos de surtos e de suavidade, o que é observado em diversas escalas de agregação. Este comportamento não pôde ser adequadamente reproduzido pelos modelos baseados em processos estocásticos markovianos, até então utilizados para dimensionamento e análise de desempenho de redes (LELAND et al., 1994).

As causas da presença de auto-similaridade no tráfego ainda não foram determinadas de modo conclusivo. Inicialmente, ela era considerada consequência do comportamento de fontes isoladas cujo tráfego é reconhecidamente auto-similar (e.g. transmissão de vídeo comprimido). No entanto, Willinger et al. (WILLINGER et al., 1995), demonstraram que um fluxo de tráfego auto-similar pode ser obtido pela superposição de muitas fontes ON/OFF identicamente distribuídas, desde que os períodos de silêncio e/ou de atividade sejam representados por variáveis aleatórias com distribuição de cauda longa, i.e., que podem assumir valores elevados com probabilidade não-negligenciável (efeito Noé – alta variabilidade ou variância infinita). Esse resultado é comprovado por outros estudos, que mostram que a agregação do tráfego de diferentes aplicações envolvendo transferência de arquivos (por exemplo aplicações cliente-servidor e *World-Wide Web*) pode resultar em um fluxo auto-similar se o tamanho dos arquivos segue uma distribuição subexponencial (CROVELLA; BESTAVROS, 1997) (PARK; WILLINGER, 2000) (PARK; KIM; CROVELLA, 1996).

Do ponto de vista da análise e do dimensionamento de redes, ainda não há um consenso em relação à relevância da auto-similaridade. Alguns autores argumentam que, sob certas condições, ela tem pouca influência no comportamento do tráfego (HEYMAND;

LAKSHMON, 1996). No entanto, a grande maioria dos estudos mostra que a auto-similaridade pode degradar o desempenho das redes de forma significativa. Neste caso, a incidência de surtos de tráfego tende a ser mais alta e o decaimento da distribuição da ocupação dos *buffers* dos nós de comutação tendem a ser muito lento (MORIN, 1995) (PARK; KIM; CROVELLA, 1996) (PAXSON; FLOYD, 1994) (PARK; WILLINGER, 2000) (PARK; KIM; CROVELLA, 1997). Isto significa que:

- a taxa de perda de pacotes por transbordo cresce rapidamente com o grau de auto-similaridade e não é possível obter significativa redução com o aumento dos *buffers* nos comutadores;

- a alta incidência de surtos no tráfego auto-similar reduz a eficiência da multiplexação estatística no aumento do uso efetivo da capacidade de transmissão e

- os atrasos de transferência tendem a aumentar significativamente, já que a ocupação média dos *buffers* cresce rapidamente com o grau de auto-similaridade do tráfego.

Modelos de tráfego com dependência de longa duração se baseiam principalmente em processos auto-similares. O termo auto-similar se refere normalmente a processos assintoticamente auto-similares de segunda ordem ou monofractais (PARK; WILLINGER, 2000).

O parâmetro de Hurst mede o grau de auto-similaridade do processo e representa basicamente uma medida do decaimento da função de auto-correlação do processo.

2.2 Processos Estocásticos Auto-Similares

Matematicamente, pode-se definir um processo estocástico auto-similar da seguinte forma:

Definição 2.1 [Processo estocástico auto-similar (PRATA, 2003)]. Seja um processo estocástico $Y(t)$, $t \in R$, onde R representa o conjunto dos números reais. Este processo é dito auto-similar com parâmetro de auto-similaridade $H \in (0;1)$ se, para todo $a > 0$ e $t \geq 0$, os processos $Y(t)$ e $a^{-H}Y(at)$ são identicamente distribuídos, i.e.

$$Y(t) \stackrel{d}{=} a^{-H}Y(at) \quad (1)$$

onde o símbolo $\stackrel{d}{=}$ denota a igualdade em distribuição.

Um exemplo de processo auto-similar é mostrado na Figura 2, onde se observa que o processo não muda suas propriedades apesar da mudança de escala.

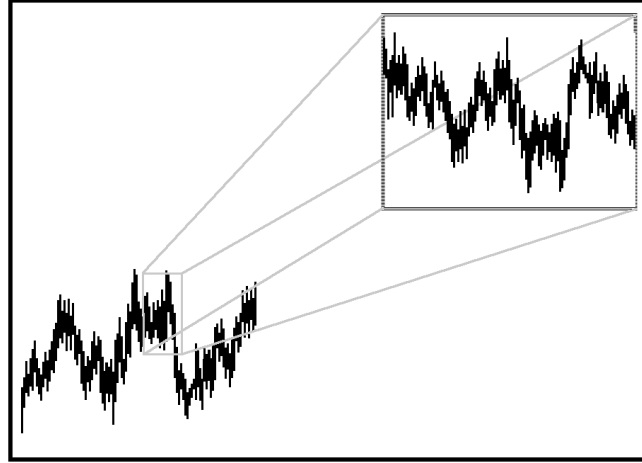


Figura 2 – Auto-similaridade estatística. Uma parte dilatada da série não pode ser estatisticamente distinguida de toda a série (PARK; WILLINGER, 2000).

Definição 2.2 (PRATA, 2003). Para um processo $Y(t)$ que seja auto-similar,

$$Y(t) \stackrel{d}{=} t^H Y(1) \quad (2)$$

o que pode ser verificado substituindo, na Equação (1), $a = t^{-1}$.

O parâmetro de auto-similaridade H é denominado parâmetro de Hurst, em homenagem a H. E. Hurst (HURST; BLACK; SIMAIKA, 1965). Um processo $Y(t)$ que seja auto-similar, segundo a Definição 2.1, é obrigatoriamente não estacionário (a menos que seja degenerado (i.e. $Y(t) = 0$, para todo $t \in R$)). No contexto da modelagem de tráfego, este processo representa o volume de tráfego acumulado até o instante t sendo, portanto, denominado processo de acumulação. A este processo pode ser associado um outro denominado processo de incrementos, que representa o volume de tráfego acumulado no intervalo $[t_0; t_0 + t]$. O conceito de auto-similaridade pode também ser definido para processos de incrementos (MANDELBROT; VAN NESS, 1968).

Definição 2.3 (PRATA, 2003). O processo de incremento $X(t)$, $t \in R$, associado a um dado processo $Y(t)$, é dito auto-similar com parâmetro de auto-similaridade $H \in (0;1)$ se, para todo $a > 0$ e $t_0 \geq 0$:

$$X(t) = Y(t_0 + t) - Y(t_0) \stackrel{d}{=} a^{-H} [Y(t_0 + at) - Y(t_0)] \quad (3)$$

É possível demonstrar que, se o processo $Y(t)$ é auto-similar, o correspondente processo de incrementos $X(t)$ também é auto-similar. No contexto da modelagem de tráfego, é interessante considerar apenas os processos $Y(t)$ cujo processo de incrementos $X(t)$ seja estacionário, ao menos no sentido amplo. Dessa forma, o processo $X(t)$ pode ser discretizado e utilizado para representar o volume de tráfego em um instante $t \in Z$, onde Z representa o conjunto dos números inteiros. Neste sentido, seja $Y(t)$ um processo auto-similar com incrementos estacionários, para o qual $E\{Y(t)\} = 0$. A partir do correspondente processo de incrementos em tempo discreto, $X(t)$, $t \in Z$, podem ser obtidos processos agregados $X^{(m)}(t)$, $t \in Z$, definidos como a média amostral do processo original $X(t)$ em blocos, não sobrepostos, de tamanho m , i.e.

$$X^{(m)}(t) = \frac{1}{m} \sum_{i=m(t-1)+1}^{mt} X(i) \quad (4)$$

Dado que o processo $X(t)$ é estacionário,

$$X^{(m)} \stackrel{d}{=} \frac{1}{m} \sum_{i=1}^m X(i) \stackrel{d}{=} \frac{1}{m} [Y(m) - Y(0)] \stackrel{d}{=} m^{H-1} [Y(1) - Y(0)] \stackrel{d}{=} m^{H-1} X \quad (5)$$

Assim, se o processo $Y(t)$ é auto-similar com incrementos estacionários, então o correspondente processo de incrementos em tempo discreto satisfaz uma relação semelhante aquela utilizada para definir a auto-similaridade em tempo contínuo, e a seguinte definição pode ser estabelecida:

Definição 2.4 [Processos estocásticos discretos e auto-similares (BIANCHI, 2005)]. Seja um processo estocástico $X(t)$, $t \in Z$. Este processo é dito auto-similar com parâmetros de auto-similaridade $H \in (0;1)$ se, para todo $m > 0$ e $t \geq 0$, os processos $X(t)$ e $m^{1-H} X^{(m)}(t)$ são identicamente distribuídos, i.e.

$$X(t) \stackrel{d}{=} m^{1-H} X^{(m)}(t) \quad (6)$$

Os processos estocásticos que satisfazem a Equação (6) são denominados exatamente auto-similares, visto que a sua distribuição se mantém invariável em todas as escalas de agregação. No entanto, é possível que a auto-similaridade se manifeste somente em escalas de agregação maiores, o que torna interessante introduzir também a seguinte definição:

Definição 2.5 [Processos estocásticos discretos e assintoticamente auto-similares (BIANCHI, 2005)]. Seja um processo estocástico $X(t)$, $t \in Z$. Este processo é dito assintoticamente auto-similar com parâmetros de auto-similaridade $H \in (0;1)$ se, para todo $t \geq 0$.

$$\lim_{m \rightarrow \infty} m^{1-H} X^{(m)}(t) \stackrel{d}{=} X(t) \quad (7)$$

A Equação (7) é utilizada quando a auto-similaridade aparece em escalas de agregações maiores.

2.3 Processos Auto-Similares de Segunda Ordem

Em muitas aplicações, apenas algumas propriedades estatísticas dos processos são considerados relevantes. No caso das redes, por exemplo, o tráfego é normalmente caracterizado em termos de estatísticas de segunda ordem, de modo que é suficiente considerar a manifestação da auto-similaridade apenas nessas estatísticas. Isso permite relaxar as Definições 2.4 e 2.5 e introduzir a seguinte definição:

Definição 2.6 [Processos estocásticos discretos e exatamente auto-similares de segunda ordem (PARK; WILLINGER, 2000)]. Seja um processo estocástico $X(t)$, $t \in Z$. Este processo é denominado exatamente auto-similar de segunda ordem com parâmetro $H \in [1/2;1)$ se for possível exprimir sua auto-correlação como:

$$R_{X^{(m)}}(\tau) = R_X(\tau) = \frac{1}{2} \left[(\tau+1)^{2H} - 2\tau^{2H} + (\tau-1)^{2H} \right], \quad \forall m \geq 1 \quad (8)$$

Definição 2.7 [Processos estocásticos discretos e assintoticamente auto-similares de segunda ordem (PARK; WILLINGER, 2000)]. Seja um processo estocástico $X(t)$, $t \in Z$. Este processo é denominado assintoticamente auto-similar de segunda ordem com parâmetro $H \in [1/2;1)$ se for possível exprimir sua auto-correlação como:

$$\lim_{m \rightarrow \infty} R_{X^{(m)}}(\tau) = R_X(\tau) = \frac{1}{2} \left[(\tau+1)^{2H} - 2\tau^{2H} + (\tau-1)^{2H} \right] \quad (9)$$

Observação: No conceito de auto-similaridade de segunda ordem, o parâmetro H é normalmente restrito ao intervalo $[1/2;1)$ por ser esta região de interesse prático – a condição $H \in (0;1/2)$ é dificilmente encontrada na prática visto que, neste caso, $\sum_{\tau} R_x(\tau) = 0$.

A principal característica dos processos auto-similares de segunda ordem é, portanto, o caráter não-degenerativo da função de auto-correlação dos seus processos agregados. A Figura 3 ilustra o comportamento da função de auto-correlação $R_X(\tau)$ para diversos valores de H .

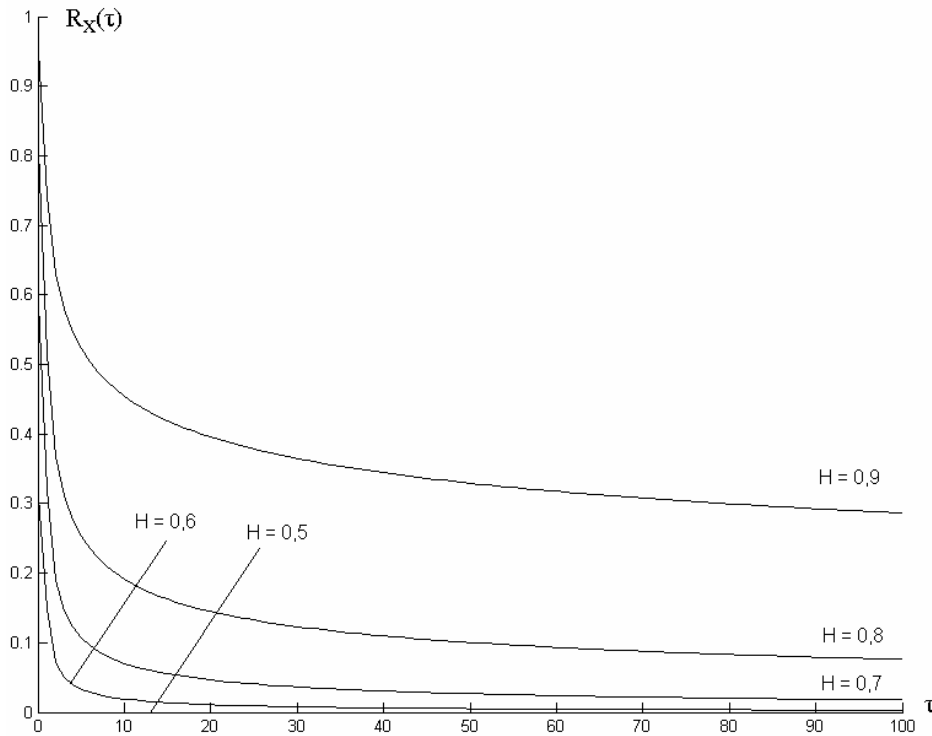


Figura 3 – Comportamento da função de auto-correlação de um processo auto-similar de segunda ordem em função do parâmetro de auto-similaridade H .

Nota-se que, para valores baixos de H , valores expressivos de auto-correlação são encontrados apenas para pequenos valores de τ . Em particular para $H = 1/2$, a função de auto-correlação é equivalente aquela do ruído gaussiano branco sendo, portanto, somável (i.e. $\sum_{\tau} R_x(\tau) < \infty$). Neste caso, diz-se que o processo $X(t)$ apresenta dependência de curta duração (*short-range dependence*, SRD) (PARK; WILLINGER, 2000).

Por outro lado, para $H > 1/2$, a relação (8) pode ser aproximada por (PARK; WILLINGER, 2000):

$$R_X(\tau) \cong H(2H-1)\tau^{2H-2}, \tau \rightarrow \infty \quad (10)$$

o que significa que, do ponto de vista assintótico, $R_X(\tau)$ se comporta como $c\tau^{-\beta}$, $0 < \beta < 1$. Isto significa que a função de auto-correlação do processo $X(t)$ apresenta decaimento hiperbólico e, portanto, não é somável (i.e. $\sum_{\tau} R_x(\tau) \rightarrow \infty$). Além disso, pode-se dizer que,

nas proximidades da origem, a sua densidade espectral de potência $S_x(w)$ se comporta como $cw^{-\gamma}$ ($0 < \gamma < 1$), de forma que $\lim_{w \rightarrow 0} S_x(w) \rightarrow \infty$. Tal comportamento não se verifica no caso de $H = 1/2$, para o qual a densidade espectral de potência é limitada na origem. Assim, diz-se que processos auto-similares com $H > 1/2$ apresentam dependência de longa duração (*long-range dependence*, LRD), cujo principal efeito é a persistência do comportamento do processo no domínio do tempo (PARK; WILLINGER, 2000).

Definição 2.8 [Processo estocástico estacionário com dependência de longa duração (BIANCHI, 2005)]. Um processo estocástico estacionário $X(t)$ apresenta dependência de longa duração se a sua função de auto-correlação apresenta decaimento hiperbólico, i.e. $R_X(\tau)$ comporta-se assintoticamente como $c\tau^{-\beta}$, $0 < \beta < 1$.

Aos processos com dependência de longa duração são geralmente associados variáveis aleatórias de distribuição de cauda longa, cuja função de distribuição de probabilidade decai de forma hiperbólica.

Definição 2.9 [Função de distribuição de cauda longa (GOLDIE; KLÜPPELBERG, 1998)]. Seja $F(x)$ uma distribuição de probabilidade definida no intervalo $(0; \infty)$, tal que $F(x) < 1$, $\forall x > 0$. A função $F(x)$ é dita uma distribuição de cauda longa se

$$\lim_{x \rightarrow \infty} \frac{\bar{F}(x-y)}{\bar{F}(x)} = 1, \quad \forall y > 0 \quad (11)$$

sendo $\bar{F}(x) = 1 - F(x)$.

É possível demonstrar que uma variável aleatória X tem distribuição de cauda longa se, para $x \rightarrow \infty$, $P\{X > x\} \approx x^{-\alpha}$, $0 < \alpha < 2$ (PARK; WILLINGER, 2000). Devido a sua relação com as distribuições de cauda longa, pode-se dizer que os processos com dependência de longa duração atingem valores elevados com probabilidade não-negligenciável. Em conjunto com o comportamento persistente no domínio do tempo, tal fato justifica os efeitos danosos do tráfego auto-similar, em particular para $H > 1/2$. Por outro lado, a dependência de longa duração torna o processo passível de predição (PARK; WILLINGER, 2000), o que pode ser utilizado no desenvolvimento de mecanismos de controle preditivo de tráfego (HIRCHOREN, 1999).

2.4 Estimação do Parâmetro de Auto-Similaridade

Diversas técnicas podem ser utilizadas para estimar o parâmetro H de uma série temporal a partir das suas amostras. Pode-se citar o estimador de Whittle, o gráfico da estatística R/S e o estimador baseado em *wavelets* (PRATA, 2003) (BIANCHI, 2005).

2.4.1 Estimador de Whittle

Esse método supõe que tanto a estrutura como os parâmetros do processo $X(t)$ são conhecidos, à exceção da variância e do parâmetro H . A densidade espectral desse processo pode ser definida, a partir de amostras da sua função de auto-correlação, como:

$$S_X(w, H) = \sum_{\tau} R_X(\tau) \exp(-i w \tau) \quad (12)$$

Define-se, então, $S_X^*(w, H)$ como uma forma normalizada de $S_X(w, H)$, na qual a variância de $X(t)$ seria unitária, e ainda $I_N(w)$ como o periodograma obtido a partir de N amostras do processo $X(t)$:

$$I_N(w) = \frac{1}{2 \cdot \pi \cdot N} \left| \sum_{k=1}^N X(k) \exp(i w k) \right|^2 \quad (13)$$

O parâmetro de auto-similaridade pode, então, ser obtido resolvendo o seguinte problema de otimização:

$$H = \arg \min_H \int_{-\pi}^{\pi} \frac{I_N(w)}{S_X^*(w, H)} dw \quad (14)$$

Se o processo $X(t)$ é reduzido a um conjunto de n amostras, a integral em (14) pode ser entendida como uma soma discreta para $w = \frac{2\pi}{N}, \frac{4\pi}{N}, \dots, \frac{2(n-1)\pi}{N}$. As definições utilizadas pelo estimador de Whittle permitem ainda obter diretamente a variância da estimativa, σ_H^2 , definida como:

$$\sigma_H^2 = 4\pi \left[\int_{-\pi}^{\pi} \left(\frac{\partial}{\partial H} \log S_X^*(w, H) \right)^2 dw \right]^{-1} \quad (15)$$

Embora o estimador de Whittle permita uma análise bastante refinada dos dados, ele tem a desvantagem de ser um método paramétrico, podendo levar a resultados incorretos caso os dados não se ajustem à estrutura previamente assumida para o processo $X(t)$.

2.4.2 Estatística R/S

Este método é baseado na definição da estatística R/S para processos estocásticos discretos, e sua principal vantagem é a independência com relação à distribuição marginal do processo em análise. A estatística R/S é definida como:

$$R/S(N) = \frac{\max_{1 \leq j \leq N} \left\{ \sum_{l=1}^j [X(l) - M(N)] \right\} - \min_{1 \leq j \leq N} \left\{ \sum_{l=1}^j [X(l) - M(N)] \right\}}{\sqrt{\frac{1}{N} \sum_{l=1}^N [X(l) - M(N)]^2}} \quad (16)$$

sendo $M(N) = \frac{1}{N} \sum_{l=1}^N X(l)$. No caso de processos auto-similares, essa razão apresenta a seguinte característica (LELAND et al., 1994):

$$R/S(N) \cong \left(\frac{N}{2} \right)^H \quad (17)$$

onde N representa o tamanho da amostra.

2.4.3 Estimador Baseado em *Wavelets*

O parâmetro de auto-similaridade de um processo pode ser estimado utilizando técnicas baseadas na transformada de *wavelets* (HIRCHOREN, 1999) (ABRY; VEITCH, 1998). Seja $Y(t)$ um processo auto-similar segundo a Definição 2.1. O desenvolvimento em *wavelets* deste processo até a escala J é dada por

$$Y(t) = 2^{-J/2} \sum_{n=-\infty}^{\infty} a_J[n] \phi(2^{-J}t - n) + \sum_{j=1}^J 2^{-j/2} \sum_{n=-\infty}^{\infty} d_j[n] \psi(2^{-j}t - n) \quad (18)$$

onde $\psi(t)$ é a função *wavelet* utilizada no desenvolvimento, e $\phi(t)$ é a correspondente função de escala. Para decomposição ortonormal,

$$d_j[n] = 2^{-j/2} \int_{-\infty}^{\infty} Y(t) \psi(2^{-j}t - n) dt \quad (19)$$

$$a_J[n] = 2^{-J/2} \int_{-\infty}^{\infty} Y(t) \phi(2^{-J}t - n) dt \quad (20)$$

onde $j, n \in Z$. As sequências $a_j[n]$ e $d_j[n]$ são denominadas, respectivamente, coeficientes de aproximação e coeficientes de detalhamento da transformada. É possível demonstrar que a variância de $d_j[n]$ é dada por (HIRCHOREN, 1999):

$$Var\{d_j[n]\} = k2^{j(2H+1)}V_\psi(H) \quad (21)$$

A função $V_\psi(H)$ depende da *wavelet* utilizada na decomposição. Para o caso da *wavelet* de Haar:

$$V_\psi(H) = \frac{1 - 2^{-2H}}{(H+1)(2H+1)} \quad (22)$$

De forma que:

$$\log_2 Var\{d_j[n]\} = (2H+1)j + \left[1 + \log_2 k + \log_2 \left(\frac{V_\psi(H)}{2} \right) \right] \quad (23)$$

A curva que representa a relação entre $\log_2 Var\{d_j[n]\}$ e a escala j é, portanto, uma reta de inclinação $2H-1$. Se o processo $Y(t)$ for gaussiano e apresentar incrementos estacionários, então $k = Var\{Y(t)\}/2$, e a variância também pode ser estimada utilizando este método (HIRCHOREN, 1999).

As principais vantagens do método baseado em *wavelets* são a sua eficiência e a capacidade de estimar o parâmetro H sem polarização.

2.5 Geração de Tráfego Auto-Similar

Muitos pesquisadores observaram que o modelo de Poisson, que era usado na análise de filas, nem sempre era capaz de modelar adequadamente o tráfego de rede (PAXSON; FLOYD, 1995). Na escolha de um modelo de tráfego apropriado, vários fatores são decisivos, tais como: generalidade, facilidade de implantação, precisão, enfim, características próximas às das fontes reais.

Como salientado no item 2.1 e nas definições 2.8 e 2.9 a agregação de tráfego de diversas aplicações envolvendo transferência de arquivos pode gerar um tráfego auto-similar se o tamanho destes arquivos segue uma distribuição de cauda pesada.

Mellia, Muscariello e Rossi realizaram diversas medidas de tráfego, usando a ferramenta TSTAT (TSTAT, 2004), na rede do Campus do Politécnico de Torino (Itália)

(ROSSI; MUSCARIELLO; MELLIA, 2004). Tais medidas comprovaram a distribuição tipo cauda pesada para o tamanho dos arquivos transmitidos pela Internet.

A distribuição presente em (ROSSI; MUSCARIELLO; MELLIA, 2004) foi discretizada em grupos conforme mostrado na Figura 4 e tomada como base para a obtenção dos dados numéricos presentes nesse trabalho.

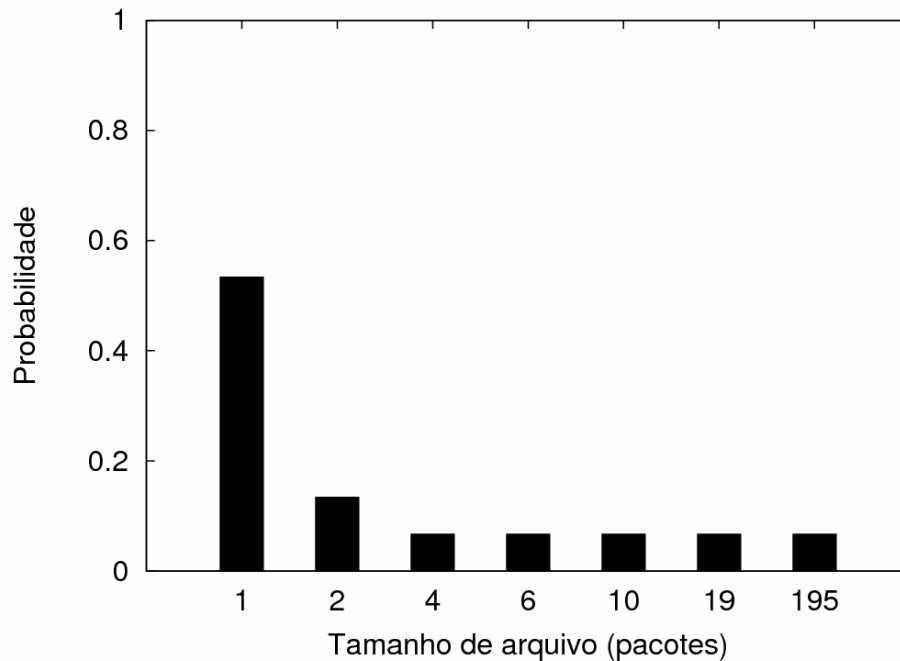


Figura 4 – Tamanho dos arquivos transmitidos em função das suas respectivas probabilidades.

A geração tráfego auto-similar se dá pela transmissão, via TCP, de arquivos cuja dimensão segue esta distribuição. Este será o processo usado para a obtenção de dados via simulação. Fixada a distribuição, também, pode-se obter as informações estatísticas necessárias ao modelo analítico proposto no capítulo 4.

A seguir utiliza-se o método R/S, descrito anteriormente, para determinar o parâmetro de Hurst do tráfego considerado.

Primeiramente foram obtidos os gráficos do volume de dados (bits por intervalo de tempo) em um intervalo de 500s nas escalas de 10ms, 100ms e 1000ms, como é mostrado na Figura 5.

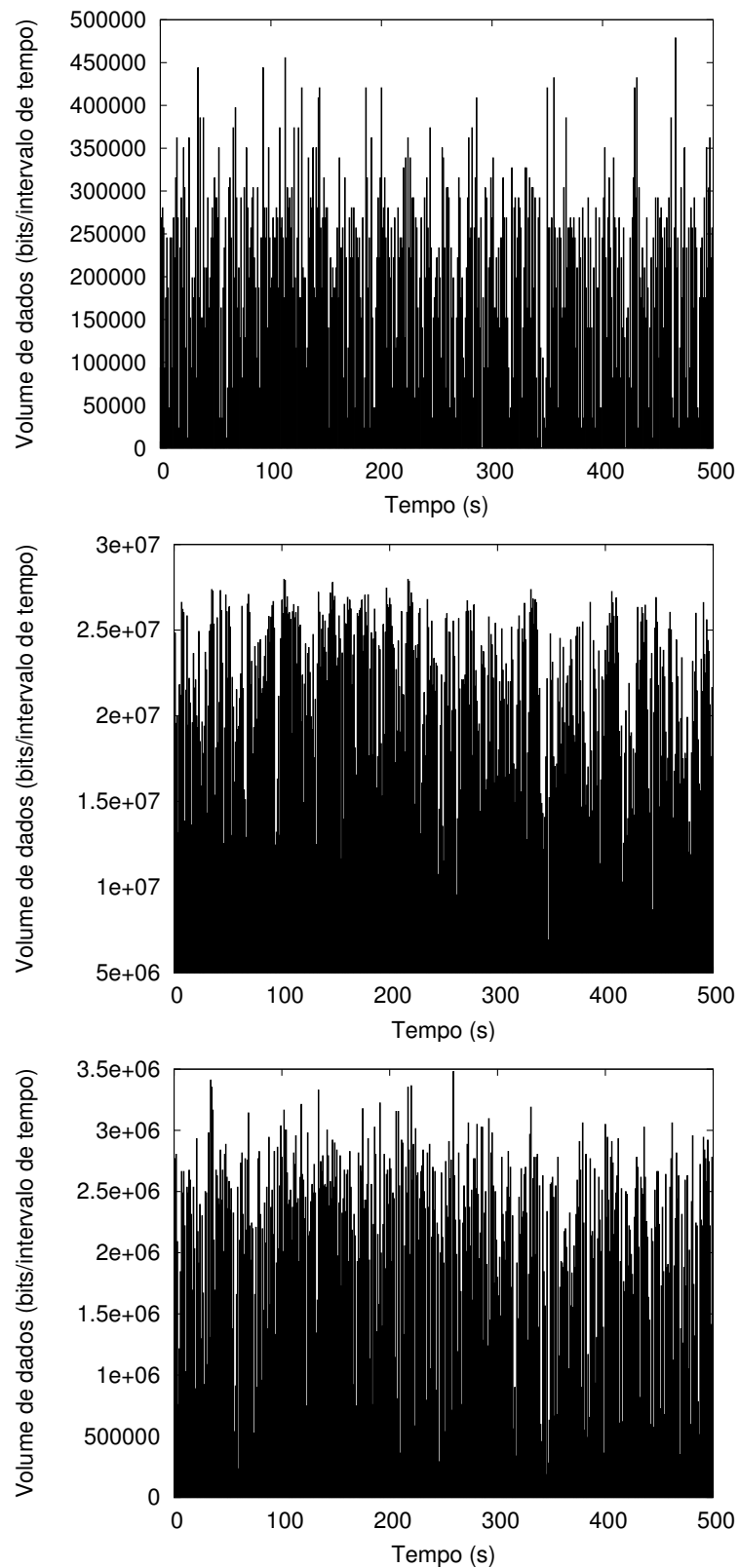


Figura 5 – Volume de dados (bits por intervalo de tempo) no buffer dos roteadores em um intervalo de 500s nas escalas de 10ms, 100ms e 1000ms.

Em seguida foram determinadas as funções de auto-correlação para cada série. A primeira providência da análise de uma série deve ser a remoção de tendências e a procura por

sazonalidades; a não observação desta regra pode levar a conclusão incorreta (PEDROSO, 2006). A partir daí irá se calcular o valor do parâmetro de Hurst para se saber o quão auto-similar é este tráfego (o Anexo A contém o programa em C utilizado para a determinação dos valores de auto-correlação).

Através da Figura 6 pode-se verificar um lento decaimento da função de auto-correlação o que indica a presença de auto-similaridade.

Com os volumes de dados, pode-se calcular os valores de $R(N)$ e $S(N)$. Os valores de N utilizados nestas situações foram, respectivamente, 50000, 5000 e 500 para as escalas de amostragem de 10ms, 100ms e 1000ms. Assim utilizou-se a Equação (24) para avaliar o parâmetro de Hurst.

$$H = \frac{\log\left(\frac{R}{S}\right)}{\log\left(\frac{N}{2}\right)} \quad (24)$$

Para a primeira situação os valores obtidos foram: $R(N) = 288199148,83$; $S(N) = 100370,21$; e o parâmetro de Hurst $H = 0,7863$ (com 95% de confiança no intervalo [0,7675; 0,8052]). Para a segunda situação os valores obtidos foram: $R(N) = 287491400,86$; $S(N) = 715758,66$; e o parâmetro de Hurst $H = 0,7663$ (com 95% de intervalo de confiança [0,7227; 0,8099]). Para a última situação tem-se $R(N)=286782614,08$; $S(N) = 4256934,13$; e o parâmetro de Hurst $H = 0,7625$ (com 95% de intervalo de confiança [0,7154; 0,8095]). Como os valores obtidos do parâmetro de Hurst foram maiores do que 0,5, pode-se comprovar que o tráfego gerado possui características auto-similares (o Anexo B contém o programa em C utilizado para calcular o parâmetro de Hurst pelo método R/S).

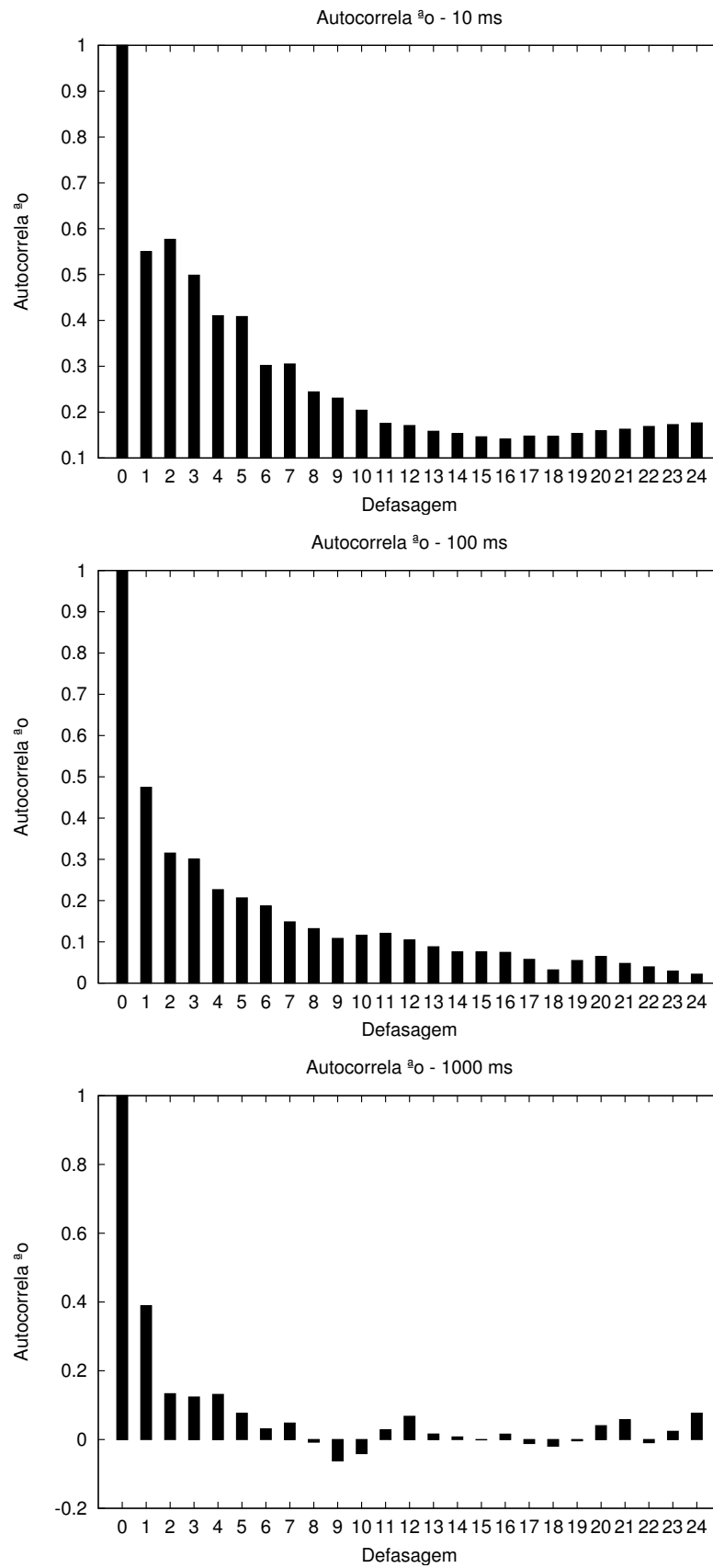


Figura 6 – Valores da auto-correlação em função do deslocamento para os sinais com taxa de amostragem de 10ms, 100ms e 1000ms respectivamente.

2.6 Conclusão do Capítulo

Neste capítulo pode-se definir auto-similaridade quando um determinado objeto preserva as mesmas características independentemente da sua escala de espaço e tempo, na seqüência foram apresentadas diversas definições importantes relacionadas à auto-similaridade. Pode-se concluir desse capítulo que o dimensionamento de uma rede de dados, sem levar em consideração o fato de o tráfego ter um comportamento auto-similar, pode levar a valores errôneos, como, por exemplo, no dimensionamento dos *buffers*.

Foi mostrado neste capítulo o processo de geração do tráfego que será utilizado nessa tese, sendo que este tráfego apresentada as mesmas características mostradas em (ROSSI; MUSCARIELLO; MELLIA, 2004). Na seqüência foram descritas três maneiras de se obter o parâmetro de Hurst (estimador de Whittle, estatística R/S e estimador baseado em *wavelets*), através do valor desse parâmetro pode-se comprovar que o tráfego utilizado nessa tese tem características auto-similares.

3 ESCALONADOR WFQ

3.1 Introdução

Em redes IP, os roteadores são elementos centrais para a obtenção de qualidade de serviço: pacotes em geral trafegam por vários roteadores antes de chegar ao seu destino final. Cada roteador pode ser considerado como um recurso compartilhado, onde cada aplicação gera requisições por uma parcela dos recursos disponíveis através do envio de pacotes a serem processados e encaminhados. Às vezes a quantidade de pacotes a ser processada é compatível com os recursos disponíveis do roteador, permitindo que sejam processados imediatamente. Outras vezes ocorre disputa pelos recursos no roteador (contenção): o número de pacotes é maior que a quantidade de recursos disponíveis para processá-los. Um aspecto crítico do gerenciamento do desempenho é a forma como cada roteador resolve a contenção de seus recursos, estando grande parte deste fator relacionada com a disciplina de fila utilizada (PAXSON et al., 1998).

A função de uma disciplina de fila é a de definir qual pacote deve ser processado após o final do processamento do pacote anterior. Durante congestionamentos em roteadores, à medida que a quantidade de recursos necessários para o processamento de novos pacotes é maior do que os disponíveis, os pacotes em excesso são enfileirados. Caso o congestionamento seja longo, a fila eventualmente atingirá um tamanho máximo a partir do qual pacotes serão descartados. Disciplinas de fila permitem o gerenciamento da contenção de recursos no roteador, adquirindo uma grande importância no tratamento ofertado ao tráfego. Há cinco características principais as quais devem ser levadas em conta na implementação e escolha de uma disciplina de fila: o atraso e o *jitter* induzidos no tráfego, a escolha dos pacotes a serem descartados, o isolamento e a justiça entre diferentes classes de tráfego (WESTERINEN; STRASSNER, 2001).

Tanto o atraso induzido quanto o *jitter* obtidos por uma determinada classe de tráfego estão diretamente relacionados ao tempo de espera dos pacotes nas filas dos roteadores. Em geral, quanto maior o tamanho das filas dos roteadores, maior serão o atraso e o *jitter* médios dos pacotes encaminhados (WESTERINEN; STRASSNER, 2001). Disciplinas de fila também podem ser configuradas para privilegiar o encaminhamento de determinadas classes de tráfego em detrimento de outras. Por exemplo, pacotes de uma aplicação X podem ser configurados para serem encaminhados apenas se não houver pacotes de uma aplicação Y na fila; o que possibilita que o tráfego de Y obtenha um nível de serviço melhor que X.

Disciplinas de fila também podem incluir gerenciamento das funções de descarte de pacotes. Quando a fila do roteador está cheia e um novo pacote chega, a disciplina de fila deve possuir um algoritmo que escolha se o novo pacote deve ser descartado ou se algum outro pacote já enfileirado deve ser escolhido para descarte para que o novo pacote possa ser processado. Em geral, quanto menor o tamanho da fila de roteadores, maior será o descarte de pacotes em caso de congestionamentos.

Disciplinas de fila podem possuir algoritmos que garantam uma alocação mínima de recursos (espaço na fila, vazão) a determinadas classes de tráfego. Em geral, a alocação de recursos é garantida apenas quando o tráfego adequa-se a um perfil de tráfego previamente definido. O perfil de tráfego representa as características que o tráfego deve possuir para que as garantias das disciplinas de filas possam ser atendidas. O isolamento é definido como a proteção fornecida a uma classe de tráfego com relação ao tráfego fora do perfil (ou tráfego em excesso) de outras classes de modo que o tráfego em excesso não degrade o tratamento oferecido pela disciplina de filas ao tráfego dentro do perfil (CLARK et al., 1998).

Uma outra característica importante é como a justiça (*fairness*) é implementada pela disciplina de filas. O termo justiça pode ser definido como a forma como a disciplina de filas divide os recursos entre requisições concorrentes e de tamanho variado, de modo a obter-se um compartilhamento por igual (*equal sharing*) dos recursos disponíveis entre as várias requisições (BLAKE et al.; 1998). A forma como o compartilhamento por igual é obtido pode variar levando a diferentes interpretações no uso do termo, sendo dependente do algoritmo utilizado pela disciplina de fila.

Uma forma de obtenção de compartilhamento por igual bastante popular é o critério max-min, o qual é utilizado por diversas disciplinas de fila de importância, como o *Weighted Fair Queuing* (WFQ).

3.2 Algoritmos de Escalonamento de Pacotes

O compartilhamento de recursos em redes de comunicações fez com que os sistemas adotassem uma estratégia de multiplexação estatística, devido a falta de recursos ofertados pela rede. Isto pode ser percebido pela contenção de pacotes existente em roteadores da rede. Em redes que suportam integração de serviços torna-se necessário providenciar QoS diferenciada por categorias de serviço, oferecendo garantias de desempenho a aplicações críticas e ao mesmo tempo permitindo uma partilha de recursos de acordo com critérios de equidade.

Os algoritmos de escalonamento são componentes essenciais para atingir estes objetivos, uma vez que determinam as disciplinas de serviço a aplicar a fluxos aceitos pela rede. Os algoritmos de escalonamento são usados em qualquer sistema ou camada protocolar em que ocorra contenção por recursos, mas em especial em elementos de rede por exemplo em roteadores. Eles também desempenham um papel importante na provisão de diferentes níveis de QoS a diferentes aplicações, permitindo controle diferenciado de atraso, largura de banda ou taxa de perdas de pacotes. Em geral pretende-se que os algoritmos de escalonamento sejam simples (fáceis de implementar) e tratem de forma equitativa tráfego do tipo melhor esforço e garantam ao restante do tráfego níveis de desempenho negociados e facilitem os mecanismos associados de controle de admissão de conexões.

Podem ser citados alguns escalonadores de pacotes: FIFOQ, FQ, CQ, GPS, WFQ, WRR e PQ.

3.2.1 FIFOQ (*First-In First-Out Queueing*)

Este é o algoritmo de escalonamento de pacotes utilizado originalmente no conjunto de protocolos Inter Rede. O algoritmo está baseado na criação de um buffer de tamanho fixo para cada interface física. Os pacotes que devem ser enviados por esta interface são armazenados neste *buffer* e enviados em ordem de chegada – o primeiro pacote a ser transmitido será o pacote mais antigo no *buffer*. Quando não há mais espaço no *buffer*, os pacotes que chegam são descartados, conforme mostrado na Figura 7. O problema deste método é a impossibilidade da separação do tráfego e divisão a banda passante entre os fluxos IP.

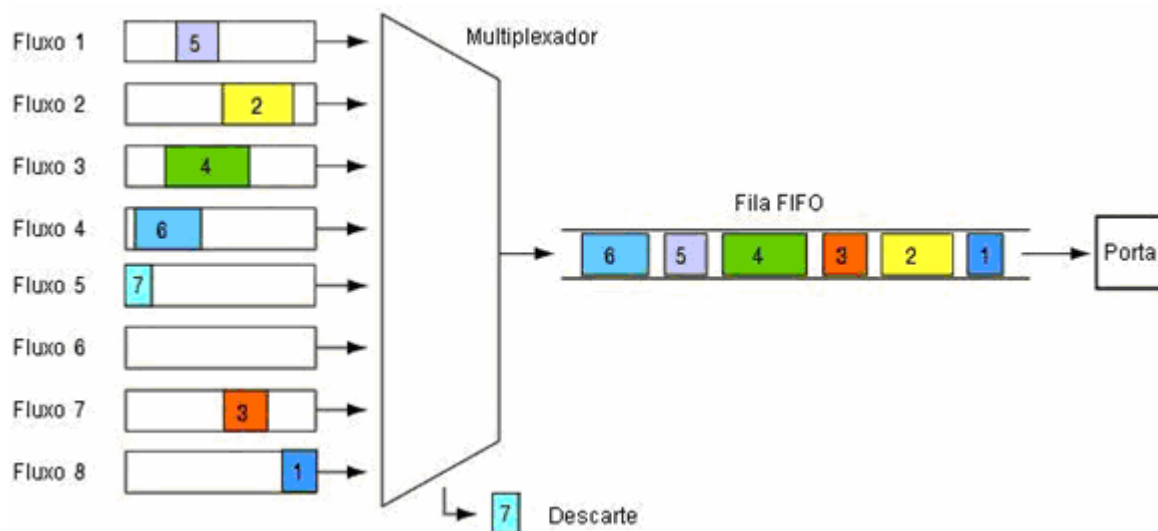


Figura 7 – Escalonador FIFOQ (BALLIACHE, 2003).

3.2.2 FQ (*Fair Queueing*)

No algoritmo FQ as mensagens são ordenadas em sessões, e para cada sessão representa um fluxo, onde estes fluxos são atendidas por *round-robin*, um pacote por fluxo. No caso de uma sessão lotar seu *buffer*, os próximos pacotes são descartados, independentemente das demais sessões estarem completas, conforme mostrado na Figura 8. Isto faz com que um fluxo não tenha uma parcela maior do meio em detrimento dos demais.

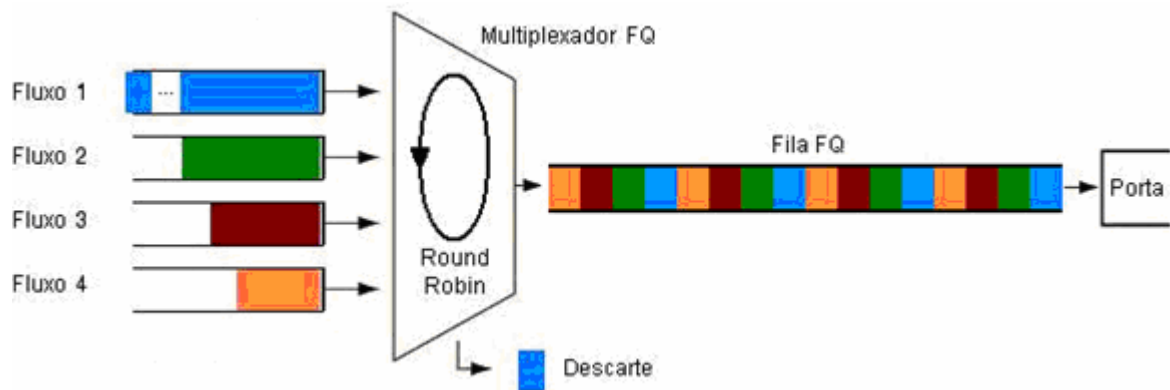


Figura 8 – Escalonador FQ (BALLIACHE, 2003).

3.2.3 CQ (*Custom Queueing*)

O CQ aloca partes da banda para determinados fluxos e a outra parte é dividida para os demais fluxos. O CQ funciona de forma semelhante ao FQ, porém a cada tempo do *round-robin* são liberados os pacotes conforme a proporção alocada inicialmente para cada fluxo.

3.2.4 GPS (*Generalized Processor Sharing*)

É o melhor escalonador possível, no entanto não é implementável. Durante qualquer intervalo de tempo em que existam N filas não vazias, o servidor serve simultaneamente os N primeiros pacotes de cada fila, cada um com um débito igual a $1/N$ da capacidade da ligação física. De forma equivalente se pode dizer que é servida uma quantidade de dados infinitesimal em cada visita a uma fila (uma aproximação consistiria em servir rotativamente um bit de cada fila não vazia).

3.2.5 WFQ (*Weighted Fair Queueing*)

O WFQ é um algoritmo FQ melhorado onde é possível estabelecer pesos para determinados tipos de fluxo. O algoritmo escalonando o tráfego prioritário para frente da fila, reduzindo o tempo de resposta. Ao mesmo tempo, compartilha o restante da banda com os outros tipos de fluxo de uma forma justa, conforme mostrado na Figura 9. O WFQ é dinâmico e se adapta automaticamente as mudanças das condições de tráfego.

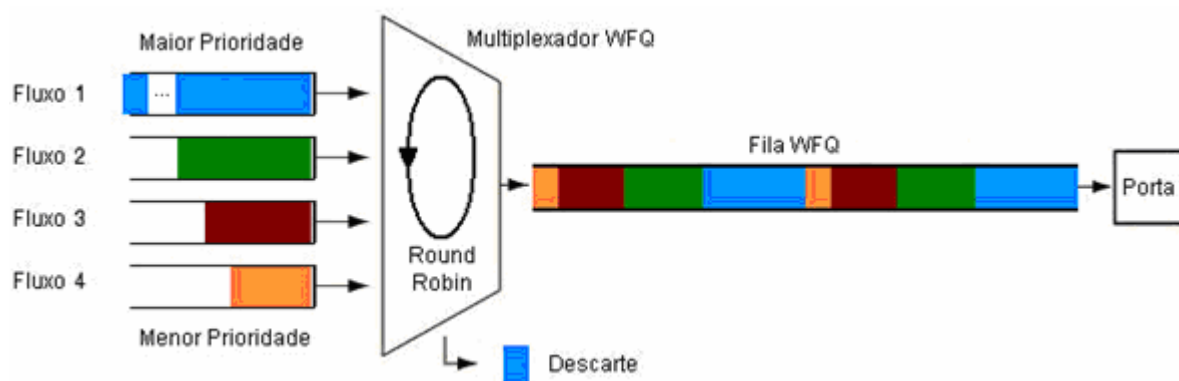


Figura 9 – Escalonador WFQ (BALLIACHE, 2003).

3.2.6 WRR (*Weighted Round Robin*)

Cada fluxo irá fazer uso de uma fila no elemento de rede. O escalonador seleciona pacotes das filas em uma seqüência circular entre as filas. A quantidade de pacotes selecionados será igual ao peso da fila. Esse escalonador tem uma aproximação razoável com o GPS se os pacotes tiverem comprimento fixo e se as filas tiverem o mesmo peso, não é justo se os pacotes tiverem diferentes comprimentos e os pesos não forem iguais.

3.2.7 PQ (*Priority Queueing*)

O PQ é um algoritmo semelhante ao WFQ, porém os níveis mais prioritários têm total preferência em relação aos de menor prioridade. Contudo esse tipo de algoritmo deve ser muito bem administrado, pois quando houver um grande fluxo de pacotes de alta prioridade, os pacotes de baixa prioridade podem sofrer um atraso muito grande ou até mesmo o descarte.

Devido às qualidades superiores em desempenho e compartilhamento de banda, o escalonador WFQ será considerado nesse trabalho.

3.3 O Escalonador WFQ

O WFQ é um escalonador de pacotes que tem se mostrado bastante popular, possuindo grande aceitação na indústria e servido como base para vários outros trabalhos. Para compreendê-lo, faz-se necessário explicar o modelo teórico do escalonador GPS ótimo, o qual o WFQ tenta aproximar.

Em 1993, Parekh e Gallanger (PAREKH; GALLANGER, 1993) demonstraram que, empregando servidores GPS nos comutadores da rede, garantias de atraso fim-a-fim poderiam ser providas para uma conexão cujo tráfego tivesse sido policiado por um mecanismo *Leaky Bucket* (GIROUX; GANTI, 1998). O algoritmo apresentado por Parekh e Gallanger apresenta um desempenho aproximado para o GPS, uma vez que o GPS assume que o servidor pode

atender várias conexões simultaneamente e que o tráfego dessas conexões pode ser dividido infinitamente (não é realizável na prática). Essa disciplina foi chamada de *Packet Generalized Processor Sharing* (PGPS), e demonstrou-se idêntica à disciplina WFQ proposta em 1989 por Demers, Keshav e Shenker (DEMERS; KHESAV; SHENKER, 1990) (LU; ROBOTHAM, 2004).

O GPS baseia-se no critério de compartilhamento por igual max-min, com uma modificação a qual atribui pesos que refletem uma diferenciação entre requisições (*weighted fair share*). Considere-se o exemplo geral de n requisições de tamanhos $s_1, s_2, s_3, \dots, s_i, \dots, s_n$ e pesos relativos associados $w_1, w_2, w_3, \dots, w_i, \dots, w_n$. Seja W a soma dos pesos relativos ($W = \sum_i^n w_i$). Considere ainda que a quantidade de recursos disponíveis seja igual a R .

Inicialmente, o servidor tentará alocar $(w_1 * R/W)$ recursos à primeira requisição. Se essa quantidade de recursos é maior que s_1 , então a demanda é completamente satisfeita e o balanço de recursos faltando será $(R-s_1)$. Se a quantidade for menor, então a demanda será atendida apenas parcialmente e o balanço dos recursos remanescentes será $(R - (w_1 * R/W))$. Em ambos os casos há agora uma nova quantidade de recursos disponíveis a serem compartilhados entre as $(n-1)$ requisições remanescentes, e o processo é aplicado novamente de forma circular (BLAKE et al., 1998).

O GPS pressupõe um modelo fluido onde o tráfego é infinitamente divisível e o escalonamento é feito com granularidade ínfima entre os fluxos ativos a cada instante. Na realidade a implementação do modelo fluido não é viável, de modo que sua principal utilidade é como referência. O GPS é especialmente útil neste sentido porque possui características ótimas de isolamento entre classes de serviço, justiça e garantias de latência máxima.

O escalonador de pacotes *Packet Generalized Processor Sharing* também conhecido como WFQ utiliza um algoritmo que procura emular o GPS com uma granularidade ao nível de pacotes. A implementação dessa emulação é realizada a partir de uma função utilizada para calcular o instante em que cada pacote seria transmitido caso o GPS estivesse implementado. Essa função, conhecida como função de tempo virtual, atribui a cada pacote um instante de tempo de início e um instante de tempo de fim, os quais correspondem aos tempos no sistema de referência GPS. A ordem em que os pacotes são transmitidos corresponde à ordem crescente dos instantes de tempo virtuais de fim de cada um deles (BLAKE et al., 1998).

A Figura 10 representa melhor este escalonador (MELO, 2001). O descarte de pacotes pode ocorrer tanto na fila de entrada como nas filas individuais dos fluxos.

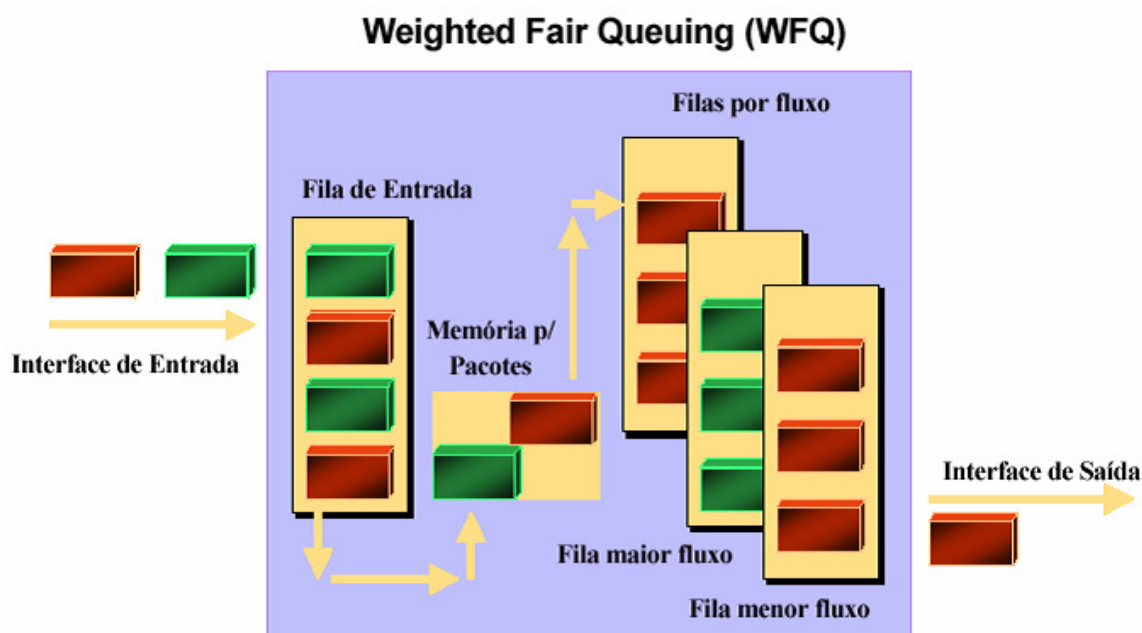


Figura 10 – Escalonador de Pacotes WFQ (MELO, 2001).

O WFQ possui características bastante interessantes de justiça e isolamento entre classes de serviço. Como no GPS, o WFQ procura implementar a justiça entre as classes de serviço utilizando o compartilhamento por igual max-min ponderado: se duas classes de serviço possuem pacotes enfileirados durante qualquer período de tempo, elas são servidas em proporção direta aos seus pesos, independentemente da quantidade de tráfego em excesso que qualquer um deles possa ter gerado. Quando uma classe de serviço requer uma quantidade menor de recursos que o efetivamente alocado, o excesso é compartilhado entre todas as outras classes em proporção direta aos pesos relativos de serviço. O compartilhamento por igual max-min também implementa o isolamento entre as diversas classes de serviço: o tráfego em excesso de uma das classes não prejudica o tratamento ofertado ao tráfego das restantes.

Deve-se frisar que o WFQ é apenas uma aproximação do GPS: enquanto o WFQ manipula pacotes de tamanho variado, o GPS pressupõe que o tráfego é infinitamente divisível. A aproximação implementada pelo WFQ é proporcional ao tamanho médio dos pacotes do tráfego servido. Mesmo assim, o WFQ pode ser considerado como uma grande melhora em precisão na justiça entre as classes de serviço e garantias de desempenho sobre outras disciplinas de fila como o *Weighted Round Robin* (WRR) (BLAKE et al., 1998).

Uma outra vantagem do WFQ é que um limite do atraso fim-a-fim pode ser computado baseado no peso atribuído a cada fluxo. Esse limite pode ser calculado para todos os enlaces entre a origem e o destino do fluxo e é independente do número de fluxos que são

multiplexados em cada enlace (REZENDE, 1999). Essa propriedade faz com que este algoritmo seja bastante utilizado em implementações de arquiteturas de qualidade de serviço que requerem limites quantitativos de atraso.

Como exemplo, na Figura 11 tem-se um sistema simples com duas fontes compartilhando uma mesma fila e um único servidor. Através de simulação, podem-se mostrar as bandas ocupadas por cada um dos fluxos (Figura 12). O enlace possui capacidade igual a 2Mbps e o WFQ está configurado com 90% para o fluxo1 e 10% para o fluxo 2.

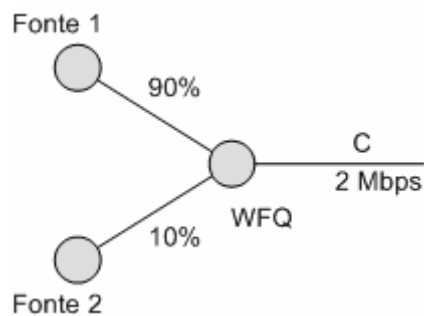


Figura 11 – Configuração do Escalonador WFQ, compartilhando duas fontes com 90% e 10%.

Quando uma das fontes não tem dados a transmitir, a outra fonte ocupa toda a banda disponível do sistema — isto é, o escalonador irá ponderar o peso das filas se e somente se as fontes tiverem dados para transmitir (NABAS; FORONDA; GODOY, 2004).

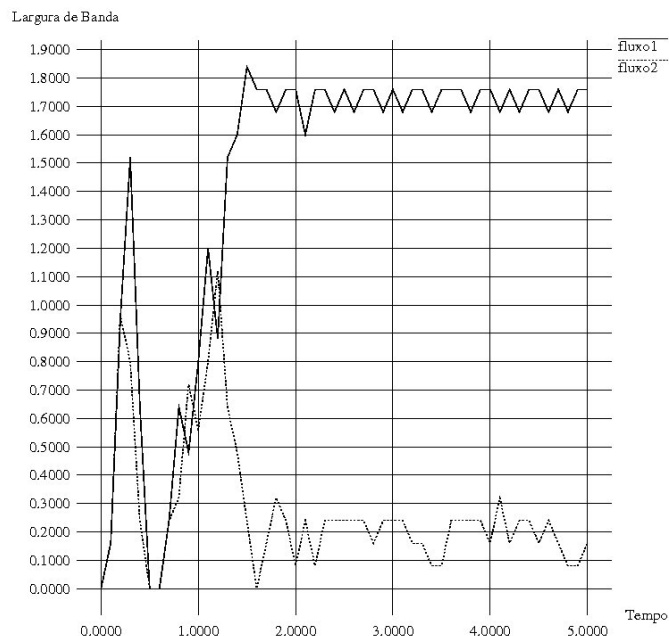


Figura 12 – Largura de Banda utilizada pelos fluxos 1 e 2 da Figura 11.

3.4 Aspectos Analíticos do Escalonador WFQ

Nessa seção são analisados os aspectos analíticos do escalonador WFQ. O escalonador WFQ pode gerenciar dois ou mais caminhos de dados e automaticamente suavizar rajadas e reduzir o tempo de atraso médio.

O WFQ classifica pacotes por ordem de chegada e prioridade para determinar a ordem de transmissão. O efeito é que o WFQ classifica sessões em maior ou menor largura de banda. Clientes aos quais é atribuída menor largura de banda são penalizados quando concorrem com outros que tem maior largura de banda (SAYENKO et al., 2003). A Figura 13 representa o modelo de um nó de rede com o escalonador WFQ.

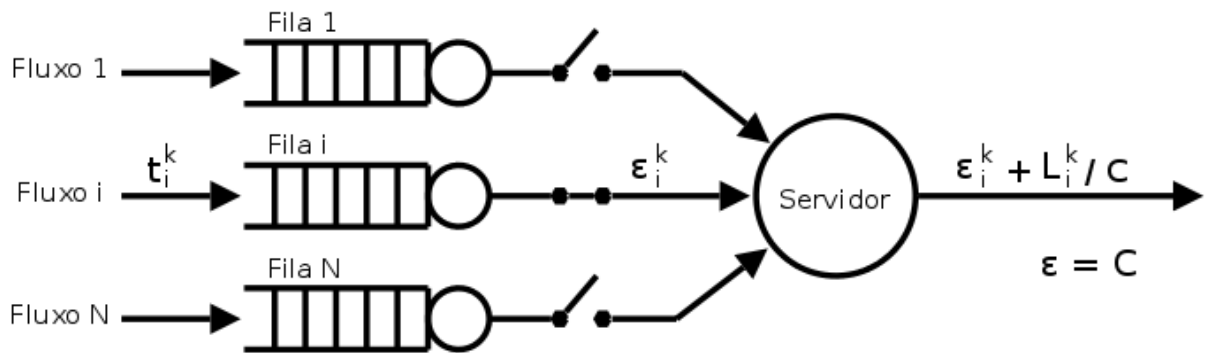


Figura 13 – Modelo de um nó com o escalonador WFQ (DEKERIS; ADOMKUS; BUDNIKAS, 2006)

Este modelo pode ser usado para descrever o atraso nas filas, aonde t_i^k é o tempo de chegada do último bit de um pacote p_i^k (i.e., o k -ésimo pacote do i -ésimo fluxo) da fila. L_i^k representa o tamanho do pacote p_i^k , ϵ_i^k é o instante em que ocorre a transmissão do último bit do pacote p_i^k e C é a capacidade do enlace. O tempo gasto pelo pacote p_i^k na fila pode, então, ser escrito como:

$$T_i^k = \epsilon_i^k - t_i^k \quad (25)$$

O atraso de pacotes pertencente ao fluxo i pode ser descrito pela função:

$$\tau_i(t) = \sum_{k=0}^{\infty} L_i^k \cdot \delta(t - \epsilon_i^k) \quad (26)$$

onde $\delta(t)$ é o delta de Dirac. Então o tempo de espera na fila em um nó de rede é:

$$T_i(s; t] = \int_s^t w_i(\tau) \cdot d\tau \quad (27)$$

A base do escalonador WFQ é o cálculo dos pesos das filas, os quais influenciam diretamente o tempo real necessário para cada pacote ser servido por um sistema GPS ideal.

Tem-se λ_i como a taxa de chegada e ρ_i como a intensidade de tráfego (fator de utilização) para uma dada classe de tráfego i . Para simplificar os cálculos, assume-se que o tempo de serviço (razão entre λ_i e ρ_i) é igual a um; então, $\lambda_i = \rho_i$. Tem-se w_i como o peso atribuído à fila correspondente. O peso relativo r_i da fila i é então definido como:

$$r_i = \frac{w_i}{\sum_{i=1}^I \rho_i} \quad (28)$$

onde w_i é o peso atribuído da fila i e I é o número da classe.

O peso relativo pode ser compreendido como a probabilidade, em longo prazo, de uma classe ser servida quando todas as classes são continuamente colocadas em espera.

O WFQ usa apenas alguns campos do cabeçalho dos pacotes para determinar o protocolo subjacente. Para o protocolo IP, usam-se o *Type of Service* (ToS) e os endereços de origem e de destino. Para os protocolos TCP e UDP, usam-se as portas de origem e destino (MAGAÑA; MORATÓ; VARAIYA, 2003).

O escalonador é configurado em roteadores de borda (MOTA et al., 2001), classificando e colocando em uma fila os pacotes que chegam (conforme mostrado na Figura 13). O escalonador então serve o enlace novamente, de maneira cíclica – isto é, servindo primeiro a classe 1, depois a classe 2, e assim por diante – respeitando o peso de cada fila.

3.5 Exemplo do Funcionamento do Escalonador WFQ

No exemplo proposto por (MARQUES, 2005), o escalonamento WFQ é aplicado a três filas com cotas iguais de 8000 bits para cada classe de fluxo (as cotas são proporcionais aos pesos). No primeiro ciclo, conforme mostrado na Figura 14, todas as variáveis de saldo são inicializadas com zero. Na lista de filas ativas, um ponteiro indica para qual fila os recursos do sistema estarão disponíveis naquele momento. Neste caso os recursos do sistema estão disponíveis para a fila 1. Como o saldo de 8000 bits é insuficiente para transmitir o pacote de 12000 bits o saldo é acumulado para o próximo ciclo em que os recursos estarão disponíveis para esta fila e encerra-se o ciclo atual.

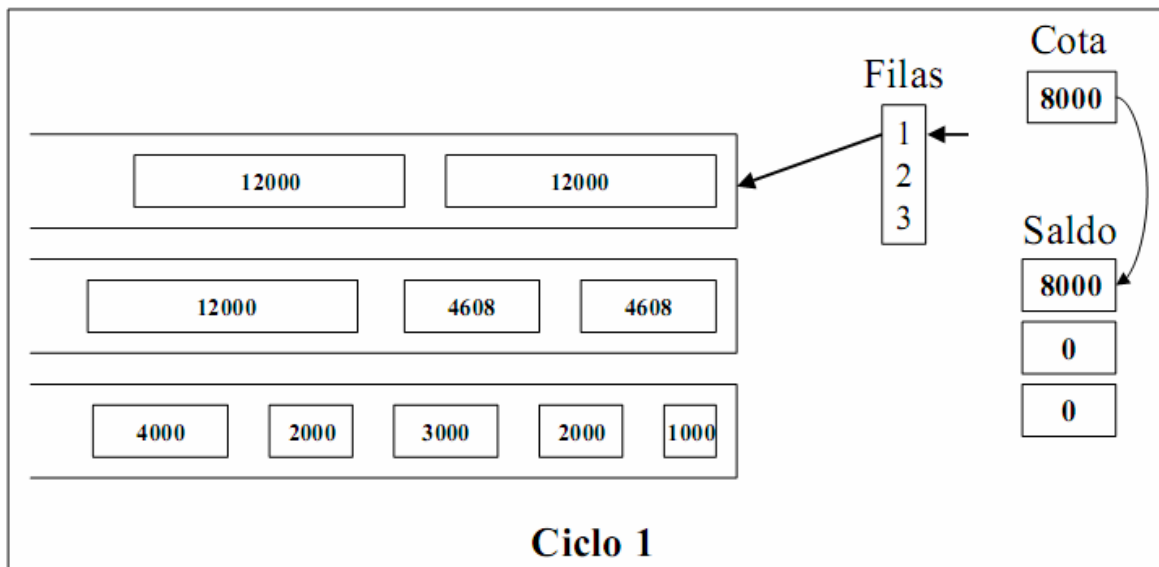


Figura 14 – Ciclo 1 do WFQ.

No ciclo 2 a fila ativa passa a ser a fila 2. Nesta fila com o saldo de 8000 bits é transmitido um pacote de 4608 bits. Desta forma, o saldo passa a ser 3392 bits e é acumulado para a transmissão do próximo pacote. Este ciclo é exibido pela Figura 15.

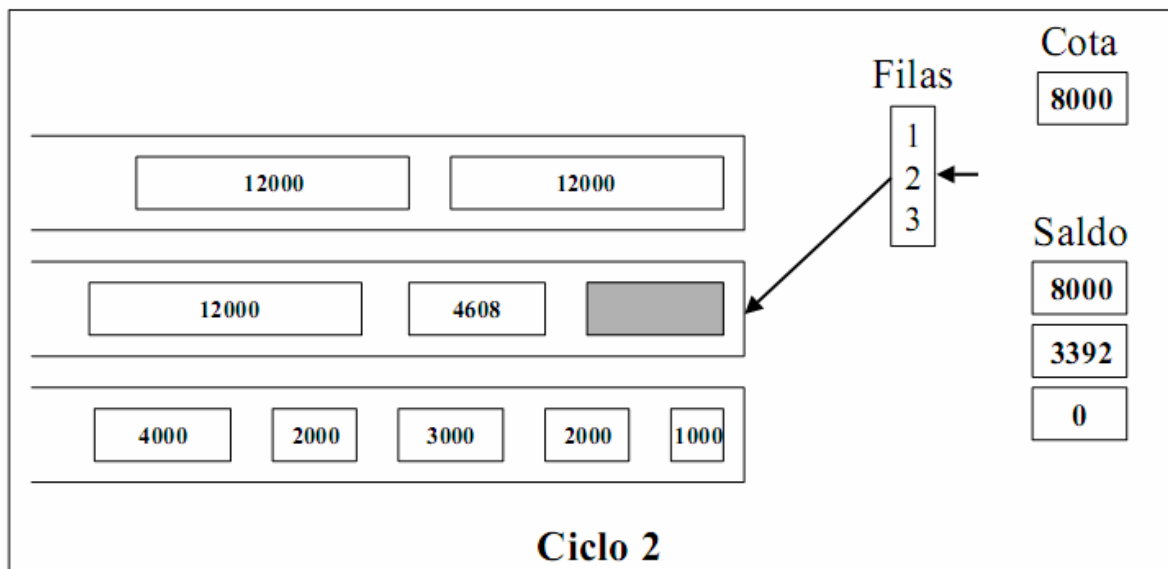


Figura 15 – Ciclo 2 do WFQ.

Já no ciclo 3 com a fila 3 ativa são transmitidos os pacotes de 1000 bits, 2000 bits, 3000 bits e 2000 bits que no total perfaz uma soma de 8000 bits. Dessa forma, o saldo é zerado e torna-se insuficiente para transmissão do último pacote presente nesta fila, tendo como consequência o fim do ciclo corrente, ilustrado pela Figura 16.

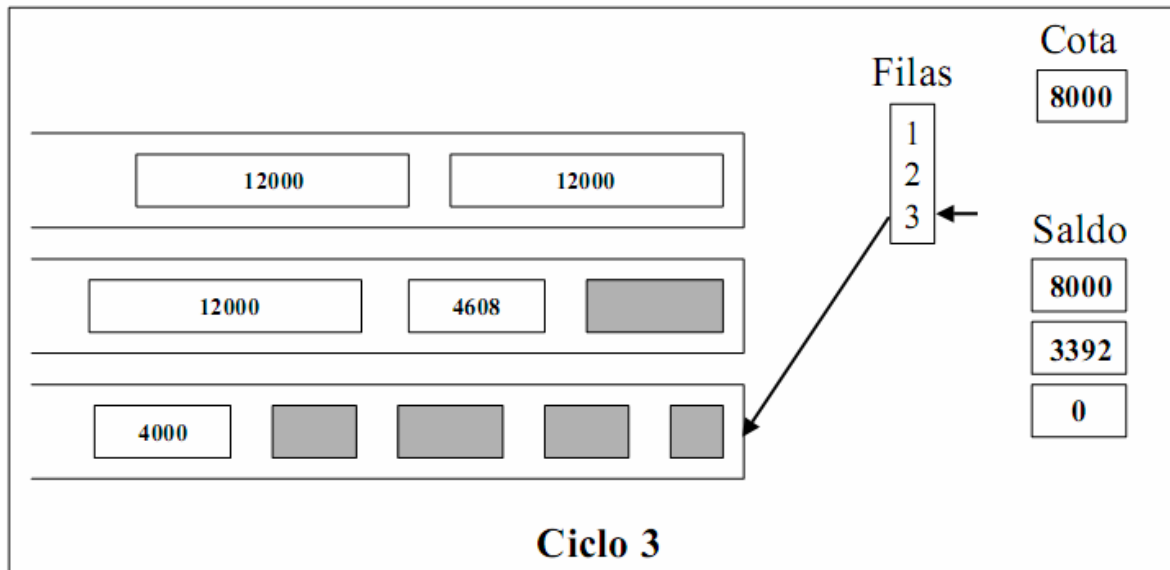


Figura 16 – Ciclo 3 do WFQ.

No ciclo 4 a fila ativa volta a ser a fila 1. O saldo remanescente de 8000 bits é somado a cota de 8000 bits. Dessa maneira, o saldo passa a ser 16000 bits viabilizando a transmissão do primeiro pacote de 12000 bits. Após esta transmissão o saldo passa a ser de 4000 bits e como não é possível a transmissão do último pacote de 12000 bits o saldo fica acumulado para o próximo ciclo desta fila. A Figura 17 ilustra esta situação.

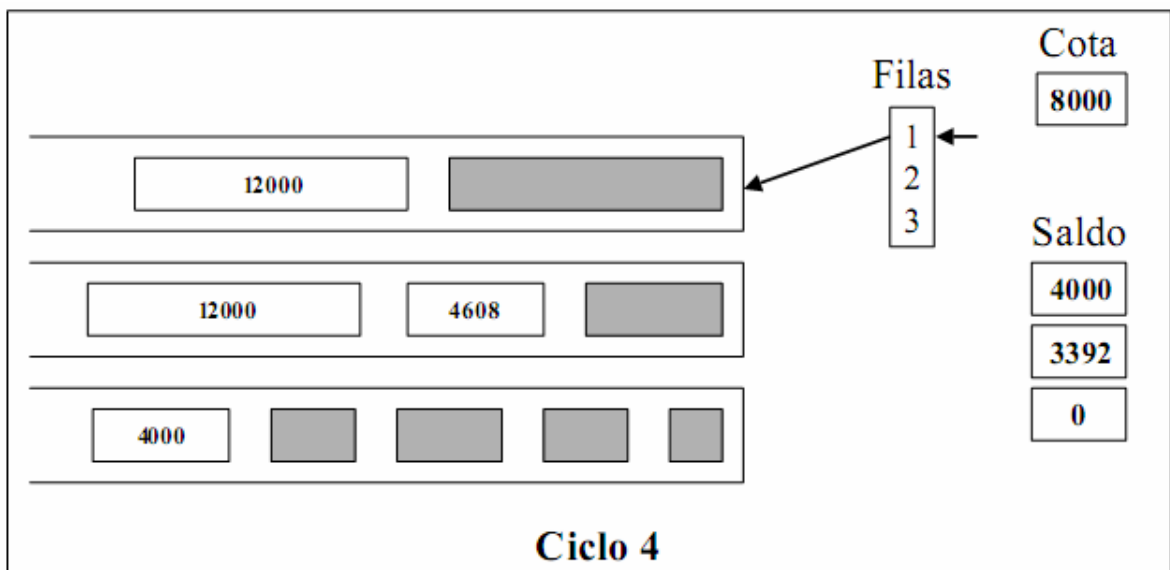


Figura 17 – Ciclo 4 do WFQ.

Na seqüência, a fila ativa passa ser a fila 2. O saldo anterior é somado a cota de 8000 bits resultando num saldo de 11392 bits. Por meio deste saldo é transmitido mais um pacote desta fila. O saldo restante é de 6784 bits e fica acumulado para o próximo ciclo desta fila uma vez que resta apenas mais um pacote na fila. Este ciclo é mostrado pela Figura 18.

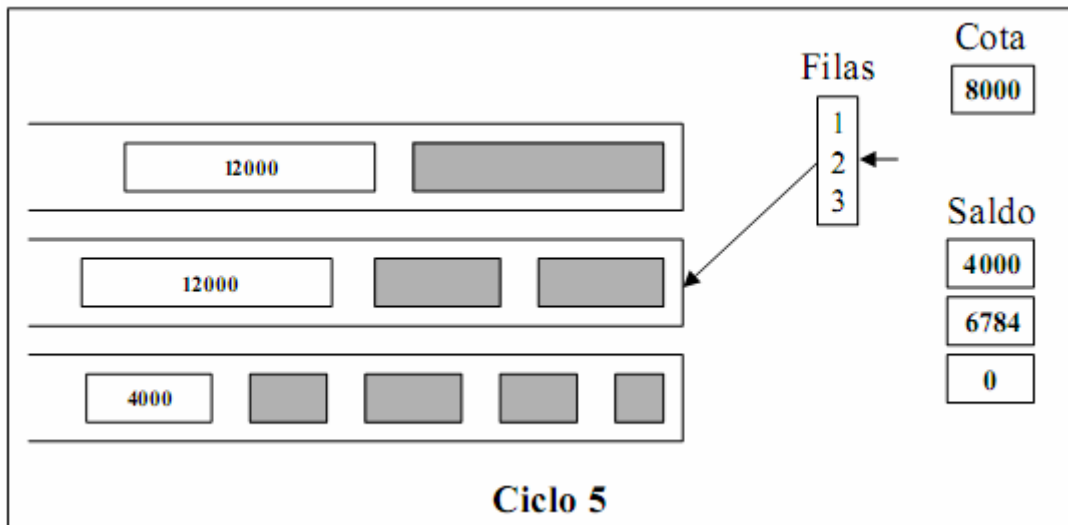


Figura 18 – Ciclo 5 do WFQ.

No ciclo 6, o saldo somado a cota denota um total de 8000 bits. Dessa forma, o pacote de 4000 bits é transmitido e como não há mais pacotes nesta fila o saldo é zerado e a lista de filas ativas passa a contar somente com as filas 1 e 2. A Figura 19 exibe este cenário.

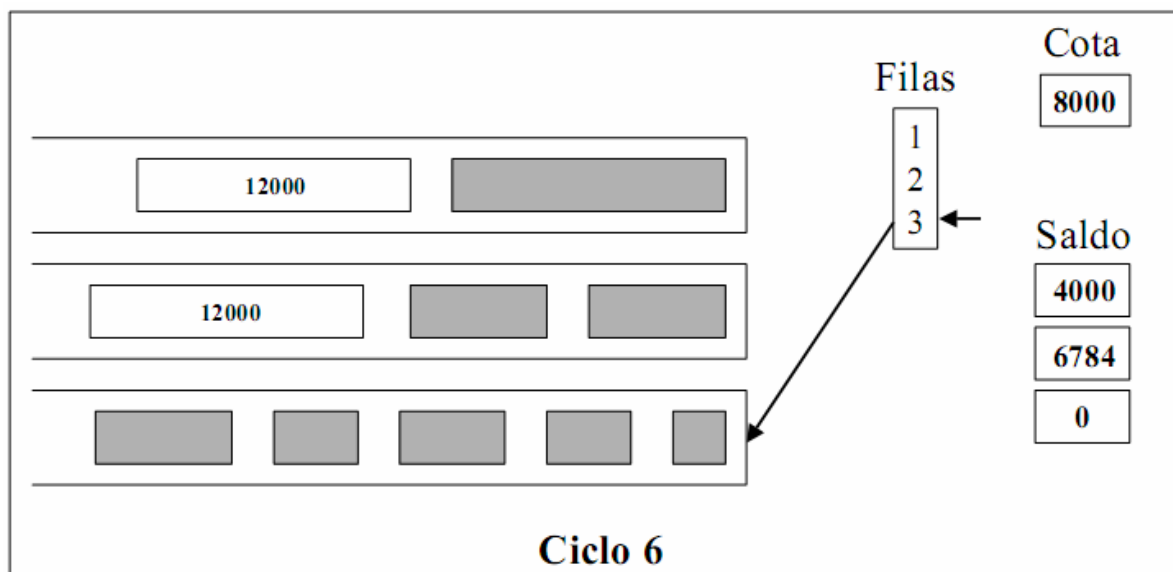


Figura 19 – Ciclo 6 do WFQ.

No ciclo 7 é transmitido o último pacote da fila 1, tendo em vista que o saldo de 12000 bits garante a transmissão do pacote. Sendo este pacote o último a ser transmitido o saldo é zerado e partir deste ciclo somente há uma fila com pacotes a serem transmitidos. A Figura 20 ilustra este ciclo.

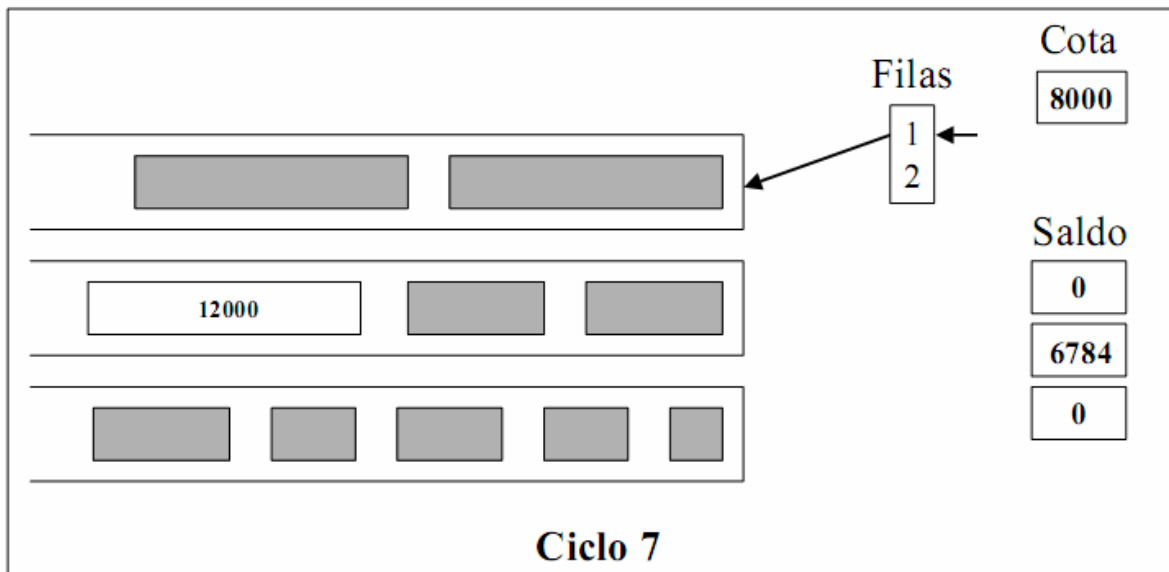


Figura 20 – Ciclo 7 do WFQ.

Por fim, o último pacote do sistema presente na fila 2 é transmitido. Desta forma, o saldo é zerado e nenhuma fila permanece ativa, pois, não há nenhum pacote no sistema. O sistema fica inoperante até o momento em que chegar algum pacote nas filas de espera podendo neste caso no máximo ter 3 filas ativas ao mesmo tempo. A Figura 21 mostra esta situação.

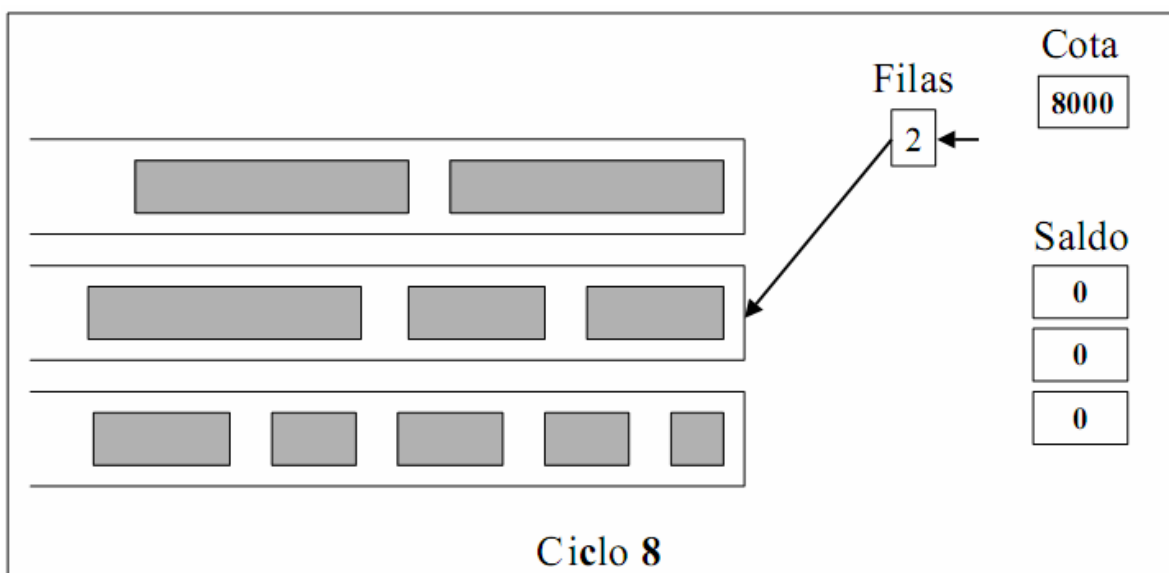


Figura 21 – Ciclo 8 do WFQ.

3.6 Conclusões do Capítulo

Nesse capítulo foi demonstrada a importância de se escolher um determinado escalonador de pacotes para um nó de rede, através da escolha do escalonador pode-se fazer uma diferenciação dos fluxos, isto é, prover Qualidade de Serviço a diferentes aplicações,

permitindo diferenciar atraso entre os fluxos, fornecer maior ou menor largura de banda e até mesmo poder limitar a taxa de perda de pacotes por fluxo. Nessa tese o escalonador utilizado foi o WFQ devido ao fato que o mesmo consegue atender a várias filas simultaneamente e diferencia-las através do peso atribuído a cada uma delas. Foi mostrado um modelamento matemático do escalonador e um exemplo passo a passo do seu funcionamento.

4 PROPOSTA DE MODELOS ANALÍTICOS PARA UM NÓ DE REDE COM O ESCALONADOR WFQ

4.1 Introdução

O desenvolvimento de modelos analíticos é de importância fundamental para a análise e projeto de sistemas e protocolos em redes de telecomunicações. Entre as características desejadas para um modelo estão a sua simplicidade, obtida ao se restringir somente aos aspectos que influenciem significativamente no comportamento que se deseja analisar, e a sua tratabilidade, geralmente associada à complexidade de se gerar resultados analisáveis a partir do modelo (GARETTO, 2004).

Atualmente, o desempenho de algumas redes não está atingindo um patamar satisfatório devido a falta de recursos físicos (largura de banda, capacidade de processamento dos roteadores, capacidade dos *buffers*, entre outros) e também devido a uma não adequada priorização de fluxos que trafegam pela rede (DEKERIS; ADOMKUS; BUDNIKAS, 2006).

O escalonador, configurado nos roteadores de rede, atua sobre os fluxos de acordo com suas prioridades e é responsável então pela qualidade de serviço obtida pelos mesmos.

O objetivo desse capítulo corresponde ao desenvolvimento de um modelo para o escalonador WFQ que seja capaz de fornecer métricas de desempenho, em função de uma dada configuração, quando o tráfego considerado apresenta características auto-similares. O modelo proposto deve apresentar baixo esforço computacional e ser acurado dentro da faixa de variação, normalmente encontrada na prática, dos seus principais parâmetros (fator de utilização, tamanho de buffer e configuração do WFQ).

Nesse trabalho o tráfego considerado é do tipo TCP devido ao fato que este protocolo é o mais utilizado na Internet atualmente (LUO; MARIN, 2005) (MEISS; MENCZER; FORTUNATO, 2008) (ERMAN et al., 2007) (BASHER et al., 2008). O intervalo de variação com relação ao fator de utilização está compreendido entre 0,40 a 0,85 visto que entre estes valores são normalmente encontrados pelas operadoras de telecomunicações (NUCCI et al., 2007) (CISCO, 2009). O tamanho do *buffer* concentra-se próximo de 80 pacotes (GARETTO, 2004) (BUSH; MEYER, 2002) (BEHESHTI et al., 2008) (BARMAN; SMARAGDAKIS; MATTA, 2004).

Esse capítulo propõe dois modelos: (1) modelo WFQ com tráfego de Poisson e (2) modelo WFQ com tráfego auto-similar. Uma descrição desses modelos assim como os

resultados obtidos pelos mesmos são apresentados e analisados. Os modelos permitem a obtenção de indicadores tais como a probabilidade de perda e atraso médio de pacotes em função dos parâmetros de configuração. Adicionalmente a latência da transmissão via TCP também é obtida por ser esta um fator importante na análise de qualidade de serviço observado por um usuário da rede.

A validação da proposta é realizada por meio de simulação com o uso do software NS-2 (versão 2.28).

Com os resultados obtidos por estes dois modelos pode-se verificar a importância de se considerar o tipo de tráfego de uma rede.

4.2 Modelo do WFQ com Tráfego de Poisson

Essa seção apresenta um modelo analítico exato para a análise de desempenho de um escalonador WFQ que considera, entretanto, tráfego Poissoniano. Este modelo está baseado na abordagem proposta por Hedge e Avrachenkov (HEDGE; AVRACHENKOV, 2004), conhecido por modelo do particionamento completo.

Considere-se um sistema com um único servidor que serve dois *buffers* separados, conforme mostrado na Figura 22. Cada classe i de pacotes possui um *buffer* de tamanho total K que é dividido internamente em dois *buffers* de tamanho M e N .

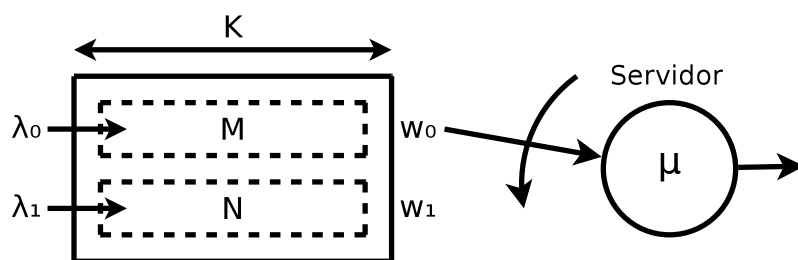


Figura 22 – Escalonador WFQ.

Para esse sistema WFQ, assume-se duas classes de fluxos, fluxo de classe 0 e fluxo de classe 1. Esses fluxos chegam de acordo com o processo de Poisson de taxa λ_i , $i \in \{0,1\}$ e requerem tempos de serviços exponenciais com uma média $1/\mu_i$, $i \in \{0,1\}$. Cada classe é atribuída a uma fila virtual de dimensão configurável. As classes são servidas segundo o gerenciamento de fila do tipo FIFO.

A fila i é servida à taxa proporcional a w_i (para $w_i > 0$) quando a fila $1-i$ não está vazia e à taxa unitária quando aquela fila está vazia. Os coeficientes (pesos) w_i são tais que $w_0 + w_1 = 1$. O servidor serve fluxos se ao menos uma fila não está vazia.

A Figura 23 ilustra o diagrama de transição de estados da cadeia de Markov que modela o escalonador WFQ. Os estados são representados por (m, n) , onde m e n são o número de clientes nas classes 0 e 1, respectivamente. Quando o processo está no estado (m, n) , ele pode transitar para o estado $(m+1, n)$, $(m, n+1)$, $(m-1, n)$ se $m > 0$ e $(m, n-1)$ se $n > 0$. A taxa de transição do estado (m, n) para $(m, n+1)$, onde $(0 \leq n \leq N-1)$, é a taxa de chegada da classe 0, isto é, λ_0 , do processo de Poisson. A taxa de transição do estado (m, n) para $(m+1, n)$, onde $(0 \leq m \leq M-1)$, é a taxa de chegada da classe 1, isto é λ_1 , do processo de Poisson. Quando não há clientes de classe 0 no sistema a taxa de transição de $(0, n)$ para $(0, n-1)$ é a taxa de serviço da classe 0, isto é, μ . Na mudança do estado $(m, 0)$ para $(m-1, 0)$, a taxa de serviço da classe 1, é μ . Entretanto a taxa de transição do estado (m, n) para $(m-1, n)$ é a taxa de serviço da classe 1 multiplicado pelo peso da classe 1, isto é $w_1\mu$. Uma transição de (m, n) para $(m, n-1)$ possui taxa de serviço da classe 0 multiplicada pelo peso da classe 0, isto é $w_0\mu$.

O gerador infinitesimal desse processo é dado por:

$$Q_{(m,0),(m-1,0)} = \mu, m \geq 1 \quad (29)$$

$$Q_{(m,n),(m-1,n)} = w_1\mu \quad (30)$$

$$Q_{(m,n),(m,n-1)} = w_0\mu, m \geq 1, n \geq 1 \quad (31)$$

$$Q_{(m,n),(m+1,n)} = \lambda_1 \quad (32)$$

$$Q_{(m,n),(m,n+1)} = \lambda_0, m \geq 0, n \geq 0 \quad (33)$$

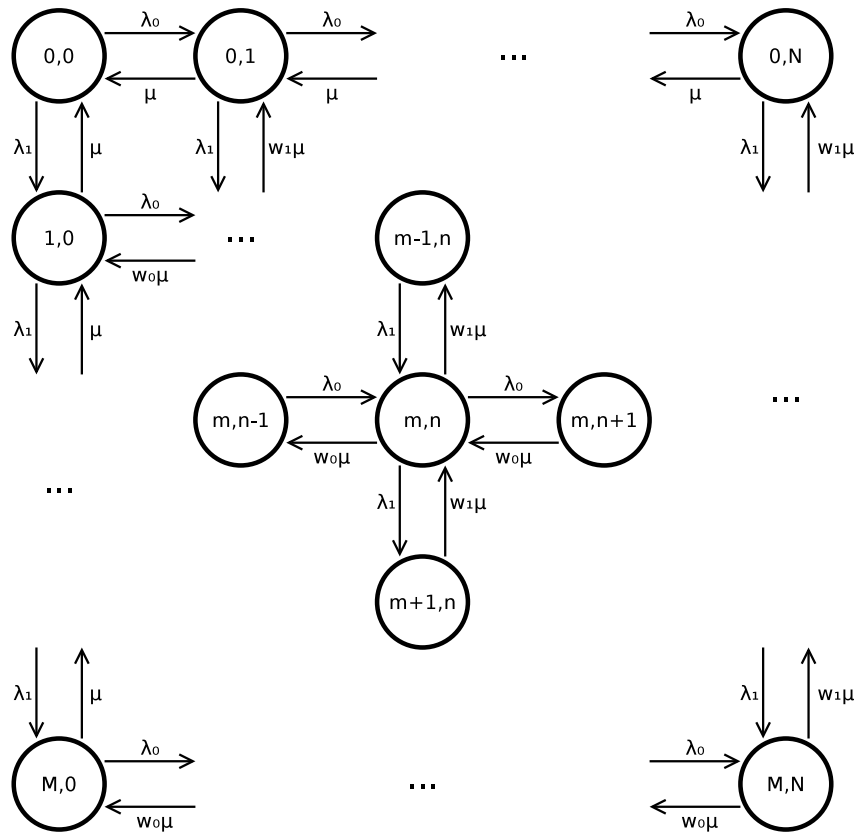
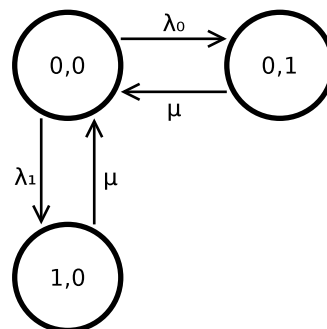


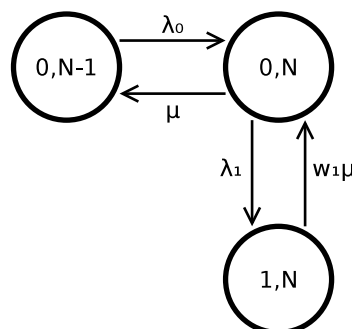
Figura 23 – Cadeia de Markov para o modelo WFQ.

As figuras a seguir apresentam as equações de equilíbrio para o sistema.



$$(\lambda_0 + \lambda_1).P(0,0) - \mu.P(0,1) - \mu.P(1,0) = 0$$

Figura 24 – Diagrama de estados e equação de equilíbrio para $m = 0$ e $n = 0$.



$$(\mu + \lambda_1).P(0, N) - \lambda_0.P(0, N - 1) - \mu.w_1.P(1, N) = 0$$

Figura 25 – Diagrama de estados e equação de equilíbrio para $m = 0$ e $n = N$.

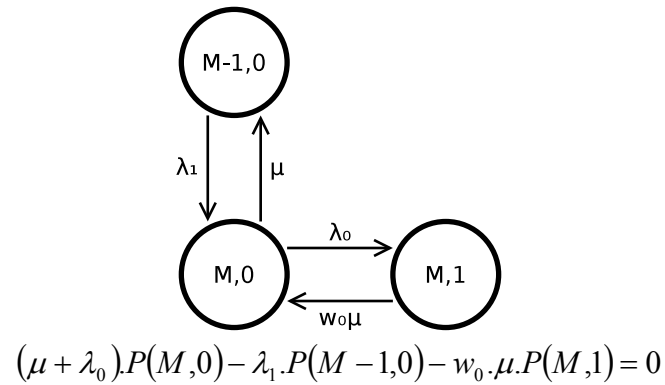


Figura 26 – Diagrama de estados e equação de equilíbrio para $m = M$ e $n = 0$.

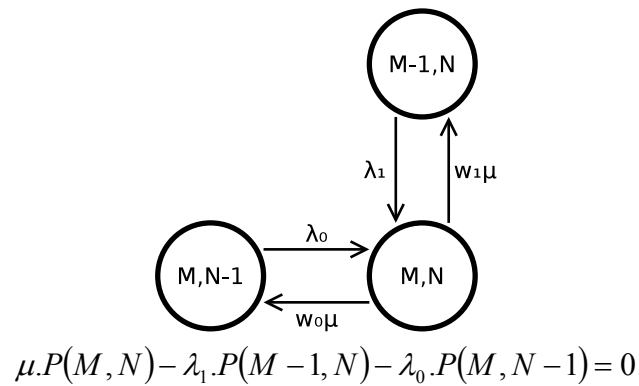


Figura 27 – Diagrama de estados e equação de equilíbrio para $m = M$ e $n = N$.

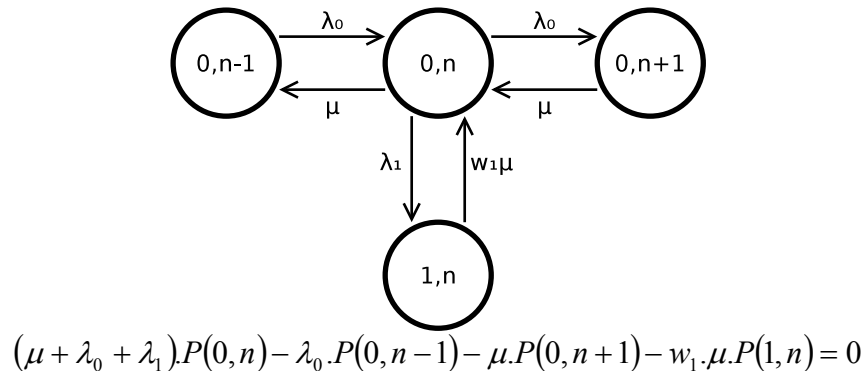
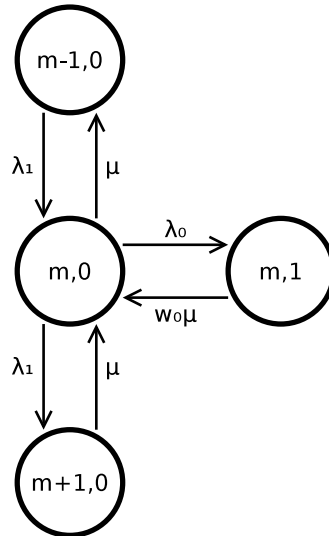
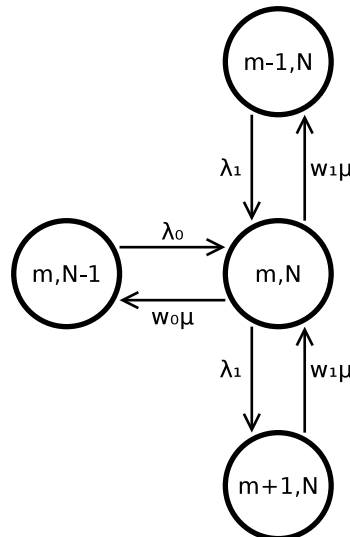


Figura 28 – Diagrama de estados e equação de equilíbrio para $m = 0$ e $1 \leq n \leq N-1$.



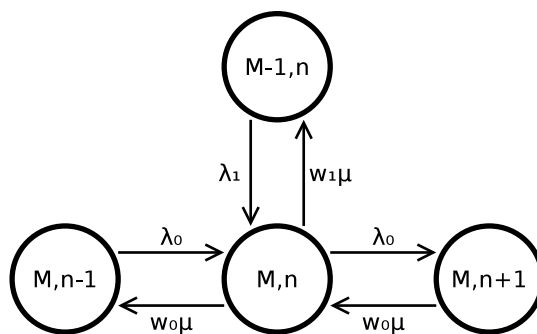
$$(\mu + \lambda_1 + \lambda_0).P(m,0) - \lambda_1.P(m-1,0) - \mu.P(m+1,0) - \mu.w_0.P(m,1) = 0$$

Figura 29 – Diagrama de estados e equação de equilíbrio para $1 \leq m \leq M-1$ e $n = 0$.



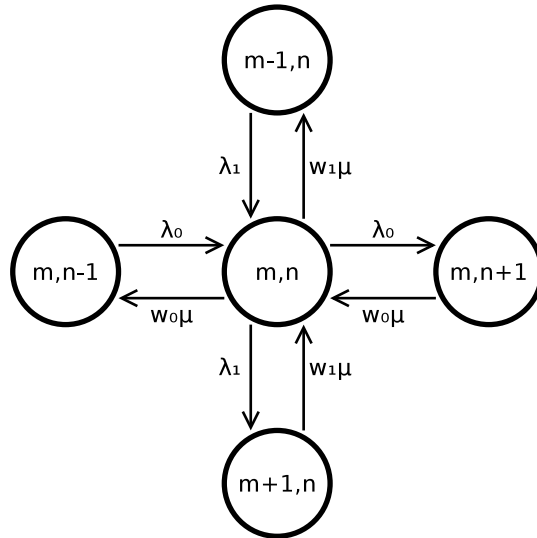
$$(\mu + \lambda_1).P(m, N) - \lambda_1.P(m-1, N) - \lambda_0.P(m, N-1) - w_1.\mu.P(m+1, N) = 0$$

Figura 30 – Diagrama de estados e equação de equilíbrio para $1 \leq m \leq M-1$ e $n = N$.



$$(\mu + \lambda_0).P(M, n) - \lambda_1.P(M-1, n) - w_0.\mu.P(M, n+1) - \lambda_0.P(M, n-1) = 0$$

Figura 31 – Diagrama de estados e equação de equilíbrio para $m = M$ e $1 \leq n \leq N-1$.



$$(\mu + \lambda_0 + \lambda_1).P(m, n) - \lambda_1.P(m-1, n) - w_0.\mu.P(m, n+1) - w_1.\mu.P(m+1, n) - \lambda_0.P(m, n-1) = 0$$

Figura 32 – Diagrama de estados e equação de equilíbrio para o caso geral.

4.2.1 Resolução da Cadeia de Markov

As probabilidades estacionárias de estado $P(m, n)$ para o sistema com o escalonador WFQ podem ser estimadas resolvendo um sistema de equações de equilíbrio escrita por:

$$P \cdot Q = 0 \quad (34)$$

A Equação (34) é um sistema homogêneo de equações lineares que pode ser resolvido utilizando a eliminação Gaussiana, desde que seja trocada qualquer uma das k equações pela equação normalizada (35).

$$\sum_{m=0}^M \sum_{n=0}^N P(m, n) = 1 \quad (35)$$

Isso permite que se encontre uma solução bem definida.

$$P \cdot Q = b, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{bmatrix} \quad (36)$$

Este sistema de equações pode ser resolvido através uma ferramenta computacional, tal como o Matlab (aqui utilizado). O pseudocódigo do algoritmo se encontra no Anexo C.

4.2.2 Cálculo das Métricas

A probabilidade de perda de pacotes de cada fila pode ser calculada como a probabilidade do *buffer* estar cheio, conforme as Eqs. (37) e (38).

$$P_{loss_0} = \sum_{m=0}^M P(m, N) \quad (37)$$

$$P_{loss_1} = \sum_{n=0}^N P(M, n) \quad (38)$$

O número médio de pacotes na fila \bar{N}_i , $i = \{0,1\}$ é dado por:

$$\bar{N}_0 = \sum_{n=0}^N \sum_{m=0}^M n \cdot P(m, n) \quad (39)$$

$$\bar{N}_1 = \sum_{m=0}^M \sum_{n=0}^N m \cdot P(m, n) \quad (40)$$

O tempo médio de espera na fila \bar{W}_i , $i \in \{0,1\}$, pode ser determinado pela Lei de Little (KLEINROCK, 1975) (KLEINROCK, 1976) (GROSS; HARRIS, 1998):

$$\bar{W}_i = \frac{\bar{N}_i}{T_i} \quad (41)$$

onde T_i é o *throughput* do fluxo i .

4.2.3 Resultados Numéricos

Nessa seção serão apresentados alguns resultados numéricos considerando uma capacidade de enlace de 25 Mbps e tamanho dos *buffers* iguais a 40 pacotes. São analisadas duas situações para os pesos do escalonador WFQ (30%-70% e 10%-90%).

A Figura 34 e a Figura 34 mostram os valores de RTT e Ploss em função do fator de utilização para as situações em que o escalonador está configurado com pesos 30%-70% e 10%-90%, respectivamente. O valor do RTT é calculado através da Equação (42).

$$RTT_i = 2 \cdot d_L + \bar{W}_i \quad (42)$$

onde dL é o retardo de propagação do enlace.

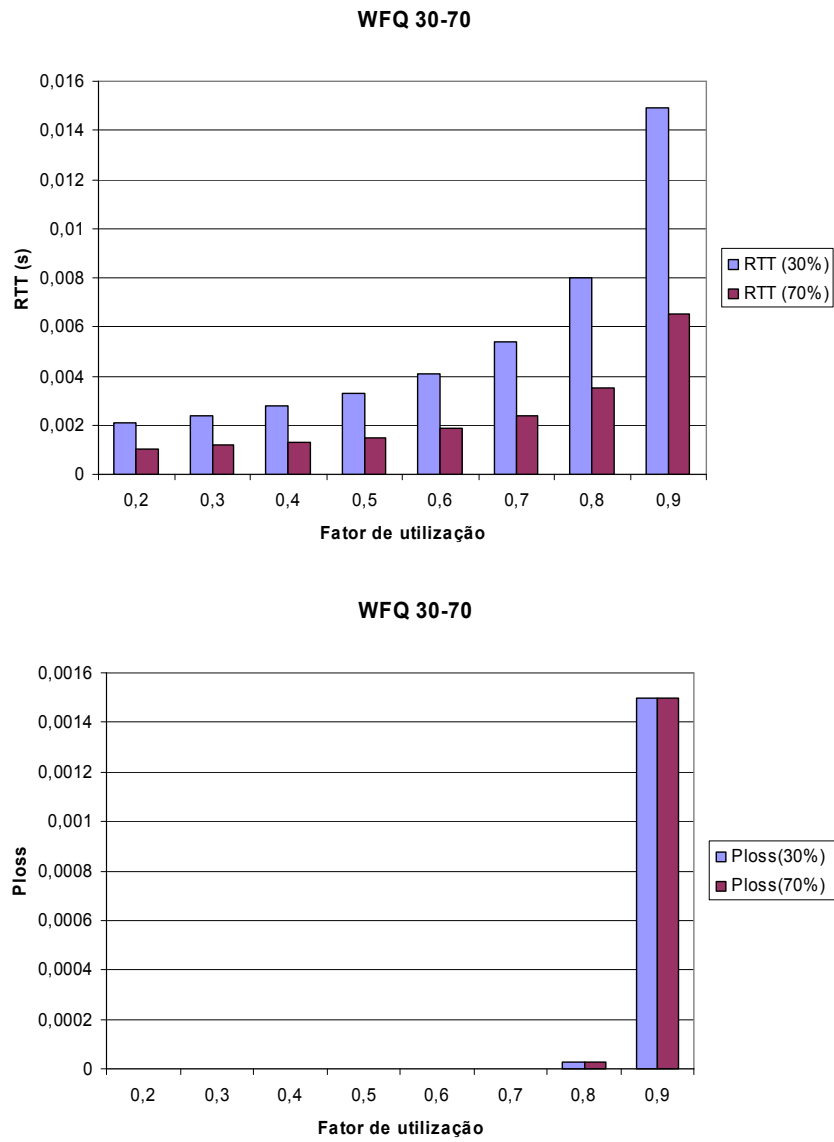


Figura 33 – Valores de RTT e Ploss em função do valor do fator de utilização para a situação 30%-70%.

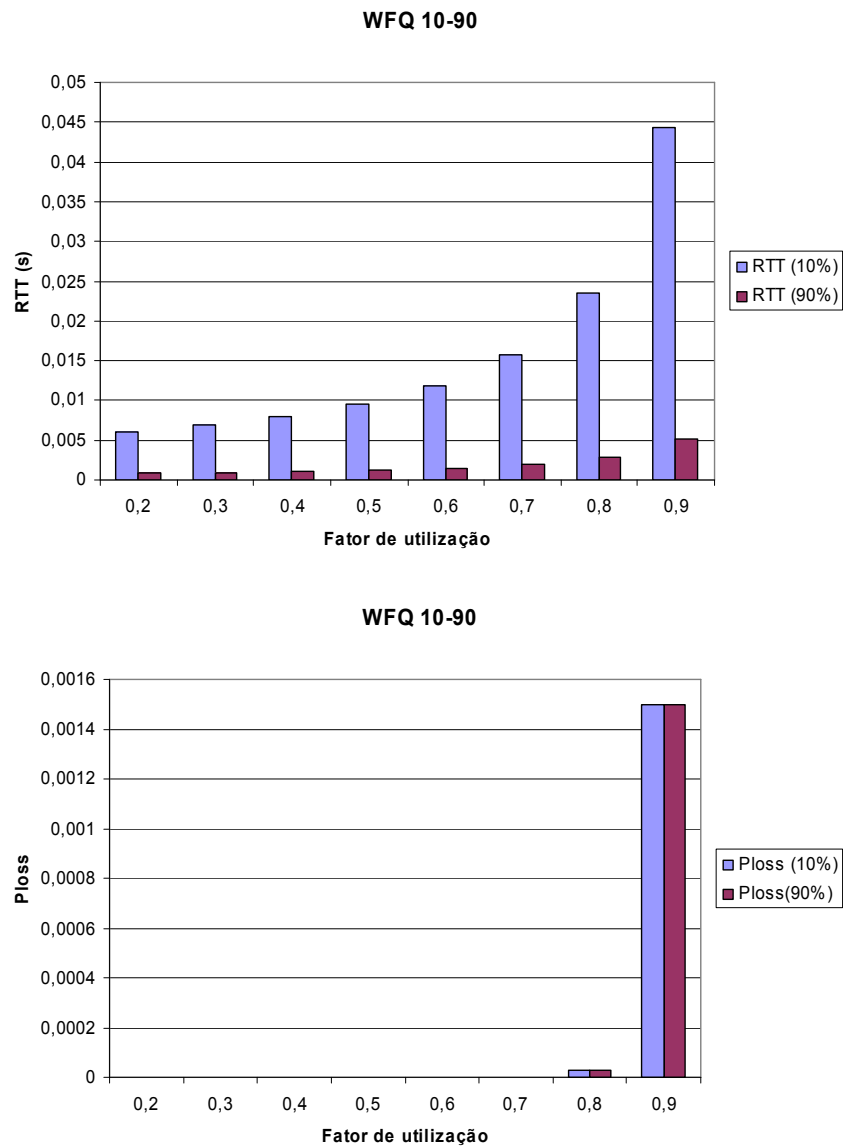


Figura 34 – Valores de RTT e Ploss em função do valor do fator de utilização para a situação 10%-90%.

Nota-se, de ambas as figuras, que o valor do RTT aumenta para maiores valores do fator de utilização e diminui quando o peso da fila do escalonador for maior. Os valores de Ploss são iguais independentemente do peso do escalonador, uma vez que o tamanho do *buffer* é o mesmo em todas as situações.

A segunda análise consistiu em manter o fator de utilização fixo em 0,8 e variar o tamanho do *buffer* de 20 a 140 pacotes. A Figura 35 e a Figura 36 mostram os valores de Ploss e RTT em função do tamanho do *buffer*, com as respectivas configurações de pesos de 30%-70% e 10%-90%.

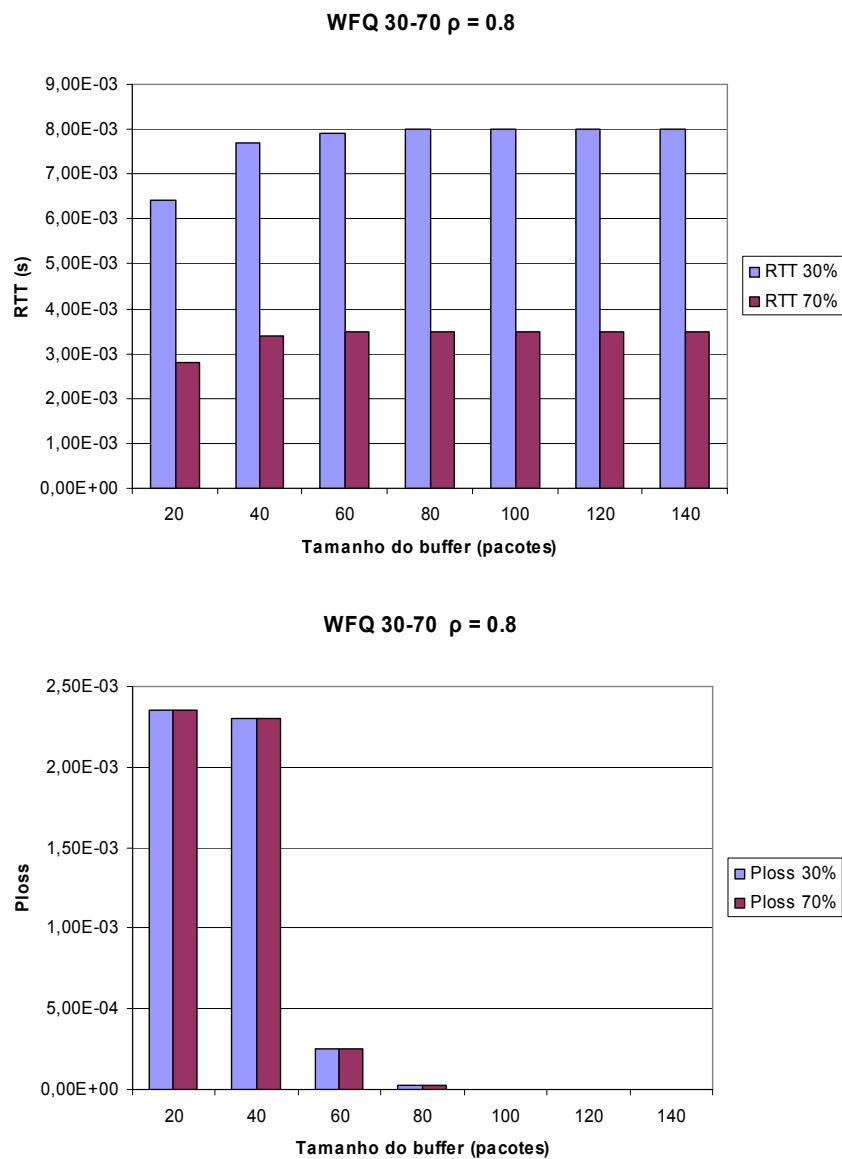


Figura 35 – Valores de RTT e Ploss para a situação 30%-70% em função do tamanho do *buffer*.

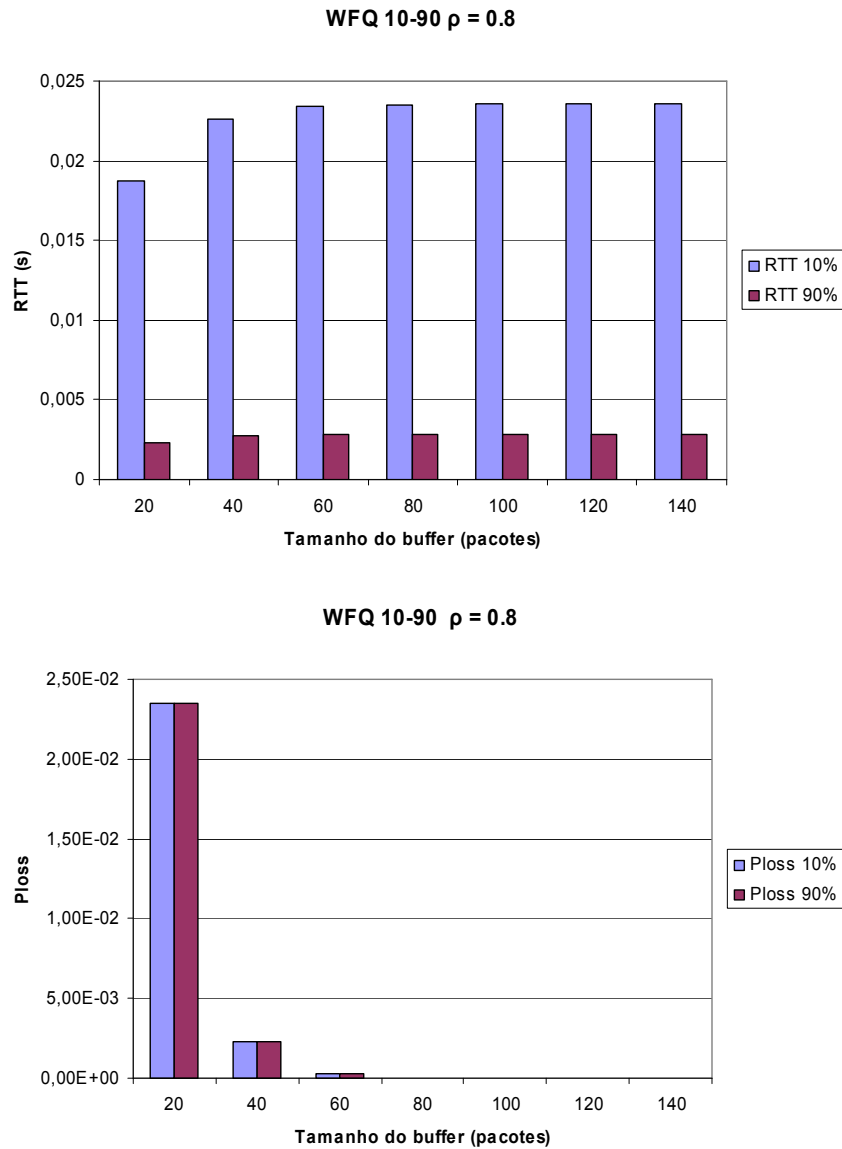


Figura 36 – Valores de RTT e Ploss para a situação 10%-90% em função do tamanho do buffer.

Observa-se nas figuras anteriores que o valor de RTT é maior para um menor peso do escalonador, e este valor se torna estável a partir do tamanho do *buffer* de 60 pacotes. Com relação ao valor de Ploss este valor diminui com o aumento do tamanho do *buffer* se tornando praticamente igual a zero a partir do tamanho do *buffer* igual a 80 pacotes.

4.3 Modelo WFQ com Tráfego Auto-Similar

Nesta seção será analisado o comportamento do escalonador WFQ quando o mesmo é alimentado com tráfego auto-similar. A motivação está baseada no fato de que o tráfego que atualmente alimenta os roteadores existentes na Internet apresenta comportamento auto-similar.

O modelo proposto para análise do sistema WFQ separa os fluxos em duas filas e estas são configuradas independentemente uma da outra, fazendo com que o tráfego de uma fila não interfira na outra. Cada fila considera o modelo Markoviano com chegada em grupos $M_{[X]}/M/1/B$, onde $[X]$ representa a distribuição do tamanho do grupo (em pacotes), com configurações de parâmetros de tempo de serviço ajustados para modelar o comportamento das filas de baixa e alta prioridades. Essa fila é utilizada nessa tese devido ao fato de que este modelo mostra boa concordância com resultados de simulação, conforme demonstrada na literatura corrente (Figura 37).

Desta forma consideram-se os aspectos que mais influenciam no comportamento do sistema WFQ e, ao mesmo tempo, permite-se a fácil obtenção das métricas associadas.

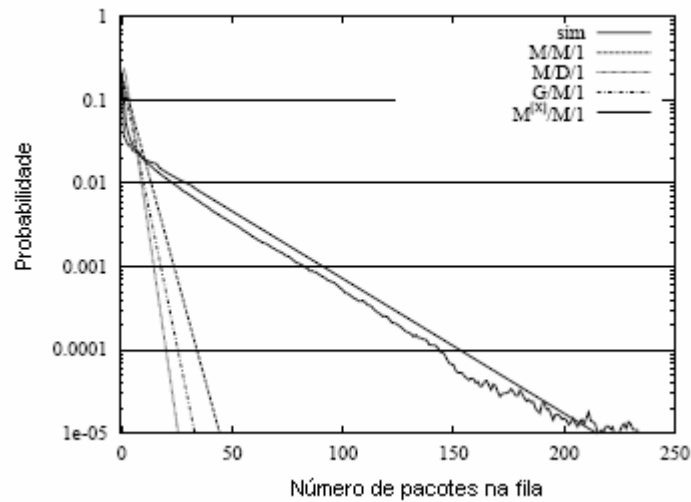


Figura 37 – Distribuição de probabilidade do comprimento da fila obtida via simulação e por diferentes modelos analíticos com $\rho=0,8$ (GARETTO, 2004).

A proposta está baseada nas seguintes considerações:

1) O *buffer* foi separado em duas partes fazendo com que se tenham dois fluxos independentes, isto é, não há uma correlação entre os fluxos 0 e 1; conforme mostra a Figura 38.

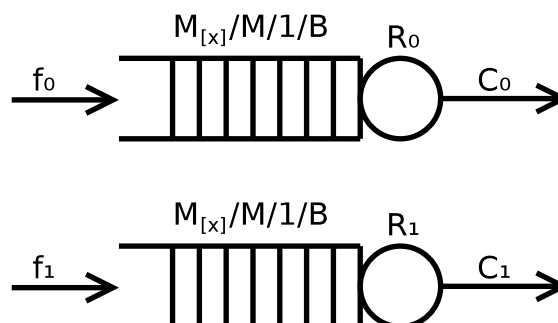


Figura 38 – Modelo Equivalente para o sistema WFQ.

2) A capacidade total do enlace foi separada de forma proporcional ao valor do WFQ fornecendo as capacidades para o modelo equivalente: $C_i = w_i \cdot C$, $i \in \{0,1\}$.

Por exemplo, se a capacidade do enlace for igual a 25 Mbps e se a configuração do escalonador WFQ for 10%- 90%, a capacidade efetiva observada pelos fluxos 0 e 1 será de 2,5 Mbps e 22,5 Mbps.

3) Para que o sistema esteja em equilíbrio é necessário que o índice de utilização seja menor que a unidade: $\rho = f_i / C_i < 1$, $i \in \{0,1\}$.

Por exemplo, na situação do item 2, se o fator de utilização for $\rho = 0,8$, o fluxo do enlace superior será igual a 2 Mbps e do enlace inferior igual a 18 Mbps.

4.3.1 Formulação Matemática da Fila M[X]/M/1/B

Nesta seção será apresentada a formulação matemática para a fila M[X]/M/1/B visando obter a probabilidade de descarte de pacotes e o tempo de espera na fila. A Figura 39 mostra o diagrama de transição de estados da fila M[X]/M/1/B, considerando a fila i no modelo equivalente.

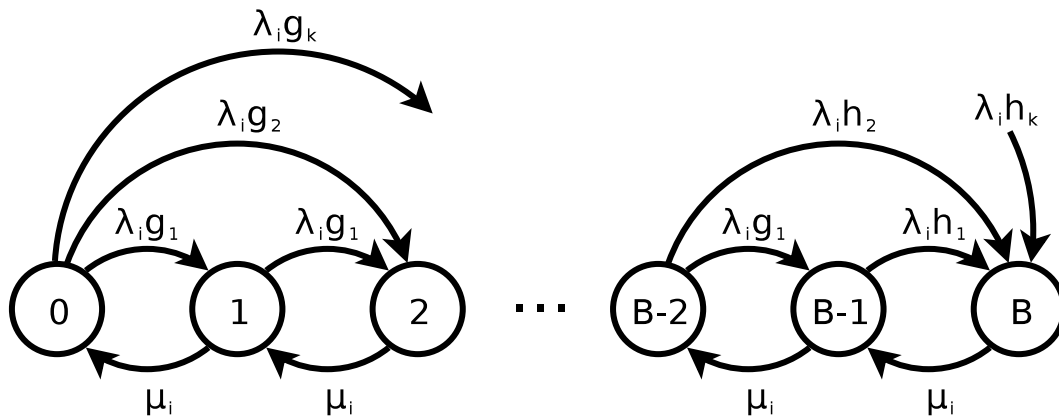


Figura 39 – Diagrama de transição de estados da fila M[x]/M/1/B.

Seja $g_k = P[X = k]$ a probabilidade de que o grupo seja formado por k pacotes e $h_k = P[X \geq k] = \sum_{m=k}^{\infty} g_m$ a probabilidade de que o grupo seja maior do que a capacidade restante de armazenamento do *buffer* em um determinado instante. Sejam λ_i a taxa de chegada e μ_i a taxa de serviço. As equações de equilíbrio do diagrama da Figura 38 são fornecidas abaixo, onde p_j são as probabilidades de estado estacionárias.

$$\lambda_i \cdot p_j = \mu_i \cdot p_{j+1}, j = 0 \quad (43)$$

$$(\lambda_i + \mu_i) \cdot p_j = \mu_i \cdot p_{j+1} + \lambda_i \cdot \sum_{k=0}^{j-1} p_k \cdot g_{j-k}, \quad 0 < j < B \quad (44)$$

$$\mu_i \cdot p_j = \lambda_i \cdot \sum_{k=0}^{j-1} p_k \cdot h_{j-k}, \quad j = B \quad (45)$$

De modo a modelar o escalonador WFQ deve-se ter então $\mu_i = w_i \cdot \mu$, $i \in \{0,1\}$. A resolução deste sistema de equações é realizada com base no método da Seção 4.2.1. O valor de Ploss é obtido encontrando-se o valor de P_B e o atraso médio na fila é calculado usando a Lei de Little.

4.4 Cenário para Simulação

O cenário configurado no NS-2 (versão 2.28) para validação da proposta é mostrado na Figura 40, e tem as seguinte características: 100 fontes TCP distribuídas em dois grupos de 50 fontes, de forma a se produzir dois fluxos agregados com prioridades diferentes ($f1$ e $f2$). O tamanho dos arquivos que são enviados do transmissor para o receptor tem distribuição conforme item 2.5 para que o tráfego tenha um comportamento que se assemelha ao tráfego real da Internet com características auto-similares. O escalonador WFQ é configurado no roteador R1, possuindo buffer para 80 pacotes e capacidade de enlace igual a 25 Mbps. Os valores buscados são: RTT e Ploss (camada de rede) e latência (camada de transporte). O fator de utilização do enlace (ρ) varia de 0,2 até 0,9. O atraso de propagação entre os dois roteadores é de 0,5 ms.

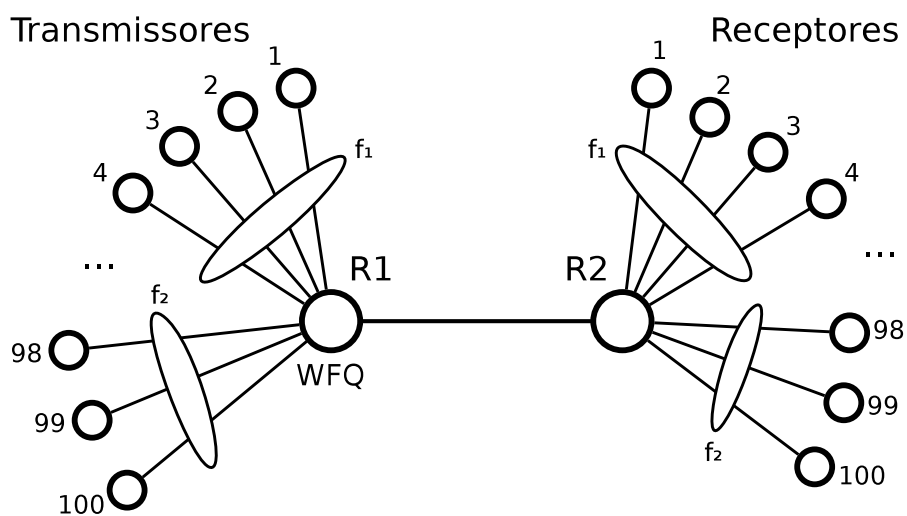


Figura 40 – Topologia Utilizada no Simulador NS-2.

4.5 Comparação de Resultados entre o Modelo Analítico e a Simulação

A seguir os resultados numéricos obtidos a partir do modelo analítico serão comparados a resultados de simulação conduzida com o NS-2 para diferentes configurações do cenário básico.

4.5.1 Resultados Obtidos na Camada de Rede

Para que se possa abranger um maior número de situações fez-se uma variação no valor do fator de utilização do enlace (ρ) de 0,2 até 0,9 e uma variação na configuração do escalonador WFQ de 10%-90%, 30%-70% e 50%-50%.

Inicialmente consideraram-se fluxos agregados de mesma prioridade (WFQ configurado com pesos 50%-50%). As simulações conduzidas com o NS-2 foram executadas até que as estimativas de interesse encontrassem, com probabilidade de 95%, um desvio padrão relativo à média igual ou menor que 10%. Os resultados para RTT e Ploss foram registrados e plotados juntamente com os valores obtidos do modelo analítico proposto. Observa-se que no modelo equivalente cada fila é configurada com um buffer de tamanho 40 pacotes.

Da Figura 41 pode-se observar que os valores obtidos pela simulação e pelo modelo foram próximos para valores do fator de utilização maiores que 60%, tanto para RTT quanto para o Ploss. Pode-se observar que todos os valores de RTT foram inferiores a 0,045s e vão aumentado a medida que o fator de utilização aumenta, porque quando se tem uma maior demanda de recursos na rede (fluxo) maior será o tempo médio em fila (maior valor de RTT). O valor de RTT é calculado como sendo duas vezes o tempo de propagação do sinal (ida e volta) mais o atraso médio na fila. No NS-2, um pacote é marcado no transmissor, e, quando este recebe um ACK, tem-se o valor do RTT. O RTT médio é calculado considerando os pacotes que são enviados e confirmados.

Com relação ao Ploss, para a situação aonde se tem um maior fluxo (maior valor do fator de utilização) o *buffer* do equipamento é preenchido mais rapidamente fazendo com que o número de pacotes descartados se torne maior. Para os valores do fator de utilização inferiores a 0,9 a probabilidade de perda é inferior a 5% tanto para o modelo quanto para o NS-2.

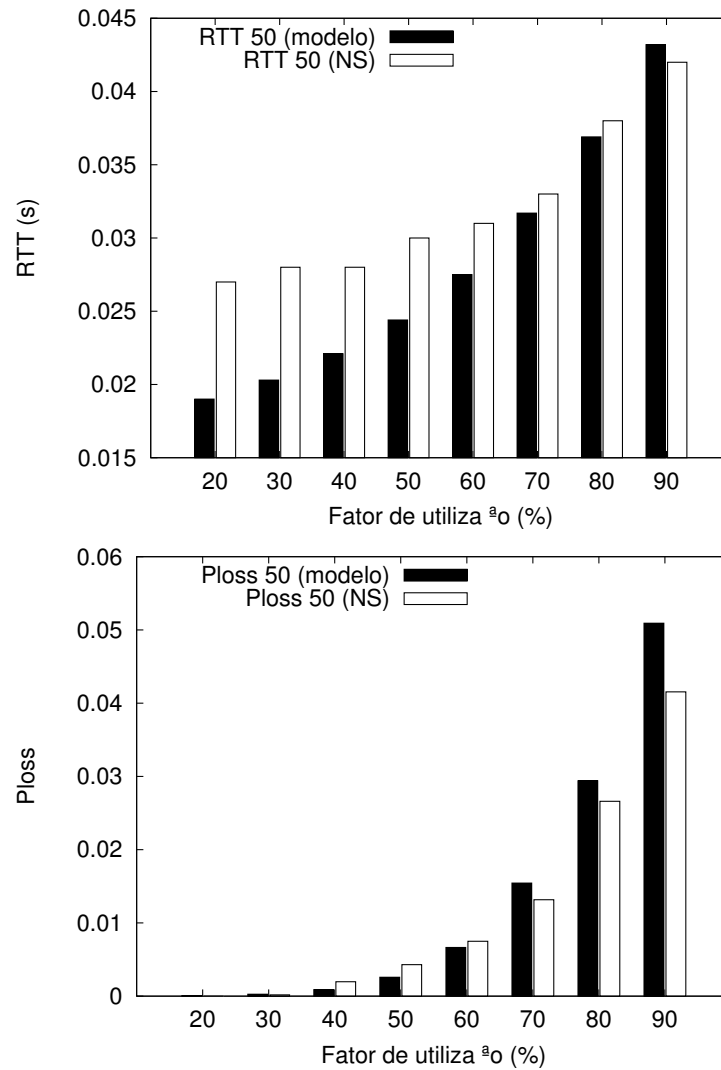


Figura 41 – Valores de RTT e Ploss em função do fator de utilização para configuração do escalonador WFQ igual a 50%-50%.

Em seguida é alterada a configuração do escalonador WFQ para que o peso das filas fique proporcional a 30% e 70%. Os resultados são apresentados na Figura 42 e na Figura 43.

Pode-se observar na Figura 42 que os valores do modelo analítico estão próximos dos valores obtidos pela simulação tanto para o fluxo 30% como para o fluxo 70%. Os valores de RTT para o fluxo que se encontra na fila com maior peso (70%) são menores em relação ao fluxo que se encontra na fila com menor peso (30%), pois a capacidade do enlace para este fluxo é maior, fazendo com que o atraso seja menor.

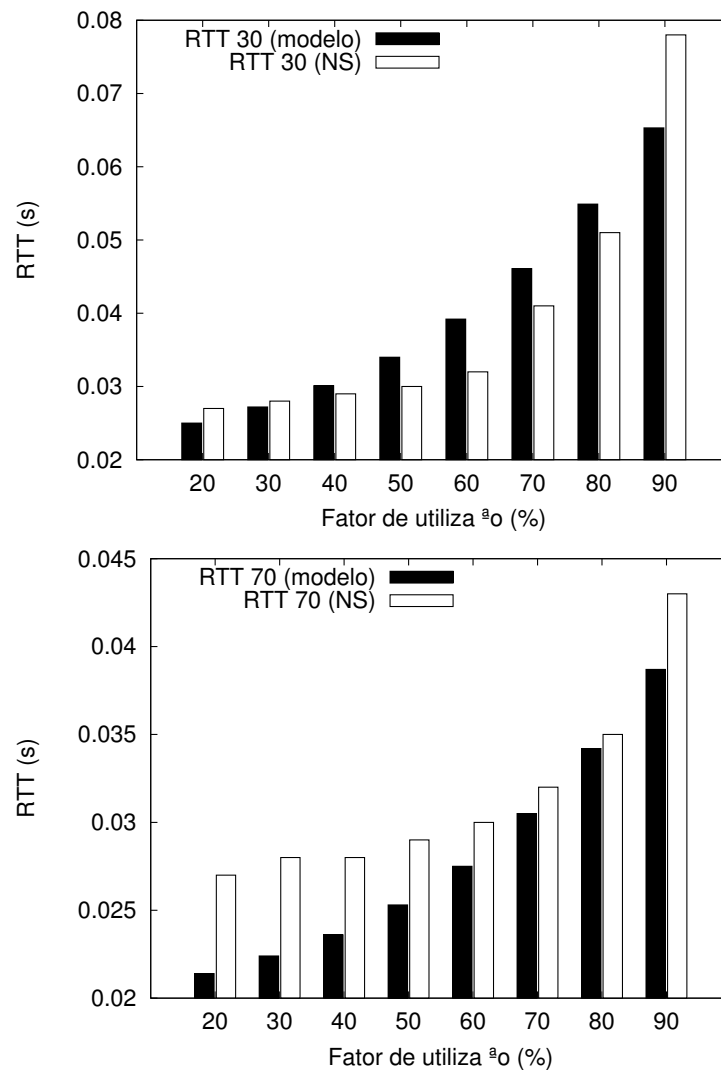


Figura 42 – Valores de RTT em função do fator de utilização para configuração do escalonador WFQ igual a 30%-70%.

Com relação ao Ploss, pode-se observar na Figura 43 que a diferença existente entre os fluxos de 30% e 70% são pequenas, visto que o fator que mais interfere neste caso é o tamanho do *buffer* (nesta situação têm-se os valores do *buffer* iguais para cada fluxo).

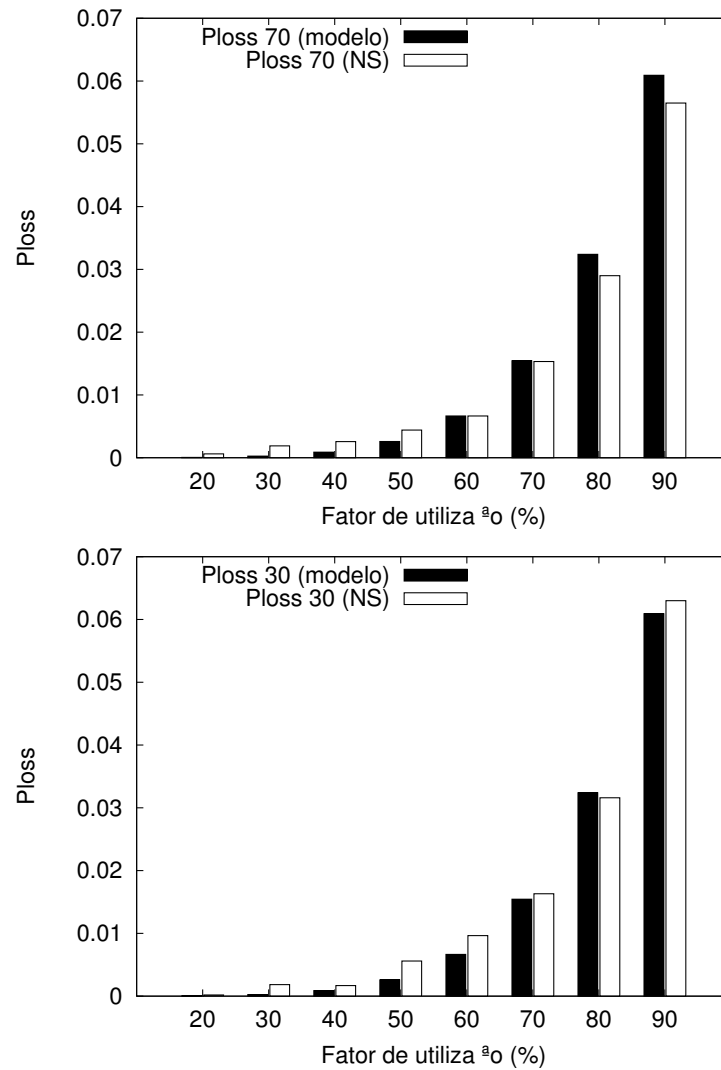


Figura 43 – Valores de Ploss em função do fator de utilização para configuração do escalonador WFQ igual a 30%-70%

Em seguida é alterada a configuração do escalonador WFQ para que o peso das filas seja proporcional a 10%-90%. Os valores obtidos estão representados na Figura 44 e na Figura 45.

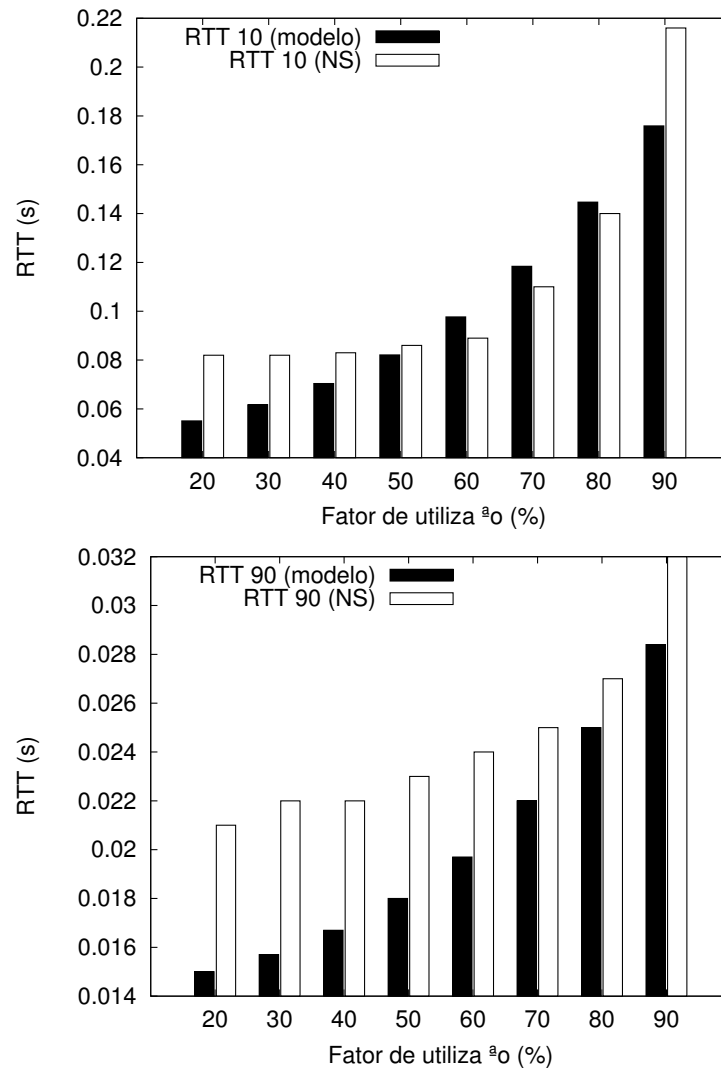


Figura 44 – Valores de RTT em função do fator de utilização para configuração do escalonador WFQ igual a 10%-90%.

A Figura 44 mostra que os valores de RTT do fluxo de 90% em comparação com o fluxo de 10% exibem uma diferença significativa devido ao fato de as bandas disponíveis para cada um dos fluxos serem muito diferentes.

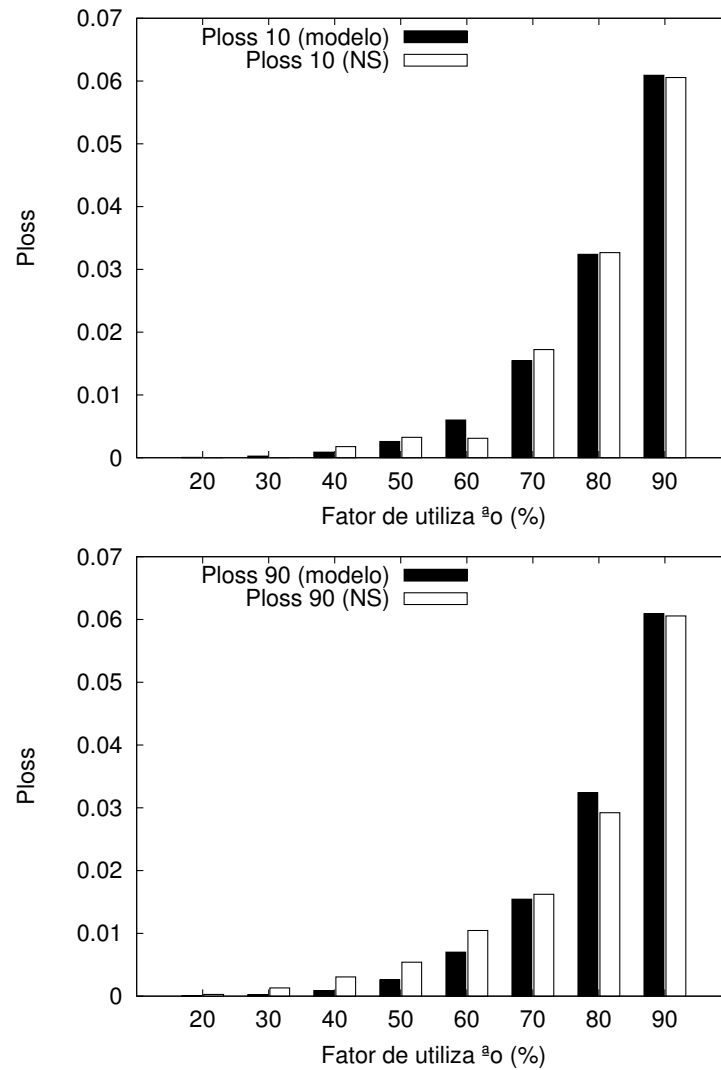


Figura 45 – Valores de Ploss em função do fator de utilização para configuração do escalonador WFQ igual a 10%-90%.

Os valores de Ploss, no entanto, são praticamente iguais devido ao fato de que o fator de maior influência sobre os valores de Ploss é o tamanho do *buffer*.

4.5.1.1 Análise de Erro

A seguir apresenta-se uma análise de erro quando se compara valores encontrados com o uso do modelo analítico e aqueles obtidos via simulação. O erro foi calculado pela seguinte fórmula:

$$|E| = \frac{|V_{\text{modelo}} - V_{\text{simulado}}|}{V_{\text{simulado}}} \quad (46)$$

As Figuras 46 e 47, a seguir, mostram o erro acumulado em função do fator de utilização, considerando várias configurações para o WFQ.

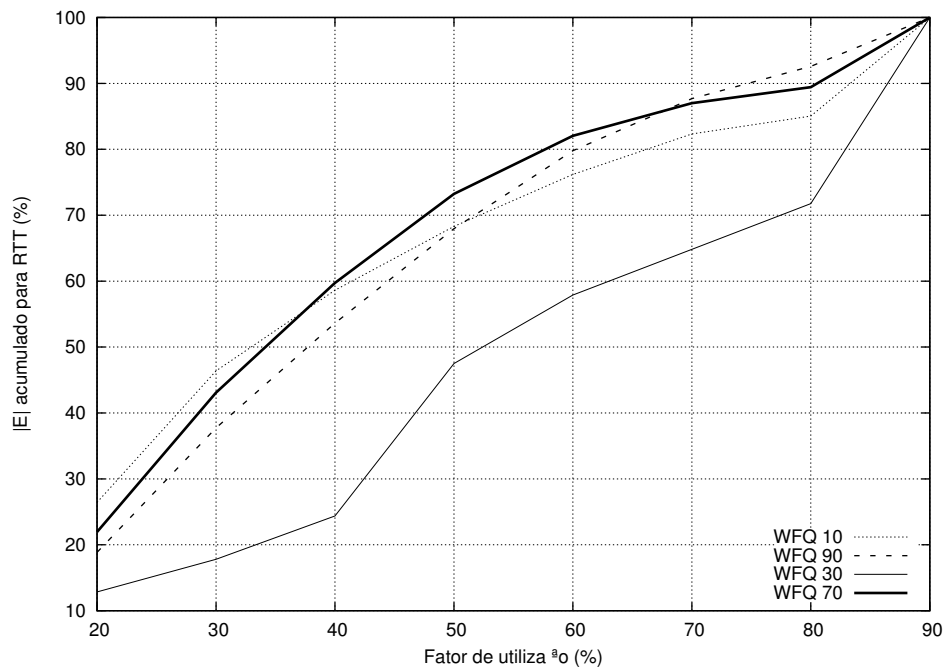


Figura 46 – Variação do valor do erro acumulado para o RTT em função do fator de utilização.

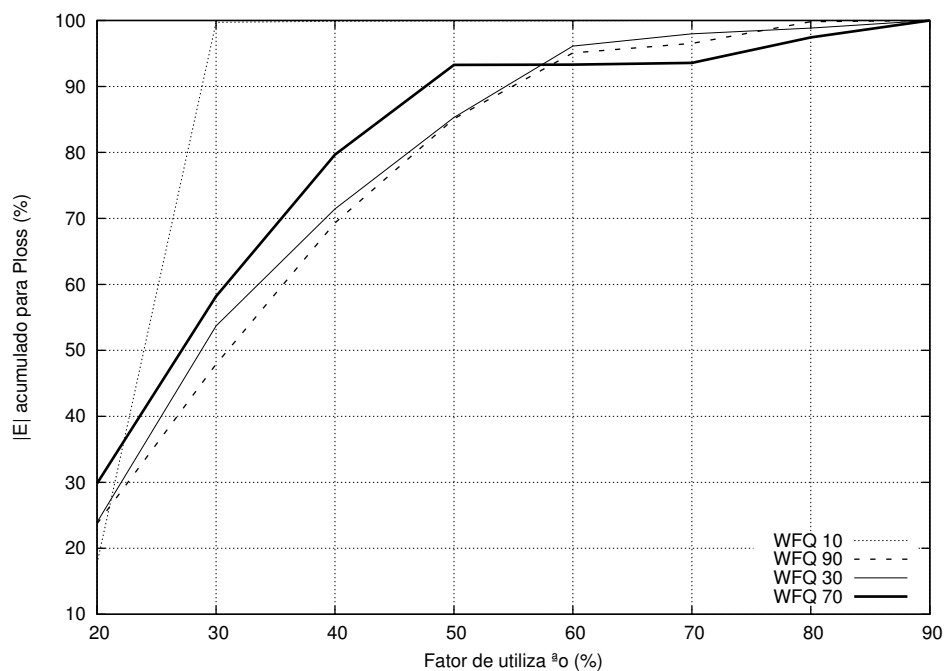


Figura 47 – Variação do valor do erro acumulado para o Ploss em função do fator de utilização.

Pode-se observar que os valores de erro acumulado crescem rapidamente até o fator de utilização de 50%. O que indica que a faixa recomendada para uso do modelo proposto deve ser aquela onde o fator de utilização é superior a 50% (nesta faixa o erro é menor do que 30% para valores de RTT e menor do que 10% para valores de Ploss).

4.5.2 Resultados Obtidos Variando o Tamanho do *Buffer*

Um segundo conjunto de resultados foi obtido para analisar o comportamento do sistema frente a uma variação no tamanho do *buffer* associado às filas no roteador. Para isto foi escolhido o tamanho do *buffer* variando de 20 a 160 pacotes com intervalos de 10 pacotes. Considerou-se o fator de utilização constante e igual a 0,8 e a configuração do escalonador WFQ constante e igual 50%-50%. Os resultados para RTT e do Ploss são mostrados na Figura 48.

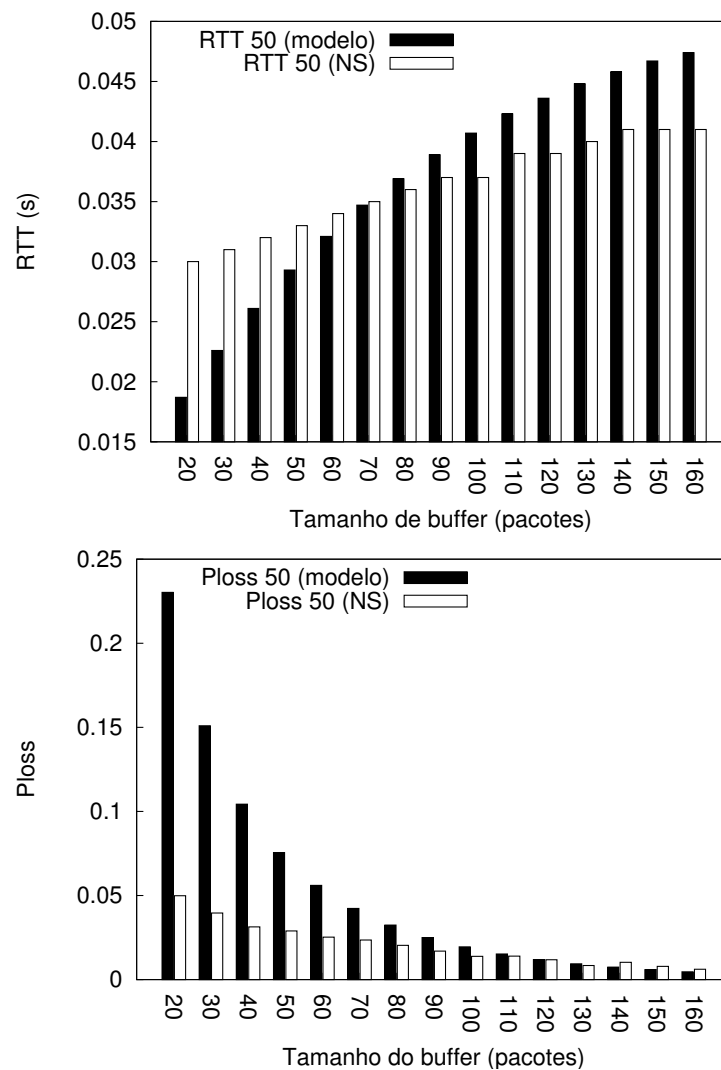


Figura 48 – Valores de RTT e Ploss em função da variação do tamanho do *buffer*.

Observa-se que o modelo equivalente apresenta uma boa aproximação no intervalo entre 70 e 120 pacotes. Pode-se observar que quando se tem um tamanho de *buffer* maior os valores de RTT serão maiores porque o tempo de atraso no *buffer* será maior e conseqüentemente o número de pacotes descartados será menor. Com estes resultados é

possível analisar o comportamento de uma rede aonde os roteadores podem ter valores diferentes de buffer.

4.5.2.1 Análise de Erro

A seguir apresenta-se uma análise de erro quando se compara valores encontrados com o uso do modelo analítico e aqueles obtidos via simulação. A metodologia adotada para se calcular o erro foi a mesma utilizado na Seção 4.2.1.1 A Figura 49 e a Figura 50 mostram o erro acumulado em função do fator de utilização.

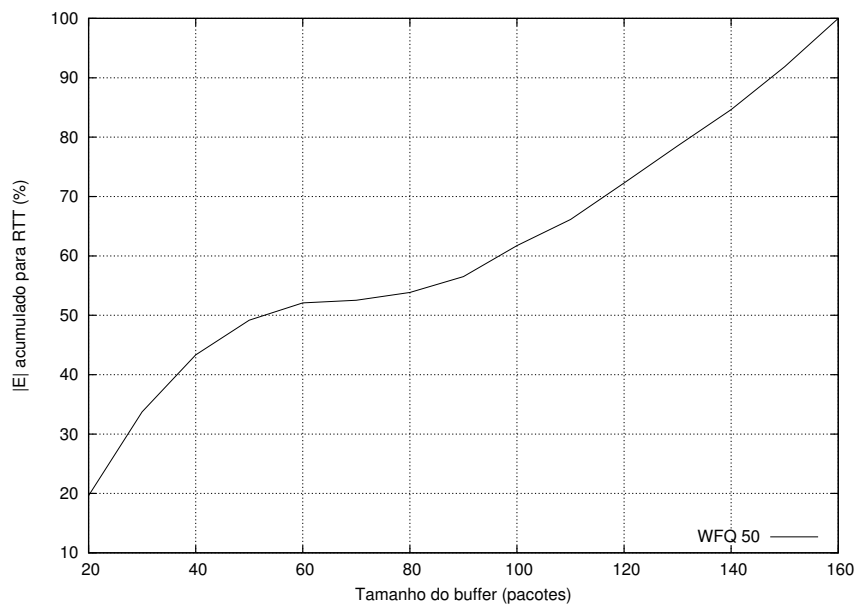


Figura 49 – Variação do valor do erro acumulado para o RTT em função do tamanho do buffer.

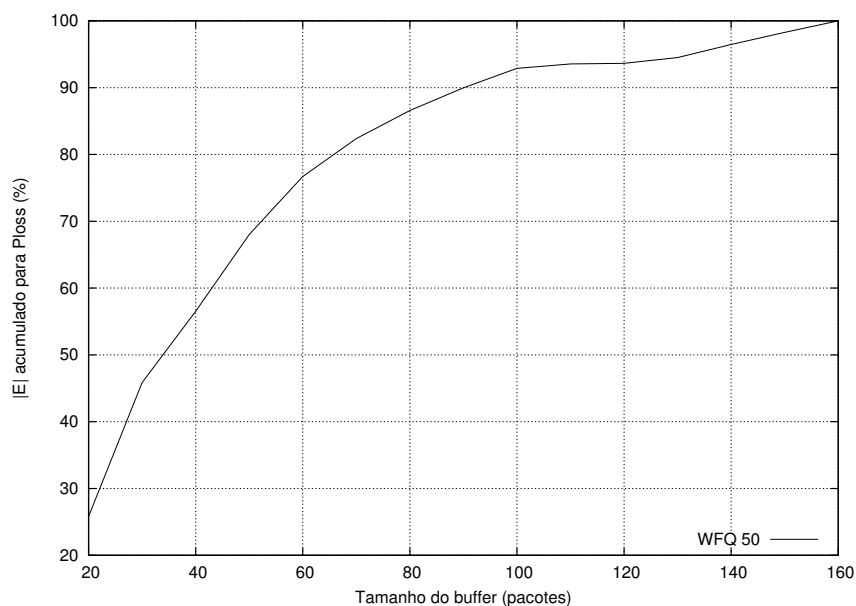


Figura 50 – Variação do valor do erro acumulado para o Ploss em função do tamanho do buffer.

Pode-se observar das figuras que os valores de erro acumulado crescem rapidamente até o valor do tamanho do *buffer* igual a 60 pacotes. O que indica que a faixa recomendada para uso do modelo proposto deve ser aquela onde o tamanho do buffer varia em torno de 80 pacotes (na faixa aonde o tamanho do *buffer* se encontra entre 60 a 100 pacotes o erro é menor do que 10% para valores de RTT e menor do que 15% para valores de Ploss).

4.5.3 Resultados Obtidos na Camada de Transporte

Para completar a avaliação do modelo proposto procurou-se verificar qual seria o valor do tempo médio de transferência de arquivos (latência) observado por uma aplicação colocada logo acima da camada de transporte. A latência é um fator importante na análise da qualidade de serviço (QoS) ofertada por uma rede.

Valores de latência obtidos com o modelo analítico do protocolo TCP proposto em (CARDWELL; SAVAGE; ANDERSON, 2000), chamado neste trabalho de modelo CSA, foram comparados a valores obtidos via simulação no NS-2. Foi considerado um tráfego onde os arquivos possuem uma distribuição conforme o item 2.5. Três cenários foram considerados levando em consideração a configuração do escalonador WFQ. No primeiro cenário 50%-50%, no segundo 30%-70% e no terceiro 10%-90%. Na Figura 51 é mostrada a latência para o primeiro cenário e é considerado o valor do fator de utilização constante e igual a 0,8.

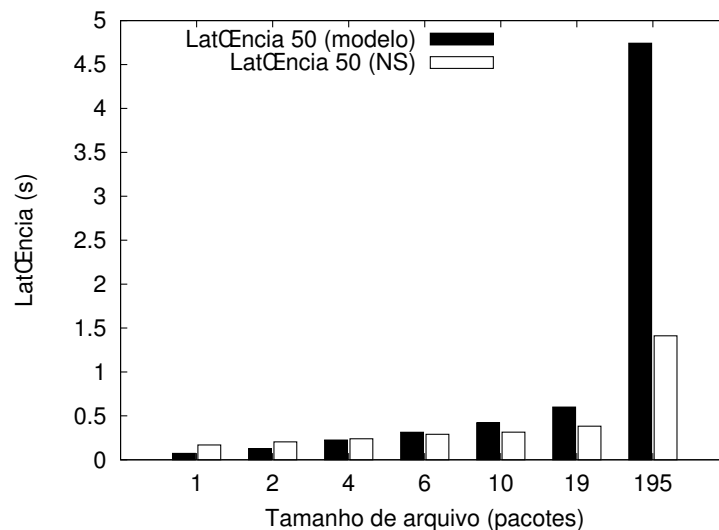


Figura 51 – Valores de latência em função do tamanho do arquivo para $\rho = 0,8$ e WFQ 50%-50%.

Os valores de latência apresentados na Figura 51 são crescentes a medida que o tamanho do arquivo aumenta, isto é, para se transferir arquivos maiores o tempo gasto também é maior. Nos gráficos de latência observa-se uma diferença significativa com relação aos arquivos com tamanho de 195 pacotes, devido ao fato que o modelo CSA superestima o

valor da latência devido a aproximação utilizada para o cálculo do tempo gasto para transmissão dos dados remanescentes após as fases de *slow start* e *fast recovery* do TCP (CARDWELL; SAVAGE; ANDERSON, 2000).

A próxima análise é feita alterando-se os valores do escalonador WFQ para 30%-70% e mantendo-se o mesmo valor para o fator de utilização.

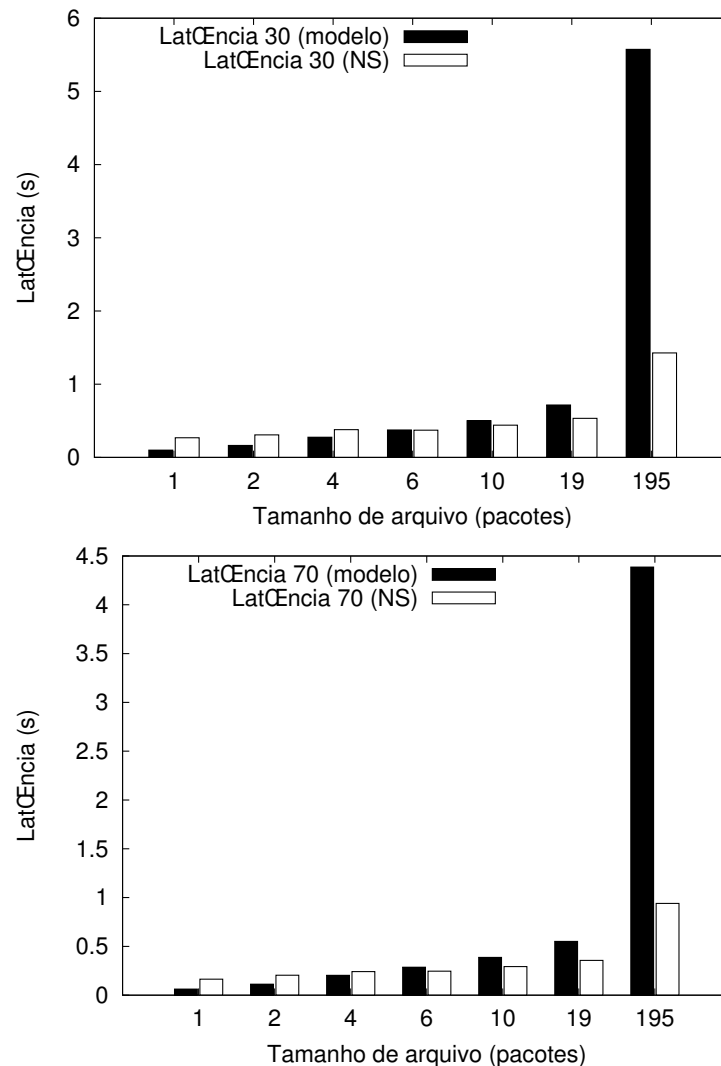


Figura 52 – Valores de latência em função do tamanho do arquivo para $\rho = 0,8$ e WFQ 30%-70%.

Os valores de latência mostrados na Figura 52 são crescentes à medida que o tamanho do arquivo aumenta, visto que é necessário mais processamento por parte do TCP. Pode-se observar também que pacotes que utilizam uma banda maior têm um tempo gasto para transferência menor. Não se tem uma relação direta com o peso do WFQ visto que outros fatores também são relevantes para esta transferência total dos arquivos, como por exemplo, o tamanho do *buffer*. Neste caso, com o valor de $\rho = 0,8$ os valores de latência são inferiores a 0,5s para arquivos de tamanho 19 pacotes. Por fim, é analisada a latência quando o

escalonador WFQ estiver configurado com os pesos 10%-90% e os valores do fator de utilização e do *buffer* são constantes, como mostra a Figura 53.

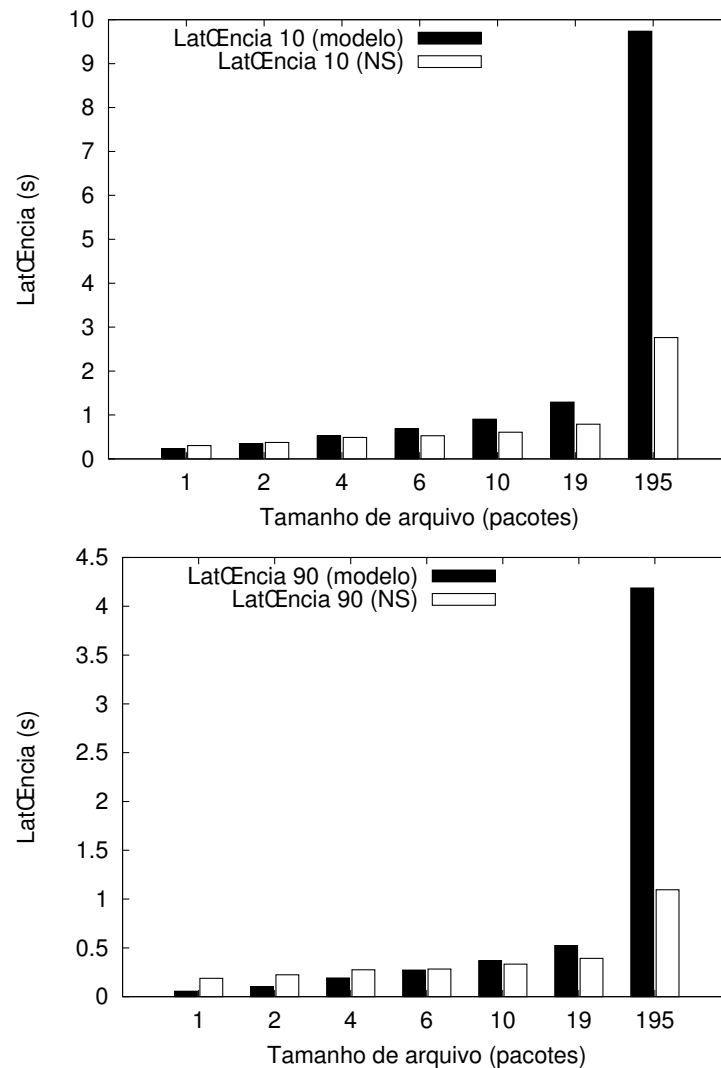


Figura 53 – Valores de latência em função do tamanho do arquivo para $\rho = 0,8$ e WFQ 10%-90%.

Pode-se observar que com o aumento do tamanho dos arquivos o valor da latência também aumenta. Sendo maior para o caso 10%, devido ao fato de que a largura de banda do fluxo de 90% é maior do que a largura de banda do fluxo de 10%, porém a diferença não é exatamente proporcional ao peso do escalonador.

4.6 Conclusão do Capítulo

Nesse capítulo foram desenvolvidos modelos para o escalonador WFQ quando o mesmo é alimentado com tráfego Poissoniano e tráfego do tipo auto-similar.

O primeiro modelo está baseado em uma cadeia de Markov bidimensional para representar o comportamento do escalonador WFQ alimentado por dois fluxos de prioridades diferentes.

O segundo modelo faz uso de duas filas separadas com taxas de serviço ajustadas de modo a simular o comportamento do escalonador WFQ. As filas são do tipo $M_{[X]}/M/1/B$ para levar em conta o comportamento auto-similar do tráfego.

Pode-se observar que os valores de RTT para o modelo que utiliza o tráfego de Poisson é aproximadamente 10 vezes menor do que o modelo que utiliza o tráfego auto-similar e para valores de Ploss o modelo que utiliza o tráfego de Poisson é mais que 20 vezes menor do que o modelo que utiliza o tráfego auto-similar. Com esta discrepância entre os modelos pode-se concluir que, em uma análise e/ou dimensionamento de uma rede real, o tipo de tráfego que alimenta os roteadores tem influência significativa.

Com os resultados obtidos neste capítulo conclui-se que é possível determinar os valores de RTT e Ploss, usando o modelo equivalente para o escalonador WFQ, com erro aceitável, se for considerada a faixa adequada de valores de índice de utilização do sistema e de dimensão do *buffer*. Tais resultados sugerem que o modelo analítico proposto fornece uma abordagem simples e eficiente para a análise e dimensionamento de redes usando o escalonador WFQ, sujeito ao tráfego auto-similar com pequeno esforço computacional.

Incluíram-se também resultados considerando a camada de transporte, pois a latência do protocolo TCP é um fator importante na análise de qualidade de serviço ofertado pela rede.

5 CONCLUSÃO

O crescimento do número de usuários e o surgimento de novas aplicações na Internet vêm exigindo das ISPs (*Internet Service Provider*), além da disponibilidade de mais largura de banda, o suporte adequado a requisitos de qualidade de serviço. Diversas soluções vêm sendo desenvolvidas como alternativas ao serviço *best effort*, destacando-se os serviços integrados (*IntServ*) e serviços diferenciados (*DiffServ*). De forma a contribuir com tais esforços, esta tese propõe e analisa modelos para o escalonador WFQ visando prover Qualidade de Serviço a determinado grupo de usuários.

Nessa tese pôde-se definir auto-similaridade quando um determinado objeto preserva as mesmas características independentemente da sua escala de espaço e tempo. Foram apresentadas diversas definições importantes relacionadas ao tema. Pode-se concluir que o dimensionamento de uma rede de dados, sem levar em consideração o fato do tráfego ter um comportamento auto-similar, pode levar a valores errôneos, como por exemplo no dimensionamento dos *buffers* e no valor da probabilidade de descarte de pacotes.

Foi mostrado o processo de geração do tráfego sintético utilizado nessa tese, sendo que este tráfego apresentada as mesmas características mostrados em (ROSSI; MUSCARIELLO; MELLIA, 2004). Na seqüência foram descritas três maneiras de se obter o parâmetro de Hurst (estimador de Whittle, estatística R/S e estimador baseado em *wavelets*), através do cálculo desse parâmetro ($0,5 < H < 1$) pôde-se comprovar que o tráfego utilizado nessa tese tem características auto-similares.

A importância de se escolher um determinado escalonador de pacotes para um nó de rede foi discutida. Através da escolha do escalonador pode-se fazer uma diferenciação dos fluxos, isto é, prover Qualidade de Serviço a diferentes aplicações, permitindo diferenciar atraso, fornecer maior ou menor largura de banda e até mesmo poder limitar a taxa de perda de pacotes por fluxo. Nessa tese o escalonador considerado foi o WFQ devido ao fato que o mesmo consegue atender a várias filas simultaneamente e diferencia-las através do peso atribuído a cada uma delas. Foi mostrado um modelamento matemático do escalonador e um exemplo passo a passo do seu funcionamento.

Como contribuição foram desenvolvidos modelos para o escalonador WFQ quando o mesmo é alimentado com tráfego Poissoniano e tráfego do tipo auto-similar. O primeiro modelo está baseado em uma cadeia de Markov bidimensional para representar o

comportamento do escalonador WFQ alimentado por dois fluxos de prioridades diferentes. O segundo modelo faz uso de duas filas separadas com taxas de serviço ajustadas de modo a simular o comportamento do escalonador WFQ. As filas são do tipo $M_{[X]}/M/1/B$ para levar em conta o comportamento auto-similar do tráfego.

Pôde-se observar que os valores de RTT para o modelo que utiliza o tráfego de Poisson é aproximadamente 10 vezes menor do que o modelo que utiliza o tráfego auto-similar e para valores de Ploss o modelo que utiliza o tráfego de Poisson é mais que 20 vezes menor do que o modelo que utiliza o tráfego auto-similar. Com esta discrepância entre os modelos pode-se concluir que, em uma análise e/ou dimensionamento de um rede real, o tipo de tráfego que alimenta os roteadores tem influência significativa.

Com os resultados obtidos nessa tese conclui-se que é possível determinar os valores de RTT e Ploss, usando o modelo equivalente para o escalonador WFQ (com 2 níveis de prioridade), com erro aceitável (conforme análise do capítulo 4), se for considerada a faixa adequada de valores de índice de utilização do sistema (entre 0,4 e 0,85) e de dimensão do *buffer* (próximo de 80 pacotes). Tais resultados sugerem que o modelo analítico proposto fornece uma abordagem simples e eficiente para a análise e dimensionamento de redes usando o escalonador WFQ, sujeito ao tráfego auto-similar com pequeno esforço computacional.

Incluiu-se também resultados considerando a camada de transporte, pois a latência do protocolo TCP é um fator importante na análise de qualidade de serviço ofertado pela rede.

Para que fosse possível cobrir uma maior diversidade de situações o escalonador WFQ foi configurado de três maneiras: 10%-90%, 50%-50% e 30%-70%, em todas as situações os valores esperados (dentro dos intervalos de interesse) foram satisfatórios.

Nessa tese o modelo analítico proposto não foi comparado diretamente com outros modelos devido ao fato de que este modelo é o único, de acordo com nosso melhor conhecimento, que trata o escalonador WFQ alimentado com tráfego auto-similar. Os modelos presentes na literatura, e citados na introdução dessa tese, ou tratam o escalonador WFQ sem levar em consideração o fato do tráfego ser auto-similar como é o caso dos trabalhos de Horváth e Telek (HORVÁTH; TELEK, 2003), Hedge e Avrachenkov (HEDGE; AVRACHENKOV, 2004) e Soo e Chung (SOO; CHUNG, 2002); ou tratam o caso de uma fila única sem distinção (priorização) entre fluxos sob tráfego auto-similar, como é o caso dos trabalhos de Garetto (GARETTO, 2004).

Com os resultados obtidos desse trabalho foram gerados alguns artigos citados abaixo:

- *Priorization of Users in MPLS network using the WFQ scheduler*, aceito no *13th International Conference on Telecommunications*; Ilha da Madeira, Portugal, ICT 2006;
- *MPLS Network Using the WFQ Scheduler and Queuing Theory*, aceito na *Convención Internacional de Ingeniería Mecánica, Eléctrica e Industrial*, CIMEI 2006;
- *Utilização do Escalonador WFQ em um nó DiffServ para Priorizar Determinados Grupos de Usuários*, aceito no *XII Congresso Internacional de Telecomunicações*, SENACITEL 2006.
- *Analysis of the WFQ Scheduler in a DiffServ Node loaded by TCP Traffic*, aceito na *XIII Convención de Ingeniería Eléctrica*, CIE 2007.
- *A Simulation Study of the WFQ Scheduler in a DiffServ Node under LRD Traffic*, aceito em *The Seventh International Conference on Networking*, ICN 2008.
- *Análise de Filas Prioritárias em uma Rede MPLS com Chegadas em Grupo*, aceito no *XIII Congresso Internacional de Telecomunicações*, SENACITEL 2008.

Como desenvolvimento futuro para este trabalho é possível considerar mais de duas filas prioritárias, como por exemplo: 10%-30%-60%, estender a análise para uma rede com vários nós WFQ, trocar a estratégia de gerenciamento de fila (por exemplo, *Drop Tail* por RED), utilizar fontes de tráfego diferentes do TCP e trocar a distribuição de arquivos transmitidos para obter outros valores do parâmetro de Hurst.

REFERÊNCIAS

- ABRY, Patrice; VEITCH, Darryl. **Wavelet Analysis of Long-Range Dependent Traffic**. IEEE Transactions on Information Theory, v. 44, n. 1, p. 2-15, 1998.
- ALQAHTANI, S. A. **Resource Allocation Schemes for 3G Mobile Service Providers with Heterogeneous Traffic**. In: IEEE INTERNATIONAL CONFERENCE ON NETWORKS, 15., 2007. Proceedings..., p. 107-111, nov. 2007.
- ANDREWS, Matthew; ZHANG, Lisa. **Satisfying Arbitrary Delay Requirements in Multihop Networks**. In: IEEE CONFERENCE ON COMPUTER COMMUNICATIONS, 27., 2008, Phoenix, EUA.
- BALLIACHE, Leonardo. **Differentiated Service on Linux HOWTO**. Ago. 2003. Disponível em: <<http://www.opalsoft.net/qos/DS.htm>>. Acesso em: 30 set. 2009.
- BARMAN, Dhiman; SMARAGDAKIS, Georgios; MATTA, Ibrahim. **The Effect of Router Buffer Size on HighSpeed TCP Performance**. In: IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE, 2004. Proceedings..., v. 3, p. 1617-1621, nov. 2004.
- BASHER, Naimul; MAHANTI, Aniket; MAHANTI, Anirban; WILLIAMSON, Carey; ARLITT, Martin. **A comparative analysis of web and peer-to-peer traffic**. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 17., 2008. Proceedings..., p. 287-296, 2008. Nova York, EUA: ACM, 2008.
- BEHESHTI, Neda; GANJALI, Yashar; GHOBADI, Monia; McKEOWN, Nick William; SALMON, Geoffrey. **Experimental Study of Router Buffer Sizing**. Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, p. 197-210, 2008. Nova York: ACM, 2008.
- BERAN, Jan; SHERMAN, Robert; TAQQU, Murad S.; WILLINGER, Walter. **Long range dependence in VBR video traffic**. IEEE Transactions on Communications, v. 43, p. 1566-1579, 1995. Nova York, EUA: IEEE Press, 1995.
- BIANCHI, Gabriel R. **Um Modelo Multifractal Aplicado à Predição de Tráfego de Redes - Proposta e Avaliação**. 2005. Dissertação (Mestrado) - Universidade Estadual de Campinas, Campinas, 2005.
- BLAKE, S.; BLACK, D.; CARLSON, M.; DAVIES, E.; WANG, Z. **RFC 2475 - An Architecture for Differentiated Services**. Internet Engineering Task Force, 1998. Disponível em: <<http://www.ietf.org/rfc/rfc2475.txt>>. Acesso em: 30 set. 2009.
- BRADEN, Robert; CLARK, David D.; SHENKER, Scott. **RFC1633 - Integrated Services in the Internet Architecture: an Overview**. Internet Engineering Task Force, 1994. Disponível em: <<http://www.ietf.org/rfc/rfc1633.txt>>. Acesso em: 30 set. 2009.
- BUSH, Randy; MEYER, David. **RFC 3439 - Some Internet Architectural Guidelines and Philosophy**. Internet Engineering Task Force, 2002. Disponível em: <<http://www.ietf.org/rfc/rfc3439.txt>>. Acesso em: 30 set. 2009.

CARDWELL, Neal; SAVAGE, Stefan; ANDERSON, Thomas. **Modeling TCP Latency**. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS AND NETWORKS, 9., Tel Aviv, Israel, Proceedings..., 2000.

CASONE, Luca. **Improving many-to-one traffic flowing in multi-hop 802.15.4 WSNs using a MAC-level fair scheduling**. In: IEEE INTERNATIONAL CONFERENCE ON MOBILE AD HOC AND SENSOR SYSTEMS, 2007. Proceedings..., p. 1-7, out. 2007.

CHANG, Kirk; KIM, Gitae; WONG, Larry; SAMTANI, Sunil; STAIKOS, Aristides; MITESH, Patel; BOWCOCK, Jefferey; SYMBOL, Jonathan. **Network Layer Congestion Control to Ensure Quality of Service (QoS) in Secure Battlefield Mobile Ad Hoc Networks**. In: IEEE MILITARY COMMUNICATIONS CONFERENCE, 2007. Proceedings..., p. 1-7, out. 2007.

CISCO, 2009. <http://www.cisco.com.br/>. Acesso em: 25 out. 2007.

CLARK, David D.; BRADEN, Robert; CROWCROFT, Jon; DAVIE, Bruce; DEERING, Steve. **RFC 2309 - Recommendations on Queue Management and Congestion Avoidance in the Internet**. Internet Engineering Task Force, 1998. Disponível em: <<http://www.ietf.org/rfc/rfc2309.txt>>. Acesso em: 30 set. 2009.

CROVELLA, Mark E.; BESTAVROS, Azer. **Self-similarity in World Wide Web traffic: evidence and possible causes**. IEEE/ACM Transactions on Networking, v. 5, n. 6, p. 835-846, dez. 1997. Piscataway, EUA: IEEE Press, 1997.

DEKERIS, Brunonas; ADOMKUS, Tomas; BUDNIKAS, Aurelijus. **Analysis of QoS assurance using Weighted Fair Queueing (WFQ) scheduling discipline with low latency queue (LLQ)**. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY INTERFACES, 28., 2006.

DEKERIS, Brunonas; NARBUTAITE, Lina; ADOMKUS, Tomas. **A new adaptive fair queueing (AFQ) scheduler for support SLA**. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY INTERFACES, 29., 2007.

DEMERS, Alan; KESHAV, Srinivasan; SHENKER, Scott. **Analysis and Simulation of a Fair Queueing Algorithm**. In: INTERNETWORKING RESEARCH AND EXPERIENCE, 1990. Journal...

DU, Hongfei; FAN, Linghang, EVANS, Barry G. **Combined Delay and Rate Differentiation Packet Scheduling for Multimedia Content Delivery in Satellite Broadcast/Multicast Systems**. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 2007. Proceedings..., p. 4444-4449, jun. 2007.

ERMAN, Jeffrey; MAHANTI, Anirban; ARLITT, Martin; WILLIAMSON, Carey. **Identifying and discriminating between web and peer-to-peer traffic in the network core**. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 16., 2007. Proceedings..., p. 883-892, 2007. Nova York, EUA: ACM, 2007.

FELDMANN, Anja; GILBERT, Anna C.; WILLINGER, Walter. **Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic**. Proceedings of the ACM/SIGCOMM, p. 25-38, 1998.

FELDMANN, Anja; GILBERT, Anna C.; WILLINGER, Walter; KURTZ, Thomas G. **The Changing Nature of Network Traffic: Scaling Phenomena**. ACM SIGCOMM Computer Communication Review, v. 28, n. 2, p.5-29, abr. 1998. Nova York, EUA: ACM, 1998.

FGEE, El-Bahlul; PHILLIPS, William J.; ROBERTSON, William. **Comparison between a Proposed QoS Mathematical Model and other IP QoS Models**. In: COMMUNICATION NETWORKS AND SERVICES RESEARCH CONFERENCE, 2008. p. 595-599.

FGEE, El-Bahlul; PHILLIPS, William J.; ROBERTSON, William; ELHOUNIE, ; SMEDA, A. **A Scalable Mathematical QoS Model for IP Networks**. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES, 3., 2008.

FREITAS, Mario Sérgio T. **Auto-similaridade e Geometria Fractal**. Disponível em: <<http://pessoal.utfpr.edu.br/msergio/cap3/index.html>>. Acesso em: 30 set. 2009.

GARETTO, Michele. **Analytical Models of IP Networks with TCP Traffic**. 2004. 121 f. Tese (Doutorado em Engenharia Eletrônica e de Telecomunicações) - Politecnico di Torino, Itália, 2004.

GARETTO, Michele; TOWSLEY, Donald F. **Modeling, simulation and measurements of queuing delay under long-tail Internet traffic**. In: ACM SIGMETRICS INTERNATIONAL CONFERENCE ON MEASUREMENT AND MODELING OF COMPUTER SYSTEMS, 2003, San Diego, EUA. Proceedings..., p. 47-57, 2003. Nova York, EUA: ACM, 2003.

GARRETT, Mark W.; WILLINGER, Walter. **Analysis, modeling and generation of self-similar VBR video traffic**. ACM SIGCOMM Computer Communication Review, v. 24, n. 4, p. 269-280, out. 1994. Nova York, EUA: ACM, 1994.

GIROUX, Natalie; GANTI, Sudhakar. **Quality of Service in ATM Networks: State-of-the-Art Traffic Management**. Upper Saddle River, EUA: Prentice Hall, 1998.

GOLDIE, Charles M.; KLÜPPELBERG, Claudia. **Subexponential distributions**. A practical guide to heavy tails: statistical techniques and applications, p. 435-459, 1998. Cambridge, EUA: Birkhauser Boston, 1998.

GROSS, Donald; HARRIS, Carl M. **Fundamentals of Queueing Theory**. Nova York: John Wiley & Sons, 1998.

HEDGE, Nidhi; AVRACHENKOV, Konstantin E. **Service Differentiation and Guarantees for TCP-based Elastic Traffic**. In: INTERNATIONAL WORKSHOPS ON QUALITY OF FUTURE INTERNET SERVICES, Zurique, Suíça, 2002.

HEYMAND, D.; LAKSHMAN, T. V. **What are the Implications of Long Range Dependence for VBR Video Traffic Engineering?** IEEE/ACM Transactions on Networking, v. 4, p. 301-317, jun. 1996. Nova York, EUA: ACM, 1996.

HIRCHOREN, Gustavo Abraham. **Predição e estimação de parâmetros de processos auto-similares para redes de faixa larga**. 1999. Tese (Doutorado em Engenharia Elétrica) - Universidade Estadual de Campinas, Campinas, 1999.

HORVÁTH, Gábor; TELEK, Miklós. **An Approximate Analysis of Two Class WFQ Systems**. In: INTERNATIONAL WORKSHOP ON PERFORMABILITY MODELING OF COMPUTER AND COMMUNICATION SYSTEMS, 6., 2003.

HURST, Harold E.; BLACK, R. P.; SIMAIKA, Y. M. **Long-term storage: an experimental study**. Londres, Reino Unido: Constable, 1965.

KLEINROCK, Leonard. **Queuing Systems, vol. I: Theory**. Nova York: John Wiley & Sons, 1975.

KLEINROCK, Leonard. **Queuing Systems, vol. II: Computer Applications**. Nova York: John Wiley & Sons, 1976.

LELAND, Will E.; TAQQU, Murad S.; WILLINGER, Walter; WILSON, Daniel V. **On the self-similar nature of Ethernet traffic (extended version)**. IEEE/ACM Transactions on Networking, v. 2, n. 1, p. 1-15, fev. 1994 Piscataway, EUA: IEEE Press, 1994.

LIN, Jin C.; CHOU, Chun L.; LIU, Cheng H. **Performance Evaluation for Scheduling Algorithms in WiMAX Network**. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 22., 2008. Proceedings..., p. 68-74, 2008.

LI, T.; REKHTER, Yakov. **RFC 2430 - A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)**. Juniper Networks and Cisco Systems, 1998. Disponível em: <<http://www.ietf.org/rfc/rfc2430.txt>>. Acesso em: 30 set. 2009.

LU, Jordan. ROBOTHAM, Robert. **On the Implementation of Weighted Fair Queuing in High Speed Networks**. In: CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING, Sheraton Fallsview, Ontario, Canadá, 2004.

LUO, Song; MARIN, Gerald A. **Realistic Internet traffic simulation through mixture modeling and a case study**. Proceedings of the Winter Simulation Conference, dez. 2005.

LU, Tsung C.; HUANG, Ching Y.; LIN, Maw C. **Priority-based EDF scheduling algorithm for real-time service**. In: INTERNATIONAL CONFERENCE ON INFORMATION, COMMUNICATIONS & SIGNAL PROCESSING, 6., 2007. Proceedings..., p. 1-5, dez. 2007.

MAGAÑA, Eduardo; MORATÓ, Daniel; VARAIYA, Pravin. **Tuning the weights in WFQ schedulers for the maximization of carried best effort traffic**. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS, 10., 2003.

MANDELBROT, Benoît B. **The Fractal Geometry of Nature**. Nova York: W. H. Freeman and Company, 1982.

MANDELBROT, Benoît B.; VAN NESS, John W. **Fractional Brownian Motions, Fractional Noises and Applications**. SIAM Review, v. 10, n. 4, p. 422-437, 1968.

MARQUES, Leandro Bento S. **Estudo de Desempenho do Sistema 3G 1xEV-DO Através de Modelos Reais de Tráfego**. 2005. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Estadual de Campinas, Campinas, 2005.

MASI, Denise M. B.; FISCHER, Martin J.; GARBIN, David A. **Modeling the performance of low latency queuing for emergency telecommunications**. In: WINTER SIMULATION CONFERENCE, 39., 2007, Washington, D.C., EUA. In: Proceedings..., p. 2266-2275. Piscataway, EUA: IEEE Press, 2007.

MEISS, Mark R.; MENCZER, Filippo; FORTUNATO, Santo. **Ranking Web Sites with Real User Traffic**. In: INTERNATIONAL CONFERENCE ON WEB SEARCH AND DATA MINING, Palo Alto, EUA, 2008. Proceedings..., p. 65-76, 2008. Nova York, EUA: ACM, 2008.

MELO, Edison Tadeu L. **Qualidade de Serviço em Rede IP com DiffServ: Avaliação Através de Medições**. 2001. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Florianópolis, 2001.

MORIN, Patrick R. **The Impact of Self-Similarity on Network Performance Analysis**. 1995. Tese (Doutorado em Ciências da Computação) - Carleton University, Ottawa, Canadá, 1995.

MOTA, Oscar T. J. D. D. L.; GOMES, Antônio Tadeu A.; COLCHER, Sérgio; SOARES, Luiz Fernando G. **Uma Arquitetura Adaptável para Provisão de QoS na Internet**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADOR, 19., 2001, Florianópolis. NABAS, Kleber Kendy H. Análise de Filas Prioritárias em uma Rede MPLS com Chegada em Grupo. In: CONGRESO INTERNACIONAL DE TELECOMUNICACIONES, 13., 2008, Chile.

NABAS, Kleber Kendy H.; FORONDA, Augusto; GODOY JUNIOR, Walter. **An Architectural Framework for Support Quality of Service in MPLS Networks**. WSEAS Transactions on Communications, v. 3, n. 2, p. 653-657, abr. 2004.

NABAS, Kleber Kendy H.; FORONDA, Augusto; GODOY JUNIOR, Walter. **Arquitetura para Provisão de QoS em uma Rede MPLS**. In: WORKSHOP DE CIÊNCIAS DA COMPUTAÇÃO E SISTEMAS DA INFORMAÇÃO DA REGIÃO SUL, 1., 2004, Palhoça. Anais do WorkComp Sul, 2004.

NABAS, Kleber Kendy H.; FORONDA, Augusto; GODOY JUNIOR, Walter; WILLE, Emilio Carlos G. **Priorization of Users in MPLS Networks Using the WFQ Scheduler and Queuing Theory**. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS, 13., 2006, Ilha da Madeira, Portugal.

NABAS, Kleber Kendy H.; GODOY JUNIOR, Walter; WILLE, Emilio Carlos G. **A Simulation Study of the WFQ Scheduler in a DiffServ Node under LRD Traffic**. In: INTERNATIONAL CONFERENCE ON NETWORKING, 7., 2008, Cancun, México.

NABAS, Kleber Kendy H.; GODOY JUNIOR, Walter; WILLE, Emilio Carlos G.; TEIXEIRA, Lincoln Herbert. **Analysis of the WFQ Scheduler in a DiffServ Node Loaded by TCP Traffic**. In: CONVENCION INTERNACIONAL DE INGENIERIA MECÁNICA, ELÉCTRICA E INDUSTRIAL, 4., 2007, Santa Clara, Cuba. Havana: Obras, 2007.

NABAS, Kleber Kendy H.; GODOY JUNIOR, Walter; WILLE, Emilio Carlos G.; TEIXEIRA, Lincoln Herbert; REGUERA, Evelio M. G. F. V. A.. **MPLS Network Using the WFQ Scheduler and Queuing Theory**. In: CONVENCION INTERNACIONAL DE

INGENERÍA MECÁNICA, ELÉCTRICA E INDUSTRIAL, 3., 2006, Santa Clara, Cuba. Havana: Obras, 2006.

NABAS, Kleber Kendy H.; GODOY JUNIOR, Walter; WILLE, Emilio Carlos G.; TEIXEIRA, Lincoln Herbert. **Utilização do Escalonador WFQ em um nó DiffServ para Priorizar Determinados Grupos de Usuários**. In: CONGRESO INTERNACIONAL DE TELECOMUNICACIONES (SENACTIEL), 12., 2006, Chile.

NETWORK SIMULATOR (NS, ver. 2), 2007. <http://www.isi.edu/nsnam/ns/>

NORROS, Ilkka. **A storage model with self-similar input**. Queueing Systems, v. 16, n. 3-4, p. 387-396, set. 1994. Springer Netherlands, 1994.

NORROS, Ilkka. **On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks**. IEEE Journal of Selected Areas in Communications, v. 13, n. 6, p. 953-962, 1995.

NUCCI, Antonio; BHATTACHARYYA, Supratik; TAFT, Nina; DIOT, Christophe. **IGP Link Weight Assignment for Operational Tier-1 Backbones**. IEEE/ACM Transactions on Networking, v. 15, n. 4, p. 789-802, ago. 2007.

PAREKH, Abhay K.; GALLAGER, Robert G. **A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single-Node Case**. IEEE/ACM Transactions on Networking, v. 1, n. 3, p. 344-357, jun. 1993.

PARK, Kihong; KIM, G.; CROVELLA, Mark E. **On the effect of traffic self-similarity on network performance**. In: CONFERENCE ON PERFORMANCE AND CONTROL OF NETWORK SYSTEMS, 1997.

PARK, Kihong; KIM, G.; CROVELLA, Mark E. **On the Relation between File Sizes, Transport Protocols and Self-Similar Network Traffic**. In: INTERNATIONAL CONFERENCE ON NETWORK PROTOCOLS, 1996, Columbus, EUA.

PARK, Kihong; WILLINGER, Walter. **Self-similar network traffic and performance evaluation**. Nova York, EUA: John Wiley & Sons, 2000.

PAXSON, Vern; ALMES, Guy; MAHDAVI, Jamshid; MATHIS, Matt. **RFC 2330 - Framework for IP Performance Metrics**. Internet Engineering Task Force, 1998. Disponível em: <<http://www.ietf.org/rfc/rfc2330.txt>>. Acesso em: 30 set. 2009.

PAXSON, Vern; FLOYD, Sally. **Wide-Area Traffic: The Failure of Poisson Modeling**. In: CONFERENCE ON COMMUNICATIONS ARCHITECTURES, PROTOCOLS AND APPLICATIONS, 1994, Londres, Reino Unido. Proceedings..., p. 257-268, 1994. Nova York, EUA: ACM, 1994.

PEDROSO, Carlos Marcelo. **Desenvolvimento de modelo de tráfego e método para melhoria de desempenho de servidores Web**. 2006. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2006.

PEREIRA, Flávio de Melo; FONSECA, Nelson L. S.; ARANTES, Dalton S. **A fair scheduling discipline for Ethernet passive optical networks**. In: IEEE GLOBAL

TELECOMMUNICATIONS CONFERENCE, 2007, Washington, D.C., EUA. Proceedings..., p. 2702-2707.

PRATA JÚNIOR, Oscavo G. **Uma Proposta de Modelagem Matemática para Tratamento de Tráfego com Característica Auto-Similar**. 2003. Dissertação (Mestrado) - Fundação Instituto Nacional de Telecomunicações, Santa Rita de Sapucaí, 2003.

REZENDE, José F. **Avaliação do Serviço Assegurado para a Diferenciação de Serviços na Internet**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 17., 1999, Salvador. Anais do SBRC, 1999. p. 339-353.

RIEDI, Rudolf H.; VÉHEL, Jacques Lévy. **TCP traffic is multifractal: a numerical study**. INRIA Research Report, 1997.

ROSSI, Dario; MELLIA, Marco; MUSCARIELLO, Luca. **Analyzing TCP Flow Arrivals**. COST279, Technical Document No. TD(04)013, 2004.

SAYENKO, Alexander; HÄMÄLÄINEN, Timo; JOUTSENSALO, Jyrki. SILTANEN, Jarmo. **An Adaptive Approach to WFQ with the Revenue Criterion**. Proceedings of the Eighth IEEE International Symposium on Computers and Communications, p. 181, 2003. Washington, DC, EUA: IEEE Computer Society, 2003.

SOO, Hooi Miin; CHUNG, Jong-Moon. **Analysis of nonpreemptive priority queueing of MPLS networks with bulk arrivals**. In: MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, 45., 2002, Oklahoma, EUA.

TCP STATISTIC AND ANALYSIS TOOL (TSTAT, ver. 2.0), 2008. <http://tstat.tlc.polito.it/index.shtml>

WANG, Y.; FAN, D. H.; TAFAZOLLI, Rahim. **Solution to weight-adaptive fair queuing**. Electronics Letters, v. 44, n. 5, p. 385-387, fev. 2008.

WESTERINEN, Andrea; STRASSNER, John. **Information Model for Describing Network Device QoS Mechanisms**. Internet Draft. Internet Engineering Task Force, 2001. Disponível em: <<http://tools.ietf.org/pdf/draft-ietf-policy-qos-device-info-model-00.pdf>>. Acesso em: 30 set. 2009.

WILLINGER, Walter; TAQQU, Murad S.; SHERMAN, Robert; WILSON, Daniel V. **Self-Similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level**. ACM SIGCOMM Computer Communication Review, v. 25, n. 4, out. 1995. Nova York, EUA: ACM, 1995.

ANEXO A — PROGRAMA PARA CÁLCULO DA AUTO-CORRELAÇÃO

```

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdarg.h>

char *progname;
double *data = NULL;
size_t ndata = 0;
unsigned pdata = 0;

void
fail(char *msg, ...)
{
    va_list v;
    va_start(v, msg);
    fprintf(stderr, "%s: ", progname);
    vfprintf(stderr, msg, v);
    fprintf(stderr, "\n");
    va_end(v);
    exit(EXIT_FAILURE);
}

void
alloc(size_t size)
{
    data = realloc(data, (ndata = size) * sizeof(double));
    if (!data) fail("malloc(), tamanho = %zu\n", ndata * sizeof(double));
}

void
put(double v)
{
    if (pdata >= ndata)
        alloc(ndata+500);
    data[pdata++] = v;
}

void
trim()
{
    alloc(pdata);
}

void
freed()
{
    if (data) free(data);
}

double
sqr(double x)
{
    return x*x;
}

double
autocorr(double *x, double ave, double var, size_t n, unsigned k)
{
    unsigned i;
    double a = 0.;
    for (i=0; i<n-k; i++)
        a += (x[i]-ave)*(x[i+k]-ave);
    return a/var;
}

```

```

}

int
main(int argc, char **argv)
{
    FILE *fp;
    double v, ave, var;
    unsigned i;
    progname = argv[0];
    if (argc < 2) {
        fprintf(stderr, "uso: %s FILENAME [BUFSIZE]\n", argv[0]);
        exit(EXIT_SUCCESS);
    }
    atexit(freed);
    if (argc > 3) {
        int s = atoi(argv[2]);
        if (s <= 0)
            fail("tamanho do buffer inválido.");
        else
            alloc((size_t)s);
    }
    fp = fopen(argv[1], "r");
    if (!fp) fail("erro ao abrir arquivo %s", argv[1]);
    while (!feof(fp)) {
        if (!fscanf(fp, "%lf ", &v)) break;
        put(v);
    }
    fclose(fp);
    trim();
    if (!ndata) {
        printf("nenhum valor.");
        exit(EXIT_SUCCESS);
    }
    ave = var = 0.;
    printf("%zu valores\n", ndata);
    for (i=0; i<ndata; i++) {
        ave += data[i];
    }
    ave /= ndata;
    for (i=0; i<ndata; i++) {
        var += sqr(data[i]-ave);
    }
    for (i=0; i<ndata; i++) {
        printf("%f\n", autocorr(data, ave, var, ndata, i));
    }
    return EXIT_SUCCESS;
}

```


ANEXO B — PROGRAMA PARA CÁLCULO DO PARÂMETRO DE HURST PELO MÉTODO R/S

```

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdarg.h>

char *programe;
FILE *fp;
double *data;
size_t datasize;
unsigned datapos;
double n, avg, var, stddev, r;

void
fail(char *msg, ...)
{
    va_list v;
    va_start(v, msg);
    fprintf(stderr, "%s: ", programe);
    vfprintf(stderr, msg, v);
    fprintf(stderr, "\n");
    va_end(v);
    exit(EXIT_FAILURE);
}

double
sqr(double x)
{
    return x*x;
}

void
calc_avg()
{
    unsigned i;
    avg = 0.0;
    for (i=0; i<datasize; i++)
        avg += data[i] / n;
}

void
calc_rs()
{
    unsigned i;
    double x, var, l, prevl, maxl, minl;
    var = l = prevl = maxl = minl = 0.0;
    for (i=0; i<datasize; i++) {
        x = data[i];
        var += (sqr(x-avg)) / n;
        if (!i) {
            l = x-avg;
            maxl = minl = l;
        } else {
            l = prevl+x-avg;
            if (l > maxl)
                maxl = l;
            if (l < minl)
                minl = l;
        }
        prevl = l;
    }
    stddev = sqrt(var);
}

```

```

        r = maxl - minl;
    }

    void
    allocdata(size_t size)
    {
        datasize = size;
        data = realloc(data, datasize * sizeof(double));
        if (!data)
            fail("malloc(), tamanho = %u\n", datasize);
    }

    void
    put(double v)
    {
        if (datapos >= datasize)
            allocdata(datasize+500);
        data[datapos++] = v;
    }

    void
    trim()
    {
        unsigned ndatasize = datapos;
        if (datasize == ndatasize)
            return;
        else
            allocdata(ndatasize);
    }

    void
    freeall()
    {
        if (data) free(data);
        if (fp) fclose(fp);
    }

    int
    main(int argc, char **argv)
    {
        data = NULL;
        datasize = datapos = 0;
        progname = argv[0];
        if (argc < 2) {
            fprintf(stderr, "uso: %s FILENAME [BUFSIZE]\n", argv[0]);
            exit(EXIT_SUCCESS);
        }
        atexit(freeall);
        if (argc > 3) {
            int s = atoi(argv[2]);
            if (s <= 0)
                fail("tamanho do buffer inválido.");
            else
                allocdata((unsigned)s);
        }
        fp = fopen(argv[1], "r");
        if (!fp) fail("erro ao abrir arquivo %s", argv[1]);
        double v;
        while (!feof(fp)) {
            if (!fscanf(fp, "%lf ", &v))
                break;
            put(v);
        }
        fclose(fp);
        fp = NULL;
        trim();
        if (!(n = datasize))
            fail("nenhum valor.");
    }

```

```
fprintf(stderr, "%u valores\n", datasize);
/* média. */
calc_avg();
printf("M =\t\t%f\n", avg);
/* variância e desvio padrão. */
calc_rs();
printf("S =\t\t%f\n", stddev);
printf("R =\t\t%f\n", r);
printf("H =\t\t%f\n", log10(r/stddev)/log10(n/2));
return EXIT_SUCCESS;
}
```

ANEXO C — PSEUDOCÓDIGO PARA RESOLUÇÃO DA CADEIA DE MARKOV DA FILA M/M/1/B

dados de entrada:

α_0, α_1 : pesos do escalonador das classes
 ρ : fator de utilização
 d : atraso do enlace
 L : tamanho (em pacotes) de ambas as filas (e número de estados no sistema)
 P : tamanho de pacote (em bits)
 C : capacidade do enlace (em megabits por segundo)

A = matriz $L+1$ por $L+1$

E = matriz $(L+1)^2$ por $(L+1)^2$

P = C = vetor com $(L+1)^2$ posições

mm = zero

para cada m em 0..L

 para cada n em 0..L

 se m = n = 0

 A[0,0] = 2ρ

 A[0,1] = -1

 A[1,0] = -1

 caso contrário, se m = 0 e n = L

 A[0,L] = $1+\rho$;

 A[0,L-1] = -1;

 A[1,L] = $-\alpha_1$;

 ...

 /* tratar cada um dos casos explicados anteriormente para construir a matriz de coeficientes */

 ...

 caso contrário /* caso geral */

 A[m,n] = $1+2\rho$

 A[m-1,n] = $-\rho$

 A[m,n-1] = $-\rho$

 A[m,n+1] = $-\alpha_0$

 A[m+1,n] = $-\alpha_1$

 seja T um vetor (temporário) com $(L+1)^2$ posições

 para cada nn em 0..L

 para cada i em 0..L

 T[L*(nn-1)+i] = A[nn,i] /* copiar uma linha */

 /* copiar o vetor T para a linha mm da matriz E */

 para cada i em 0..L

 E[mm,i] = T[i]

 incrementar mm

/* preencher uma linha com 1s, já que $\sum P(m,n) = 1$ */

seja l algum número em 0..L

para cada i em 0..L

 E[l,i] = 1

 C[l] = 1

/* resolver o sistema das matrizes E (coeficientes) e C (termos independentes) por eliminação Gaussiana e armazenar o resultado no vetor P (probabilidades) */

P = C · (matriz inversa de E^T)

$N_0 = N_1 = 0$

para cada m em 0..L

$N_0 = N_0 + m \cdot \Sigma P[m, 0..L]$

para cada n em 0..L

$N_1 = N_1 + n \cdot \Sigma P[0..L, n]$

/* ploss é igual à soma das probabilidades de todos os estados de buffer cheio */

ploss = $\Sigma P[L \cdot (0..L)]$

rtt₀ = $2 \cdot d + 8 \cdot P \cdot N_0 / (\alpha_0 \cdot C)$

rtt₁ = $2 \cdot d + 8 \cdot P \cdot N_1 / (\alpha_1 \cdot C)$

saída = ploss, rtt₀ e rtt₁

ANEXO D — MODELO ANALÍTICO PARA O TCP

O TCP fornece um serviço de dados confiável orientado à conexão em redes de comutação de pacotes. Isto é alcançado confirmando os pacotes recebidos (ACKs), números de seqüência e temporizadores. O TCP emprega mecanismos de controle de congestionamento baseado no ajuste do tamanho da sua janela de congestionamento (*cwnd*). *cwnd* é a máxima quantidade de dados que o emissor pode transmitir antes de receber um ACK. O receptor também informa um limite (*rwnd*) sobre a quantidade de dados suportada. A transmissão de dados é sempre determinada pelo tamanho da janela $W_m = \min(cwnd, rwnd)$.

Diversos pesquisadores propuseram modelos analíticos que caracterizam o desempenho do TCP em função dos valores de RTT e a probabilidade de perda de pacotes (CASSETTI; MEO, 2000) (MATHIS et al., 1997) (KUMAR, 1998) (PADHYE et al., 1998) (MISRA; OTT, 1999) (PADHYE; FIROIU; TOWSLEY, 1999). Este anexo apresenta de forma sucinta abordagem proposta por Carwell, Savage e Anderson, que foi utilizada nessa tese.

Em (CARDWELL; SAVAGE; ANDERSON, 2000) foi desenvolvido um modelo analítico para o cálculo do tempo esperado de transferências (latência) de arquivos de comprimento arbitrário via TCP. Este modelo é uma extensão do trabalho proposto em (PADHYE et al., 2000) para incluir as fases de estabelecimento da conexão e *slow start*. A latência esperada $E[L_{CE}]$ do *three-way handshake* para estabelecimento da conexão (CE) é:

$$E[L_{CE}] = RTT + RTO_0 \cdot \left(\frac{1 - p_r}{1 - 2p_r} + \frac{1 - p_f}{1 - 2p_f} - 2 \right)$$

onde: RTO_0 é o intervalo de retransmissão de *timeout*, p_f é a probabilidade de perda de segmentos no caminho direto do servidor para o cliente e p_r é a probabilidade de perda no caminho reverso. A latência média $E[L]$ para completar uma transferência de N segmentos é calculada como a soma de quatro componentes:

$$E[L] = E[L_{SS}] + E[L_{loss}] + E[L_{ca}] + E[L_{delack}]$$

onde L_{SS} é a latência do *slow start*, L_{loss} é o custo esperado das perdas de segmento que ocorrem no final do *slow start*, L_{ca} é o tempo esperado para transferir os segmentos restantes $(N - E[d_{SS}])$ e L_{delack} é o custo do primeiro ACK atrasado se $cwnd_0$ for igual a 1.

Ambos $E[L_{SS}]$ e $E[L_{loss}]$ são funções de RTT , b , W_m , p , N , $E[t_{TO}]$ e o tamanho da janela inicial $cwnd_0$. $E[L_{ca}] = (N - E[d_{SS}]) / T$, onde T é a vazão do TCP. $E[d_{SS}]$ é o número esperado de segmentos enviados antes que uma perda seja encontrada durante do *slow start*. Finalmente $E[L_{delay}]$ é igual a 150ms para plataformas Windows ou 100ms para BSD UNIX. Dada a complexidade das equações para $E[L_{SS}]$, $E[L_{loss}]$ e $E[L_{ca}]$, as mesmas não são detalhadas neste anexo, podendo ser encontradas em (CARDWELL; SAVAGE; ANDERSON, 2000).