

Gerenciamento Ativo de Filas para o protocolo “High Speed Transmission Control Protocol” em Redes com Produto Banda-Atraso Elevado

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Juliana de Santi e aprovada pela Banca Examinadora.

Campinas, 26 de Fevereiro de 2008.



Prof. Dr. Nelson Luis Saldanha da Fonseca
IC - UNICAMP (Orientador)

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

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**
Bibliotecária: Maria Júlia Milani Rodrigues CRB8a / 2116

Santi, Juliana de
Sa59g Gerenciamento ativo de filas para o protocolo "High Speed Transmission Control Protocol" em redes com produto banda-atraso elevado / Juliana de Santi -- Campinas, [S.P. :s.n.], 2008.

Orientador : Nelson Luis Saldanha da Fonseca
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Transmission control protocol (TCP). 2. Highspeed TCP (HSTCP). 3. Controladores ótimos. 4. Teoria de controle. 5. Controle de congestionamento. 6. Gerenciamento ativo de filas (AQM). 7. Redes de alta velocidade. I. Fonseca, Nelson Luis Saldanha da. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Active queue management to the High Speed Transmission Control Protocol in high bandwidth-delay networks.

Palavras-chave em inglês (Keywords): 1. Transmission control protocol (TCP). 2. Highspeed TCP (HSTCP). 3. Optimal controllers. 4. Control theory. 5. Congestion control. 6. Active queue management (AQM). 7. High speed networks.

Área de concentração: Redes de Computadores

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Prof. Dr. Nelson Luis Saldanha da Fonseca (IC-UNICAMP)
Prof. Dr. José Claudio Geromel (FEEC-UNICAMP)
Prof. Dr. Edmundo Roberto Mauro Madeira (IC-UNICAMP)
Prof.ª Dra. Islene Calciolari Garcia (IC-UNICAMP)

Data da defesa: 26/02/2008

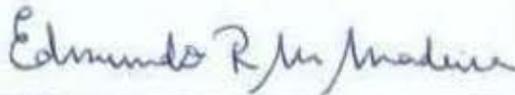
Programa de pós-graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 26 de fevereiro de 2008, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. José Claudio Geromel
FECC/UNICAMP



Prof. Dr. Edmundo Roberto Mauro Madeira
IC/UNICAMP



Prof. Dr. Nelson Luis Saldanha da Fonseca
IC/UNICAMP

Gerenciamento Ativo de Filas para o protocolo “High Speed Transmission Control Protocol” em Redes com Produto Banda-Atraso Elevado

Juliana de Santi¹

Fevereiro de 2008

Banca Examinadora:

- Prof. Dr. Nelson Luis Saldanha da Fonseca
IC - UNICAMP (Orientador)
- Prof. Dr. José Claudio Geromel
FEEC - UNICAMP
- Prof. Dr. Edmundo Roberto Mauro Madeira
IC - UNICAMP
- Prof. Dra. Islene Calciolari Garcia
IC - UNICAMP

¹Suporte financeiro: CAPES

© Juliana de Santi, 2009.
Todos os direitos reservados.

Resumo

A utilização eficiente da banda passante em redes de alta velocidade e grandes atrasos, denominadas redes com produto banda-atraso elevado (PBA), tornou-se um grande desafio. Isto ocorre devido aos ajustes do protocolo *Transmission Control Protocol* (TCP). O *High Speed TCP* (HSTCP), uma variante do TCP para redes com PBA elevado, emprega ajustes mais agressivos permitindo, assim, que a utilização da banda seja escalável. As políticas de Gerenciamento Ativo de Filas ou *Active Queue Management* (AQM), monitoram o nível de ocupação das filas nos roteadores e notificam o congestionamento incipiente aos emissores TCP através do descarte/marcação de pacotes.

O sistema de controle de congestionamento apresenta natureza de retroalimentação, na qual, a taxa de transmissão dos nós fontes é ajustada em função do nível de ocupação da fila. Os controladores AQM determinam a probabilidade de descarte/marcação para maximizar a vazão e minimizar perdas, garantindo, assim, a estabilidade do tamanho da fila independentemente das variações das condições da rede.

Neste trabalho, define-se a política de gerenciamento ativo de filas HSTCP-H2 para redes com PBA elevado que utilizam o protocolo HSTCP. Para a derivação de HSTCP-H2, são utilizadas técnicas de Teoria de Controle Ótimo. A principal característica desta política é considerar o atraso do sistema o que permite melhor utilização dos recursos disponíveis. A estabilidade e os objetivos de desempenho do sistema são expressos e solucionados através de Desigualdades Matriciais Lineares, permitindo que os parâmetros do controlador possam ser calculados através da solução de um problema convexo simples.

Diferentes controladores foram derivados considerando-se diferentes objetivos de desempenho, os quais consideram as características de redes com produto banda-atraso elevado. Através de simulações, os desempenhos dos controladores derivados são avaliados e a eficácia do controlador que apresentou o melhor desempenho foi comparado com o desempenho da política de AQM RED. São considerados cenários com enlace gargalo único e com múltiplos gargalos.

Abstract

The efficient utilization of bandwidth in high speed and large delay networks, called high bandwidth-delay product networks (BDP), has become a major challenge. This is due to adjustments of the Transmission Control Protocol (TCP). The High Speed TCP (HSTCP), a TCP variant to high BDP networks, employs more aggressive adjustments, allowing scalable bandwidth utilization. The Active Queue Management (AQM) policies monitor the queue length in the routers and notify incipient congestion to TCP source by marking or dropping packets.

The congestion control system presents intrinsic feedback nature, where the transmission rates of the sources are adjusted according to the level of congestion inferred by the queue occupancy. The AQM controllers determine the dropping/marketing probability values to maximize throughput and minimize losses, giving guarantees to stabilize the queue length independent of network conditions.

In this work, it is defined HSTCP-H2, an active queue management policy to high BDP networks, which adopt the HSTCP as their transport protocol. Optimal control theory is used to conceive HSTCP-H2. The novelty of the proposed approach lies in consider the delay of the system which allows better use of available resources. Furthermore, in the proposed approach, stability and performance objectives are completely expressed as Linear Matrix Inequalities (LMIs), thus requiring the solution of a single convex problem for the computation of the controller parameters.

Different controllers are derived considering different design goals, which take into account the characteristics of the high bandwidth-delay product networks. The performance produced by different optimal controllers was investigated. The efficacy of the controller with the best performance was then compared to the performance of RED policy. The simulation experiments were carried out using topologies with single and multiple bottleneck.

Dedicatória

"Se vi mais longe, foi porque me apoiei sobre os ombros de gigantes."
(Isaac Newton)

*Aqueles que acreditam em mim todos os dias:
meus pais Jacir Pedro de Santi e Jovelina Gaspar de Santi*

Agradecimentos

Agradeço a Deus pela vida, por todas as oportunidades e realizações.

A meus pais Jacir Pedro de Santi e Jovelina Gaspar de Santi e meu irmão Jeferson C. de Santi pelo apoio incondicional.

A Rodrigo Minetto pelo carinho e compreensão.

A meu orientador Prof. Nelson Fonseca pelos direcionamentos que tornaram possível a realização deste trabalho.

A Michele Lima pelo auxílio na fase inicial deste trabalho.

A Leonardo Rangel pelas valiosas discussões que contribuíram de forma significativa para este trabalho.

A André Drummond e Leyza Dorini pela ajuda “de última hora”.

A Evandro C. Bracht por me receber em Campinas, pelos bolos, pipoca,..., por tudo.

A todos os amigos, pelas conversas sérias e também pelas não tão sérias assim.

Aos professores do Instituto de Computação.

Aos funcionários do IC, pela disponibilidade em auxiliar sempre que necessário.

A CAPES pelo suporte financeiro.

Sumário

Resumo	vii
Abstract	ix
Dedicatória	xi
Agradecimentos	xiii
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Principais Contribuições	4
1.3 Organização	5
1.4 Publicações	6
2 Mecanismos de Controle de Congestionamento TCP	9
2.1 O Mecanismos de Janela Deslizante do TCP	11
2.2 O Mecanismo de Controle de Congestionamento do TCP Reno	15
2.3 Problemas do TCP em Redes com Produto Banda-Atraso Elevado	16
2.3.1 Lei do Inverso da Raiz Quadrada	17
2.3.2 Comportamento Oscilatório da Janela de Transmissão	18
2.3.3 Penalização de Conexões com RTTs longos	19
2.3.4 Dependência de Perdas	19
2.4 Controle de Congestionamento TCP para Redes com PBA Elevado	20
2.4.1 <i>High Speed</i> TCP	20
2.4.2 FAST TCP	24
2.4.3 Binary Increase Congestion Control	28
2.4.4 CUBIC	31
2.4.5 <i>Scalable</i> TCP	33

3	Gerenciamento Ativo de Filas	37
3.1	DropTail	38
3.2	Gerenciamento Ativo de Filas - AQM	38
3.3	Requisitos para Políticas de Gerenciamento Ativo de Filas	39
3.4	A Política de AQM Random Early Detection - RED	41
3.4.1	Funcionamento da Política RED	42
3.4.2	Avaliação de RED	45
3.5	Políticas de AQM Baseadas em Teoria de Controle	48
3.6	ECN - Explicit Congestion Notification	52
4	Introdução a Teoria de Controle e Análise de Sistemas Dinâmicos e Controle	53
4.1	Conceitos e Definições	53
4.2	Modelagem Matemática	55
4.3	Modelagem no Estado de Espaço	56
4.4	Controladores Ótimos	57
4.5	Desigualdades Matriciais Lineares	58
5	Sistema de Controle de Congestionamento HSTCP/AQM	61
5.1	Modelo dinâmico para o Controle de Congestionamento HSTCP/AQM	62
5.2	Linearização do Sistema	63
5.3	Controle de Congestionamento como um Problema de Controle	68
6	Projeto de um Controlador Ótimo AQM	71
6.1	Obtenção do Controlador para a Política de AQM	72
6.2	Síntese dos Controladores Ótimos	76
6.3	Determinação do Ponto de Equilíbrio	77
6.4	Determinação dos Parâmetros dos Controladores	78
6.5	Implementação Digital dos Controladores	79
6.6	Considerações em relação a Política HSTCP-H2 AQM	80
7	Efetividade do Controlador HSTCP-H2-AQM	83
7.1	Cenários das Simulações com um Único Enlace Gargalo	83
7.2	Comparação entre os Controladores Ótimos Propostos	85
7.2.1	Experimentos Utilizando tráfego de Longa Duração	86
7.2.2	Experimentos Utilizando tráfego de Curta Duração	88

7.3	Avaliação da Eficácia do Controlador HSTCP-H2 com Enlace Gargalo Único	91
7.3.1	Experimentos Utilizando tráfego de Longa Duração	93
7.3.2	Experimentos Utilizando tráfego de Curta Duração	99
7.4	Cenários das Simulações com Múltiplos Enlaces Gargalo	104
7.4.1	Experimentos Utilizando tráfego de Longa Duração	108
7.4.2	Experimentos Utilizando tráfego de Curta Duração	113
8	Conclusões e Trabalhos Futuros	119
8.1	Conclusões	119
8.2	Trabalhos Futuros	121
	Bibliografia	123

Lista de Tabelas

7.1	Parâmetros Utilizados na Implementação dos Controladores Ótimos	85
-----	---	----

Lista de Figuras

2.1	Janela Deslizante do Emissor	13
2.2	Janela Deslizante do Receptor	13
2.3	Comportamento dos algoritmos do TCP Reno	15
2.4	Função de Resposta do HSTCP	21
2.5	Comportamento do HSTCP	23
2.6	Componentes do mecanismo de controle de congestionamento do FAST TCP	25
2.7	Função de Crescimento de BIC	28
2.8	Função de Crescimento do CUBIC	32
2.9	Função de Resposta do STCP e TCP	34
3.1	Comportamento do RED	43
5.1	Controle de Congestionamento como um Sistema de Controle com Retro- alimentação	69
7.1	Topologia utilizada nos experimentos com enlace gargalo único	84
7.2	Tráfego FTP: coeficiente de variação do tamanho da fila em função da carga	86
7.3	Tráfego FTP: coeficiente de variação da probabilidade de des- carte/marcação em função da carga	87
7.4	Tráfego FTP: coeficiente de variação da janela de congestionamento em função da carga	88
7.5	Tráfego WEB: coeficiente de variação do tamanho da fila em função da carga	89
7.6	Tráfego WEB: coeficiente de variação da probabilidade de des- carte/marcação em função da carga	90
7.7	Tráfego WEB: coeficiente de variação da janela de congestionamento em função da carga	90
7.8	Tráfego FTP: tamanho médio da fila em função da carga	94
7.9	Tráfego FTP: vazão média por conexão ativa em função da carga	95

7.10	Tráfego FTP: tamanho médio da janela de congestionamento por conexão ativa em função da carga	95
7.11	Tráfego FTP: goodput médio por conexão ativa em função da carga	96
7.12	Tráfego FTP: número médio de RTO por conexão ativa em função da carga	97
7.13	Tráfego FTP: RTT médio por conexão ativa em função da carga	98
7.14	Tráfego FTP: tempo de transferência por conexão ativa em função da carga	98
7.15	Tráfego FTP: média do número de conexões ativa em função da carga . . .	99
7.16	Tráfego WEB: tamanho médio da fila em função da carga	100
7.17	Tráfego WEB: vazão média por conexão ativa em função da carga	101
7.18	Tráfego WEB: tamanho médio da janela de congestionamento por conexão ativa em função da carga	102
7.19	Tráfego WEB: goodput médio por conexão ativa em função da carga . . .	103
7.20	Tráfego WEB: RTT médio por conexão ativa em função da carga	103
7.21	Tráfego WEB: número médio de RTO's por conexão ativa em função da carga	104
7.22	Tráfego WEB: tempo médio de transferência por conexão ativa em função da carga	105
7.23	Tráfego WEB: número médio de conexões em função da carga	105
7.24	Topologia utilizada nos experimentos com múltiplos enlaces gargalo	106
7.25	Tráfego FTP: vazão média por conexão ativa em função da carga	108
7.26	Tráfego FTP: tamanho médio da janela de congestionamento por conexão ativa em função da carga	109
7.27	Tráfego FTP: goodput médio por conexão ativa em função da carga	110
7.28	Tráfego FTP: número médio de RTO por conexão ativa em função da carga	111
7.29	Tráfego FTP: RTT médio por conexão ativa em função da carga	112
7.30	Tráfego FTP: tempo de transferência por conexão ativa em função da carga	112
7.31	Tráfego FTP: média do número de conexões ativa em função da carga . . .	113
7.32	Tráfego WEB: vazão média por conexão ativa em função da carga	114
7.33	Tráfego WEB: tamanho médio da janela de congestionamento por conexão ativa em função da carga	115
7.34	Tráfego WEB: goodput médio por conexão ativa em função da carga . . .	116
7.35	Tráfego WEB: número médio de RTO por conexão ativa em função da carga	116
7.36	Tráfego WEB: RTT médio por conexão ativa em função da carga	117
7.37	Tráfego WEB: tempo de transferência por conexão ativa em função da carga	118

7.38 Tráfego WEB: média do número de conexões ativa em função da carga . . 118

Lista de Abreviaturas e Siglas

ACK	<i>Acknowledgment</i>
AIMD	<i>Additive Increase Multiplicative Decrease</i>
AQM	<i>Active Queue Management</i>
BIC	<i>Binary Increase Congestion Control</i>
CWND	<i>Congestion Window</i>
DCCP	<i>Datagram Congestion Control Protocol</i>
FAST TCP	<i>Fast Active Management Scalable Transmission Control Protocol</i>
FTP	<i>File Transfer Protocol</i>
HSTCP	<i>High Speed TCP</i>
IETF	<i>The Internet Engineering Task Force</i>
LMI	<i>Linear Matrix Inequality</i>
PBA	<i>Produto Banda-Atraso</i>
QoS	<i>Quality of Service</i>
RED	<i>Random Early Detection</i>
RTO	<i>Retransmission Timeout</i>
RTT	<i>Round Trip Time</i>
SSTHRESH	<i>Slow-Start threshold</i>
STCP	<i>Stable TCP</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>

Capítulo 1

Introdução

1.1 Contexto e Motivação

Novas tecnologias de enlace para a Internet proporcionam redes com grande disponibilidade de banda passante e, em alguns casos, com atraso elevado, como é o caso das redes Satélite. O avanço destas novas tecnologias tem sido um processo rápido e contínuo, e termos como *Gigabit* e *Terabit*, largamente difundidos no ambiente acadêmico, tornam-se cada vez mais conhecidos no ambiente comercial.

As redes de alta velocidade e/ou atrasos elevados são denominadas redes com produto banda-atraso elevado. O produto banda-atraso refere-se ao produto da capacidade do enlace, em *bits* por segundo, vezes o atraso fim-a-fim, em segundos. Estes produto determina a quantidade de dados necessária para preencher completamente o canal entre o emissor e o receptor.

Em [Subramanyan, 1995], uma rede é definida como sendo de alta velocidade quando sua capacidade é igual ou superior a 100Mbits/s. Em [Cottrell et al., 2003], onde são analisadas medidas para transferência de grande quantidade de dados em redes de alta velocidade entre o *Stanford Linear Accelerator Center* e sites remotos espalhados nos continentes, é verificado que 95% dos RTTs medidos estão na faixa de 166 a 170ms. Estudos feitos em [Shakkottai et al., 2004] apontam para um atraso médio de cerca de 200ms para o tempo de viagem em transmissões internacionais e intercontinentais. No geral, observa-se que a comunidade científica têm considerado a faixa de valores utilizados em [Cottrell et al., 2003, Shakkottai et al., 2004] como atraso elevado.

O desenvolvimento de novas aplicações para Internet, que possuem altos requisitos de banda passante, também, tem crescido rapidamente. Muitas destas aplicações tem

origem em centros de pesquisas científicas internacionais, tais como o *European Council for Nuclear Research* (CERN), e, tipicamente, transmitem vários fluxos de 10 Gbps ao longo de distâncias superiores a 10,000 Km. Pode-se citar, como exemplos, aplicações em *e-Science*, aplicações em grades e aplicações multimídia.

No entanto, ao contrário do que se poderia imaginar, a enorme disponibilidade de recursos não soluciona os problemas relacionados às necessidades de capacidade de transmissão, podendo ser, na realidade, o gargalo da Internet. Este gargalo surge em decorrência da forma como o enlace é utilizado, ou seja, como o controle da transmissão de dados é coordenado pelo protocolo *Transmission Control Protocol* (TCP). Um outro fator que contribui para a degradação no funcionamento do protocolo TCP é a longa distância sobre a qual os dados devem trafegar, o que implica em valores de atraso elevados. Desta forma, a grande dificuldade em se utilizar de modo eficiente os recursos em redes com produto banda-atraso elevado é gerada pelo protocolo TCP, que é incapaz de reagir de forma adequada ao aumento da banda e/ou do atraso.

Proposto para prevenir os colapsos de congestionamento ocorridos na década de 80, o TCP Reno, o protocolo de camada de transporte predominando na Internet, é composto por quatro algoritmos: *Slow-Start*, *Congestion Avoidance*, *Fast Recovery* e *Fast Retransmit*. *Slow-Start* e *Congestion Avoidance* são utilizados pelo TCP emissor para controlar a quantidade de dados a inserir na rede, de forma a evitar o congestionamento. Os algoritmos de *Fast Recovery* e *Fast Retransmit* são utilizados na recuperação de perdas de segmentos.

O TCP utiliza a estratégia AIMD (*Additive-Increase-Multiplicative-Decrease*), para governar o comportamento da sua janela de transmissão. Esta estratégia é inerentemente oscilatória, levando a alguns momentos de congestionamento intenso e outros de subutilização do enlace. Além disso, no TCP as reduções da janela de congestionamento, em decorrência de perdas, são excessivamente drásticas, enquanto que o incremento linear de um segmento a cada *Round Trip Time* (RTT) faz com que a janela cresça de forma muito lenta. Desta forma, o TCP subutiliza uma grande quantidade de recursos por um longo período de tempo. Um outro problema é a injustiça de RTT, onde fluxos com grande RTT são prejudicados em relação aos fluxos com RTT pequeno, o que prejudica a utilização da banda de forma equitativa. A vazão do TCP é inversamente proporcional à raiz da probabilidade de descarte/marcação, a qual aumenta com o quadrado do número de conexões ativas. Desta forma, para que o TCP alcance grandes janelas de congestionamento, e utilize de forma eficiente a banda passante disponível, a probabilidade de descarte/marcação

deve ter valores muito pequenos, valores estes inferiores ao limite teórico de taxa de erro e, portanto, irrealistas com as atuais tecnologias disponíveis. Ademais, a detecção do congestionamento é feita exclusivamente por perdas, ou seja, uma perda é sempre considerada como sinal de congestionamento. No entanto, não necessariamente uma perda é decorrente de congestionamento, como é o caso das redes sem fio que apresentam alta taxa de erros, e, assim, a taxa de transmissão é reduzida de forma errônea e desnecessária.

Para contornar os problemas apresentados pelo TCP Reno em redes com produto banda-atraso elevado, um conjunto de TCPs variantes foram propostos. Algumas destes TCPs variantes são: HSTCP, o Fast TCP, o BIC, o CUBIC, o STCP e o XCP.

O protocolo *High Speed TCP* (HSTCP), uma das primeiras variantes do TCP para redes de alta velocidade e atraso elevado, opera em duas fases diferentes. Nas situações em que a taxa de eventos de perdas é maior do que 10^{-3} , o desempenho do TCP Reno é satisfatório e, portanto, o mecanismo de ajuste do HSTCP procede da mesma forma que o do TCP Reno. Por outro lado, quando perdas são eventos raros, o HSTCP emprega uma função de crescimento da janela mais agressiva permitindo, assim, que a utilização da banda disponível seja escalável. Os parâmetros do HSTCP são ajustados de forma a estabelecer um relacionamento linear em escala log-log entre a taxa de envio e os eventos de congestionamento.

Embora sejam indispensáveis na prevenção de colapsos de congestionamento, os mecanismos de controle de congestionamento do TCP Reno e das variantes do TCP não são suficientes para prevenir o congestionamento, devido ao controle limitado que as fontes TCP exercem sobre o tráfego agregado na presença de fluxos que não estão sujeitos ao controle de congestionamento. O desempenho do mecanismo de controle de congestionamento do TCP é dependente de outros mecanismos, tais como políticas de Gerenciamento Ativo de Filas (AQM), implementadas nos roteadores. Os mecanismos de controle de congestionamento presente nos roteadores detectam o congestionamento através do monitoramento do tamanho da fila, podendo detectar o congestionamento incipiente e tomar decisões sobre a notificação e o descarte de pacotes. Ao serem notificados, os emissores TCP reduzem a taxa de transmissão antes que a fila transborde e ocorra um grande número de perdas.

Random Early Detection (RED) é a política recomendada pela *Internet Engineering Task Force* (IETF) para ser implementada na Internet. RED estima o tamanho médio da fila (Q_{avg}) e compara com dois limiares: min_{th} e max_{th} . Quando o tamanho médio da fila é menor que min_{th} nenhum pacote é descartado/marcado. Quando Q_{avg} está entre min_{th} e max_{th} , tenta-se prevenir o congestionamento. Marca-se/descarta-se cada pacote que

chega com uma certa probabilidade p , que aumenta linearmente com o tamanho médio da fila. No caso em que Q_{avg} é maior que max_{th} , a fase é de controle de congestionamento. Nesta fase, todos os pacotes que chegam a fila são descartados.

O grande desafio ao utilizar RED é ajustar os seus parâmetros, pois quando os valores dos limiares não são corretamente definidos, o desempenho de RED degrada-se, podendo ser pior do que o da tradicional política *DropTail*. Para superar estas dificuldades, numerosos estudos baseados em heurísticas tem sido conduzidos. No entanto, estes estudos não asseguram que um ponto de equilíbrio seja alcançado, nem garantem estabilidade do tamanho da fila. Pesquisas têm sido conduzidas na tentativa de derivar configurações para RED de uma maneira mais sistemática. Uma destas abordagens usa Teoria de Controle para projetar políticas de AQM que assegurem estabilidade em torno do ponto de equilíbrio. Políticas baseadas em Teoria de Controle consideram a natureza intrínseca de retroalimentação do congestionamento na rede, no qual a taxa de transmissão dos nós fontes deve ser ajustada de acordo com o estado de congestionamento, que, por sua vez, é determinado em função da ocupação da fila. Assim, os controladores são responsáveis por determinar o valor da probabilidade de descarte/marcação adequada para estabilizar o tamanho da fila independentemente das variações das condições da rede.

1.2 Principais Contribuições

Neste trabalho, são empregadas técnicas de Teoria de Controle Ótimo no sistema de controle de congestionamento da Internet. O controle de congestionamento é tratado como um sistema de controle com retroalimentação, no qual o controlador é o responsável por determinar a probabilidade de descarte/marcação, que maximiza a vazão e minimiza o número de perdas, ou seja, a taxa de transmissão deve ser ajustada em função do estado de congestionamento da rede. A probabilidade de descarte/marcação deve garantir, ainda, a estabilidade do tamanho da fila independentemente das condições da rede. As principais contribuições deste trabalho são:

- O projeto e derivação de um controlador AQM ótimo, para redes com produto banda-atraso elevado que usam o protocolo HSTCP. Ademais, a estabilidade e os objetivos de desempenho do sistema de congestionamento são expressos e solucionados através de Desigualdades Matriciais Lineares. A planta utilizada na derivação do controlador representa o sistema de congestionamento de forma detalhada, garantindo sua estabilidade independentemente das condições de rede;

- Avaliação dos objetivos de projeto de um controlador ótimo AQM que obtenha o melhor desempenho em redes com produto banda-atraso elevado;
- Avaliação, através de simulações, da capacidade do controlador HSTCP-H2 em atingir os objetivos de desempenho quando comparado ao da política de AQM padrão, RED;
- Utilização de cenário com múltiplos enlaces gargalos, tráfego cruzado e de ruído e sob diferentes cargas, permitindo, assim, a investigação do desempenho dos controladores HSTCP-H2 e RED em cenários mais próximos das redes operacionais;

1.3 Organização

Este trabalho está organizado da seguinte forma: no Capítulo 2, apresenta-se o mecanismo de controle de congestionamento do TCP Reno, os algoritmos que o compõem e suas deficiências em redes com produto banda-atraso elevado. Apresenta-se também o mecanismo de controle de congestionamento de TCPs variantes para as redes de alta velocidade e atraso elevado.

No Capítulo 3, apresenta-se o mecanismo de controle de congestionamento presente nos roteadores. São abordadas as vantagens e desvantagens da política de AQM RED, recomendada pelo IETF para a Internet. São apresentados, também, alguns trabalhos de políticas de AQM existentes. Ademais, é apresentado, o mecanismo *Explicit Congestion Notification* (ECN), que possibilita a dissociação da notificação de congestionamento de perdas de pacotes.

No Capítulo 4, introduz-se os conceitos de modelagem matemática e teoria de controle necessários para o entendimento do trabalho desenvolvido.

No Capítulo 5, o sistema de controle de congestionamento é apresentado como um problema de controle. O modelo simplificado da dinâmica do sistema de controle de congestionamento, que captura a variação da janela do HSTCP em função da variação do tamanho da fila é apresentado.

No Capítulo 6, discorre-se sobre o desenvolvimento do projeto de um controlador ótimo, denominado HSTCP-H2/AQM, utilizando técnicas de controle ótimo, ao invés de controle clássico. A planta utilizada no desenvolvimento do controlador HSTCP-H2 representa a dinâmica detalhada do sistema, garantindo a sua estabilidade independentemente da variação das condições da rede.

Para a derivação do controlador, é utilizada uma abordagem não-racional. Além disso, a estabilidade e os objetivos de desempenho do sistema de AQM são completamente expressos e solucionados através de Desigualdades Matriciais Lineares ou LMIs (*Linear Matrix Inequalities*), de forma que os parâmetros do controlador podem ser calculados através da solução de um simples problema convexo.

Neste capítulo, são discutidos quais devem ser os objetivos de desempenho de projeto de um controlador ótimo AQM, de forma a obter o melhor desempenho em redes com produto banda-atraso elevado. O melhor ponto de equilíbrio para o controlador é, também, investigado.

No Capítulo 7, apresentam-se os resultados da comparação entre os controladores obtidos, no intuito de verificar qual deles produz o melhor desempenho diante de tráfego de longa e curta duração em redes com produto banda-atraso elevado. O controlador que produziu o melhor resultado é comparado com RED. A eficácia do controlador é verificada em um ambiente dinâmico de rede, através de simulações exaustivas utilizando o simulador de redes *Network Simulator*. Foram utilizados no estudo a topologia *Dumb-bell*, com um único enlace gargalo, e a topologia *Parking-Lot*, com múltiplos enlaces gargalos e tráfego cruzado. A utilização de uma topologia com múltiplos enlaces gargalos permite a verificação da eficácia do controlador em cenários que apresentam maior proximidade aos cenários das redes operacionais. Para verificar a robustez da política sob diferentes condições de rede, variou-se a carga e tráfego de ruído foi inserido. Um gerador de tráfego foi utilizado para gerar tráfegos específicos FTP e WEB.

No Capítulo 8, são apresentadas as considerações finais e os trabalhos futuros.

1.4 Publicações

Esta seção apresenta as publicações originadas da dissertação.

- Juliana de Santi, Nelson L. S. da Fonseca, and Michele M. A. E. Lima, “AQM Controllers for High Speed TCP Networks”. In: III Workshop TIDIA, 2006, São Paulo. Proceedings of III Workshop TIDIA 8587837125, p. 87–89, 2006 [Santi et al., 2006].
- Juliana de Santi, Nelson L. S. da Fonseca and Michele M. A. E. Lima, “Projeto de Controladores Ótimos para Redes com Produto Banda-Atraso Elevado”. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2007, Belém. Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Belém do Pará, p. 899–912, 2007 [Santi et al., 2007].

- Juliana de Santi, Nelson L. S. da Fonseca, and Michele M. A. E. Lima, “Active Queue Management Controller for the High Speed TCP Protocol”, In: IEEE International Conference on Communications (IEEE ICC), Beijing China, p. 6–11, 2008 [Santi et al., 2008a].
- Juliana de Santi, Nelson L. S. da Fonseca, and Michele M. A. E. Lima, “Avaliação de Desempenho de Gerenciamento Ativo de Filas para Redes de Alta Velocidade”. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Rio de Janeiro, p. 203–216, 2008 [Santi et al., 2008b].

Capítulo 2

Mecanismos de Controle de Congestionamento TCP

Em resposta a série de colapsos de congestionamento ocorridos na década de 80, Jacobson e Karels desenvolveram um mecanismo de controle de congestionamento denominado TCP Tahoe [Jacobson and Karels, 1988].

No TCP Tahoe, novas informações são inseridas na rede na mesma proporção de recebimento de reconhecimentos (*ACKnowledgment* - ACK) pelo receptor. Nesta solução, a recuperação de perdas é feita após a ocorrência de expiração do intervalo de temporização (*timeout*), forçando a redução drástica da janela de congestionamento do emissor. Visando suprir esta deficiência foi proposto o algoritmo denominado TCP Reno, o qual é considerado a implementação TCP padrão.

O *Transmission Control Protocol*, TCP, responde por $(83 \pm 11)\%$ dos *bytes* e $(75 \pm 12)\%$ dos pacotes transmitidos e $(56 \pm 15)\%$ dos fluxos gerados na Internet [Fomenkov et al., 2004]. Estas informações demonstram que embora seja crescente o número de aplicações que utilizam o protocolo UDP, o TCP continua sendo o protocolo de transmissão de dados predominante na Internet. Desta forma, o controle de congestionamento na Internet está diretamente ligado ao funcionamento dos mecanismos de controle de congestionamento TCP, e, portanto, é imprescindível o estudo destes mecanismos.

O TCP Reno, ou simplesmente TCP, é composto por quatro algoritmos: *Slow-Start*, *Congestion Avoidance*, *Fast Recovery* e *Fast Retransmit* [Allman et al., 1999]. *Slow-Start* e *Congestion Avoidance* são utilizados pelo TCP emissor para controlar a quantidade de dados a inserir na rede, de forma a evitar o congestionamento. Os algoritmos de *Fast Recovery* e *Fast Retransmit* são utilizados na recuperação de perdas de segmentos.

Para controlar o congestionamento, o TCP utiliza o mecanismo de janela deslizante, que ajusta a taxa de transmissão em função da estimativa de banda passante disponível. O ajuste é feito de acordo com o recebimento de ACKs enviados pelo receptor.

Embora seja o mecanismo mais utilizado na Internet e apresente relativa eficiência, o mecanismo de controle de congestionamento do TCP apresenta uma série de problemas, os quais são agravados em redes com produto banda-atraso elevado [Li et al., 2007, Chen and Bensaou, 2007]. O TCP utiliza a estratégia AIMD (*Additive Increase Multiplicative Decrease*), para governar o comportamento da sua janela de transmissão. Esta estratégia é inerentemente oscilatória, o que faz com que o uso da banda passante disponível também oscile significativamente, levando a alguns momentos de congestionamento intenso e outros de subutilização do enlace. Um outro problema é a injustiça de RTT, ou seja, fluxos com grande RTT são prejudicados em relação aos fluxos com RTT pequeno, o que prejudica a utilização da banda de forma equitativa. Ademais, a detecção do congestionamento é feita exclusivamente por perdas, ou seja, perdas não necessariamente caracterizam congestionamento. No entanto, o TCP sempre considera perda como sinal de congestionamento e, assim, reduz de forma errônea sua taxa de transmissão. Com o desenvolvimento das redes de alta velocidade e grande atraso, observou-se que estes problemas se tornam ainda mais críticos em redes com produto banda-atraso elevado.

Para melhorar a capacidade do TCP em estimar a banda disponível, aprimorar o seu desempenho em redes com grande produto banda-atraso, garantir a estabilidade da janela, e, conseqüentemente, a taxa de transmissão, variações do TCP têm sido propostas, dentre estas o HSTCP [Floyd et al., 2002], o Fast TCP [Jin et al., 2003], o BIC [Xu et al., 2004], o CUBIC [Rhee and Xu, 2005], o STCP [Kelly, 2003] e o XCP [Katabi et al., 2002].

Neste Capítulo, é apresentado o mecanismo de controle de congestionamento do TCP Reno, seus conceitos e modo de funcionamento. São discutidos os problemas relacionados ao TCP Reno em redes com produto banda atraso-elevado. Posteriormente, são apresentadas os TCPs variantes HSTCP, Fast TCP, BIC, CUBIC e STCP que visam superar as deficiências de desempenho apresentadas pelo TCP Reno em redes com produto banda-atraso elevado.

2.1 O Mecanismos de Janela Deslizante do TCP

O protocolo TCP fornece tanto controle de fluxo como controle de congestionamento. O controle de fluxo torna compatível a taxa de envio do remetente com a taxa de recebimento do receptor, evitando, assim, que sejam enviados mais dados do que o receptor é capaz de receber. O controle de congestionamento refere-se ao controle de envio de dados pelo emissor de modo a evitar que a rede torne-se congestionada e ocorram perdas.

Para proporcionar controle de fluxo e controle de congestionamento o TCP utiliza um mecanismo denominado janela deslizante, que ajusta a taxa de transmissão em função da estimativa de banda passante disponível e da capacidade de recepção do receptor. No mecanismo de janela deslizante são utilizadas três informações contidas no cabeçalho IP [Postel, 1981]: o número de seqüência, *NumSeq*; o número de reconhecimento, ACK (*ACKnowledgment*); e a janela do receptor, *RcvWnd*.

Cada segmento enviado recebe um número de seqüência, *NumSeq*, para que o receptor possa reordenar os segmentos, caso eles tenham sido recebidos fora de ordem, e para solicitar a retransmissão de segmentos não recebidos.

O número de ACK é utilizado para que o receptor possa informar ao emissor que determinados segmentos foram recebidos; este campo indica que o receptor recebeu corretamente todos os *bytes* até o número ACK-1, e que espera receber o segmento com número de seqüência ACK. O receptor faz a confirmação para cada segmento recebido, entretanto, se segmentos forem recebidos fora de ordem, o receptor continua enviando o ACK com o valor do número de seqüência do segmento que espera receber. Se três ou mais segmentos ACKs duplicados forem recebidos, o emissor infere que o segmento esperado pelo receptor foi perdido. Os segmentos recebidos fora de ordem são armazenados no *buffer* do receptor até que os segmentos perdidos sejam recebidos, quando, então, são repassados para a camada de aplicação. Embora o receptor utilize reconhecimento cumulativo, ou seja, um ACK pode indicar o recebimento correto de mais de um segmento, por questões de simplicidade, será assumido que cada segmento é reconhecido por ACKs distintos.

O TCP receptor possui um *buffer* interno, no qual os segmentos são temporariamente armazenados e processados até que possam ser repassados para a aplicação na ordem correta. A capacidade do *buffer* será denotada por *RcvBuf*. O valor da janela do receptor, *RcvWnd*, informa a disponibilidade no *buffer* para armazenar segmentos recebidos. Este valor é usado pelo emissor para fazer o controle de fluxo. Seja o último *byte* recebido pelo receptor, *RcvLastByteRcvd*, e seja o último *byte* processado pela aplicação no receptor,

$RcvLastByteRead$, o valor da janela $RcvWnd$, é definido por:

$$RcvWnd = RcvBuff - (RcvLastByteRcvd - RcvLastByteRead) \quad (2.1)$$

Seja $SndNextByteSent$ o número de seqüência do próximo segmento a ser enviado pelo emissor e seja $SndNextByteAcked$ o último número de seqüência reconhecido pelo receptor. O número de segmentos em transito, $PacketsOut$, ou seja, segmentos enviados, mas não reconhecidos, é dado por:

$$PacketsOut = SndNextByteSent - SndNextByteAcked \quad (2.2)$$

Uma vez que o emissor conhece a capacidade de armazenamento do receptor, pode determinar a quantidade de dados que pode enviar sem sobrecarregar o receptor, ou seja, o valor de $PacketsOut$ não deve exceder o valor de $RcvWnd$. A taxa de transmissão do emissor é, também, limitada pela capacidade da rede, que é determinada pela quantidade de dados que se pode transmitir em um determinado momento, em *bytes* por segundo. Entretanto, o núcleo da rede é transparente ao emissor, ou seja, não informa ao emissor sobre a disponibilidade de recursos. O emissor possui uma variável denominada janela de congestionamento, $SndCwnd$, que determina a quantidade de dados que o emissor pode enviar, baseada em uma estimativa de disponibilidade de banda passante. Assim, o valor da janela é determinado dinamicamente, de forma a prevenir a ocorrência de congestionamento. Caso ocorram perdas a janela é reduzida, e caso o recebimento de segmentos seja confirmado por ACKs, então a janela é aumentada. A forma como o TCP realiza os ajustes da janela de congestionamento será apresentada na seção seguinte.

Os mecanismos de janela deslizante do emissor e do receptor são apresentados nas Figuras 2.1 e 2.2. A quantidade de dados que o emissor pode transmitir dentro de uma mesma janela em dado momento, $SndUseableWnd$, é dada por:

$$SndUseableWnd = SndCwnd - PacketsOut \quad (2.3)$$

Para cada segmento enviado há um temporizador que é utilizado para a detecção de perdas. Caso não haja o reconhecimento do segmento enviado dentro do intervalo de temporização, diz-se que ocorreu um evento de *timeout*, ou seja, o intervalo de temporização expirou. O segmento que sofreu o *timeout* é considerado perdido, e deve ser retransmitido.

O Intervalo de temporização deve ser maior que um RTT (*Round Trip Time*), ou seja, o tempo de envio do segmento e recebimento de seu reconhecimento. Desta forma, a determinação do intervalo de temporização é dependente de uma estimativa adequada

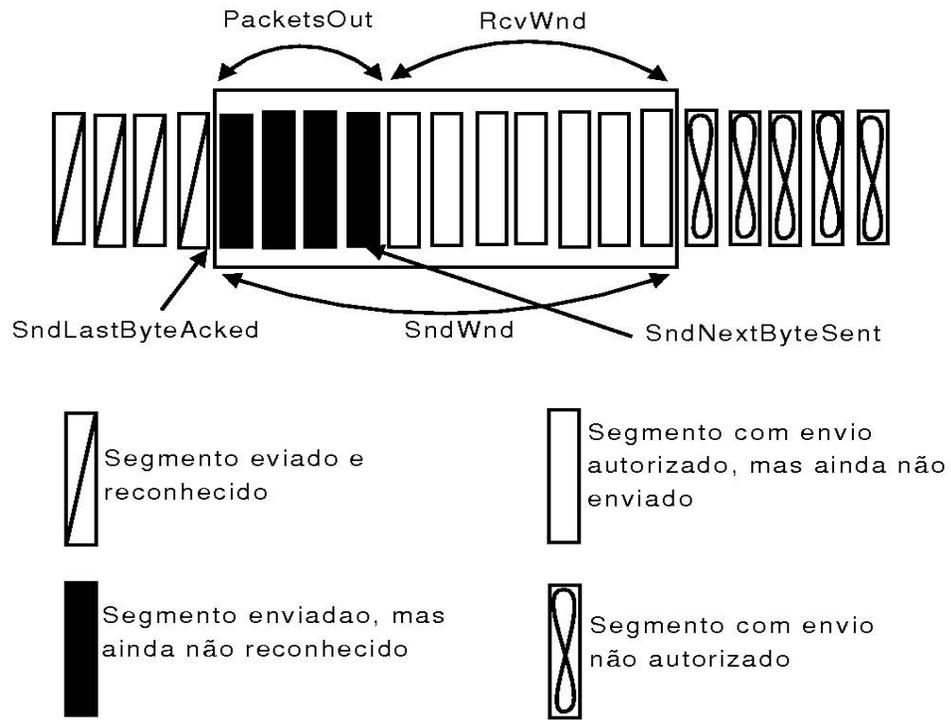


Figura 2.1: Janela Deslizante do Emissor

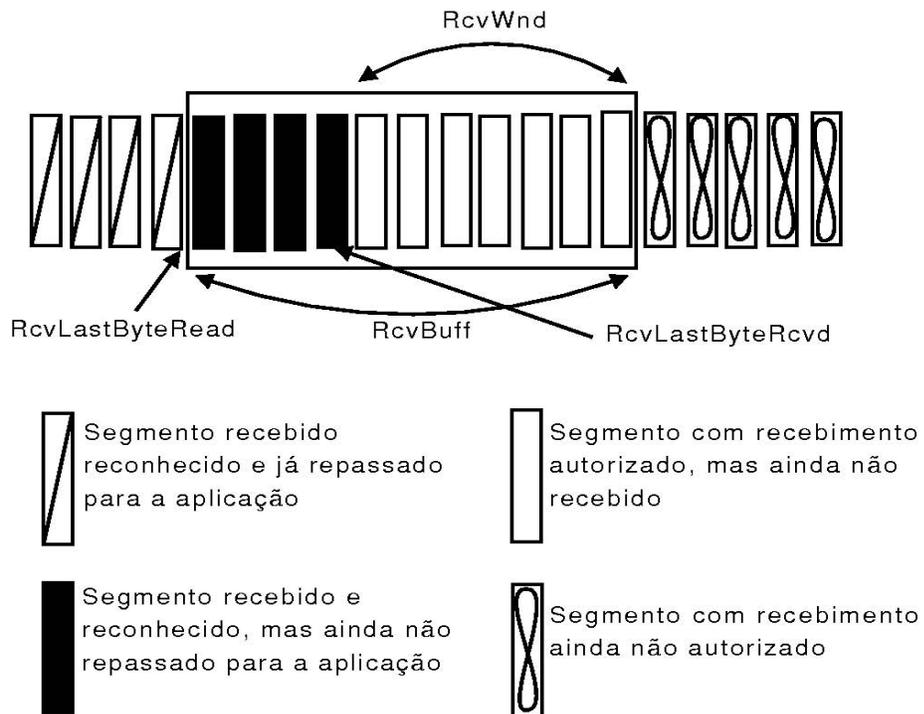


Figura 2.2: Janela Deslizante do Receptor

do RTT. Se o intervalo for menor do que o RTT, então retransmissões desnecessárias irão ocorrer. Por outro lado, intervalos muito maiores que o RTT, levariam a um tempo adicional para descobrir a ocorrência da perda e fazer a retransmissão, introduzindo, assim, atrasos desnecessários na transferência de dados.

Seja $SndSampleRTT$, o intervalo decorrido entre o envio de um segmento e seu respectivo reconhecimento por ACK. Os valores obtidos por cada segmento podem ser distintos, já que o ambiente de rede é dinâmico. Assim, para ter maior precisão na determinação do intervalo de temporização, o TCP emissor mantém uma estimativa do valor típico de RTT, $SndEstimatedRTT$, a qual é atualizada cada vez que um ACK é recebido, obtendo-se, assim, uma nova amostra de $SndSampleRTT$. A variação do RTT é definida pela variável $SndVarRTT$. O cálculo da estimativa do RTT e sua variação, é baseado em uma média móvel exponencial (MMEP). Para o cálculo de $SndEstimatedRTT$, atribui-se um peso maior aos valores das amostras de $SndSampleRTT$ obtidas mais recentemente, permitindo, assim, que o valor reflita com maior exatidão o estado atual da rede. Para o cálculo de $SndVarRTT$, verifica-se o quanto $SndSampleRTT$ se desvia do valor $SndEstimatedRTT$. Os valores utilizados são $\alpha = 1/8$ e $\beta = 1/4$ conforme recomendação de [Jacobson and Karels, 1988, Paxson and Allman, 2000]. Os valores de $SndEstimatedRTT$ e $SndVarRTT$ são dados por:

$$SndEstimatedRTT = (1 - \alpha) * SndEstimatedRTT + \alpha * SndSampleRTT \quad (2.4)$$

$$SndVarRTT = (1 - \beta) * SndVarRTT + \beta * |SndEstimatedRTT - SndSampleRTT| \quad (2.5)$$

Conhecidos os valores da estimativa do RTT e sua variação, é possível determinar o intervalo de temporização, $SndRTO$ (*Sender Retransmit TimeOut interval*). Se o valor de $SndVarRTT$ for alto, implica que há grande variação nos valores obtidos para os RTTs, e, assim, o valor de $SndRTO$ deve ser bem maior que o valor de $SndEstimatedRTT$. Se a variação é pequena, então $SndEstimatedRTT$ reflete o estado atual da rede, e o valor de $SndRTO$ deverá ser apenas um pouco maior que o valor de $SndEstimatedRTT$. O valor de $SndRTO$ é determinado por [Paxson and Allman, 2000]:

$$SndRTO = SndEstimatedRTT + 4 * SndVarRTT \quad (2.6)$$

2.2 O Mecanismo de Controle de Congestionamento do TCP Reno

O TCP Reno, a implementação TCP padrão, é composto por quatro algoritmos: *Slow-Start*, *Congestion Avoidance*, *Fast Recovery* e *Fast Retransmit* [Allman et al., 1999]. Enquanto os dois primeiros são responsáveis por coordenar os dados a serem enviados pelo emissor, os dois últimos algoritmos respondem pela recuperação de segmentos perdidos. Embora o mecanismo de controle de congestionamento do TCP Reno seja composto por quatro algoritmos, na prática eles são implementados como se fossem dois: *Slow-Start/Congestion Avoidance* e *Fast Recovery/Fast Retransmit*. O comportamento destes algoritmos pode ser observado na Figura 2.3.

No TCP a quantidade de dados injetados na rede é controlada através do *Slow-Start* e do *Congestion Avoidance*. Para este fim, é utilizada a variável *cwnd* (janela de congestionamento), que especifica a quantidade de segmentos de dados ainda não reconhecidos que o emissor pode ter na conexão, e a variável *ssthresh* (*Slow-Start threshold*) usada como limiar para o crescimento de *cwnd*.

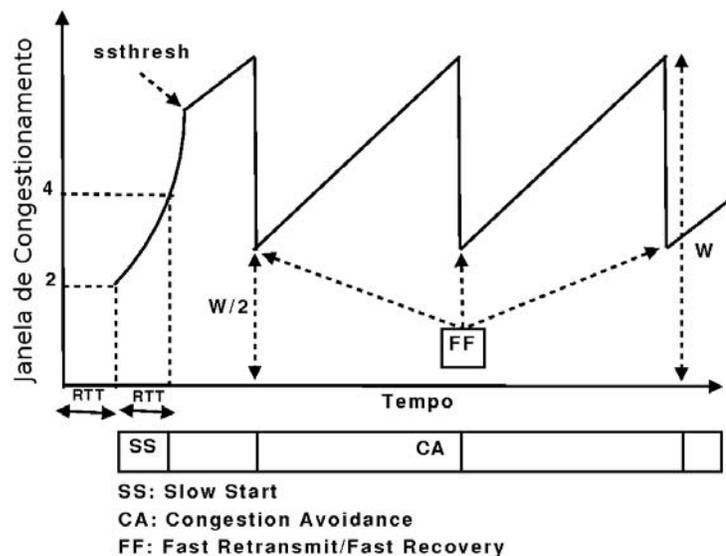


Figura 2.3: Comportamento dos algoritmos do TCP Reno

Inicialmente a disponibilidade de banda na rede é desconhecida. Para estimar esta disponibilidade, a fase de *Slow-Start* começa lentamente a injetar segmentos na rede. A medida que o receptor vai confirmando o recebimento dos segmentos (ACK), a quantidade de dados transmitidos é dobrada.

A fase de *Congestion Avoidance* passa a guiar a conexão no lugar da fase de *Slow-Start* em duas situações. A primeira ocorre quando o tamanho da janela de congestionamento ultrapassa o valor do *ssthresh*. A segunda inicia-se quando ocorre o congestionamento, o qual é indicado através de pacotes perdidos. A detecção de que dados foram perdidos dá-se pelo recebimento de ACKs duplicados ou pelo esgotamento do temporizador.

Na fase de *Congestion Avoidance* a janela de congestionamento é dividida pela metade e passa a ser incrementada linearmente por um segmento a cada *Round Trip Time* (RTT).

O *Fast Retransmit* é usado para a recuperação de perdas. Quando três ACKs identificando o mesmo segmento são recebidos, este segmento é considerado perdido, sendo então retransmitido sem ter que esperar pela ocorrência de *timeout*. Posteriormente a esta retransmissão, o *Fast Recovery* passa a guiar a transmissão de dados até que um ACK não-duplicado seja recebido. Este algoritmo faz com que o *Congestion Avoidance* seja executado ao invés do *Slow-Start*. A razão para isto é que ACKs duplicados, além de poderem indicar segmentos perdidos, podem indicar, também, que pacotes estão trafegando na rede, e desta forma, dados podem continuar a serem transmitidos, mas a uma taxa menor.

A fase de *Congestion Avoidance* permanece até que perdas sejam detectadas. Quando a perda é detectada pela ocorrência de *timeout*, o valor do limiar é ajustado para metade do valor corrente da janela de congestionamento e *Cwnd* volta ao seu valor inicial de um segmento. Neste caso, o emissor passa, novamente, a aumentar a janela de congestionamento exponencialmente, usando o procedimento de *Slow-Start* até que a janela atinja o limiar *ssthresh*.

2.3 Problemas do TCP em Redes com Produto Banda-Atraso Elevado

As tecnologias relacionadas a capacidade de transmissão de dados estão em um processo de evolução rápido e contínuo. Os termos *Gigabits* e *Terabits* tornaram-se familiares tanto no ambiente acadêmico quanto no ambiente comercial. No entanto, ao contrário do que se poderia imaginar, a enorme disponibilidade de recursos não soluciona os problemas relacionados as necessidades de capacidade de transmissão, podendo ser, na realidade, o gargalo da Internet. Este gargalo, surge em decorrência da forma como o enlace é utilizado, ou seja, como o controle da transmissão de dados é coordenado pelo protocolo da camada de transporte. Um outro fator que contribui para a degradação no funcionamento do

protocolo TCP é a longa distância sobre a qual os dados devem trafegar, o que implica em valores de atraso elevados.

As redes que apresentam grande capacidade de enlace e/ou atrasos elevados são denominadas redes com produto banda-atraso elevado. O produto banda-atraso refere-se ao produto da capacidade do enlace, em *bits* por segundo, vezes o atraso fim-a-fim, em segundos. Este produto determina a quantidade de dados necessária para preencher completamente o canal entre o emissor e o receptor. Desta forma, a grande dificuldade em se utilizar de modo eficiente os recursos em redes com produto banda-atraso elevado, é gerada pelo protocolo TCP que é incapaz de reagir de forma adequada ao aumento da banda e/ou do atraso.

Nesta seção, são abordados os problemas que degradam o desempenho do TCP, e que são agravados em redes com produto banda-atraso elevado.

2.3.1 Lei do Inverso da Raiz Quadrada

A taxa máxima, aproximada, de transmissão de uma fonte TCP é dada por [Mathis et al., 1997]:

$$SndBw \leq \frac{SndMSS}{RTT} \frac{Const}{\sqrt{p}} \quad (2.7)$$

onde, $SndMSS$ é o tamanho do segmento, p é a probabilidade de descarte/marcação e $Const$ é uma constante.

Dado que a taxa de transmissão $SndBw = \frac{Cwnd * SndMSS}{RTT}$, tem-se que:

$$Cwnd = \frac{Const}{\sqrt{p}} \quad (2.8)$$

Pelo princípio de equidade do TCP, tem-se que, no equilíbrio todas as conexões ativas sobre um enlace de capacidade C terão o valor de janela igual a $Cwnd = \frac{C}{N}$, onde N é o número de conexões ativa. Como a capacidade do enlace é um valor constante, a Equação (2.8) pode ser reescrita por:

$$p = N^2 Const \quad (2.9)$$

A Equação (2.8) é denominada “Lei do Inverso da Raiz Quadrada”. Esta lei determina que a vazão do TCP é inversamente proporcional a raiz quadrada da probabilidade de descarte/marcação p , a qual aumenta com o quadrado do número de conexões ativas (Equação 2.9). A derivação/validação deste modelo são feitas, respectivamente, em [Mathis et al., 1997] e [Ott et al., 1999].

Observando a Equação (2.8), é possível constatar que para que o valor do tamanho da janela, $Cwnd$, seja elevado, é necessário um valor extremamente baixo para a probabilidade de descarte/marcação p . Para que uma conexão TCP possa manter uma taxa de envio de 10Gbps, seria necessária uma janela de congestionamento de 83.333 segmentos, e uma probabilidade de descarte/marcação de $p \approx 2 * 10^{-10}$, ou seja, um evento de congestionamento a cada 5.000.000.000 pacotes transmitido, o equivalente a no máximo um evento de congestionamento a cada 1h e 40m. Tais valores são inferiores ao limite teórico de taxa de erro, e claramente, irrealista com as atuais tecnologias disponíveis [Floyd et al., 2002].

2.3.2 Comportamento Oscilatório da Janela de Transmissão

É conhecido que a abordagem AIMD (*Additive Increase Multiplicative Decrease*), utilizada pelo mecanismo de ajustes da janela de congestionamento do TCP, é inerentemente oscilatória, o que pode ser observado no aspecto serrilhado apresentado na Figura 2.3. Assim, a taxa de transmissão do TCP oscila consideravelmente, não estabilizando no valor da taxa de transmissão ideal, que é igual a sua porção justa de banda-passante [Chiu and Jain, 1989].

O mecanismo de controle do TCP pode ser expresso pelas equações (2.10) e (2.11).

$$cwnd \leftarrow cwnd + \frac{1}{cwnd} \quad (2.10)$$

Na Equação (2.10), a cada ACK confirmando o recebimento de pacotes, a janela de congestionamento do TCP é aumentada por um segmento.

$$cwnd \leftarrow cwnd - \frac{1}{2} * cwnd \quad (2.11)$$

A Equação (2.11) modela o ajuste da janela quando eventos de perda ocorrem. A cada sinalização de pacotes perdidos a janela de congestionamento é dividida pela metade.

Através das equações que modelam a dinâmica da janela de congestionamento do TCP, é possível verificar que o incremento linear de um pacote por RTT é muito lento, enquanto que o decréscimo multiplicativo é excessivamente drástico. Além disso, o TCP utiliza sinalização de congestionamento binária (perda de pacotes), e com isso, a oscilação torna-se inevitável.

A variação na taxa de transmissão faz com que a utilização da banda passante sofra, também, oscilações. Desta forma, em alguns momentos o canal está sobrecarregado, acarretando perdas, e em outros momentos está subutilizado, deixando recursos ociosos. Estas oscilações no tamanho da janela também levam a variações no tamanho da fila e, conseqüentemente, do RTT e do *jitter*.

2.3.3 Penalização de Conexões com RTTs longos

Ao receber segmentos de dados o TCP receptor envia ACKs de confirmação de recebimento ao TCP emissor. Através da sinalização por ACKs, o emissor infere a disponibilidade de banda passante. O tempo decorrido entre o envio do segmento e o recebimento da confirmação é denominado RTT (*Round Trip Time*). No cálculo do RTT, são considerados o tempo de propagação, característico do enlace utilizado, e o atraso em filas de roteadores. Desta forma, para ajustar a janela de transmissão o emissor TCP deve necessariamente esperar o tempo de um RTT. Assim, fluxos com RTTs longos demoram um intervalo de tempo maior para receber ACKs, logo diminuem a taxa de crescimento da janela. A consequência é que fluxos com RTT curto obtêm vazão maior do que fluxos com RTT longo. Uma vez que fluxos com RTT curto aumentam com maior rapidez a sua taxa de transmissão, os fluxos com RTT longo terão dificuldade em obter sua porção justa de banda passante. Em [Lakshman and Madhow, 1997] e [Barakat et al., 2000], verificou-se que a vazão de uma conexão é inversamente proporcional ao RTT^α , onde $1 \leq \alpha \leq 2$.

Valores elevados de RTT, prejudicam, também, a detecção de perdas, pois o tempo para que os emissores recebam a informação de perda de segmentos é maior para fluxos com RTT maior e, conseqüentemente, o tempo de recuperação de perdas para estes fluxos, é maior. Além disso, como a janela de congestionamento do TCP é dividida pela metade após um evento de perda na fase de *Congestion Avoidance*, o tempo necessário para que a conexão com RTT longo volte a utilizar a banda disponível, é consideravelmente maior do que o tempo para fluxos com RTT curto.

A fase de *Slow-Start*, é um outro problema para conexões com RTT longo. Esta fase, que deveria ser rápida e transitória, é demasiadamente prolongada, conseqüentemente aumenta-se a latência na transferência, o que diminui o desempenho destes fluxos.

2.3.4 Dependência de Perdas

Para transmitir dados sem que haja congestionamento e, conseqüentemente, perdas, o TCP precisa conhecer a disponibilidade de banda passante da rede. No entanto, a rede não informa a disponibilidade de recursos ao emissor. Desta forma, o emissor deve estimar o quanto de banda passante está disponível. Para isso, o TCP utiliza um mecanismo reativo de controle de congestionamento, no qual precisa gerar perdas para inferir o estado da rede, para então readequar sua taxa de transmissão [Gunnarsson, 2000].

Durante a transmissão, o TCP emissor sempre aumenta sua taxa de transmissão ao

ser notificado do recebimento de segmentos por meio de ACKs. Desta forma, mesmo que o TCP emissor tenha atingido a taxa de transmissão ideal, por desconhecer esta informação, continua incrementando sua janela de congestionamento, até que eventos de perdas ocorram e o TCP possa, então, reagir, diminuindo sua taxa de transmissão. Desta forma, o TCP precisa, necessariamente, gerar perdas para determinar que está transmitindo a uma taxa excessiva.

Um outro problema relacionada a notificação de congestionamento por meio de perdas, é que uma perda não necessariamente significa que a rede está congestionada. Em redes sem fio, as transmissões estão sujeitas à altas taxas de erro, que corrompem pacotes, e levam ao descarte. Além disso, desconexões ocorrem com grande frequência, devido a obstáculos físicos e a movimentação de dispositivos móveis. Todas estas perdas são, erroneamente, interpretadas como sinal de congestionamento e, conseqüente, redução na taxa de transmissão. As desconexões freqüentes, podem, ainda, gerar *timeout* em série, o que pode levar ao fim de uma conexão ou a sua inatividade, mesmo quando o nó móvel volta a se reconectar [Rezende et al., 2002].

2.4 Mecanismos de Controle de Congestionamento TCP para Redes com Produto Banda-Atraso Elevado

Reconhecendo as limitações do TCP, a comunidade de pesquisa em redes de computadores rapidamente deu respostas propondo novos mecanismos para a camada de transporte, no intuito de solucionar as deficiências do TCP em ambientes com produto banda-atraso elevado. Algumas destas propostas de TCP variantes são: *High Speed* TCP (HSTCP), FAST, BIC, CUBIC e *Scalable* TCP (STCP). Estes TCPs variantes serão abordados nesta seção.

2.4.1 *High Speed* TCP

O protocolo *High Speed* TCP (HSTCP) [Floyd et al., 2002][Souza and Agarwal, 2003] é uma modificação do mecanismo de controle de congestionamento do protocolo TCP para melhorar o desempenho das conexões TCP com grandes janelas de congestionamento e taxa de exclusão muito baixa em ambientes com produto banda-atraso elevado. A idéia deste mecanismo é que sejam feitas pequenas alterações na forma de ajuste dos parâmetros

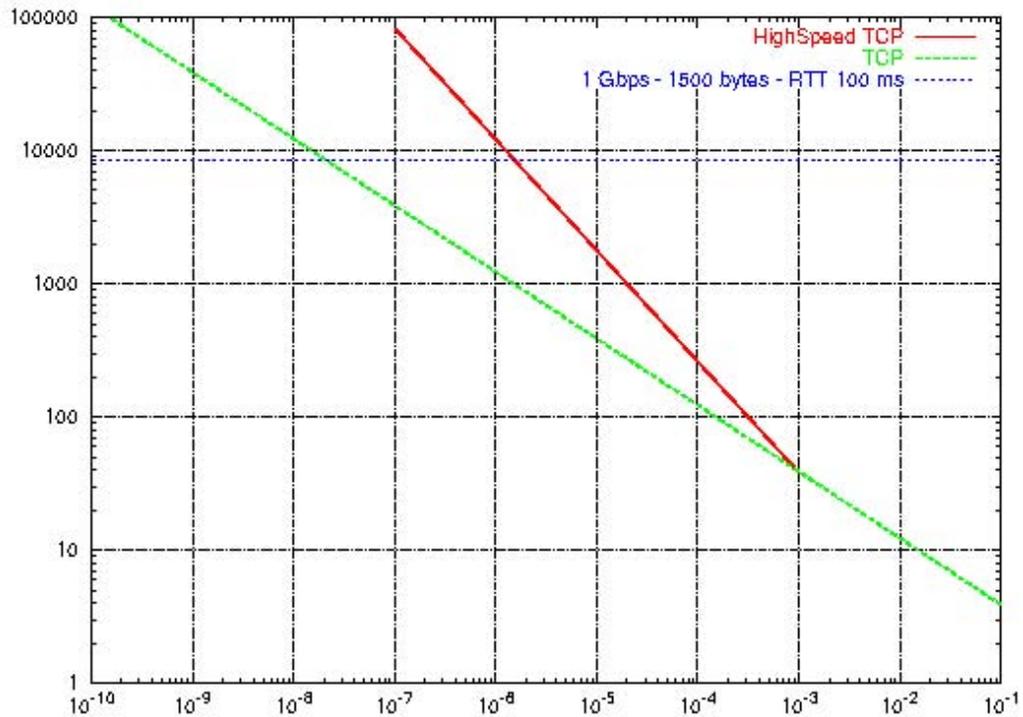


Figura 2.4: Função de Resposta do HSTCP

de incremento e decremento do TCP.

Conforme foi apresentado na Seção 2.3.2, na fase de *Congestion Avoidance*, os ajustes da janela de congestionamento do TCP Reno ($cwnd$) são caracterizados por Incremento Aditivo e Decremento Multiplicativo - AIMD. Na abordagem AIMD, a variável $cwnd$ é incrementada de um pacote por *Round Trip Time* (RTT), e a cada notificação de pacote marcado/descartado a janela é decrementada pela metade. Além disso, para atingir grandes valores para a janela de congestionamento o valor da probabilidade de descarte/marcação deve ser extremamente pequeno (Seção 2.3.1).

Para contornar este problema o HSTCP utiliza duas funções de resposta (Figura 2.4). A razão para isso é que o HSTCP só deve ter efeito com grandes janelas de congestionamento, ou seja, nas situações de congestionamento pesado onde o TCP Reno apresenta desempenho satisfatório, ele será o mecanismo de controle de congestionamento utilizado. Caso contrário, utiliza-se a função de resposta modificada do HSTCP e, desta forma, não é introduzido qualquer perigo de colapso de congestionamento.

Para especificar a função de resposta modificada do HSTCP, é necessário definir parâmetros auxiliares:

- High_Window: representa o limite superior para a janela de congestionamento;
- High_P: é a probabilidade de descarte/marcação necessária para alcançar uma janela de congestionamento de High_Window;
- Low_Window: representa o limite inferior para a janela de congestionamento, e representa o ponto de transição e compatibilidade entre as funções de resposta;
- Low_P: é a probabilidade de descarte/marcação para Low_Window;
- High_Decrease: especifica o parâmetro de decremento da janela de congestionamento, $b(cwnd)$, para $cwnd = \text{High_Window}$.

Em [Floyd et al., 2002], são sugeridos valores para as parâmetros: High_Window = 83.000; High_P = 10^{-7} ; Low_Window = 38; Low_P = 10^{-3} e High_Decrease = 0.1.

Desta forma, o procedimento para a utilização das funções de resposta é feito da seguinte forma: quando o tamanho da janela atual é no máximo *Low_Window*, ou seja, $cwnd < \text{Low_Window}$ e p é maior que 10^{-3} , a função TCP Reno é utilizada; quando o tamanho da janela atual é maior ou igual a *Low_Window*, $cwnd \geq \text{Low_Window}$, ou seja, probabilidade de descarte/marcação para o ambiente deve ser muito pequena (menor que 10^{-3}), a função HSTCP com uma agressividade maior é utilizada. A Figura 2.5 apresenta este comportamento.

Para obter a função de resposta do HSTCP, os parâmetros de incremento ($a(cwnd)$) e redução ($b(cwnd)$) da janela do TCP são modificados. Assim, na fase de *Congestion Avoidance*, a cada *round trip time* na ausência de congestionamento, o comportamento da janela do HSTCP é dado por:

$$ACK : cwnd \leftarrow cwnd + \frac{a(cwnd)}{cwnd} \quad (2.12)$$

e diante de congestionamento, em resposta ao descarte de pacotes, o comportamento da janela é definido por:

$$DROP : cwnd \leftarrow cwnd - b(cwnd) * cwnd \quad (2.13)$$

Os valores de $a(cwnd)$ e $b(cwnd)$ são ajustados em função dos parâmetros definidos para o HSTCP. Para o TCP Reno, $a(cwnd) = 1$ e $b(cwnd) = 1/2$. Assim, quando $cwnd < \text{Low_Window}$ os valores padrão de $a(cwnd)$ e $b(cwnd)$ para TCP Reno são utilizados.

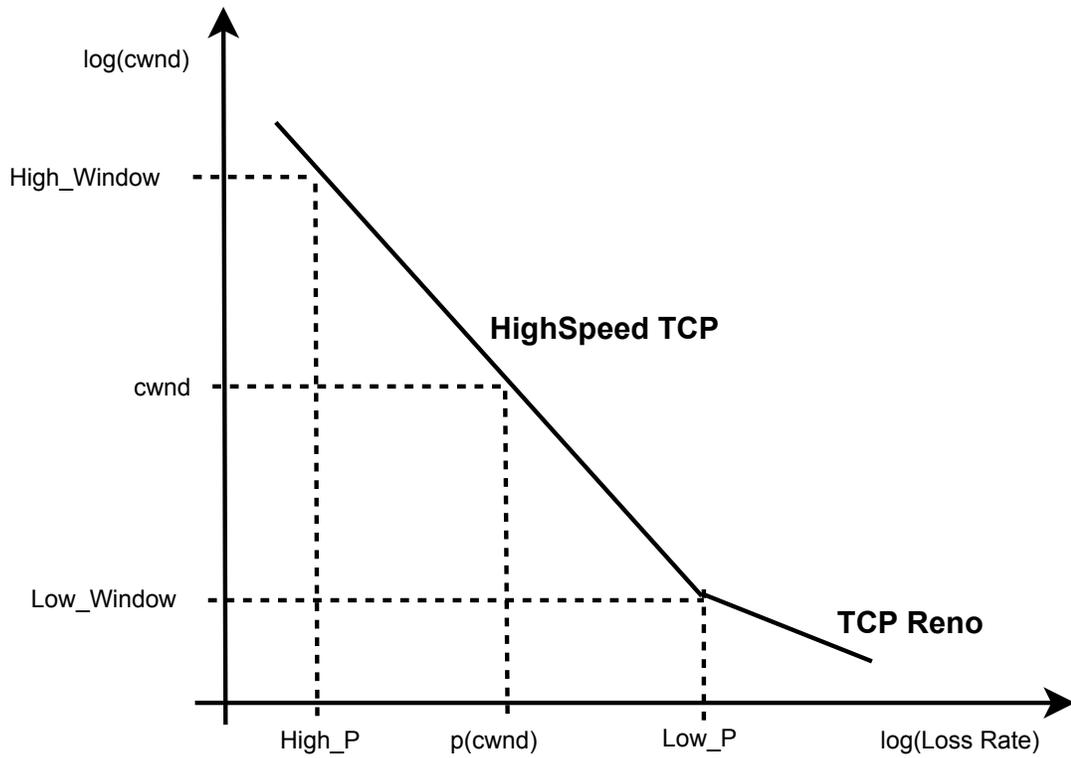


Figura 2.5: Comportamento do HSTCP

Por questões de simplicidade, o *High Speed* TCP mantém a propriedade de escala log-log na função de resposta [Floyd, 2003]. Desta forma, a função de resposta do HSTCP, para os valores de janela de congestionamento maiores que *Low_Window*, é dada por [Floyd et al., 2002]:

$$cwnd = \left(\frac{p}{Low_P} \right)^S Low_Window \quad (2.14)$$

onde p é taxa de descarte de pacotes para o tamanho médio da janela $cwnd$, e S é uma constante definida por:

$$S = \frac{\log(High_Window) - \log(Low_Window)}{\log(High_P) - \log(Low_P)} \quad (2.15)$$

Desta forma, $a(cwnd)$ é definido de acordo com:

$$a(cwnd) = \frac{(High_Window)^2 * High_P * 2 * b(cwnd)}{2 - b(cwnd)} \quad (2.16)$$

Para o TCP Reno, o parâmetro de redução da janela é $b(cwnd) = 1/2$ para $cwnd = Low_Window$. Logo, para o HSTCP, é necessário especificar $b(cwnd)$, tal que $b(cwnd) = High_Decrease$ para $cwnd = High_Window$. Em [Floyd et al., 2002], é definido que

$b(cwnd)$ deve variar linearmente com o log de $cwnd$. Assim:

$$b(cwnd) = \frac{(High_Decrease - 0.5) * (\log(cwnd) - \log(Low_Window))}{\log(High_Window) - \log(Low_Window)} + 0.5 \quad (2.17)$$

então, $a(cwnd)$ é:

$$a(cwnd) = cwnd^2 * p(cwnd) * 2 * \frac{b(cwnd)}{(2 - b(cwnd))} \quad (2.18)$$

De forma simplificada, $a(cwnd)$ e $b(cwnd)$ são [Huang et al., 2004]:

$$b(cwnd) = -0.12\log(cwnd) + 0.69 \quad (2.19)$$

$$a(cwnd) = 0.16cwnd^{0.8} \frac{b(cwnd)}{2 - b(cwnd)} \quad (2.20)$$

2.4.2 FAST TCP

O FAST TCP (*Fast Active Management Scalable Transmission Control Protocol*) [Jin et al., 2003] é uma variação TCP para redes com produto banda-atraso elevado, mas ao contrário de outras variações TCP que são baseadas no TCP Reno, o FAST TCP é baseado no TCP Vegas [Brakmo et al., 1994], e usa tanto atraso de fila quanto perda de pacotes para estimar o estado de congestionamento e fazer os ajustes da janela.

Mecanismos de controle de congestionamento TCP Reno e suas variações dependem exclusivamente de sinalização binária (um pacote é ou não perdido) para inferir a existência de congestionamento. Assim, é necessário que o *buffer* transborde periodicamente para que perdas sejam geradas, e os emissores possam detectar a existência do congestionamento. A utilização desta abordagem, inevitavelmente leva a oscilações na fila, introduzindo comportamento de *jitter*. Além disso, em redes de alta velocidade eventos de perda são eventos raros (probabilidade na ordem de 10^{-8} ou menor), e ao considerar caminhos onde pacotes podem ser excluídos ou corrompidos em decorrência de razões que não necessariamente devido a congestionamento, o desempenho destes mecanismos TCP pode ser ainda mais crítico.

No FAST TCP o mecanismo de controle de congestionamento é separado em quatro componentes funcionalmente independentes, que podem ser projetados e atualizados separadamente [Jin et al., 2005] [Wei et al., 2007]: componente de controle de dados, componente de controle de janela, componente de controle de rajada e componente de estimativa. Conforme pode-se observar na Figura 2.6, o componente de estimativa é a base para os demais componentes, ou seja, ele é responsável por gerar as informações utilizadas pelos outros três componentes na tomada de decisões.

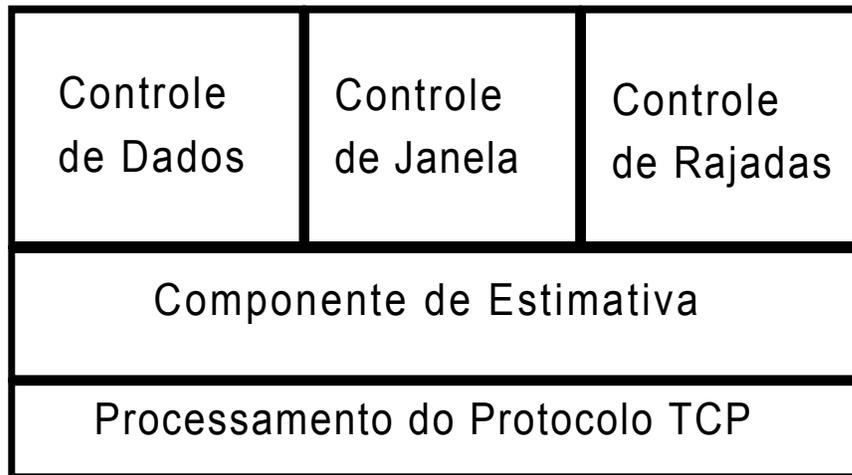


Figura 2.6: Componentes do mecanismo de controle de congestionamento do FAST TCP

Para cada pacote de dados enviado, o **componente de estimativa** calcula o atraso de fila e as indicações de perdas, fornecendo a base de informações para a tomada de decisão dos demais componentes. Quando um ACK positivo é recebido, este componente calcula o RTT para o pacote correspondente, e então atualiza o atraso médio de fila, calcula um RTT mínimo e um RTT médio exponencialmente suavizado (*exponentially smoothed average*). O RTT mínimo e médio são utilizados no componente de controle de janela. Quando um ACK negativo (sinalizado por três ACKs duplicados ou *timeout*) é recebido, ele gera uma indicação de perda para este pacote, o qual é utilizado pelo componente de controle de dados. Desta forma, o componente de estimativa gera tanto a sinalização de atraso de fila *multi-bit* como a sinalização de um *bit* baseado em perda de pacotes.

O **componente de controle de dados** seleciona os pacotes a serem enviados de um conjunto de pacotes candidatos: novos pacotes, pacotes considerados perdidos e pacotes transmitidos e ainda não reconhecidos. Quando não há perdas, novos pacotes são enviados em seqüência conforme os pacotes antigos são reconhecidos. Isto é conhecido como *self-clocking* ou *ack-clocking*. Por outro lado, durante a recuperação de perdas, uma decisão deve ser tomada com relação a quais pacotes transmitir entre: retransmitir pacotes perdidos, transmitir pacotes novos ou retransmitir pacotes antigos ainda não reconhecidos nem marcados como perdidos. O componente de controle de dados decide como compor a seleção de pacotes a enviar, o que é especialmente importante quando o produto banda-atraso é elevado, pois quando há um simples evento de perda podem, na verdade, ter sido perdidos um grande número de pacotes, e estes pacotes precisam ser retransmitidos rapidamente, de modo a manter um fluxo de ACKs estável e evitar posteriores medidas

de RTT ambíguas.

O **componente de controle de janela** determina o número de pacotes a serem transmitidos sendo, assim, o regulador da transmissão instantânea de pacotes. A forma de ajustar a janela é considerada o núcleo do FAST TCP, onde o atraso de fila é o principal fator de ajustes. Na abordagem baseada em perdas, é necessário que o *buffer* transborde para gerar descarte de pacotes, o que induz ao comportamento de *jitter*. Por outro lado, informações baseadas em atraso permitem a convergência das fonte para um estado de equilíbrio.

Sobre condições normais, FAST TCP periodicamente atualiza a janela *cwnd* baseado no RTT médio de acordo com a seguinte equação:

$$cwnd \leftarrow \min \left(2cwnd, (1 - \gamma)cwnd + \gamma \left(\frac{baseRTT}{RTT} cwnd + \alpha \right) \right) \quad (2.21)$$

onde γ é uma constante entre 0 e 1, RTT é o RTT médio atual, *baseRTT* é o RTT mínimo, e α é um parâmetro do protocolo que controla a justiça e o número de pacotes de cada fluxo que estão enfileirados na rede, e é definido em função da janela de congestionamento e do atraso de fila [Jin et al., 2003].

O modelo matemático simplificado da Equação (2.21) é dado por:

$$cwnd_i(t + 1) = cwnd_i(t) + \gamma(\alpha_i - x_i(t)q_i(t)) \quad (2.22)$$

onde $cwnd_i(t)$ é o tamanho da janela do fluxo i no tempo atual t , $x_i(t)$ é a vazão atual, e $q_i(t)$ é atraso de fila atual. O FAST TCP é denominado uma versão do TCP Vegas para redes de alta velocidade, e desta forma, ajusta a janela por uma grande quantidade (crescimento ou redução) quando o número de pacotes na fila está distante do alvo α_i , e ajusta por pequenas quantidades quando está próximo de α_i .

O **componente de controle de rajadas** determina quando transmitir pacotes, e o momento da transmissão é determinado em função da chegada de ACKs e conseqüente liberação de espaço na janela de congestionamento. Ele faz com que a transmissão de pacotes seja semelhante a um fluxo, permitindo, assim, a detecção da banda passante disponível. Este componente é particularmente importante em redes com produto banda-atraso elevado, onde o tráfego pode ser composto de muitas rajadas devido a eventos na rede e nos sistemas finais. Um simples ACK pode reconhecer uma quantidade muito grande de pacotes, abrindo espaço na janela para uma grande rajada, ou ainda, a CPU pode estar ocupada com pacotes que estão chegando e gerando interrupções; enquanto isso, uma grande quantidade de dados pode ser acumulado na fila para ser transmitida

em uma grande rajada quando houver disponibilidade da CPU. Rajadas extremamente grandes criam longas filas e aumentam a probabilidade de descarte em massa.

Uma forma de resolver o problema de rajadas no emissor é o *pacing*, onde grandes incrementos na janela são divididos em incrementos menores. Isto é realizado pelo escalonamento de sucessivas transmissões de pacotes em um intervalo de tempo constante, obtido pela divisão da janela de congestionamento pelo RTT atual. Entretanto, em uma grande janela de congestionamento somente o *pacing* pode não resolver o problemas das rajadas. Nestas situações, os mecanismos de controle de rajadas *Burtiness reduction* e *Window pacing* são utilizados. O *Burtiness reduction* decide quantos pacotes enviar quando um *ack* avança a janela por uma grande quantidade, e tenta limitar o tamanho da rajada em escalas de tempo menores do que um RTT. No *Window pacing*, quando há um período de ausência de reconhecimentos, a janela de congestionamento é aumentada para o alvo determinado pelo componente de controle de janela.

O objetivo em tomar decisões baseado no atraso de fila é manter a fila estável no roteador do enlace gargalo, ao invés de periodicamente induzir perdas, permitindo, assim, manter alta utilização. FAST TCP usa atraso de fila para controle de congestionamento quando o congestionamento é moderado e medidas confiáveis de RTT estão disponíveis.

O objetivo em reagir a perda de pacotes é reduzir a transmissão de pacotes rapidamente quando um congestionamento severo ocorre. FAST TCP reduz o congestionamento, trazendo o sistema para um estado de congestionamento moderado no qual pode-se estabilizar o atraso em torno de um atraso de fila alvo. Quando medidas confiáveis de RTT estão novamente disponíveis, a janela pode ser ajustada em função do atraso de fila, conforme descrito anteriormente.

Desta forma, uma grande vantagem ao usar atraso de fila como medida de congestionamento, é que o atraso pode ser estimado de maneira mais precisa do que a probabilidade de perda, pois amostras de perda provêem informações pouco refinadas (um *bit*), enquanto que cada medida de atraso de fila prove informações *multi-bits*, tornando mais fácil para uma implementação baseada em equações estabilizar a rede em torno de um ponto de equilíbrio de forma justa e com taxa de utilização alta. Ademais, a dinâmica do atraso de fila é escalável em relação a capacidade da rede, e isto ajuda a manter a estabilidade conforme a rede vai crescendo em capacidade.

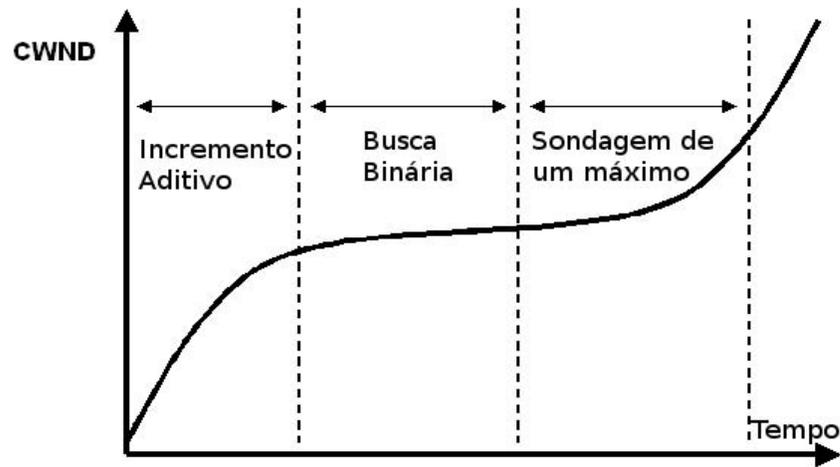


Figura 2.7: Função de Crescimento de BIC

2.4.3 Binary Increase Congestion Control

O protocolo *Binary Increase Congestion Control* (BIC) [Xu et al., 2004] é um TCP variante que modifica a função de crescimento da janela para redes de alta velocidade. O BIC busca encontrar um equilíbrio entre justiça, taxa de transmissão eficiente e estabilidade. Este mecanismo combina as técnicas de incremento de busca binária e de incremento aditivo. A Figura 2.7 apresenta as fases da função de crescimento do BIC.

No **Incremento de Busca Binária**, o controle de congestionamento é tratado como um problema de busca no qual o sistema sinaliza a perda de pacotes se a taxa de envio atual é maior do que a capacidade da rede. A janela mínima ($cwnd_min$) é estimada como o tamanho da janela na qual o fluxo não percebe qualquer perda de pacote. Se o tamanho máximo da janela ($cwnd_max$) é conhecido, a técnica de busca binária é aplicada para ajustar o tamanho da janela desejada para o ponto médio ($cwnd_target$) entre o máximo e mínimo. Conforme o tamanho da janela se aproxima da janela alvo, se houver perda, a janela atual é tratada como um novo máximo e o tamanho reduzido da janela é considerado o novo mínimo. O ponto médio entre estes novo valores torna-se o novo alvo.

A razão para esta abordagem é que perdas podem ocorrer em torno de um novo máximo, mas não em torno de um novo mínimo, então, o tamanho desejado deve ser a média entre os dois valores. Se a janela alvo é atingida e não ocorrer perdas de pacotes, então o tamanho atual da janela torna-se o novo mínimo e um novo valor alvo é calculado. Este procedimento é repetido com a atualização do mínimo e do máximo até que a diferença entre os dois fique abaixo de um limiar de incremento mínimo (S_{min}).

O incremento de busca binária permite que a banda seja explorada de maneira mais agressiva na fase inicial, quando a diferença do tamanho da janela atual para o tamanho da janela alvo é maior, e diminui a taxa de aumento conforme o tamanho da janela atual se aproxima do tamanho da janela alvo. O protocolo apresenta uma função de crescimento logarítmica, ou seja, ele reduz sua taxa de crescimento conforme o tamanho da janela se aproxima de um ponto de saturação. Outros protocolos escaláveis tendem a crescer sua taxa de forma que o crescimento no ponto de saturação é o máximo na etapa atual. Tipicamente, o número de pacotes perdidos é proporcional ao tamanho do último crescimento antes da perda. Desta forma, o incremento de busca binária pode reduzir consideravelmente a perda de pacotes.

Para assegurar rápida convergência e justiça de RTT, o incremento de busca binária é combinado com a estratégia de **incremento aditivo**. Quando a distância entre o ponto médio e um mínimo é muito grande, incrementa-se o tamanho da janela diretamente para o ponto médio, abordagem que pode adicionar muita perturbação na rede. Quando a distância entre o tamanho da janela atual e o tamanho da janela alvo, na fase de incremento de busca binária, é maior que um incremento máximo predefinido (S_{max}), ao invés de incrementar a janela diretamente para o ponto médio no próximo RTT, incrementa-se por S_{max} até a distância se tornar menor que S_{max} , quando a janela é incrementada diretamente para o valor da janela desejado. Em resumo, depois de uma grande redução na janela, incrementa-se inicialmente a janela linearmente; após, incrementa-se de forma logarítmica. Esta combinação de incremento de busca binária com incremento aditivo é chamada de **incremento binário**.

Combinada com uma estratégia de decremento multiplicativo, o incremento binário torna-se muito parecido com o incremento aditivo puro para grandes janelas. Isto porque uma grande janela resulta em um grande redução pelo decremento multiplicativo e, portanto, um longo período de incremento aditivo.

Se o crescimento da janela passar do máximo atual, então o máximo torna-se desconhecido. Neste momento, a busca binária ajusta o seu máximo para ser o máximo padrão (uma constante com valor elevado) e o tamanho da janela atual passa a ser o mínimo. Assim, o ponto médio alvo pode estar muito distante. De acordo com a busca binária, se o ponto médio alvo é muito grande, ele incrementa linearmente pelo incremento máximo. Ao invés disso, o *slow start* é executado para verificar por um novo limiar máximo S_{max} , esta é a terceira fase do BIC, a **sondagem por um novo máximo** mostrada na Figura 2.7. Se $cwnd$ é a janela atual e o incremento máximo é S_{max} , então ele incrementa em

cada RTT em passos de $cwnd + 1$, $cwnd + 2$, $cwnd + 4, \dots$, $Cwnd + S_{max}$. A razão para tal é que é provável que ele esteja no ponto de saturação e como o máximo é desconhecido, BIC experimenta a banda disponível utilizando o algoritmo de *slow start* até que ele esteja seguro para incrementar a janela por S_{max} . Depois do *slow start*, ele volta para o incremento binário.

Na Figura 2.7, é possível observar que a função de crescimento da janela na fase de sondagem é simétrica aquela usada no incremento binário, mas em ordem inversa: ele usa o inverso da busca binária, ou seja, substitui o crescimento logarítmico pelo exponencial, e então usa incremento aditivo. Durante a sondagem, a janela cresce lentamente no início para encontrar um novo máximo nas proximidades, e depois de algum tempo de crescimento lento, se um novo máximo não for encontrado, ou seja, pacotes não foram perdidos, então aumenta-se a região de busca por um novo máximo. Desta forma, utiliza-se o incremento aditivo onde a janela é aumentada por um valor fixo elevado.

Em um modelo de perdas completamente sincronizado, incremento de busca binária combinado com decremento multiplicativo convergem para a equidade. Suponha dois fluxos com janelas diferentes, mas com RTT igual. Uma vez que janelas maiores tem uma redução maior no decremento multiplicativo (com um fator fixo β), o tempo para alcançar o alvo é maior para janela maior. Entretanto, o tempo de convergência pode ser muito longo. No incremento de busca binária, são gastos $\log(d) - \log(S_{min})$ RTT para atingir a janela máxima depois de uma redução d . Uma vez que a janela é incrementada em passos de \log , grandes janelas e pequenas janelas podem alcançar seus respectivos máximos muito rápido e quase ao mesmo tempo. Assim, uma conexão com janela pequena tira somente uma pequena quantidade de banda de uma conexão com janela grande antes da próxima redução. Para lidar com este comportamento, foram introduzidas modificações no incremento de busca binária.

No incremento de busca binária depois de uma redução, novos máximos e mínimos são ajustados. Supondo que estes valores sejam max_wind_i e min_win_i para fluxos i ($i = 1, 2$). Se o novo máximo é menor que o anterior, esta janela está numa tendência de redução (provavelmente por ter uma janela maior que a equidade). Então, o novo máximo é reajustado para ser o mesmo valor da nova janela alvo, ou seja, $max_win_i = (max_win_i - min_wind_i)/2$, e o alvo é reajustado. Aplica-se posteriormente o incremento binário normal. Esta é a estratégia de convergência rápida.

Embora alcance boa escalabilidade, justiça e estabilidade em ambientes de alta velocidade, a função de crescimento do BIC pode ser muito agressiva para o TCP especialmente

na presença de RTT baixo e em redes de baixa velocidade. Além disso, as diferentes fases de controle da janela adicionam complexidade à análise do protocolo.

2.4.4 CUBIC

O TCP CUBIC [Rhee and Xu, 2005] é um TCP variante para redes de alta velocidade. Ele é uma versão melhorada do BIC que simplifica o controle de janela enquanto melhora a justiça de RTT e a compatibilidade com o TCP.

O crescimento da janela do CUBIC é determinado por uma função cúbica (Figura 2.8), a qual é muito parecida com a função de crescimento do BIC (Figura 2.7). A janela de congestionamento de CUBIC é dada pela seguinte equação:

$$cwnd_{cubic} = C(t - K)^3 + cwnd_max \quad (2.23)$$

onde, C é o fator de escala, t é o tempo decorrido desde a última redução da janela, $cwnd_max$ é o tamanho da janela exatamente antes da última redução da janela, e $K = \sqrt[3]{cwnd_max \frac{\beta}{C}}$, onde β é uma constante que representa o fator de decréscimo multiplicativo da janela quando perdas de pacotes ocorrem, ou seja, a janela é reduzida para $\beta * cwnd_max$.

Todo o crescimento da janela de CUBIC é descrito por uma única função. Desta forma, não existem diferentes fases de controle da janela de congestionamento como acontece no BIC (incremento aditivo, busca binária e sondagem). Esta característica simplifica o CUBIC de maneira expressiva.

Como pode-se observar na Figura 2.8, a janela do CUBIC cresce rapidamente após uma redução, mas ao se aproximar de $cwnd_max$ seu crescimento é lento, chegando a ser quase zero. Acima de $cwnd_max$ o CUBIC inicia um processo de sondagem por disponibilidade de banda passante, na qual a janela cresce vagarosamente na fase inicial e acelera o crescimento conforme se afasta de $cwnd_max$. Este forma de crescimento lento em torno de $cwnd_max$ melhora a estabilidade do protocolo e aumenta a utilização da rede, enquanto que o crescimento rápido, quando a janela está distante de $cwnd_max$, assegura a escalabilidade do protocolo.

A função cúbica assegura a justiça intra-protocolos entre os fluxos concorrentes do mesmo protocolo. Por exemplo, supondo que dois fluxos estão competindo no mesmo caminho fim-a-fim. Os dois fluxos convergem para a equidade desde que eles reduzam a janela pelo mesmo fator multiplicativo β , de forma que o fluxo com $cwnd_max$ maior reduza mais, e a função de crescimento faça com que o fluxo com $cwnd_max$ maior

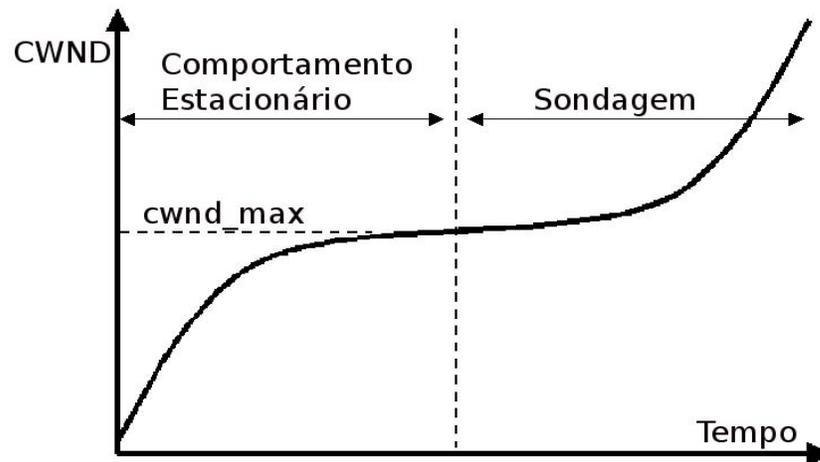


Figura 2.8: Função de Crescimento do CUBIC

incremente a janela de maneira mais lenta. Desta forma, os dois fluxos eventualmente convergem para o mesmo tamanho de janela[Rhee and Xu, 2005].

Para assegurar a propriedade de justiça, o CUBIC usa uma estratégia na qual a taxa de crescimento da janela é determinada pelo tempo decorrido t . Isto assegura justiça de RTT linear uma vez que os fluxos concorrentes com diferentes RTT terão o mesmo valor de t depois de uma perda de pacotes sincronizada. Para melhorar a justiça e a estabilidade, o crescimento da janela não deve ser maior que S_{max} por segundo. Esta característica mantém crescimento linear da janela quando ela está longe de $cwnd_max$, fazendo com que a função de crescimento esteja de acordo com a função do BIC. A diferença é que o incremento linear da janela do CUBIC é dependente do tempo real, desta forma, diante de RTT curto o incremento linear do CUBIC por RTT é menor, embora permaneça constante em tempo real.

O incremento da janela em tempo real melhora consideravelmente a compatibilidade com o TCP. Enquanto a função de crescimento da janela de outros protocolos dependentes de RTT (por exemplo o TCP Reno), crescem proporcionalmente mais rápido (em tempo real) quando o RTT é menor, o CUBIC cresce independentemente do RTT. Desta forma, diante de RTT menor o TCP apresenta uma agressividade maior enquanto CUBIC permanece inalterado, sendo, portanto, mais amigável para o TCP.

No CUBIC quando o tamanho da janela é pequeno, o valor de $cwnd_max$ é também pequeno. Assim, K é menor e, portanto, a taxa de crescimento da função do CUBIC é muito lenta. Desta forma, mesmo quando o produto banda-atraso da rede é pequeno, CUBIC é TCP-friendly.

Para manter a mesma taxa de crescimento do TCP, emula-se o algoritmo de ajustes da janela do TCP após um evento de perda. Uma vez que CUBIC reduz a janela por um fator β depois de um evento de perda, o incremento aditivo do TCP-*fair* seria $3((1-\beta)/(1+\beta))$ por RTT. Isto porque a taxa de envio de um protocolo AIMD é dada por:

$$\frac{1}{RTT} \sqrt{\frac{\alpha}{2} \frac{1+\beta}{1-\beta} \frac{1}{p}} \quad (2.24)$$

onde α é o incremento aditivo da janela, e p é a taxa de perda. Para o TCP, os valores de α e β são 1 e $\frac{1}{2}$, respectivamente, assim, a taxa de envio média do TCP é:

$$\frac{1}{RTT} \sqrt{\frac{3}{2} \frac{1}{p}} \quad (2.25)$$

Para atingir a mesma taxa de envio do TCP com um β arbitrário, α deve ser $3((1-\beta)/(1+\beta))$. Se β for ajustado para 0.8, o fator de incremento aditivo é 0.5. Dada esta taxa de crescimento por RTT, o tamanho da janela do TCP emulado no tempo t é:

$$cwnd_{tcp} = cwnd_{max} \beta + 3 \left(\frac{1-\beta}{1+\beta} \frac{t}{RTT} \right) \quad (2.26)$$

Se $cwnd_{tcp}$ é maior do que $cwnd_{cubic}$ (Eq. 2.23), então $cwnd_{tcp}$ é ajustada. Caso contrário, $cwnd_{cubic}$ é o tamanho da janela de congestionamento atual. Se o período de duração do congestionamento é menor que $(1/C * RTT)$, ou se a taxa de perda de pacotes é maior que $(0.35 * C * RTT * 3)$, então o CUBIC é TCP-*friendly* [Rhee and Xu, 2005].

2.4.5 Scalable TCP

O Scalable TCP - STCP [Kelly, 2003] é um TCP variante para redes de alta velocidade, cuja idéia principal é tornar o tempo de recuperação, após um evento de perda, independente do tamanho da janela de congestionamento. O Scalable TCP baseia-se no HSTCP e em técnicas de controle de congestionamento por engenharia estável.

No STCP, quando não há congestionamento, para cada reconhecimento recebido em um RTT a janela é aumentada por:

$$cwnd \rightarrow cwnd + a \quad (2.27)$$

onde a é uma constante com $0 < a < 1$. Quando há detecção de congestionamento a janela é ajustada da seguinte forma:

$$cwnd \rightarrow cwnd - [b * cwnd] \quad (2.28)$$

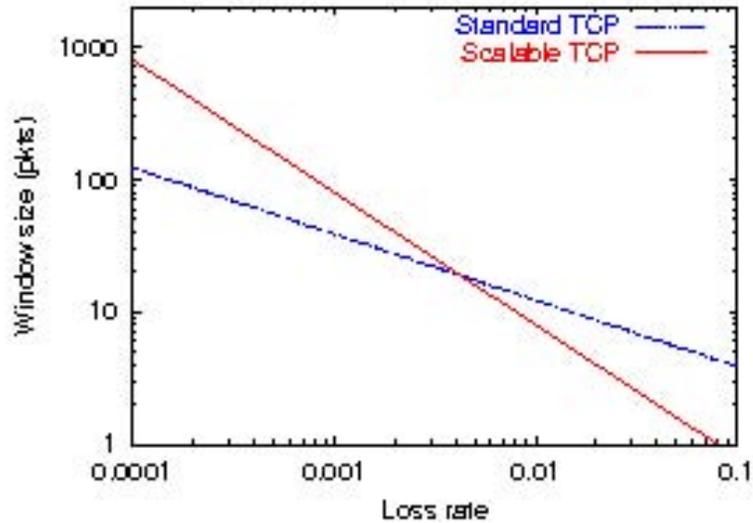


Figura 2.9: Função de Resposta do STCP e TCP

sendo b uma constante com $0 < b < 1$.

O tempo de recuperação de pacotes perdidos na implementação TCP é proporcional ao tamanho da janela e ao RTT da conexão. No STCP o tempo de recuperação é proporcional, somente, ao RTT. Esta diferença permite ao STCP se sobressair em relação ao TCP em redes de alta velocidade. A propriedade de escala é aplicável a quaisquer valores das constantes a e b , sendo que estes valores são determinados em função das restrições de implementação e desenvolvimento [Kelly, 2003]. Recomenda-se que $a = 0.01$ e $b = 0.125$, o que foi definido levando-se em consideração o tráfego legado, as propriedades de alocação de banda, a variação da taxa do fluxo, propriedades de convergência e teorias de controle de estabilidade.

Conforme foi verificado na Seção 2.3.1, as conexões TCP não alcançam grandes janelas de congestionamento em redes com produto banda-atraso elevado. Seja l_cwnd (*legacy window*) a janela máxima que uma conexão TCP consegue atingir, e seja, l_p (*legacy loss rate*) a taxa de perda máxima necessária para atingir l_cwnd . Então, o princípio do STCP é, utiliza-se o algoritmo de ajustes do TCP nas situações que ele é efetivo ($cwnd \leq l_cwnd$), e utiliza o algoritmo de atualização do STCP quando o TCP tem seu desempenho degradado ($cwnd > l_cwnd$). Desta forma, o STCP considera dois estados: quando o nível de congestionamento é tal que a taxa de exclusão é maior que l_p , o TCP é utilizado; para nível de congestionamento com taxa de exclusão menor que l_p , o STCP é utilizado. No segundo caso, as conexão STCP terão uma janela de

congestionamento muito maior do que as conexões TCP. No entanto, isto não deve ser considerado como injustiça, pois as conexões TCP não sofrerão *starvation*, e, além disso, os recursos utilizados pelas conexões STCP, de outra forma seriam subutilizados.

O valor recomendado para l_cwnd é 16 [Kelly, 2003], o que leva p_l ser igual a $5.86 * 10^{-3}$. Considerando as funções de resposta do STCP ($cwnd = \frac{a}{b \cdot p}$) e do TCP $cwnd = \sqrt{\frac{1.5}{p}}$, o comportamento do STCP e do TCP são mostrados na Figura 2.9

Capítulo 3

Gerenciamento Ativo de Filas

É notória a importância dos mecanismos de controle de congestionamento presentes no protocolo de transporte *Transmission Control Protocol* (TCP). No entanto, é conhecido, também, que somente estes mecanismos não são suficientes para garantir qualidade de serviços em todas as situações, uma vez que outros protocolos não diminuem sua taxa de transmissão diante de congestionamento. Desta forma, há a necessidade de mecanismos auxiliares para controlar o congestionamento.

Os mecanismos de controle de congestionamento presentes nos roteadores detectam o congestionamento através do monitoramento do tamanho da fila e, assim, podem detectar o congestionamento incipiente e tomar decisões sobre a notificação e o descarte de pacotes. Estes mecanismos de controle de congestionamento presentes nos roteadores são os algoritmos de gerenciamento de fila.

Neste capítulo são apresentados mecanismos de gerenciamento de fila em redes TCP/IP. Primeiramente, são apresentadas as características e os problemas de *DropTail*, a mais simples das políticas de gerenciamento de fila. Discorre-se sobre a necessidade da utilização de um novo mecanismo de gerenciamento de fila, denominado gerenciamento ativo de filas ou AQM. Na Seção 3.3, são discutidos os requisitos para políticas de AQM. Na Seção 3.4, é apresentado RED, a política de AQM padrão utilizada na Internet. Na Seção 3.5, são apresentadas algumas políticas de AQM baseadas em Teoria de Controle. Na Seção 3.6, é apresentado o conceito do mecanismo de *Explicit Congestion Notification* (ECN).

3.1 DropTail

A política de AQM mais simples é conhecida como *DropTail*. Com *DropTail*, os roteadores aceitam pacotes até que a capacidade máxima da fila seja atingida, quando, então os pacotes que chegam são descartados, ou seja, pacotes são aceitos enquanto existir espaço livre na fila [Braden et al., 1998]. *DropTail* foi utilizada por muitos anos como política de gerenciamento de fila na Internet. Entretanto, ela apresenta algumas deficiências:

- **Justiça:** *DropTail* permite que uma única conexão ou um pequeno número de conexões monopolize todo o espaço na fila, enquanto outras conexões tem seus pacotes descartados por falta de espaço. Além disso, o descarte não é proporcional a utilização da fila, logo, um tráfego pode ter seus pacotes descartados mesmo sem ter utilizado nenhum recurso. Ademais, como apenas alguns fluxos utilizam a fila, os fluxos que transmitem em rajadas são prejudicados, uma vez que a fila está cheia e todos os seus pacotes são descartados, sua taxa de perda é, conseqüentemente, maior;
- **Filas Cheias:** com *DropTail* a fila pode ficar completamente ocupada por um longo período de tempo, dado que a sinalização de congestionamento ocorre somente quando a fila está cheia. Nesta situação, o tráfego em rajadas é descartado o que causa injustiça. Além disso, filas cheias favorecem o fenômeno de sincronização global, que ocorre quando diversos fluxos TCP tem pacotes perdidos, o que leva a redução simultânea da taxa de transmissão.

3.2 Gerenciamento Ativo de Filas - AQM

Diante das deficiências da política *DropTail*, duas abordagens foram apresentadas: realizar descarte aleatório de um pacote ou descartar o primeiro pacote da fila [Braden et al., 1998]. Nas duas propostas, um pacote é removido da fila para dar espaço a um pacote que chega, quando a fila está cheia. O descarte aleatório apresenta um custo de $O(n)$, onde n é o tamanho da fila, sendo, portanto, elevado. Estas abordagens, no entanto, não supera todas as dificuldades de *DropTail*.

O principal objetivo em se utilizar *buffers* em redes é permitir que dados enviados em forma de rajadas possam ser absorvidos temporariamente para serem depois transmitidos, aumentando, assim, a vazão. Se as filas estiverem sempre cheias, não há espaço para absorver as rajadas. Aumentar o tamanho da fila também não é uma alternativa ideal, já

que aumentaria o atraso. O tamanho máximo da fila deve refletir o tamanho da rajada máxima que se deseja absorver. Além disso, o tamanho médio da fila deve ser mantido baixo para diminuir o atraso fim-a-fim dos pacotes. Flutuações no tamanho da fila devem ser permitidos para absorver rajadas e, também, para não penalizar o congestionamento passageiro.

Quando *DropTail* é utilizado os emissores diminuem sua taxa de transmissão ao serem notificados da existência de congestionamento, o que acontece somente quando a fila está cheia. O intervalo de tempo entre o descarte do pacote pela política de gerenciamento de fila até o momento em que o emissor é notificado é consideravelmente grande, e o emissor durante este período continua a transmitir pacotes que também serão descartados devido ao congestionamento existente. Desta forma, são necessários mecanismos que evitem ao máximo o descarte de pacotes. Para tal, o emissor precisa ser notificado a fim de que diminua sua taxa de transmissão antes que a fila esteja completamente cheia.

Uma outra necessidade é a manutenção do tamanho da fila baixo, e para isso é necessário que pacotes sejam descartados de forma aleatória, e os nós emissores sejam notificados da existência de congestionamento, para que possam reduzir a taxa de transmissão, e evitem o transbordo da fila. O mecanismo de gerenciamento de filas permite que os roteadores controlem quando e como devem descartar pacotes, e tenta solucionar os problemas relacionados à justiça e transbordo de filas apresentados por *DropTail*. Tal mecanismo é denominado de gerenciamento ativo de filas, ou *Active Queue Management* (AQM).

O propósito de AQM é a detecção e notificação de congestionamento incipiente, de forma a permitir que os emissores TCP possam reduzir sua taxa de transmissão antes que as filas transbordem evitando, assim, a degradação de desempenho.

3.3 Requisitos para Políticas de Gerenciamento Ativo de Filas

Nesta Seção, os principais objetivos para políticas de gerenciamento ativo de filas são abordados.

Ao detectar uma perda a conexão TCP reduz sua taxa de transmissão pela metade. Se a política de AQM descartar um pacote de cada conexão ativa, então, haverá a redução simultânea da taxa de transmissão de todas as conexões, e a utilização da banda passante será baixa. Isto caracteriza o fenômeno de sincronização global [Floyd and Jacobson, 1993]. A ocorrência da sincronização global, apesar de ser um fenô-

meno raro, degrada a utilização da rede e, portanto, deve ser evitada, através do descarte de pacotes de forma seletiva e limitando o número de fluxos afetados por este descarte.

Para que uma política de AQM possa ser considerada boa, ela deve também garantir que conexões que compartilham as mesmas condições de rede, tenham um descarte proporcional a sua utilização da fila de pacotes. Além disso, quando perdas múltiplas forem inevitáveis, a política de AQM deve prevenir que estas perdas sejam em rajadas, de forma a permitir a ativação dos mecanismos de recuperação do TCP, evitando, assim, as conseqüências decorrentes de *timeout*. A política de AQM deve, também, penalizar com um número maior de perdas as conexões mal comportadas, ou seja, aquelas que utilizam mais recursos do que deveriam. Além disso, fluxos que transmitem em rajadas não devem ser penalizados, pois transmitem uma grande quantidade de dados em um curto espaço de tempo.

Para maximizar o aproveitamento dos recursos de rede, pacotes que já utilizaram uma grande quantidade de recursos em comparação com outros pacotes, ou que já percorreram um longo caminho devem ter prioridade sobre os outros pacotes, ou seja, devem ser descartados só em último caso. Apesar de desejável, este não é um requisito fácil de se alcançar, pois não é uma tarefa trivial descobrir a quantidade de recursos que um determinado pacote já consumiu, ou o caminho percorrido pelo mesmo.

Outros dois objetivos de desempenho almeçados para políticas de AQM são apresentados em [Hollot et al., 2002]: utilização eficiente da fila e garantias de baixo retardo e baixa variação no retardo. Utilizar eficientemente a fila significa evitar a existência de períodos alternados de transbordo e ociosidade da fila. Quando a fila transborda, pacotes são perdidos, há a necessidade de retransmissões e o tráfego em rajadas é prejudicado. Por outro lado, quando a fila está vazia, o enlace é subutilizado. Para se ter valores de retardo baixos é necessário que o tamanho da fila seja mantido pequeno, o que, em contrapartida, pode gerar a subutilização do enlace. Ademais, é desejável que grandes variações no tamanho da fila sejam evitadas para prevenir a variação do retardo (*jitter*), o que é prejudicial a algumas aplicações.

A principal função dos roteadores é rotear pacotes o mais rápido possível. Sendo assim, um algoritmo de AQM deve ser o mais simples e eficiente possível para não degradar o desempenho da rede. Isto significa que o número de parâmetros, os critérios, a dinâmica do algoritmo e as tarefas a serem realizadas devem ser levadas em consideração na definição do algoritmo.

A política de AQM deve ser definida de forma que o aumento do tráfego ou o número

de conexões não afetem o desempenho provido pelas políticas de descarte de pacotes, ou seja, o crescimento do sistema deve ser permitido sem limitações.

Outro requisito é que a política seja robusta, ou seja, o sistema deve manter-se estável e com desempenho esperado independentemente das condições da rede, que incluem RTT (*Round Trip Time*) e o tipo de tráfego.

De forma resumida, a política de AQM deve prevenir e controlar o congestionamento, ser escalável, simples, justa, evitar o fenômeno de sincronização global e ainda determinar o valor ideal da probabilidade de descarte/marcação que estabiliza o sistema, maximiza as taxas de transmissão e minimiza o tamanho da fila sujeito às condições do meio, de forma a evitar perdas de pacotes desnecessárias.

Embora todos os requisitos apresentados sejam importantes para uma política de AQM, quais devem ser utilizados no projeto de uma política é uma questão que precisa ser analisada e respondida para que uma política efetiva possa ser definida. A resposta para esta questão dependerá basicamente do tipo de tráfego e das necessidades das aplicações que se deseja atender.

3.4 A Política de AQM *Random Early Detection* - RED

Nesta Seção, apresenta-se a política de AQM RED, recomendada pela IETF [Braden et al., 1998] e largamente implementada nos roteadores comerciais.

Proposta em [Floyd and Jacobson, 1993], *Random Early Detection* (RED) é uma política de AQM responsável por detectar e notificar o estado de congestionamento da rede aos TCPs emissores. A detecção do congestionamento incipiente se dá através do monitoramento do tamanho médio das filas de saída no núcleo da rede, então, de forma aleatória, uma conexão é escolhida e notificado. A notificação pode ser feita através do descarte de pacotes ou da marcação de pacotes caso o protocolo de transporte tenha suporte para isto. Ao receber a notificação, o emissor TCP diminui sua taxa de transmissão permitindo, assim, a redução do congestionamento. Além do objetivo principal de prevenir congestionamento, RED foi projetado para levar em consideração os seguintes objetivos: prevenir a sincronização global, prevenir a penalização de tráfego em rajadas e manter um limite máximo para o tamanho médio da fila.

A **sincronização global** [Floyd and Jacobson, 1994] é um fenômeno que ocorre quando todas as conexões são notificadas e reduzem simultaneamente a sua janela de congestionamento, levando a redução da quantidade de dados transmitidos e, conseqüen-

temente, deixando recursos subutilizados.

Em mecanismos que não fazem controle de congestionamento incipiente, quando o fila está na sua capacidade máxima, o **tráfego em rajadas** é simplesmente descartado, e em uma posterior retransmissão estará novamente utilizando recursos da rede, havendo, portanto, a necessidade de gerenciar o tamanho da fila de maneira a permitir a absorção deste tipo de tráfego.

Para prevenir a sincronização global e penalização de tráfego em rajadas, RED escolhe aleatoriamente o pacote a ser marcado/descartado. Desta forma, a probabilidade de que um pacote de uma conexão específica seja marcado/descartado é proporcional a quantidade de banda que a conexão está utilizando.

É necessário **manter um limite máximo para o tamanho médio da fila**, mesmo na presença de tráfego não-adaptativo, para que o atraso da conexão possa ser reduzido. RED realiza este controle, marcando/descartando pacotes que chegam quando o tamanho médio da fila excede um limiar máximo.

3.4.1 Funcionamento da Política RED

O congestionamento incipiente é informado aos emissores através do descarte de pacotes ou ainda por meio da marcação de um *bit* no cabeçalho do pacote. Desta forma, os emissores diminuem a quantidade de dados injetados na rede, contribuindo para a queda na taxa de perda de pacotes. Com menor número de pacotes perdidos, a quantidade de retransmissões será menor, ou seja, a quantidade de dados transmitidos com sucesso (*goodput*) aumenta.

A tomada de decisão sobre a ação a ser tomada em relação a um pacote é definida através do cálculo do tamanho médio da fila (Q_{avg}), e da comparação deste valor com um limiar máximo (max_{th}) e um limiar mínimo (min_{th}). Na Figura 3.1 mostra-se o comportamento do RED, que segue o seguinte procedimento:

- Quando $min_{th} \leq Q_{avg} < max_{th}$, está ocorrendo a prevenção de congestionamento e, portanto, os pacotes são descartados ou marcados com uma probabilidade p_a , que é calculada em função do tamanho médio da fila;
- Quando $Q_{avg} \geq max_{th}$, o algoritmo está na fase de controle de congestionamento e, portanto, todos os pacotes que chegam são descartados;
- Caso $Q_{avg} < min_{th}$, então não há indicativo de que a rede esteja congestionada e todos os pacotes são aceitos.

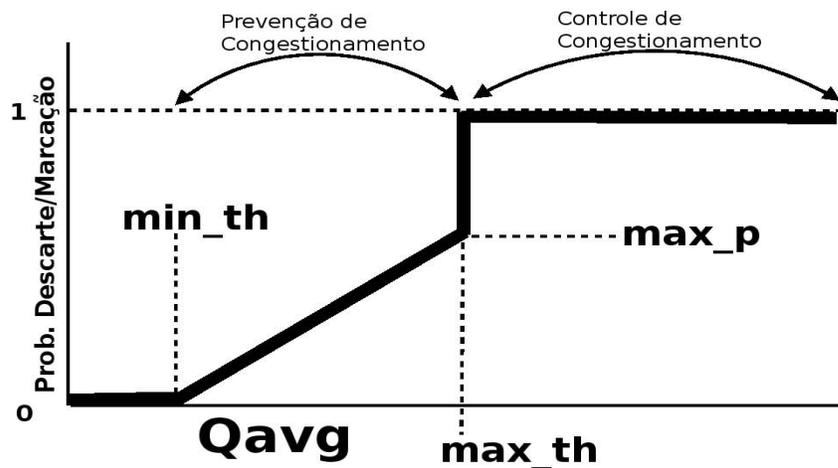


Figura 3.1: Comportamento do RED

O Algoritmo 3 representa o comportamento do RED. Pode-se observar neste algoritmo que para os ajustes do RED são necessários dois algoritmos auxiliares, um para o cálculo do tamanho médio da fila Q_{avg} (linha 2) e outro para o cálculo da probabilidade de descarte p_a (linha 4).

Algoritmo 1 Random Early Detection - RED

```

1: for (Cada pacote que chega) do
2:   Calcula  $Q_{avg}$ 
3:   if ( $min_{th} \leq Q_{avg} < max_{th}$ ) then
4:     Calcula  $p_a$ 
5:     Marque/descarte os pacotes com  $p_a$ 
6:   else if ( $max_{th} \leq Q_{avg}$ ) then
7:     Descarte os pacotes que chegam
8:   end if
9: end for
   { onde:
      $Q_{avg}$  = tamanho médio da fila
      $min_{th}$  = limiar mínimo
      $max_{th}$  = limiar máximo
      $p_a$  = probabilidade de descarte/marcação para um pacote
   }

```

Para obter o tamanho médio da fila, RED utiliza um filtro passa baixa, que é uma média ponderada móvel exponencial (*exponential weighted moving average*) com peso W_q , apresentada no Algoritmo 4. Desta forma, o congestionamento passageiro é filtrado, não tendo influência significativa no incremento do tamanho médio da fila.

O peso da fila (W_q) determina a forma como o algoritmo responde às mudanças no

Algoritmo 2 Calcula Q_{avg}

```

1: if (Fila não estiver vazia) then
2:    $Q_{avg} := (1 - W_q)Q_{avg} + W_q * Q$ 
3: else
4:    $Num_{Pkg} := f(time - Q_{time})$ 
5:    $Q_{avg} := (1 - W_q)^{Num_{Pkg}} * Q_{avg}$ 
6: end if
   { onde:
    $Q_{avg}$  = tamanho médio da fila
    $W_q$  = peso da fila
    $Q$  = tamanho corrente da fila
    $Num_{Pkg}$  = estimativa do número de pacotes transmitidos durante o período em que
   a fila esteve vazia
    $time$  = Valor corrente do tempo
    $Q_{time}$  = período de tempo em que a fila esteve vazia
    $f(t)$  = função linear no tempo t
   }

```

tamanho da fila [Lima and Fonseca, 2002]. Desta forma, quando W_q é muito alto o congestionamento passageiro não é filtrado. Quando W_q é muito baixo a reação ao congestionamento será muito lenta. O valor de W_q é determinado conforme a Inequação 3.1, onde P é o número de pacotes que se deseja suportar nas rajadas. O peso da fila adequadamente escolhido, garante um limite ao tamanho médio da fila, mesmo na presença de tráfego não-adaptativo e, desta maneira, rajadas e congestionamento passageiro são permitidos. Em [Floyd, 1997a] recomenda-se a utilização de $W_q \leq 0.002$ para o peso da fila.

$$P + 1 + \frac{(1 - W_q)^{(P+1)} - 1}{W_q} < min_{th} \quad (3.1)$$

Para encontrar Q_{avg} leva-se, também, em consideração o período em que a fila ficou vazia (*idle period*). Para isso, é necessário estimar o número de pacotes que deveriam ter sido transmitidos (Num_{Pkg}) no período no qual a fila fica vazia. Depois do *idle period* o roteador calcula o tamanho médio da fila como se Num_{Pkg} pacotes tivessem chegado na fila durante o período em que ela esteve vazia.

No Algoritmo 5, mostra-se o cálculo da probabilidade de descarte/marcação. Neste algoritmo, são considerados dois fatores: um contador (*cont*) que é incrementado a cada vez que um pacote chega no roteador, sendo reinicializado quando um pacote é descartado; e uma probabilidade intermediária p_b , com valor variando entre zero e max_p , sendo que max_p é atingido quando o tamanho médio da fila é a igual a max_{th} . A probabilidade de descarte/marcação (p_a) de pacotes é diretamente proporcional ao tamanho médio da fila

Q_{avg} , ou seja, quanto maior o tamanho da fila maior será p_a . Além disso, o descarte de pacotes é proporcional ao uso do canal pela conexão, prevenindo-se, assim, a sincronização global. O valor *default* para o limite superior da probabilidade p_b é $max_p = 0.1$.

Algoritmo 3 Probabilidade

```

1:  $p_b := max_p(Q_{avg} - min_{th}) / (max_{th} - min_{th})$ 
2:  $p_a := p_b / (1 - cont * p_b)$ 
   { onde:
      $p_a$  = probabilidade de descarte/marcação para um pacote
      $p_b$  = função linear do tamanho médio da fila, fica entre 0 -  $max_{th}$ 
      $max_p$  = máximo  $p_b$  quando  $Q_{avg} = max_{th}$ 
   }

```

3.4.2 Avaliação de RED

Desde que foi introduzido em [Floyd and Jacobson, 1993], RED tem sido largamente estudado, e ajustes de parâmetros [Floyd, 1997a][Floyd, 2000] e modificações[Feng et al., 1999] [Lin and Morris, 1997] tem sido propostos para melhorar o desempenho desta política de AQM.

Uma das propostas relacionadas a ajustes de parâmetros, e que tem se destacado, é a utilização de RED com a versão “gentle”[Floyd, 2000]. Em RED com a versão “gentle” habilitada, a variação da probabilidade de descarte/marcação p_a é modificada. Quando o tamanho médio da fila varia entre min_{th} e max_{th} , p_a varia linearmente entre 0 e max_p . Quando o tamanho médio da fila varia entre max_{th} e $2 * max_{th}$, p_a varia linearmente entre max_p e 1. Com a opção “gentle”, o comportamento de RED é melhorado, dando maior robustez a política de AQM em relação aos valores definidos para os parâmetros max_{th} e max_p .

RED possui um conjunto de vantagens que a levaram a ser a política de AQM recomendada e mais implementada nos roteadores. No entanto, apresenta algumas desvantagens, as quais são discutidas na literatura. Estas vantagens e desvantagens de RED são apresentadas a seguir.

Vantagens:

- **Eficiência na prevenção de Congestionamento:** se W_q for ajustado de forma correta, RED garante um limite no tamanho médio da fila, garantindo, assim, que rajadas e congestionamento passageiros sejam permitidos;

- **Prevenção da sincronização global:** com RED, a taxa de exclusão de pacotes aumenta em função do aumento do nível de congestionamento e a probabilidade de um pacote de um determinado fluxo ser descartado é proporcional ao uso do enlace por este fluxo;
- **Boa razão vazão/retardo:** RED consegue controlar o tamanho médio da fila enquanto acomoda congestionamento passageiro, garantindo, assim, alta vazão e baixo retardo;
- **Reage bem ao tráfego em rajada:** ao garantir um limite para o tamanho médio da fila, RED suporta, também, tráfego em rajadas;
- **Identifica conexões mal comportadas:** como o descarte é feito de forma aleatória e proporcional a utilização da banda passante pelo fluxo, é possível identificar os fluxos com maior quantidade pacotes descartados, ou seja, os fluxos mal comportados;
- **Utilização de operações simples e pouca memória:** RED utiliza apenas variáveis globais, não armazenando informações para cada fluxo, o que faz com que ele gaste pouca memória. Ademais, o algoritmo utiliza operações simples, sendo, assim, rápido;
- **Implementação gradual:** a utilização de RED não implica na necessidade de alterações no protocolo de transporte, tão pouco exige que todos os roteadores o implementem, ou seja, ele pode ser implementado de forma gradual. Atualmente, a maioria dos roteadores comerciais implementa RED.

Problemas e desvantagens:

- **Injustiça:** Um dos grandes problemas de RED é a injustiça. A seguir são apresentadas as deficiências de RED que acarretam este problema:
 - **Penalização de conexões TCP com janelas pequenas:** O grau de tolerância a perda de pacotes é diretamente relacionado com o tamanho da janela. Enquanto que conexões com uma grande janela se recuperam de perdas em apenas um RTT, uma conexão com janela pequena precisa esperar por um *timeout* relativamente grande para se recuperar de apenas uma perda. Ao escolher um pacote de forma aleatória, RED pode ter escolhido um pacote de uma conexão com janela pequena, conseqüentemente, esta conexão

sofre uma penalização muito maior do que uma conexão que apresenta janela grande [Lin and Morris, 1997];

- **Orientação à tráfegos adaptativos:** O TCP diminui a sua taxa de transmissão na presença de congestionamento. Em contrapartida, a existência de tráfego não-adaptativo, que envia altas taxas independentemente da existência de perdas, faz com que a taxa de descarte aumente em todas as conexões e, conseqüentemente, as conexões adaptativas reduzem sua taxa de transmissão enquanto as não-adaptativas tomam uma maior quantidade de banda passante [Lin and Morris, 1997];
 - **Descarte não-proporcional:** Quando um pacote de uma nova conexão é aceito, a probabilidade de descarte dos futuros pacotes a serem transmitidos pelas outras conexões que já estavam ativas aumenta, mesmo que estas conexões consumam menos banda. Isto pode causar, temporariamente, o descarte não proporcional mesmo entre tráfegos idênticos.
- **Dificuldade em determinar corretamente os limiares min_{th} e max_{th} :** Os valores adequados dependem do valor do tamanho médio de fila que se deseja manter, o qual é dependente do tipo de tráfego. Como dificilmente se tem, a priori, as características do tráfego sob o qual se estará sujeito, torna-se uma tarefa complicada determinar quais os valores ótimos dos limiares. Se o tráfego for em rajadas, o valor de min_{th} deve ser alto o suficiente para manter uma alta taxa de utilização da rede. O valor de max_{th} deve ser determinado de acordo com o valor médio máximo de atraso permitido nos roteadores. A determinação deste valor depende do tipo de aplicação a ser transportada na rede. Se o valor de max_{th} for muito alto, aplicações sensíveis ao atraso serão severamente penalizadas. Foi comprovado, empiricamente, que o valor de max_{th} deve ser igual a pelo menos três vezes o valor de min_{th} [Floyd, 1997a]. Além disso, a diferença entre min_{th} e max_{th} deve ser maior que o incremento típico do cálculo da média em um RTT para que RED funcione mais eficientemente, como também para evitar a sincronização global. O ajuste dos parâmetros é o argumento mais usado para a não utilização de RED, já que uma má escolha dos parâmetros pode levar RED a um comportamento inadequado. O ajustes inadequados dos parâmetros podem, ainda, levar o desempenho de RED a ser similar ao de *DropTail*. Por este motivo, alguns estudos desaconselham a utilização de RED [Bonald et al., 2000][May et al., 1999].

- **Grande quantidade de fluxos ativos:** Quando uma grande quantidade de fluxos estão ativos, o tráfego agregado é composto, basicamente, por rajadas, o que degrada o comportamento de RED, dado que o tamanho médio da fila varia rapidamente antes que o mecanismo de RED possa reagir. Dependendo da explosividade das rajadas e da quantidade de *buffers* disponíveis no enlace, podem ocorrer várias perdas, causando subutilização do enlace [Feng, 1999]. Uma solução proposta para resolver este problema é utilizar *buffer* de tamanho duas vezes o produto banda-atraso. O grande inconveniente é que quando o produto banda-atraso é elevado, a utilização do dobro deste produto, elevará tanto o atraso fim-a-fim quanto a variação do retardo (*jitter*), o que é prejudicial as aplicações iterativas.

Quando o número de fluxos ativos é muito grande, um outro problema é a possibilidade de sincronização global. A probabilidade de descarte de RED é uma função linear do tamanho médio da fila, Q_{avg} , e como este valor varia muito quando há um grande número de fluxos ou conexões ativas, a probabilidade de descarte varia, consideravelmente, em um curto espaço de tempo fazendo com que RED falhe em marcar os pacotes de forma aleatória, não sendo tão eficiente em prevenir o fenômeno de sincronização global.

- **Ponto operacional ideal:** RED atinge um ponto operacional ideal, somente quando há *buffers* suficientes e os limiares estão definidos de forma adequada.

3.5 Políticas de AQM Baseadas em Teoria de Controle

Embora seja a política de AQM recomendada pela IETF, a utilização de RED tem sido amplamente desaconselhada devido às suas limitações. A principal limitação de RED está relacionada a correta determinação de seus parâmetros, de modo a atingir um ponto operacional ideal. Uma definição inadequada destes parâmetros pode degradar seriamente o desempenho de RED, levando a um comportamento similar ou pior ao de *DropTail*. Além disso, diante de uma grande quantidade de fluxos ativos, o tráfego agregado é em rajadas, o que degrada o comportamento de RED, pois o tamanho médio da fila varia rapidamente, causando grandes variações de retardo e baixa utilização da banda passante. Quando a probabilidade de descarte sofre grande variação em um curto intervalo de tempo, a aleatoriedade na marcação dos pacotes é degradada e RED não consegue evitar o fenômeno de sincronização global.

Com o intuito de superar as dificuldades de determinar os parâmetros de RED, estudos baseados em heurísticas e simulações vêm sendo realizados [Floyd and Jacobson, 1993, Feng, 1999, Floyd, 1997a, Floyd, 1997b, Floyd, 2000, Feng et al., 1999, May et al., 1999]. No entanto, nenhum destes garantem que um ponto de equilíbrio seja atingido, nem garantem estabilidade do tamanho da fila, que é de fundamental importância para evitar variações de retardo e baixa utilização de enlace. Existem, ainda, estudos para derivar configurações de RED de uma forma mais sistemática [Bonald et al., 2000, Firoiu and Borden, 2000, Kuusela et al., 2001].

Para se projetar e desenvolver políticas de AQM que garantam estabilidade em torno do ponto de equilíbrio, técnicas de Teoria de Controle têm sido utilizadas como uma alternativa promissora.

Nas políticas de AQM baseadas em Teoria de Controle, o controle de congestionamento é visto como um sistema de controle de retroalimentação, no qual a taxa de transmissão dos emissores é ajustada em função do nível de congestionamento, inferido pelos roteadores através da monitoração da ocupação das filas e retornado para os emissores através de descarte/marcação de pacotes. Desta forma, a função do controlador é calcular a probabilidade de descarte/marcação adequada que estabiliza o tamanho da fila independentemente das condições do meio.

O crescimento em pesquisas de projetos de mecanismos de controle de congestionamento para a Internet utilizando abordagens analíticas tem sido significativo, resultando em progresso expressivo na modelagem matemática destes mecanismos [Low and Srikant, 2004, Srikant, 2004]. Ferramentas de Teoria de Controle têm representado um papel fundamental para o desenvolvimento da Teoria de Controle de Congestionamento para a Internet [Liu et al., 2003].

A seguir, são apresentados alguns trabalhos relacionados a projeto de mecanismos de AQM estáveis, desenvolvidos a partir de Teoria de Controle.

As políticas de AQM baseadas em Teoria de Controle consideram a natureza intrínseca de retroalimentação do sistema de controle de congestionamento. Busca-se compreender a dinâmica do congestionamento, para poder garantir estabilidade e robustez das condições de equilíbrio das métricas de desempenho desejadas. A função dos controladores é determinar a probabilidade de descarte/marcação adequada, que maximize a vazão e minimize as perdas e que garanta a estabilidade do sistema independentemente das variações das condições do meio. A estabilidade é importante em sistema de AQM para evitar a variação de retardo e baixa utilização da banda passante.

A maioria das políticas de AQM baseadas em Teoria de Controle utilizam apenas informações da dinâmica atual do sistema, ou seja, desconsideram os termos dependentes de atraso. Algumas exceções são apresentadas em [Lima, 2005, Kim et al., 2003a, Kim and Low, 2002a, Agrawal and Granelli, 2004, Kim et al., 2003b], e a política HSTCP-H2-AQM desenvolvida neste trabalho. O uso de controle independente de atraso possui limitações de desempenho na presença de grandes atrasos, quando comparado com o controle dependente de atraso. No contexto de redes, isto significa que a utilização da abordagem dependente do atraso ou não-razional faz com que os recursos de rede possam ser usados eficientemente [Hollot et al., 2002].

As políticas de AQM desenvolvidas baseadas em Teoria de Controle utilizam, na sua maioria, controladores clássicos tais como controladores P (*Proportional*), I (*Integral*), PI (*Proportional Integrator*), PD (*Proportional-Derivative*) ou PID (*Proportional-Integral-Derivative*). Os demais trabalhos utilizam outras ferramentas de controle tais como compensação de retroalimentação e apenas um pequeno número de propostas utilizam técnicas de controle ótimo ou robusto [Lima, 2005, Agrawal and Granelli, 2004, Vinnicombe, 2002, Peng et al., 2005].

Em [Hollot et al., 2001a] foi apresentada uma investigação analítica de RED através da linearização do modelo do sistema de congestionamento que captura o comportamento da variação da janela do TCP em função da variação do tamanho da fila [Misra et al., 2000]. Verificou-se que RED é um controlador do tipo P e, a partir desta análise, configurações adequadas para os parâmetros de RED, que levam o sistema à estabilidade, puderam ser determinadas de forma sistemática.

A política PI -AQM, *Proportional Integrator* AQM [Hollot et al., 2001b], utiliza um controlador do tipo PI , e foi desenvolvida utilizando o modelo dinâmico para o comportamento do sistema de congestionamento TCP/AQM.

A política de AQM PD -RED introduzida em [Jinsheng et al., 2003], visa melhorar RED fazendo alterações mínimas no seu algoritmo através do uso de um controlador do tipo PD . A proposta é adaptar o parâmetro max_p de RED, que determina o valor máximo para a probabilidade de descarte/marcação, de modo a estabilizar o tamanho da fila.

Introduzida em [Kim et al., 2003b], PD -AQM é uma política de AQM que considera a dependência de atraso e utiliza compensação de retroalimentação. PD -AQM utiliza uma estrutura de memória de controle para fazer as compensações necessárias decorrentes da dependência do atraso nas medidas de congestionamento obtidas.

Adicionar controle do tipo I ou D pode aumentar de forma significativa o desempenho

do controladores do tipo P . O controle do tipo I elimina o erro em torno do ponto de equilíbrio, e o controle do tipo D aprimora a estabilidade e determina como uma pequena perturbação na entrada afeta a saída do sistema, ou seja, há um controle da variação em torno do ponto de equilíbrio. Controladores P e I atuam somente sobre erros passados, sem capacidade para prever erros futuros, o que limita o desempenho de controladores do tipo PI . Por outro lado, o controle do tipo D , que prevê erros futuros, podendo tomar medidas corretivas antes que estes erros aconteçam, é utilizado para aprimorar o tempo de resposta. Desta forma, a utilização conjunta destes tipos de controle possibilitam a obtenção de políticas de AQM que tenham um tempo de resposta mínimo e garantam estabilidade em torno da fila.

Em [Xidong et al., 2003], o modelo de congestionamento utilizado para analisar RED e para derivar PI-AQM é estendido para derivar três plantas distintas: a primeira considera a existência de suporte à ECN, um mecanismo que permite que o congestionamento seja notificado sem, no entanto, descartar o pacote; a segunda considera que a rede não apresenta suporte a ECN; e a terceira planta considera a existência de fluxos não-adaptativos, ou fluxos que não diminuem a taxa de transmissão diante de congestionamento. Estas plantas são analisadas e um controlador do tipo PID é desenvolvido.

A política RHA (*Recending Horizon AQM policy*), a qual considera a dependência de atraso em seu desenvolvimento, faz a compensação do atraso nas medidas de congestionamento obtidas utilizando uma estrutura de memória de controle [Kim and Low, 2002a]. Então, é derivado a versão ótima de RHA [Kim and Low, 2002b] que estabiliza o sistema de congestionamento com custo mínimo.

A política VS-AQM (*Variable Structure AQM*) [Peng et al., 2005] é derivada utilizando a técnica de controle robusto SMVS - *Sliding Mode Variable Structure Control*. VS-AQM foi projetada diretamente a partir do modelo não linear do sistema de congestionamento TCP/AQM. Assim, o controlador não terá que lidar com incertezas do modelo decorrentes do processo de linearização.

Em [Lima, 2005], a política H2-AQM, baseada no TCP Reno, é derivada utilizando técnicas de controle ótimo. A vantagem em utilizar controladores ótimos é que os objetivos de desempenho desejados para o sistema são considerados no projeto do controlador, e uma vez determinados estes objetivos são atendidos de forma ótima. Outra política de AQM derivada utilizando teoria de controle ótimo é dsH2-AQM [Augusto et al., 2007], a qual é, também, baseada no sistema de controle de congestionamento TCP/AQM, mas diferentemente de H2-AQM, dsH2-AQM é utilizada para prover qualidade de serviço em

redes.

3.6 ECN - Explicit Congestion Notification

Com a utilização de mecanismos de gerenciamento ativo de filas, AQM, o congestionamento é detectado antes que as filas estejam cheias e é, então, notificado aos emissores através do descarte de pacotes. O detecção/notificação de congestionamento incipiente evita os problemas apresentados por *DropTail* e ainda garante um tamanho médio de fila baixo, podendo, assim, absorver tráfego em rajadas, diminuir o atraso de pacotes e evitar o problema de sincronização global.

A notificação de congestionamento pelas políticas de AQM não necessariamente precisa ser realizada por meio de descarte de pacotes. Ao invés de notificar a existência de congestionamento ao emissor por meio de um pacote descartado, pode ser utilizado um mecanismo onde um *bit* do cabeçalho do pacote é utilizado para sinalizar o congestionamento. Este mecanismo é denominado *Explicit Congestion Notification* (ECN) [Floyd, 1994, Ramakrishnan and Floyd, 1999, Ramakrishnan et al., 2001]. Quando o roteador verifica que há congestionamento, ele simplesmente sinaliza o congestionamento através da marcação do pacote ao invés de descartá-lo, e caso não exista suporte a ECN, o pacote é simplesmente descartado [Lima, 2005].

Capítulo 4

Introdução a Teoria de Controle e Análise de Sistemas Dinâmicos e Controle

Neste capítulo, são introduzidos conceitos e definições de modelagem matemática e teoria de controle necessários para a compreensão do desenvolvimento deste trabalho.

4.1 Conceitos e Definições

A teoria de controle lida com o comportamento de sistemas dinâmicos, os quais evoluem ao longo do tempo em função de regras matemáticas. O emprego de teoria de controle em sistemas tem desempenhado papel fundamental no avanço tecnológico. Através dela é possível maximizar ou minimizar certas medidas, de modo a atingir desempenho ótimo em sistemas dinâmicos.

A teoria de controle é subdividida em três fases. A fase inicial dá-se através do *Controle Clássico*, o qual consiste de métodos no domínio de frequência e tem como base as teorias de Bode, Nyquist e outros. Além disso, a teoria de controle clássico é baseada em métodos gráficos para sintetizar controladores, sendo, portanto, heurística.

Para suprir as limitações do controle clássico surge a *Teoria de Controle Moderna*, a qual utiliza formalismo matemático, emprega equações diferenciais e representa o sistema na forma de estado de espaço. Desta forma, o controlador pode ser formulado como um problema de otimização. Apesar do formalismo para o desenvolvimento de controladores eficientes, o controle moderno é freqüentemente sensível a erros na modelagem, o que levou

ao desenvolvimento do *Controle Pós-moderno* (década de 80 e 90) também chamado de método para síntese de controladores ótimos.

Quando nos referimos a sistemas de controle algumas definições são necessárias [Ogata, 1998]:

- A **variável controlada** é a grandeza ou a condição que é medida e controlada. A **variável manipulada** é a grandeza ou condição variada pelo controlador de modo a afetar o valor da variável controlada;
- O **valor de referência** é o valor desejado da variável a ser controlada;
- Quando nos referimos a sistemas com retroalimentação, **controlar** significa medir o valor da variável controlada e aplicar o valor conveniente da variável manipulada ao sistema de modo a corrigir ou limitar o desvio entre o valor medido e o valor desejado da variável controlada;
- **Planta** é qualquer objeto físico a ser controlado;
- **Processo** é uma operação que evolui progressivamente e se constitui em uma série de ações controladas ou de movimentos sistematicamente dirigidos para alcançar um determinado resultado ou meta;
- **Sistema** é um conjunto de componentes interconectados harmonicamente para realizar um certo objetivo. A interação em um sistema se dá por meio de uma variável de entrada (controle), uma variável de saída (manipulada) e a relação entre elas (função de transferência);
- Quando inserido em um sistema, o **distúrbio** tende a afetar de modo adverso o valor da variável de saída de um sistema;
- O **controlador** manipula o sinal de erro, gerando um sinal de controle que será aplicado ao sistema, corrigindo o valor da variável manipulada;
- **Controle com retroalimentação** refere-se ao processo que, em presença de distúrbios, tende a reduzir a diferença entre o sinal de saída e o sinal de referência de um sistema. O controle opera com base nesta diferença. Apenas os distúrbios não conhecidos a priori são considerados, uma vez que as perturbações conhecidas podem ser compensadas no sistema. A retroalimentação deve atender aos requisitos de estabilidade, desempenho e robustez para o sistema controlado.

- O **ponto de equilíbrio** é um ponto onde o estado do sistema está em equilíbrio e não se move.

Projetar um sistema de controle é um processo que envolve um número de escolhas e decisões que dependem das propriedades do sistema a ser controlado, bem como dos requisitos a serem satisfeitos pelo sistema controlado [Lima, 2005]. Tais decisões requerem que seja garantido o compromisso entre alguns requisitos conflitantes. O projeto de sistemas de controle visa, basicamente, aprimorar o desempenho do sistema de forma a garantir que os seguintes requisitos sejam atendidos:

- **Estabilidade:** refere-se ao quão bem comportado é o sistema controlado; ela é definida de acordo com o ponto de equilíbrio dado. Um sistema é estável se quando o vetor de estado se afasta ligeiramente do ponto de equilíbrio, ele tende a retornar ao ponto ou, ao menos, não continuar se afastando dele;
- **Boa Resposta Transitória:** refere-se a habilidade do controlador em garantir que o sistema apresente uma boa resposta transitória, que traduz o curto tempo de resposta que um sistema deve apresentar frente a ruídos ou perturbações para voltar ao ponto de equilíbrio;
- **Erro nulo ou baixo:** refere-se a habilidade do controlador em conseguir para o sistema uma saída muito próxima do valor desejado. Isto implica em dizer que o sistema deve apresentar erro nulo ou baixo;
- **Robustez:** é a habilidade do controlador em permanecer estável diante de alterações imprevistas.

4.2 Modelagem Matemática

Para analisar as características de um sistema dinâmico, há necessidade de um modelo matemático. Esta modelagem matemática é composta por um conjunto de equações diferenciais que representam a dinâmica do sistema.

A escolha do modelo matemático depende das particularidades do sistema que se deseja modelar. Estas particularidades determinam se o sistema a representar é linear ou não-linear.

Um **sistema** é **linear** se ele atende ao princípio da superposição. Neste princípio, as respostas produzidas pela aplicação simultânea de duas excitações diferentes é igual à

soma das suas respostas individuais a cada uma das excitações. No **sistema não-linear**, a resposta a duas entradas não pode ser calculada tratando-se uma entrada de cada vez e adicionando-se os resultados.

Na maioria dos casos os problemas reais são não-lineares. A determinação de soluções quando se tem não-linearidade é extremamente complicada. Por este motivo, estes sistemas são **linearizados** de modo a obter um sistema linear aproximado, ao qual se pode aplicar métodos lineares que produzem informações sobre o comportamento do sistema não-linear.

Em teoria de controle, as relações entre entradas e saídas de um sistema são denominadas **função de transferência**. A função de transferência, definida em (4.1), de um sistema representado por equações diferenciais lineares e invariantes no tempo, é definida como a relação entre a transformada de Laplace do sinal de saída (função resposta) e a transformada de Laplace do sinal de entrada (função de excitação), em que as condições iniciais são nulas [Ogata, 1998] [Distefano et al., 1972].

$$G(s) = \frac{\mathcal{L}(Saida)}{\mathcal{L}(Entrada)} \Big|_{Condições\ Iniciais=0} \quad (4.1)$$

Uma vez estabelecida, a função de transferência fornece uma descrição completa das características dinâmicas do sistema.

4.3 Modelagem no Estado de Espaço

A Teoria de Controle Moderno é baseada no conceito de estado, onde, o sistema é representado na forma de estado espaço.

O **estado** de um sistema dinâmico é o menor conjunto de valores de variáveis de estado de modo que o conhecimento destes valores em $t = t_0$, junto com o conhecimento dos valores de entrada para $t \geq t_0$, determina completamente o comportamento do sistema em qualquer instante $t \geq t_0$.

As **variáveis de estado** de um sistema dinâmico são as grandezas cujo conjunto de valores determina o estado do sistema. As variáveis necessárias para descrever completamente o comportamento de um sistema, são as componentes do vetor de estado \mathbf{x} . Um **vetor de estado** determina univocamente o estado $\mathbf{x}(t)$ para qualquer instante $t \geq t_0$, uma vez conhecido o estado em $t = t_0$ e a função de entrada $\mathbf{u}(t)$ para $t \geq t_0$.

O espaço n -dimensional cujos eixos coordenados consistem nos eixos x_1, x_2, \dots, x_n é chamado **estado de espaço**. Qualquer estado pode ser representado por um ponto no

estado de espaço.

A análise no estado de espaço envolve as variáveis de entrada, de saída e variáveis de estado na modelagem de sistemas dinâmicos. Um sistema pode ter várias representações no estado de espaço, mas em todos o número de variáveis de estado é o mesmo.

Seja o sistema com vetor de entrada \mathbf{u} de dimensão r , vetor de saída \mathbf{y} , m -dimensional e o vetor de variáveis de estado \mathbf{x} , n -dimensional. Em (4.2) e em (4.3) tem-se a representação da equação de estado e da equação de saída, respectivamente.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (4.2)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \quad (4.3)$$

Linearizando-se estas equações em torno do ponto de equilíbrio, obtém-se as equações lineares para o estado e a saída, respectivamente:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_0 \mathbf{x}_t + \mathbf{B} \mathbf{u}(t) \quad (4.4)$$

$$\mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) + \mathbf{D} \mathbf{u}(t) \quad (4.5)$$

onde \mathbf{A}_0 é a matriz de estado, $\mathbf{B} \mathbf{u}$ é a matriz de entrada, $\mathbf{C} \mathbf{x}$ é a matriz de saída e $\mathbf{D} \mathbf{u}$ é a matriz de transmissão direta.

4.4 Controladores Ótimos

O conceito do que é um bom controlador depende dos objetivos definidos para o sistema específico em questão. Entretanto, alguns pontos tais como as características do processo a ser controlado e as restrições impostas pelos requisitos de operação, sempre devem ser considerados. Além destas considerações, é necessário também estabelecer limitações aos valores mínimo e máximo para entradas e saídas controladas, limitar a variação da taxa máxima de entrada, e ainda as saídas não diretamente controladas.

Estas metas devem ser explicitamente modeladas como uma expressão matemática, a qual é chamada de função objetivo ou de custo. Desta forma, o comportamento do sistema desejado é formulado em termos de critérios de custo a serem minimizados.

O conhecimento do estado inicial e a correta modelagem do sistema são fundamentais para a exatidão do controlador [Willigenburg, 2003]. Assim, se o modelo e o estado inicial estão bem definidos e o desempenho do controlador atende aos objetivos desejados, então, diz-se que o controlador é ótimo.

A estratégia *Linear Quadratic Optimal Control*, comumente chamada de H_2 , é um controlador ótimo que objetiva minimizar a magnitude do erro na saída do sistema. Considera-se como **erro** a diferença entre uma saída esperada e uma saída controlada obtida.

Controladores ótimos não são sempre tolerantes as alterações no sistema controlado ou no ambiente, ou seja, a presença de distúrbios e incertezas no modelo podem degradar o desempenho do controlador.

Em diferentes aplicações, é necessário garantir que possíveis perturbações de controle não influenciem, de maneira proibitiva, as variáveis de interesse sob controle.

A busca por soluções de sistemas que envolvem incertezas é denominada controle robusto. O controlador robusto H_∞ objetiva minimizar o pior erro que pode resultar de alguma entrada mantendo, assim, a estabilidade do sistema, garantia de alto desempenho e degradação suave diante de alterações no sistema. Desta forma, o H_∞ provê um controle mais rigoroso do que o H_2 .

Os controladores robustos consideram os problemas de incertezas causadas por falhas no modelo do sistema e distúrbios externos, tentando ajustar o controlador aos requisitos desejados para o sistema à todos as situações que possam ocorrer. Diante destes ajustes, para amenizar a degradação, o objetivo é identificar qual a faixa de valores em que o controlador permanece estável e com o desempenho requerido, independentemente das alterações no sistema ou do ambiente.

As técnicas de Controle Robusto, H_∞ , são muito mais rigorosas do que as do H_2 . No entanto, devem ser utilizadas de maneira cuidadosa, pois o projeto para o pior caso pode levar a degradação do desempenho do sistema sob condições normais, principalmente, quando o sistema opera a maior parte do tempo sob estas condições.

4.5 Desigualdades Matriciais Lineares

Desigualdades Matriciais Lineares ou LMIs *Linear Matrix Inequalities* possuem a forma:

$$F(x) \triangleq F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (4.6)$$

onde: $x \in R^m$ é a variável e as matrizes simétricas $F_i = F_i^T \in R^{n \times m}$, $i = 0, \dots, m$ são dadas.

A desigualdade significa que $F(x)$ é definida positiva, o que implica que $u^T F(x) u > 0$ para todo u diferente de zero e $u \in R^n$. A LMI (4.6) representa um problema de restrição

convexa sobre x , ou seja, o conjunto $x|F(x) > 0$ é convexo [Boyd et al., 1981]. Tal representação pode indicar uma grande variedade de restrições convexas sobre x .

A existência de programas computacionais especializados que convergem em tempo polinomial para a resolução de LMIs implica que formular um problema em termos de LMIs equivale basicamente a resolver o problema. Várias áreas do conhecimento tem usado LMIs para formular e resolver problemas numericamente; em particular, vários problemas de Teoria de Controle podem ser expressos na forma de LMIs.

Em [Oliveira and Geromel, 2004], a síntese de controladores para sistemas lineares com atraso são expressos e solucionados como LMIs, possibilitando, assim, que os parâmetros do controlador possam ser obtidos através da resolução de um problema convexo simples, que pode ser resolvido de forma eficiente.

Capítulo 5

Sistema de Controle de Congestionamento HSTCP/AQM

O *High Speed* TCP, um dos primeiros TCP's variantes propostos na literatura, é um protocolo projetado para obter altas taxas de vazão em redes com produto banda-atraso elevado. Para isso, ele apresenta uma função de resposta modificado sem, no entanto, deixar de ser compatível com o TCP. Vários estudos tem sido realizados, no intuito de verificar as vantagens e as vulnerabilidades do HSTCP [Li et al., 2007, Zhu et al., 2005, Souza and Agarwal, 2003, Ha et al., 2006, Barman et al., 2004, Long Chengnian and Xinping, 2006]. Considerando as características e o desempenho deste protocolo em redes com produto banda-atraso elevado, o HSTCP é o mecanismo de controle de congestionamento TCP utilizado na derivação do controlador ótimo deste trabalho.

Neste capítulo, apresenta-se o sistema de controle de congestionamento como um problema de controle. O comportamento da janela do HSTCP e do tamanho da fila é capturado e representado por uma simplificação do modelo dinâmico do sistema de congestionamento HSTCP/AQM. O modelo é, então, linearizado para que se possa fazer o projeto e a análise de políticas de AQM, de maneira eficiente. Em seguida, a dinâmica é analisada em função dos parâmetros de rede e o problema de controle de congestionamento é descrito como um problema de controle.

5.1 Modelo dinâmico para o Controle de Congestionamento HSTCP/AQM

O comportamento da janela de congestionamento do TCP em função da variação da fila é capturado por meio de um modelo dinâmico baseado em análise de fluidos e equações diferenciais estocásticas apresentado em [Misra et al., 2000] e [Hollot et al., 2002].

Em [Sonkoly et al., 2005] o modelo apresentado em [Misra et al., 2000] e [Hollot et al., 2002] é generalizado para redes que usam mecanismo de controle de congestionamento *High Speed* TCP. Desta forma, o comportamento da janela do HSTCP, da fila e do RTT é modelado pelas seguintes equações diferenciais:

$$\dot{W}(t) = \frac{a(t)}{R(t)} - b(t)W(t) \frac{W(t - R(t))}{R(t - R(t))} p(t - R(t)); \quad (5.1)$$

$$\dot{q}(t) = \frac{N(t)W(t)}{R(t)} - C + \omega_q(t); \quad (5.2)$$

$$R(t) = \frac{q(t)}{C(t)} + T; \quad (5.3)$$

onde:

$W(t)$ é o tamanho médio da janela HSTCP em pacotes;

$q(t)$ é o tamanho médio da fila em pacotes;

$R(t)$ é o *round trip time* (RTT) em segundos;

$a(t)$ é o parâmetro de crescimento da janela de congestionamento, o qual leva em consideração o valor da janela atual $W(t)$;

$b(t)$ é o parâmetro de redução da janela de congestionamento, o qual leva em consideração o valor da janela atual $W(t)$;

$p(t)$ é a probabilidade de descarte/marcação de pacotes;

$N(t)$ é o número de conexões HSTCP;

C é a capacidade do enlace em pacotes/segundo;

$\omega_q(t)$ é o ruído produzido pelos fluxos UDP;

T é o atraso de propagação em segundos.

Este sistema de equações descreve o comportamento AIMD do *High Speed* TCP na fase de *Congestion Avoidance*. O Modelo é consideravelmente preciso para tráfego de longa duração, onde a fase se *Slow Start* é curta podendo, assim, ser desconsiderada.

A equação (5.1) representa a dinâmica da janela de congestionamento, onde o primeiro termo caracteriza o crescimento aditivo e o segundo termo representa a redução

multiplicativa. A Equação (5.2), representa a variação do tamanho da fila em função do tempo, que é definida pela diferença entre a taxa de chegada de pacotes $\frac{N(t)W(t)}{R(t)} + \omega_q(t)$ e a capacidade do enlace C . A Equação (5.2) difere da equação para a modelagem do comportamento da fila apresentada em [Sonkoly et al., 2005] pela inclusão do termo $\omega_q(t)$, que corresponde ao impacto dos fluxos UDP na variação da fila, permitindo uma abordagem mais realista do ambiente de rede, dado que os fluxos UDP são não-adaptativos, ou seja, não diminuem sua taxa de transmissão diante de congestionamento.

5.2 Linearização do Sistema

Neste seção, o sistema de equações (5.1-5.3), que representa o sistema de congestionamento, é linearizado. A modelagem do sistema de congestionamento e a linearização do modelo permitem que se possa fazer o projeto e a análise de políticas de AQM de maneira mais eficiente.

Para linearizar o sistema (5.1-5.3), considera-se o número de conexões HSTCP, $N(t) \equiv N_0$, a capacidade do enlace, $C(t) \equiv C_0$, e o ruído, $\omega_q(t) = \omega_{q_0}$ como constantes. Considera-se ainda que os fatores de crescimento e redução da janela do HSTCP são constantes e dados, respectivamente, por $a(t) \equiv a_0$ e $b(t) \equiv b_0$.

Seja (W, q) o estado do sistema a ser controlado (5.1-5.3), e p a entrada do sistema. O ponto de equilíbrio $(W_0, q_0$ e $p_0)$ é definido como o ponto no qual a taxa de transmissão é máxima e a taxa de perda é mínima. O ponto de equilíbrio é obtido através da solução de $\dot{W}(t) = 0$, $\dot{q}(t) = 0$ e $R_0 = T + \frac{q_0}{C_0}$. Desta forma, o ponto de equilíbrio é dado por:

$$W_0 = \sqrt{\frac{a_0}{b_0 p_0}} = \frac{R_0 C_0}{N_0} = \frac{C_0 T + q_0}{N_0}; \quad (5.4)$$

$$q_0 = C_0(R_0 - T) = N_0 \sqrt{\frac{a_0}{b_0 p_0}} - C_0 T; \quad (5.5)$$

$$p_0 = \frac{a_0 N_0^2}{b_0 (R_0 C_0)^2} = \frac{a_0 N_0^2}{b_0 (C_0 T + q_0)^2}; \quad (5.6)$$

No processo de linearização do sistema, foram ignoradas as dependências em relação ao atraso dos argumentos $(t - R)$, no cálculo da variação da fila. Assume-se que este atraso é fixo e igual a $(t - R_0)$. Desta forma, a dinâmica simplificada do comportamento do HSTCP é representada conforme segue:

$$\begin{aligned}\dot{W}(t) &= \frac{a}{\frac{q(t)}{C_0} + T} - bW(t) \frac{W(t - R(t))}{\frac{q(t - R_0)}{C_0} + T} p(t - R(t)) \\ \dot{q}(t) &= \frac{N(t)W(t)}{\frac{q(t)}{C_0} + T} - C\end{aligned}\quad (5.7)$$

Para a linearização do sistema (5.7) são definidas as funções $f = \dot{W}(t)$ e $g = \dot{q}(t)$ e avaliadas as suas derivadas parciais no ponto de equilíbrio (W_0, q_0, p_0) . Considerando $W = W(t)$, $W^* \doteq W(t - R_0)$, $q = q(t)$, $q^* \doteq q(t - R_0)$ e $p^* \doteq p(t - R_0)$, as funções f e g são definidas por:

$$f(W, W^*, q, q^*, p^*) \doteq \frac{a}{\frac{q(t)}{C_0} + T} - bW \frac{W^*}{\frac{q(t - R_0)}{C_0} + T} p^*; \quad (5.8)$$

$$g(W, q) \doteq \frac{N_0 W}{\frac{q(t)}{C_0} + T} - C_0; \quad (5.9)$$

São apresentadas, a seguir, as derivadas parciais de $f(W, W^*, q, q^*, p^*)$ e $g(W, q)$, no ponto de equilíbrio (W_0, q_0, p_0) . Neste ponto, tem-se $W = W^* = W_0$, $q = q^* = q_0$ e $p = p^* = p_0$, ou seja, os valores de W , W^* , q , q^* e p^* são substituídos por W_0 , q_0 e p_0 . São utilizados, também, os valores obtidos em (5.4-5.6) para colocar o resultado obtido em função dos parâmetros C_0 , N_0 e R_0 .

Derivadas parciais da função $f(W, W^*, q, q^*, p^*)$:

$$\begin{aligned}\frac{\partial f}{\partial W} &= -b_0 W \frac{W^*}{\frac{q(t - R_0)}{C_0} + T} p^* \\ &= -b_0 W \frac{W^*}{R_0} p^* \\ &= \frac{-(\frac{b_0(W + \Delta W)W^*p^*}{R_0}) - (-\frac{b_0 W W^* p^*}{R_0})}{\Delta W} \\ &= -\frac{b_0 \Delta W W^* p^*}{R_0 \Delta W} \\ &= -\frac{b_0 W^* p^*}{R_0} \\ &= -\frac{b_0 W_0 p_0}{R_0}\end{aligned}$$

substituindo $p_0 = \frac{a_0}{b_0 W_0^2}$ tem-se:

$$\begin{aligned}&= -\frac{b_0 W_0 \left(\frac{a_0}{b_0 W_0^2} \right)}{R_0} \\ &= -\frac{a_0}{W_0 R_0}\end{aligned}$$

uma vez que $W_0 = \frac{R_0 C_0}{N_0}$:

$$\begin{aligned} &= -\frac{a_0}{\left(\frac{R_0 C_0}{N_0}\right) R_0} \\ &= -\frac{a_0 N_0}{R_0^2 C_0}; \end{aligned}$$

$$\begin{aligned} \frac{\partial f}{\partial W^*} &= \frac{b_0 W W^* p^*}{\frac{q}{C_0} + T} \\ &= \frac{-\frac{b_0 W (W^* + \Delta W^*) p^*}{\frac{q}{C_0} + T} - \frac{(-b_0 W W^* p^*)}{\frac{q^*}{C_0} + T}}{\Delta W^*} \\ &= \frac{-\frac{b_0 W W^* p^*}{\frac{q^*}{C_0} + T} - \frac{b_0 W \Delta W^* p^*}{\frac{q^*}{C_0} + T} + \frac{b_0 W W^* p^*}{\frac{q^*}{C_0} + T}}{\Delta W^*} \\ &= -\frac{b_0 p^* W}{\frac{q^*}{C_0} + T} \end{aligned}$$

sendo $R_0 = \frac{q}{C_0} + T$:

$$= -\frac{b_0 p^* W}{R_0} = \frac{\partial f}{\partial W}$$

$$\frac{\partial f}{\partial q} = \frac{a_0}{\frac{q}{C_0} + T}$$

$$\begin{aligned} &= a_0 \left(\frac{q}{C_0} + T\right)^{-1} \\ &= -a_0 \left(\frac{q}{C_0} + T\right)^{-2} \left(\frac{\left(\frac{q+\Delta q}{C_0}\right) + T - \frac{q}{C_0} - T}{\Delta q}\right) \\ &= -a_0 \left(\frac{q}{C_0} + T\right)^{-2} \left(\frac{\frac{\Delta q}{C_0}}{\Delta q}\right) \\ &= -a_0 \frac{1}{C_0} \frac{1}{\left(\frac{q}{C_0} + T\right)^2} \\ &= -\frac{a_0}{c_0 \left(\frac{q}{C_0} + T\right)^2} \end{aligned}$$

dado que $R_0 = \frac{q}{C_0} + T$:

$$= -\frac{a_0}{C_0 R_0^2}$$

$$\begin{aligned}
\frac{\partial f}{\partial q^*} &= -\frac{b_0 W W^* p^*}{\frac{q^*}{C_0} + T} \\
&= -b_0 W W^* p^* \left(\frac{q^*}{C_0} + T \right)^{-1} \\
&= b_0 W W^* p^* \left(\frac{q^*}{C_0} + T \right)^{-2} \left(\frac{\frac{q+\Delta q}{C_0} + T - \frac{q}{C_0} + T}{\Delta q} \right) \\
&= b_0 W W^* p^* \left(\frac{q^*}{C_0} + T \right)^{-2} \frac{\Delta q}{C_0 \Delta q}
\end{aligned}$$

sendo $R_0 = \frac{q}{C_0} + T$:

$$= \frac{b_0 W_0^2 p_0}{C_0 R_0^2}$$

dado, $W_0^2 = \frac{a_0}{b_0 p_0}$, então $p_0 = \frac{a_0}{b_0 W_0^2}$ e:

$$\begin{aligned}
&= \frac{b_0 W_0^2 \frac{a_0}{b_0 W_0^2}}{C_0 R_0^2} \\
&= \frac{a_0}{C_0 R_0^2}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f}{\partial p^*} &= -\frac{b_0 W W^* p^*}{\frac{q^*}{C_0} + T} \\
&= \frac{-\frac{b_0 W W^* (p^* + \Delta p^*)}{\frac{q^*}{C_0} + T} - \left(-\frac{b_0 W W^* p^*}{\frac{q^*}{C_0} + T} \right)}{\Delta p^*} \\
&= -\frac{b_0 W W^*}{R_0} \\
&= -\frac{b_0 W_0^2}{R_0}
\end{aligned}$$

onde $W_0 = \frac{R_0 C_0}{N_0}$:

$$\begin{aligned}
&= -\frac{b_0 \left(\frac{R_0 C_0}{N_0} \right)^2}{R_0} \\
&= -\frac{b_0 R_0^2 C_0^2}{N_0^2 R_0} \\
&= -\frac{b_0 R_0 C_0^2}{N_0^2}
\end{aligned}$$

Derivadas parciais de $g(W, q)$:

$$\begin{aligned}\frac{\partial g}{\partial W} &= \frac{\frac{N(W+\Delta W)}{\frac{q}{C_0}+T} - \frac{NW}{\frac{q}{C_0}+T}}{\Delta W} \\ &= \frac{N\Delta W}{\left(\frac{q}{C_0} + T\right)\Delta W} \\ &= \frac{N}{\frac{q}{C_0} + T} \\ &= \frac{N_0}{R_0}\end{aligned}$$

$$\begin{aligned}\frac{\partial g}{\partial q} &= \frac{N}{\frac{q}{C_0} + T} \\ &= NW\left(\frac{q}{C_0} + T\right)^{-1} \\ &= -NW\left(\frac{q}{C_0} + T\right)^{-2} \frac{1}{C_0} \\ &= -\frac{N_0W_0}{C_0R_0^2}\end{aligned}$$

dados $W_0 = \frac{R_0C_0}{N_0}$:

$$\begin{aligned}&= -\frac{N_0 \frac{R_0C_0}{N_0}}{C_0R_0^2} \\ &= -\frac{R_0C_0}{C_0R_0^2} \\ &= -\frac{1}{R_0}\end{aligned}$$

Com as derivadas parciais, a linearização da dinâmica simplificada do comportamento do HSTCP (5.7) em torno do ponto de equilíbrio (W_0, q_0, p_0) é dada por:

$$\begin{aligned}\dot{x}_1(t) &= -\frac{a_0N_0}{R_0^2C_0}(x_1(t) + x_1(t - R_0)) - \frac{b_0C_0^2R_0}{N_0^2}u(t - R_0) \\ &\quad - \frac{a_0}{C_0R_0^2}(x_2(t) - x_2(t - R_0)); \\ \dot{x}_2(t) &= \frac{N_0}{R_0}x_1(t) - \frac{1}{R_0}x_2(t);\end{aligned}\tag{5.10}$$

onde $x_1(t)$, $x_2(t)$ e $u(t)$ expressam a perturbação da janela, da fila e da probabilidade de

perda em torno do ponto de equilíbrio:

$$\begin{aligned}x_1(t) &\doteq W(t) - W_0; \\x_2(t) &\doteq q(t) - q_0; \\u(t) &\doteq p(t) - p_0;\end{aligned}$$

5.3 Controle de Congestionamento como um Problema de Controle

O HSTCP mantém o princípio do TCP Reno de detectar congestionamento através de perdas. Desta forma, a estimativa de banda passante disponível e os ajustes da taxa de transmissão são realizados em função do reconhecimento de ACKs enviados pelo receptor. O HSTCP infere a existência de congestionamento quando seus pacotes são descartados pelos roteadores, e utiliza a falta de reconhecimento como uma indicação de que perdas ocorreram. As políticas de AQM são responsáveis por notificar a existência de congestionamento aos emissores. Para detectar o estado do congestionamento da rede as políticas de AQM monitoram o nível de ocupação da fila, e em função do tamanho da fila notificam os emissores através do descarte/marcação de pacotes.

O controle de congestionamento pode ser visto como um sistema de controle por retroalimentação, no qual a taxa de transmissão dos nós emissores é ajustada de acordo com o estado de congestionamento da rede, detectado pelos roteadores através do monitoramento do nível de ocupação das filas e notificação para os emissores através do descarte/marcação de pacotes.

As políticas de AQM baseadas em Teoria de Controle consideram a natureza intrínseca de retroalimentação do sistema de controle de congestionamento. Busca-se compreender a dinâmica do congestionamento, para poder garantir estabilidade e robustez das condições de equilíbrio das métricas de desempenho desejadas.

Os emissores podem, então, ajustar sua taxa de transmissão em função do estado de congestionamento da rede. Desta forma, os controladores são responsáveis por determinar a probabilidade de descarte/marcação que maximize a vazão e minimize as perdas, enquanto garantem estabilidade do tamanho da fila independentemente das variações das condições do meio.

A estabilidade é necessária devido ao fato de que sistemas AQM instáveis produzem grandes variações de retardo e baixa utilização de enlace.

A Figura 5.1 apresenta o sistema de controle de congestionamento como um sistema com retroalimentação. A função do controlador $C(s)$, é marcar os pacotes randomicamente com probabilidade p , em função do tamanho da fila medido (q), e também estabilizar a planta, denotada pela função de transferência $P(s)$, irracional em s , que descreve como a probabilidade de descarte/marcação de pacotes afeta a taxa de transmissão e, conseqüentemente, o tamanho da fila [Lima, 2005].

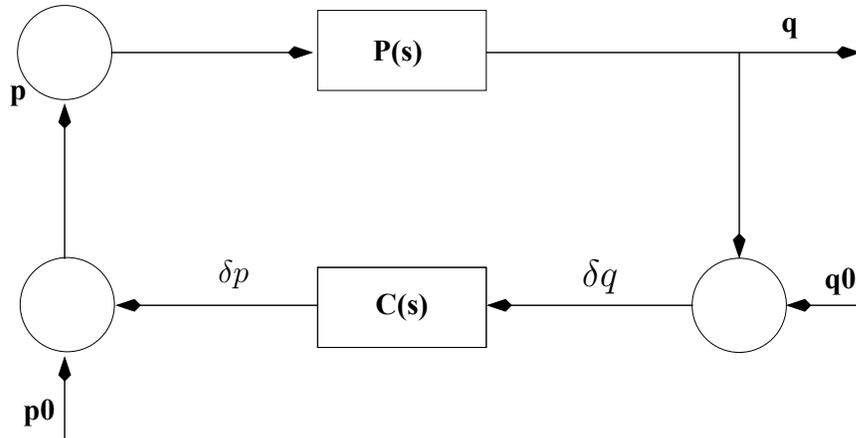


Figura 5.1: Controle de Congestionamento como um Sistema de Controle com Retroalimentação

A planta $P(s)$ pode ser determinada através da análise da dinâmica do sistema de controle de congestionamento descrita em (5.10). Tal dinâmica pode ser analisada em termos de parâmetros de rede tais como o número de fluxos HSTCP, N_0 , o tempo de viagem (RTT), R_0 , a capacidade do enlace, C_0 , e em termos da natureza de retroalimentação do AQM [Hollot et al., 2002].

O controlador $C(s)$ (Figura 5.1) estima a distância que o tamanho da fila medido q está do valor no ponto operacional q_0 , e utiliza esta diferença para determinar quanto deve variar a probabilidade de descarte. Esta variação é, então, utilizada para determinar o valor de p em função do valor da probabilidade no ponto de equilíbrio p_0 . Desta forma, o controlador monitora o tamanho da fila para determinar um valor adequado para a probabilidade de descarte/marcação, de modo a estabilizar o sistema em torno do ponto de equilíbrio independentemente das condições da rede.

Capítulo 6

Projeto de um Controlador Ótimo AQM

Neste capítulo, apresenta-se o desenvolvimento do projeto do controlador ótimo para AQM, denominado HSTCP-H2-AQM, que utiliza o modelo simplificado da dinâmica do sistema de controle de congestionamento utilizado no Capítulo 5.

Na sua grande maioria, as políticas de AQM baseadas em Teoria de controle utilizam apenas informações da dinâmica atual do sistema, desconsiderando os termos dependentes de atraso. Ao desconsiderar o atraso, a abordagem apresenta algumas limitações de desempenho, na presença de grandes atrasos, quando comparada com o controle dependente de atraso [Park, 1999]. Considerando o ambiente de redes, isto significa que a utilização da abordagem dependente de atraso ou não-racional faz com que os recursos de rede sejam usados de modo eficiente [Kim et al., 2003b]. Desta forma, neste capítulo, o sistema de controle de congestionamento apresentado em (5.10) é representado na forma de estado de espaço como um sistema linear com atraso e uma abordagem não-racional é utilizada para derivar o controlador ótimo HSTCP-H2. A utilização de uma abordagem não-racional permite a superação da principal dificuldade de se projetar controladores racionais para sistemas com atraso, que é incorporar no projeto do problema a matriz de multiplicação utilizada para provar a estabilidade com relação a parte com atraso do sistema [Oliveira and Geromel, 2004].

A síntese do controlador é baseada nos resultados apresentados em [Oliveira and Geromel, 2004], onde os projetos de controladores para sistemas lineares com atraso são expressos e solucionados como desigualdades matriciais lineares, LMIs (*Linear Matrix Inequalities*), possibilitando que os parâmetros do controlador

possam ser obtidos através da resolução de um simples problema convexo. Seguindo a abordagem apresentada em [Lima, 2005], a forma do controlador foi cuidadosamente escolhida de modo a reproduzir a estrutura da planta do sistema linear com atraso contínuo no tempo. A análise de estabilidade do sistema é feita através da conexão da planta com o controlador. Tal escolha foi considerada por possibilitar a análise de estabilidade independente de atraso que gera condições de síntese que podem ser expressas e solucionadas como LMIs.

A forma como o controlador atinge o ponto de equilíbrio é determinada pelos objetivos de desempenho do projeto. Desta forma, neste capítulo, são discutidos quais devem ser os objetivos de projeto de um controlador ótimo para redes com produto banda-atraso elevado, de modo a obter o melhor desempenho. Para a derivação dos controladores, foi investigado qual o melhor ponto de equilíbrio para o sistema de controle de congestionamento.

6.1 Obtenção do Controlador para a Política de AQM

Da linearização em (5.10), o sistema é representado na forma de estado de espaço com atraso contínuo no tempo:

$$\begin{aligned}\dot{x}(t) &= A_0x(t) + A_1x(t - R_0) + B_uu(t); \\ z(t) &= C_{z0}x(t) + C_{z1}x(t - R_0) + D_{zu}u(t); \\ y(t) &= C_yx(t - R_0) + D_{yw}w(t);\end{aligned}\tag{6.1}$$

onde:

$x(t)$ é o vetor de estado;

$u(t)$ é a entrada a ser controlada e representa a probabilidade $p(t)$;

$w(t)$ é o ruído externo produzido pelos tráfegos UDP;

$z(t)$ é a saída esperada para o sistema;

$y(t)$ é a saída obtida para o sistema.

Dado um sistema dinâmico e suas variáveis envolvidas, o controlador é responsável por atingir o ponto de equilíbrio do sistema e estabilizar em torno do mesmo. O que diferencia controladores é a forma como alcançam o ponto de equilíbrio, no qual se estabilizam. O caminho percorrido para atingir a estabilidade depende dos objetivos utilizados para projetar o controlador, ou seja, o comportamento desejado para o sistema controlado é

determinado pelos objetivos de desempenho para a política de AQM, os quais devem estar explícitos no projeto do controlador.

Em [Hollot et al., 2002] são sugeridos alguns objetivos para políticas de AQM:

- **Utilizar a fila de forma eficiente:** a sobrecarga na fila leva a perda de pacotes e, conseqüentemente, a necessidade de retransmissões e penalização de rajadas. No outro extremo está o esvaziamento do *buffer*, situação esta que subutiliza o enlace;
- **Atraso de fila:** o atraso da rede é dado pelo tempo que os pacotes permanecem em filas de roteadores e o tempo de propagação no meio. Para o bom desempenho das aplicações, é desejável que tanto o atraso como a variação do mesmo (*jitter*) mantenha-se baixos;
- **Robustez:** independente das variações do ambiente de rede, sejam número de conexões, tempo de propagação ou capacidade do enlace, os objetivos de desempenho devem ser alcançados.

Para o problema em questão, onde são abordadas redes com produto banda-atraso elevado, são considerados os seguintes objetivos de desempenho:

- Prevenir subutilização do enlace e minimizar a ocorrência de *Jitter*;
- Atingir vazão ideal mais rapidamente e minimizar a ocorrência de *Jitter*;
- Atingir vazão ideal mais rapidamente e prevenir subutilização do enlace.

Definidos os objetivos, os mesmos devem ser expressos nas matrizes do sistema (6.1) conforme segue:

$$A_0 = \begin{bmatrix} -\frac{a_0 N_0}{R_0^2 C_0} & -\frac{a_0}{C_0 R_0^2} \\ \frac{N_0}{R_0} & -\frac{1}{R_0} \end{bmatrix},$$

A matriz A_0 refere-se aos termos do sistema sem atraso. Obtida do sistema linearizado, sua primeira linha é dada por $\frac{\lambda f}{\lambda W}$ e $\frac{\lambda f}{\lambda q}$ e a segunda linha correspondendo a $\frac{\lambda g}{\lambda W}$ e a $\frac{\lambda g}{\lambda q}$.

$$A_1 = \begin{bmatrix} -\frac{a_0 N_0}{R_0^2 C_0} & \frac{a_0}{C_0 R_0^2} \\ 0 & 0 \end{bmatrix},$$

Na matriz A_1 são representados os termos com atraso. Obtida, também, diretamente da linearização do sistema, na primeira linha estão $\frac{\lambda f}{\lambda W^*}$ e $\frac{\lambda f}{\lambda q^*}$. A segunda linha de A_1 é nula, uma vez que $\frac{\lambda g}{\lambda W^*} = 0$ e $\frac{\lambda g}{\lambda q^*} = 0$.

$$B_u = \begin{bmatrix} -\frac{b_0 C_0^2 R_0}{N_0^2} \\ 0 \end{bmatrix},$$

A matriz B_u , dada por $\frac{\lambda f}{\lambda u}$ na primeira linha e $\frac{\lambda g}{\lambda p} = 0$ na segunda linha, representa a entrada a ser controlada, ou seja, a probabilidade de descarte/marcação p .

$$B_w = \begin{bmatrix} 0 & 0 \\ 0.2C_0 & 0 \end{bmatrix}, \quad D_{yw} = \begin{bmatrix} 0 & 0.02C_0 \end{bmatrix},$$

O ruído existente no sistema, devido aos fluxos UDP, é controlado por B_w . O valor $0.2C_0$ permite que os fluxos UDP utilizem até 20% da capacidade do enlace. A matriz D_{yw} é responsável por ponderar o ruído na saída, e consiste de 10% do valor presente em B_w .

$$C_y = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

A matriz c_y indica que o valor a ser medido na saída é o tamanho da fila no RTT anterior.

Os objetivos, previamente determinados, são representados nas matrizes C_{z0} , C_{z1} e D_{zu} :

Objetivo 1: Prevenir subutilização do enlace e minimizar a ocorrência de *jitter*

$$C_{z0} = \begin{bmatrix} 0 & 1 \\ \frac{N_0}{R_0} & \frac{-1}{R_0} \\ 0 & 0 \end{bmatrix}, \quad C_{z1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix},$$

Para que o enlace não seja subutilizado, a diferença entre o valor medido para o tamanho da fila (q) e o valor ideal para a fila no ponto de equilíbrio (q_0) deve ser minimizada, o que está representado na primeira linha da matriz C_{z0} . A ocorrência de *jitter* refere-se a variação de atraso, a qual é consequência da oscilação no tamanho da fila. Na segunda linha de c_{z0} , é expressa a minimização da variação no tamanho da fila, ou seja, minimiza-se o *jitter*.

Os valores esperados na saída do sistema $z(t)$, são os valores obtidos no intervalo corrente e não no intervalo anterior. Logo, C_{z1} é uma matriz nula.

D_{zu} é a matriz responsável por ponderar o valor da probabilidade de descarte/marcação na saída. Diferentes valores de pesos foram verificados, variando de 0.3 a 0.9. Os resultados obtidos foram similares, sendo 0.5 o valor adotado.

Objetivo 2: Atingir vazão ideal mais rapidamente e minimizar a ocorrência

de *jitter*

$$C_{z0} = \begin{bmatrix} 1 & 0 \\ \frac{N_0}{R_0} & \frac{-1}{R_0} \\ 0 & 0 \end{bmatrix}, \quad C_{z1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix},$$

Para atingir vazão ideal o valor medido para a janela de congestionamento (w) deve ser igual ao valor da janela de congestionamento no ponto de equilíbrio (w_0). Deseja-se, ainda, que a vazão ideal seja atingida rapidamente, ou seja, independentemente das oscilações na taxa de transmissão. Em C_{z0} na primeira linha é representada a minimização da diferença no tamanho da janela de congestionamento ($w - w_0$). Para minimizar a ocorrência de *Jitter*, na segunda linha é representado o objetivo de minimizar a variação da fila em torno do ponto de equilíbrio (q_0).

As matrizes C_{z1} e D_{zu} são as mesmas apresentadas no Objetivo 1.

Objetivo 3: Atingir vazão ideal mais rapidamente e prevenir subutilização do enlace

$$C_{z0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad C_{z1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{zu} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix},$$

A minimização da diferença entre o valor medido para a janela de congestionamento (w) e o valor no ponto de equilíbrio (w_0) está representado na primeira linha de C_{z0} . Para prevenir subutilização do enlace, o valor medido para o tamanho da fila (q) deve ficar próximo do valor no ponto de equilíbrio (q_0), o que é representado na segunda linha de C_{z0} .

Os valores para C_{z1} e D_{zu} são os mesmos dos objetivos anteriores.

Definidos os objetivos, o sistema em (6.1) é conectado ao controlador em (6.2):

$$\begin{aligned} \hat{x}(t) &= \hat{A}_0 \hat{x}(t) + \hat{A}_1 \hat{x}(t - R_0) + \hat{B}y(t); \\ u(t) &= \hat{C}_0 \hat{x}(t) + \hat{C}_1 \hat{x}(t - R_0) + \hat{D}y(t). \end{aligned} \quad (6.2)$$

No domínio da frequência, o controlador é representado pela função de transferência não-racional em (6.3).

$$C(s) = (\hat{C}_0 + \hat{C}_1 e^{sR_0})(sI - \hat{A}_0 - \hat{A}_1 e^{-sR_0})^{-1} \hat{B} + \hat{D} \quad (6.3)$$

O controlador $C(s)$, definido em (6.2) ou (6.3), foi cuidadosamente escolhido de forma a reproduzir em detalhes a estrutura da planta do Sistema (6.1). Desta forma, a estabilização da planta pelo controlador implica na estabilidade do sistema de congestionamento independentemente das condições da rede.

6.2 Síntese dos Controladores Ótimos

A síntese do controlador segue a abordagem apresentada em [Oliveira and Geromel, 2004], na qual os projetos de controladores para sistemas lineares com atraso são expressos e solucionados como desigualdades matriciais lineares, LMI (*Linear Matrix Inequalities*).

Seja o vetor de estado aumentado $\bar{x}(t)$ que contém o vetor de estado $x(t)$ e o vetor de estado do controlador $\hat{x}(t)$:

$$\bar{x}(t) = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}, \quad (6.4)$$

Da conexão do sistema (5.10) com o controlador (6.2) tem-se o sistema linear com atraso:

$$\begin{aligned} \dot{\bar{x}}(t) &= \mathcal{A}_0 \bar{x}(t) + \mathcal{A}_1 \bar{x}(t - R_0) + \mathcal{B}w(t); \\ z(t) &= \mathcal{C}_0 \bar{x}(t) + \mathcal{C}_1 \bar{x}(t - R_0) + \mathcal{D}w(t). \end{aligned} \quad (6.5)$$

onde:

$$\begin{aligned} \mathcal{A}_0 &= \begin{bmatrix} A_0 & B_u \hat{C}_0 \\ 0 & \hat{A}_0 \end{bmatrix}, \quad \mathcal{A}_1 = \begin{bmatrix} A_1 + B_u \hat{D}C_y & B_u \hat{C}_1 \\ \hat{B}C_y & \hat{A}_1 \end{bmatrix}, \\ \mathcal{B} &= \begin{bmatrix} B_w + B_u \hat{D}D_{yw} \\ \hat{B}D_{yw} \end{bmatrix}, \quad \mathcal{C}_0 = \begin{bmatrix} C_z & D_{zu} \hat{C}_0 \end{bmatrix}, \\ \mathcal{C}_1 &= \begin{bmatrix} D_{zu} \hat{D}C_y & D_{zu} \hat{C}_1 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} D_{zu} \hat{D}D_{yw} \end{bmatrix} \end{aligned}$$

Para garantir a estabilidade no sistema com atraso em (6.5), o Teorema 4-b apresentado em [Oliveira and Geromel, 2004] é usado. Este teorema especifica que um sistema como (6.5) é assintoticamente estável e $\|H_{wz}\|_2^2 < \gamma$, se existem matrizes simétricas e definidas positivas W , Y_0 e X_j , e matrizes F , R , L_j e Q_j , com $j = 0, 1$, tais que os seguintes LMIs tenham uma solução factível:

$$0 > \begin{bmatrix} \mathbf{A}_0 + \mathbf{A}_0^T & (\bullet)^T & (\bullet)^T \\ \mathbf{A}_1^T & -X_1 & (\bullet)^T \\ C_0 & C_1 & -I \end{bmatrix}, \quad (6.6)$$

$$0 < \begin{bmatrix} W & (\bullet)^T \\ \mathbf{B} & P_0 \end{bmatrix}, \quad \text{trace}(W) < \gamma \quad (6.7)$$

onde:

$$\mathbf{A}_0 = \begin{bmatrix} A_0X_0 + B_uL_0 & A_0 \\ Q_0 & Y_0A_0 \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} A_1X_0 + B_uL_1 & A_1 \\ Q_1 & Y_0A_1 + FC_y \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} B_w \\ Y_0B_w + FD_{yw} \end{bmatrix}, \quad \mathbf{P}_0 = \begin{bmatrix} X_0 & I \\ I & Y_0 \end{bmatrix},$$

$$\mathbf{C}_0 = \begin{bmatrix} C_zX_0 + D_{zu}L_0 \\ C_z \end{bmatrix}, \quad \mathbf{C}_1 = \begin{bmatrix} D_{zu}L_1 & 0 \end{bmatrix}$$

6.3 Determinação do Ponto de Equilíbrio

Para resolver o problema de programação convexa representado pelos LMIs (6.6) e (6.7), faz-se necessário definir o ponto de equilíbrio para o sistema, ou seja, devem ser atribuídos valores para os parâmetros de rede N_0 , R_0 e C_0 de modo a obter o ponto de equilíbrio (W_0, q_0, p_0) .

O ponto de equilíbrio foi escolhido com o intuito de se obter um ponto de equilíbrio para quando o sistema de congestionamento está mais suscetível, ou seja, quando a carga da rede é alta. O problema de programação convexa foi numericamente resolvido utilizando a rotina LMISol [Oliveira and Geromel, 2008]. A capacidade de enlace considerada foi de 1Gbps com pacotes de 1500 *bytes*, desta forma, $C_0 = 83333$ pacotes/segundo. Para definir o parâmetro $R_0 = 200$ ms, foram consideradas medidas realizadas em redes com transmissões internacionais e intercontinentais [Shakkottai et al., 2004], bem como redes com transferência de grandes quantidades de dados em redes de alta velocidade [Cottrell et al., 2003] e a influência do RTT na estabilidade relativa do HSTCP [Huang et al., 2004]. Em [Martin-Flatin and Ravot, 2002], o tráfego definido como de “muito longa duração” é responsável por transmitir *Terabytes* de dados com número de conexões variando entre 1 e 10, sendo o principal responsável por eventos de congestionamento. Desta forma, o parâmetro número de conexões considerado foi de $N_0 = 10$ conexões HSTCP.

Através da resolução do problema de programação convexa, pode-se afirmar que o sistema (6.5) é estável, ou seja, foram encontrados valores para as matrizes W , Y_0 , X_j , F , R , L_j e Q_j , com $j = 0, 1$, que levam a uma solução factível para o sistema.

6.4 Determinação dos Parâmetros dos Controladores

O Teorema 4 – b [Oliveira and Geromel, 2004] especifica que se for encontrada uma solução factível para as LMI's (6.6) e (6.7) o sistema (6.5) é estável e as matrizes encontradas W , Y_0 , X_j , F , R , L_j e Q_j podem ser utilizadas para determinar os parâmetros do controlador (6.2). Para definir os parâmetros do controlador, primeiramente, as matrizes arbitrárias U_0 e V_0 devem ser escolhidas de forma que $V_0 * U_0 = I - Y_0 X_0$, assim, as matrizes utilizadas foram $U_0 = X_0$ e $V_0 = X_0^{-1} - Y_0$. Considerando-se as definições em [Oliveira and Geromel, 2004], os parâmetros do controlador podem ser determinados por:

$$\begin{bmatrix} \hat{A}_0 & \hat{A}_1 & \hat{B} \\ \hat{C}_0 & \hat{C}_1 & \hat{D} \end{bmatrix} = \mathcal{K}.\mathcal{M}.\mathcal{N}. \quad (6.8)$$

onde:

$$\mathcal{K} = \begin{bmatrix} V_0^{-1} & V_0^{-1}Y_0B_u \\ 0 & I \end{bmatrix};$$

$$\mathcal{M} = \begin{bmatrix} Q_0 - Y_0A_0X_0 & Q_1 - Y_0A_1X_0 & F \\ L_0 & L_1 & 0 \end{bmatrix};$$

$$\mathcal{N} = \begin{bmatrix} U_0^{-1} & 0 & 0 \\ 0 & U_0^{-1} & 0 \\ -C_{y0}X_0U_0^{-1} & -C_{y1}X_0U_0^{-1} & I \end{bmatrix}$$

Nas soluções obtidas as matrizes \hat{A}_1 e \hat{C}_1 são aproximadamente zero, sendo portanto ignoradas. Desta forma, o termo de atraso do sistema é cancelado e os controladores se tornam racionais. O Cancelamento do termo de atraso, quando possível, é a estratégia global ótima para solucionar o problema de minimização da norma H_2 [Oliveira and Geromel, 2004].

As funções de transferência no domínio da frequência para os controladores, de acordo com os objetivos definidos são dadas por:

$$C_{HSTCP-H2_1}(s) = \frac{0.0002836s + 0.0009829}{s^2 + 3.687e^5s + 1.278e^6} \quad (6.9)$$

$$C_{HSTCP-H2_2}(s) = \frac{0.0002712s + 6.243e^{-5}}{s^2 + 3.52e^5s + 8.105e^4} \quad (6.10)$$

$$C_{HSTCP-H2_3}(s) = \frac{1.383e^{-6}s + 0.0001493}{s^2 + 1800s + 1.949e^5} \quad (6.11)$$

onde $C_{HSTCP-H2_i}(s)$ representa o controlador derivado para o i_{esimo} objetivo.

6.5 Implementação Digital dos Controladores

Para implementar os controladores é necessário representa-los no domínio- z através da definição de uma freqüência de amostragem f_s . Esta freqüência de amostragem deve ser um valor tal que evite descarte desnecessário de pacotes e estabilize o tamanho da fila. Desta forma, a freqüência do cálculo da probabilidade de descarte/marcação deve proporcionar ao mecanismo de controle de congestionamento a capacidade de reagir de forma apropriada a mudanças abruptas, mas deve fazer, também, com que sua reação não seja excessivamente severa, de modo a permitir que mudanças transientes sejam desconsideradas [Floyd, 2007].

Para determinar a freqüência adequada para a representação dos controladores no domínio- z foram testados valores de freqüência variando de 10% a 90% da capacidade do enlace, C_0 . A variação destes valores não apresentou mudanças significativas nos controladores obtidos, escolhendo-se, portanto, a freqüência de amostragem de $f_s = 8333Hz$, equivalente a 10% da capacidade de enlace C_0 .

Considerando-se 10% da capacidade do enlace como a freqüência de amostragem para a representação no domínio- z , os controladores obtidos são:

$$C_{HSTCP-H2_1}(z) = \frac{az^2 + bz - c}{z^2 - dz - e} = \frac{7.361e^{-10}z^2 + 3.06e^{-13}z - 7.358e^{-10}}{z^2 - 0.08609z - 0.9131}$$

$$C_{HSTCP-H2_2}(z) = \frac{az^2 + bz - c}{z^2 - dz - e} = \frac{7.357e^{-10}z^2 + 2.032e^{-14}z - 7.357e^{-10}}{z^2 - 0.09038z - 0.9096}$$

$$C_{HSTCP-H2_3}(z) = \frac{az^2 + bz - c}{z^2 - dz + e} = \frac{7.533e^{-11}z^2 + 9.699e^{-13}z - 7.436e^{-11}}{z^2 - 1.803z + 0.8052}$$

As funções de transferência entre $\delta p = p - p_0$ e $\delta q = q - q_0$, podem ser convertidas nas equações de diferenças no tempo discreto kT , onde $T = \frac{1}{f_s}$. Os controladores no domínio- z para os objetivos 1 e 2 apresentam equações de diferença da forma:

$$\delta p_1(kT) = a\delta q(kT) + b\delta q((k-1)T) - c\delta q((k-2)T) + d\delta p_1((k-1)T) + e\delta p_1((k-2)T); \quad (6.12)$$

O controlador no domínio- z para o objetivo 3 apresenta equação de diferença da forma:

$$\delta p_2(kT) = a\delta q(kT) + b\delta q((k-1)T) - c\delta q((k-2)T) + d\delta p_1((k-1)T) - e\delta p_1((k-2)T); \quad (6.13)$$

Dado que os controladores para os objetivos 1 e 2 convergem para uma mesma equação de diferença, então são necessários apenas dois algoritmos para implementar o cálculo da probabilidade de descarte/marcação dos controladores. O Algoritmo 4 corresponde a implementação da Equação (6.12) e o Algoritmo 5 corresponde a implementação da Equação (6.13).

Algoritmo 4 Algoritmo utilizado no cálculo da probabilidade para os controladores 1 e 2

```

1: HSTCP-H2-12-AQM-Probabilidade()
2:  $p_0 \leftarrow a_0 N_0^2 / b_0 (C_0 R_0)^2$ 
3:  $p_1 \leftarrow q_0(c - a - b) + a * q + b * q_{old} - c * q_{old1} + p_0(1 - d - e) + d * p_{old} + e * p_{old1}$ 
4:  $p_{old1} \leftarrow p_{old}$ 
5:  $q_{old1} \leftarrow q_{old}$ 
6:  $p_{old} \leftarrow p_0$ 
7:  $q_{old} \leftarrow q_0$ 

```

Algoritmo 5 Algoritmo utilizado no cálculo da probabilidade para o controlador 3

```

1: HSTCP-H2-3-AQM-Probabilidade()
2:  $p_0 \leftarrow a_0 N_0^2 / b_0 (C_0 R_0)^2$ 
3:  $p_1 \leftarrow q_0(c - a - b) + a * q + b * q_{old} - c * q_{old1} + p_0(1 - d - e) + d * p_{old} - e * p_{old1}$ 
4:  $p_{old1} \leftarrow p_{old}$ 
5:  $q_{old1} \leftarrow q_{old}$ 
6:  $p_{old} \leftarrow p_0$ 
7:  $q_{old} \leftarrow q_0$ 

```

É possível verificar que os dois algoritmos são bastante simples, e são executados a cada intervalo de amostragem de $1/f_s$. Inicialmente p_0 é calculado baseado nos parâmetros de rede dados (N_0, C_0, R_0) . Depois, o valor da probabilidade de descarte/marcação é calculado baseado nas equações de diferença. Quatro variáveis auxiliares q_{old} , p_{old} , q_{old1} e p_{old1} são utilizadas. As variáveis q_{old} e p_{old} armazenam os valores de q e p , respectivamente, no intervalo anterior. Por sua vez, as variáveis q_{old1} e p_{old1} armazenam os valores de q_{old} e p_{old} , respectivamente, no intervalo anterior.

6.6 Considerações em relação a Política HSTCP-H2 AQM

HSTCP-H2 é uma política derivada para redes que utilizam o protocolo HSTCP em ambientes com produto banda-atraso elevado, que segue a abordagem introduzida em

[Lima et al., 2004]. Algumas considerações podem ser feitas ao comparar HSTCP-H2 com outras políticas de AQM:

- HSTCP-H2 é baseada em Teoria de Controle Ótimo, sendo uma política analítica, que garante a estabilidade em torno do ponto de equilíbrio;
- A planta utilizada para derivar o controlador representa a dinâmica do sistema de congestionamento em detalhes, considerando também a existência de fluxos não-adaptativos. Desta forma, independentemente das condições da rede, o controlador obtido garante a estabilidade do sistema;
- O ponto de equilíbrio foi cuidadosamente investigado, de modo que fosse o melhor para o sistema de controle de congestionamento;
- Utilizou-se na derivação uma abordagem não-racional, permitindo que os recursos disponíveis sejam utilizados de maneira eficiente;
- HSTCP-H2 leva em consideração tanto as características de retroalimentação dos sistema, quanto o seu estado representado pelo tamanho da janela de congestionamento e pelo tamanho da fila (W, q);
- Embora tenha sido utilizada uma abordagem não-racional, o cancelamento do termo com atraso, levou a obtenção de um controlador racional.

Capítulo 7

Efetividade do Controlador HSTCP-H2-AQM

Através de simulações realizadas utilizando o software Simulink verificou-se que os três controladores derivados possuem as características de desempenho esperadas para o sistema linear [Santi et al., 2006]. É necessário, também, verificar as não-linearidades do sistema, através de simulações cuidadosas de modo a validar as características de desempenho para o sistema não linear. Desta forma, os Algoritmos (4) e (5) foram implementados no NS [Floyd, 2008], de forma a permitir a verificação/validação de desempenho do sistema de controle de congestionamento em um ambiente de rede dinâmico.

Neste capítulo, os algoritmos dos controladores derivados no Capítulo 6 são comparados entre si através de resultados obtidos em um ambiente dinâmico de redes com o objetivo de verificar qual deles apresenta melhor desempenho em redes com produto banda-atraso elevado. O controlador correspondente ao objetivo que apresentou melhor resultado é, então, comparado ao desempenho da política de AQM RED.

7.1 Cenários das Simulações com um Único Enlace Gargalo

Esta seção apresenta as configurações e o cenário utilizados nos experimentos para enlace com um único gargalo.

Na Figura 7.1 estão representados a topologia, a capacidade dos enlaces e o tempo de propagação utilizados para os experimentos com enlace gargalo único, também conhecida como topologia *Dumb-bell*. O enlace entre os nós R_1 e R_2 é o enlace gargalo.

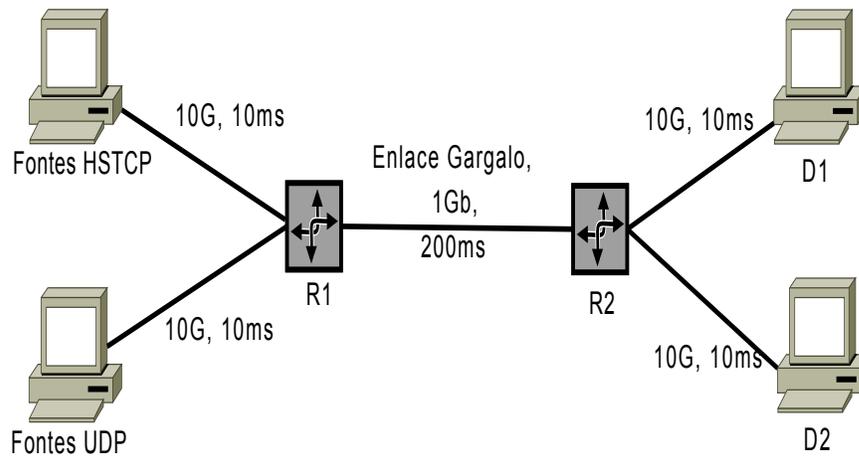


Figura 7.1: Topologia utilizada nos experimentos com enlace gargalo único

O tamanho do *buffer* utilizado é de 3333 pacotes o que corresponde a 20% do produto banda-atraso. Estudos realizados em [Barman et al., 2004] demonstram que tamanho de *buffer* equivalente a 20% do produto banda-atraso levam a uma utilização superior a 98% da capacidade de enlace pelo HSTCP.

Para verificar a eficiência das políticas de AQM variou-se o grau de congestionamento na rede, utilizando cargas de 0.4 até 1.0 da capacidade do enlace. Para gerar as cargas específicas foi empregado o gerador de tráfego TrafficGen [Cardoso and Rezende, 2000]. Foram considerados tráfego de longa duração (FTP) e curta duração (WEB). De modo a considerar um ambiente de rede realista, foram inseridos fluxos não-adaptativos CBR/UDP para gerar o tráfego de ruído.

As simulações foram compostas por 80% de tráfego de longa duração ou de curta duração como tráfego principal, e 20% de tráfego não-adaptativos como tráfego de ruído. Para a definição deste valores foram consideradas as medidas apresentadas em [Fomenkov et al., 2004], as quais demonstram que $(83 \pm 11)\%$ dos *bytes* que atravessam a Internet são do tipo TCP.

A variação TCP utilizada foi o *High Speed TCP*. O tamanho dos segmentos gerados foi de 1500 *bytes*, ou seja, o padrão Ethernet .

O tráfego principal HSTCP foi gerado dos nós *FontesHSTCP* para o nó D_1 e o tráfego de ruído foi gerado dos nós *FontesUDP* para o nó D_2 . O tráfego WEB é gerado usando uma distribuição híbrida Lognormal/Pareto. O corpo da distribuição corresponde a uma área de 0.88 e é modelado por uma distribuição Lognormal com média de 7247 *bytes*, já a cauda é modelada por uma distribuição Pareto com média de 10558 *bytes*

Controlador	a	b	c	d	e
$C1(z)$	$7.361e^{-10}$	$3.06e^{-13}$	$7.358e^{-10}$	0.08609	0.9131
$C2(z)$	$7.357e^{-10}$	$2.032e^{-14}$	$7.357e^{-10}$	0.09038	0.9096
$C3(z)$	$7.533e^{-11}$	$9.699e^{-13}$	$7.436e^{-11}$	1.803	0.8052

Tabela 7.1: Parâmetros Utilizados na Implementação dos Controladores Ótimos

[Barford et al., 1999]. O tráfego FTP é gerado usando uma distribuição exponencial com média de 512 *KBytes*.

O controle de fluxo foi desabilitado; ajustou-se a janela do receptor em um valor elevado (100000 segmentos), de forma que o crescimento da janela do transmissor fosse governada apenas pela rede e não pelo receptor.

7.2 Comparação entre os Controladores Ótimos Propostos

Para verificar o desempenho dos controladores obtidos no Capítulo 6, os Algoritmos 4 e 5 foram implementados no simulador de redes *Network Simulator* [Floyd, 2008]. O ponto de equilíbrio apresentado na Seção 5.2 foi utilizado. Para derivar os controladores foi utilizada uma frequência de amostragem de 8333 Hz, gerando os valores para os parâmetros dos controladores conforme apresentado na Tabela 7.1. O cenário apresentado na Seção 7.1 é utilizado nas simulações.

Os três controladores obtidos atingem o ponto de equilíbrio e o que os diferencia é apenas o caminho percorrido para atingir o ponto de equilíbrio. Desta forma, todos os controladores apresentam desempenho equivalente para todas as métricas, quando a média é utilizada para compará-los. A forma como os controladores atingem o ponto de equilíbrio é determinada pelos objetivos de projeto definidos, os quais influenciam a variação do desempenho em torno do seu valor médio. Assim, uma medida estatística para comparar o desempenho dos diferentes controladores é o coeficiente de variação, que é a razão entre o desvio padrão e a média ($CV = s/\bar{X}$).

Tanto para tráfego FTP como WEB, o CV das métricas relacionadas ao ponto de equilíbrio, tais como janela de congestionamento por conexão ativa (*cwnd*), probabilidade de descarte/marcação e tamanho da fila em função da carga na rede. O controlador com o melhor resultado é o que apresenta os valores mais baixos de CV para as métricas, o que

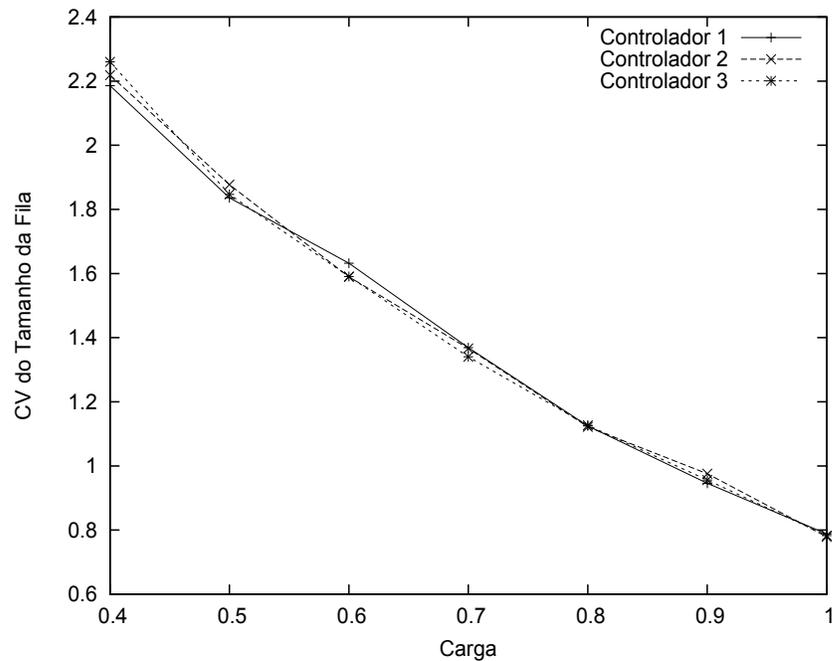


Figura 7.2: Tráfego FTP: coeficiente de variação do tamanho da fila em função da carga implica em uma pequena variação em torno do ponto de equilíbrio e, conseqüentemente, garante maior estabilidade para o sistema.

7.2.1 Experimentos Utilizando tráfego de Longa Duração

Nesta seção, são apresentados os resultados dos valores de coeficiente de variação, CV , para o tamanho da janela por conexão, para o tamanho da fila e para a probabilidade de descarte/marcação em função da carga, para as simulação que utilizaram tráfego de longa-duração (FTP).

A Figura 7.2 apresenta os CV para o tamanho da fila em função da carga. É possível verificar que os CV diminuem conforme a carga da rede é aumentada, o que significa que quanto maior a carga, maior a capacidade dos controladores em manter o valor do tamanho da fila mais próximo do ponto de equilíbrio. A diferença entre os controladores é muito pequena, o que ocorre devido ao fato de que o objetivo de minimizar *jitter* está presente nos três controladores. Para cargas acima de 0.7, a proximidade entre os resultados dos controladores é ainda maior. A escolha do ponto de equilíbrio para cargas grandes, leva os controladores a serem similares para valores elevados de carga. Embora a diferença entre os controladores seja muito pequena, para cargas elevadas, o Controlador C_1 apresenta menor CV para o tamanho da fila.

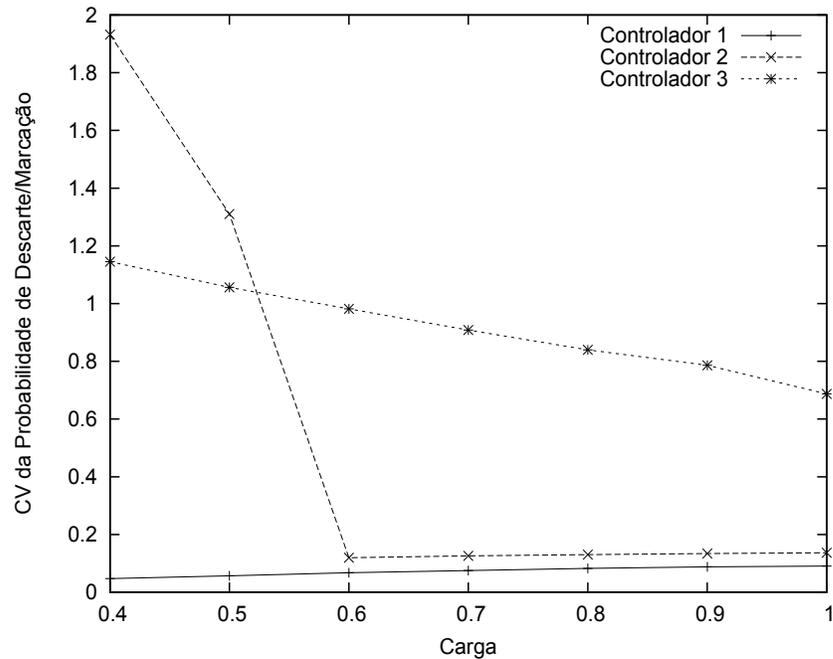


Figura 7.3: Tráfego FTP: coeficiente de variação da probabilidade de descarte/marcação em função da carga

Na Figura 7.3, é apresentado o CV da probabilidade de descarte/marcação em função da carga. Pode-se observar que, enquanto o Controlador C_1 consegue manter a probabilidade de descarte/marcação com valores baixos para todo o intervalo de cargas testado, os Controladores 2 e 3 apresentam valores elevados para o CV . Valores baixos de CV implicam em uma pequena variação da probabilidade de descarte/marcação em torno do ponto de equilíbrio, e conseqüentemente em uma menor variação da taxa instantânea de perda. Assim, para a métrica de probabilidade de descarte/marcação o Controlador C_1 é o que apresenta melhor desempenho.

A Figura 7.4 apresenta o CV do tamanho da janela de congestionamento em função da carga. Os valores de CV para todos os controladores são praticamente equivalentes. É possível observar, também, que os valores de CV aumentam com o aumento da carga. Com aumento da carga, aumenta, também, o número de conexões ativas o que leva a maiores oscilações e conseqüentemente ao aumento do CV . Pode-se observar que o Controlador 3 apresenta desempenho um pouco melhor do que os outros controladores, o que acontece devido ao objetivo do Controlador 3 que tenta minimizar a diferença entre o valor medido para a janela e seu valor ideal no ponto de equilíbrio.

O Controlador C_1 , que corresponde ao objetivo de Prevenir a Subutilização do Enlace e Minimizar a Ocorrência de *jitter*, foi o que apresentou melhores valores CV para a mai-

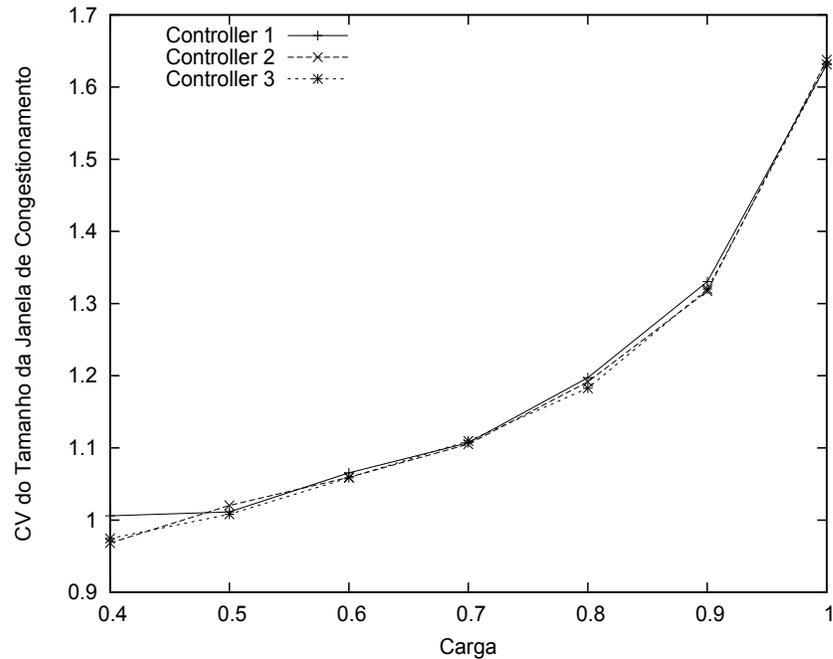


Figura 7.4: Tráfego FTP: coeficiente de variação da janela de congestionamento em função da carga

oria dos experimentos utilizando tráfego de longa-duração. Por apresentar menor CV do tamanho da fila, é o controlador que apresenta menores valores para *jitter*, e conseqüentemente, apresenta maior estabilidade do tamanho da fila. Além disso, o Controlador C_1 produz os menores valores de CV para a probabilidade de descarte/marcação, ou seja, apresenta a menor variação na taxa instantânea de perda.

7.2.2 Experimentos Utilizando tráfego de Curta Duração

Nesta seção, são apresentados os resultados dos valores de coeficiente de variação, CV , para o tamanho da janela por conexão, para o tamanho da fila e para a probabilidade de descarte/marcação em função da carga, para as simulação que utilizaram tráfego de curta-duração (WEB).

O sistema de equações utilizado para derivar os controladores considera a fase de *Congestion Avoidance* da HSTCP, portanto, o modelo apresenta grande precisão para tráfego de longa-duração, no qual a fase de *Slow Start* é curta e pode ser desconsiderada. Entretanto, o modelo não é tão preciso para tráfego de curta duração, dado que para este tipo de tráfego a fase de *Slow Start* tem um papel fundamental na janela do emissor. Desta forma, a eficiência dos controladores derivados pode ser reduzida quando o tráfego

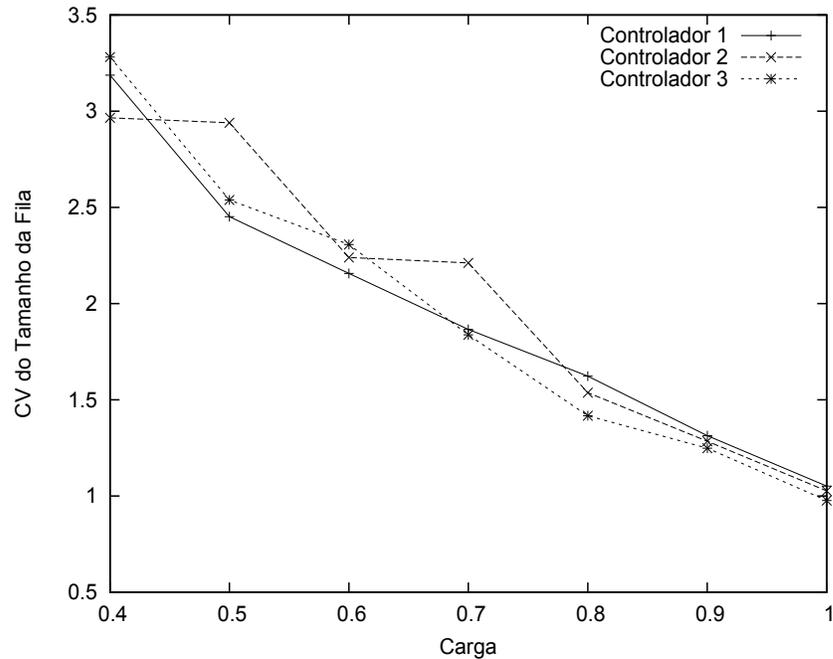


Figura 7.5: Tráfego WEB: coeficiente de variação do tamanho da fila em função da carga

utilizado é do tipo WEB. Mesmo conhecendo estas limitações, também foi investigado a eficiência dos controladores para tráfego de curta-duração.

A Figura 7.5 apresenta o CV do tamanho da fila em função da carga. O comportamento dos CV para o tamanho da fila dos controladores para tráfego WEB é igual ao apresentado para FTP, no entanto, para tráfego WEB passa a existir uma diferença considerável entre os CV s dos controladores. A maior variação nos CV do tamanho da fila apresentados pelos controladores sob tráfego WEB ocorre devido a grande oscilação na taxa de chegada de pacotes, o que é uma consequência da fase de *Slow Start* que dobra a taxa de transferência a cada RTT. Para valores baixos de carga o Controlador C_1 se sobressai aos demais; no entanto, a medida que a carga aumenta o Controlador C_3 apresenta menores valores para o CV do tamanho da fila.

Na Figura 7.6 é apresentado o CV da probabilidade de descarte/marcação em função da carga. É possível observar que o comportamento da CV para tráfego WEB é o mesmo apresentado pelo tráfego FTP. Além disso, para todo o intervalo de cargas testada, o Controlador C_1 é o que apresenta melhores CV de variação.

Através da Figura 7.7, pode-se observar que o comportamento do CV para tráfego WEB é diferente do comportamento para tráfego FTP. Para a maior parte do intervalo de cargas testado o Controlador C_1 é o que apresenta menores valores de CV .

Assim como para tráfego FTP, os Controladores C_1 e C_3 se sobressaem nos resul-

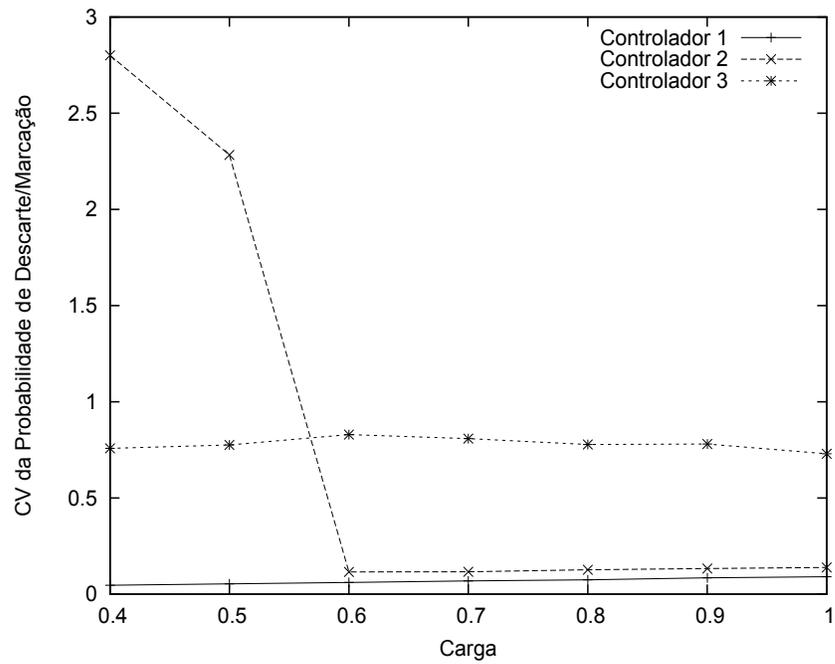


Figura 7.6: Tráfego WEB: coeficiente de variação da probabilidade de descarte/marcação em função da carga

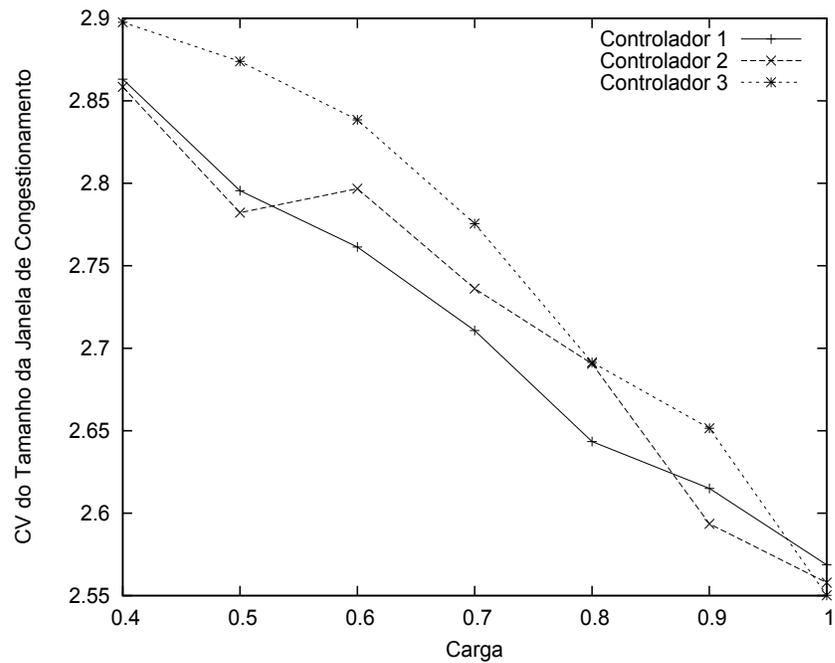


Figura 7.7: Tráfego WEB: coeficiente de variação da janela de congestionamento em função da carga

tados de CV . Curiosamente, os dois controladores consideram o objetivo de “prevenir a subutilização do enlace”. No entanto, considerando principalmente a métrica de CV de descarte/marcação, o Controlador C_1 apresenta uma larga vantagem em relação ao Controlador C_3 .

Considerando o desempenho dos controladores, de uma forma geral tanto para tráfego FTP como WEB, o Controlador C_1 é o controlador escolhido como o que apresenta melhor desempenho. O objetivo de desempenho definido para a derivação deste controlador é “prevenir a subutilização do enlace e minimizar a ocorrência de *jitter*”. Maximizar a utilização dos recursos disponíveis é um dos grandes desafios ao considerar redes com produto banda-atraso elevado, e portanto, ter uma política de AQM que auxilie neste propósito é de fundamental importância ao considerar estes ambientes. Além disso, a minimização do *jitter*, implica em estabilização da fila, e conseqüentemente na estabilização do sistema de controle de congestionamento como um todo.

O Controlador C_1 será referenciado daqui em diante por HSTCP-H2-AQM ou HSTCP-H2. A eficácia do HSTCP-H2 será avaliada, nas próximas seções, através da comparação com a política RED, considerando tanto enlace com um único gargalo, como enlaces com múltiplos gargalos.

7.3 Avaliação da Eficácia do Controlador HSTCP-H2 com Enlace Gargalo Único

Nesta Seção, a eficácia da política de AQM HSTCP-H2 (Algoritmo 4) é avaliada através de simulações no NS. Comparando o seu desempenho com o da política de AQM padrão RED. Os ajustes de RED foram realizados considerando-se o limiar max_th equivalente a 20% do produto banda-atraso, ou seja, 3333 pacotes. Os ajustes dos demais parâmetros foram feitos levando em consideração as recomendações em [Floyd, 1997a] e [Floyd, 2000]; que indicam para os seguintes valores $min_th = \frac{1}{3}max_th = 1111$ pacotes, $max_p = 0.1$ e $w_q = 0.002$ e a opção *gentle_* habilitada.

As simulações foram realizadas utilizando tanto tráfego de longa-duração (FTP), como tráfego de curta-duração (WEB). O cenário utilizado é o mesmo da Figura 7.1, Seção 7.1. Para cada tipo de tráfego, várias métricas foram consideradas e seus valores médios obtidos em função da carga na rede são apresentados nos gráficos que seguem.

Para derivar os valores mostrados nos gráficos, utilizou-se o método de replicação independente com intervalo de confiança de 95%

As métricas consideradas para coleta e análise são:

- **Tamanho médio da fila:** esta métrica permite verificar como a política de AQM afeta o tráfego agregado que chega ao enlace gargalo. O tamanho médio da fila indica como a política de AQM atua preventivamente no controle do congestionamento. Esta é uma métrica de grande importância dado que o RTT e o *jitter* estão diretamente relacionados a ela. O tamanho médio da fila é dado em número de pacotes de tamanho de 1500bytes ;
- **Vazão:** esta métrica e as subsequentes estão relacionadas às conexões ativas, ou seja, como as políticas de AQM afetam o desempenho das conexões HSTCP. A vazão é apresentada em percentagem e é a razão entre a quantidade total de *bytes* enviados e a capacidade do enlace;
- **Tamanho médio da janela de congestionamento(cwnd):** O valor da vazão média decorre do tamanho média da janela, ou seja, quanto maior o tamanho médio da janela, maior a vazão média obtida. O valor do tamanho da janela é dado em pacotes de 1500 bytes ;
- **Goodput médio:** esta métrica indica o quão eficiente é a utilização da banda passante, ou seja, qual a percentagem de dados enviados que realmente chegam ao receptor, ou seja, que não foram perdidos. O *goodput* ou vazão útil é dado por: $(nbytes - nretbytes)/nbytes$, onde *nbytes* é o número de *bytes* enviados, e *nretbytes* é quantidade de *bytes* retransmitidos. A vazão útil é um valor no intervalo $[0, 1]$;
- **RTT médio:** determina o atraso sofrido pelos pacotes da conexões devido a variação do tamanho da fila gerado por uma determinada política de AQM;
- **RTO médio:** quando uma perda não é detectada por meio de ACKs, ela será detectada por meio de expiração do intervalo de temporização (*timeout*). Quando o congestionamento é intenso podem acontecer perdas de rajadas de pacotes, o que aumenta a probabilidade de ocorrência de *timeouts*, com a consequente degradação do desempenho das conexões. Perdas de múltiplos segmentos podem afetar a vazão e o tempo de transferência das conexões. As políticas de AQM são diretamente responsáveis pelo descarte de pacotes, portanto, a política de AQM deve apresentar uma taxa de descarte de pacotes proporcional, de modo a evitar a continuidade de perdas de pacotes, assim como, reduzir a ocorrência de RTOs;

- **Tempo médio de transferência:** o HSTCP reduz sua taxa de transmissão na presença de congestionamento, indicado pelo descarte de pacotes. Desta forma, as perdas sofridas pelas conexões impactam na vazão, e conseqüentemente na latência da transferência do arquivo;
- **Número de conexões ativas:** esta métrica avalia a eficiência da política de AQM em fazer o melhor uso dos recursos da rede de forma a permitir que um maior número de conexões utilize os recursos para um mesmo período de simulação.

7.3.1 Experimentos Utilizando tráfego de Longa Duração

Nesta seção, são apresentados os resultados obtidos nos experimentos realizados com tráfego de longa duração (FTP), usando enlace com um único gargalo.

Na Figura 7.8, apresenta-se o tamanho médio da fila em função da carga. Conforme o esperado, o aumento do congestionamento no enlace leva ao crescimento do tamanho da fila. A fila do HSTCP-H2 é no mínimo 42% maior do que a fila de RED, chegando a ser quase três vezes maior com carga de 1.0. Filas maiores significam que um volume maior de dados foi transferido pela rede e, conseqüentemente, a utilização da banda passante é, também, maior. Desta forma, o HSTCP-H2 apresenta maior utilização dos recursos disponíveis para todo o intervalo de cargas testado. É importante ressaltar que um dos objetivos de projeto do controlador é, "prevenir a subutilização do enlace", ficando, assim, evidente a eficácia do controlador em atingir o objetivo almejado. Além de apresentar tamanho de fila menor, é possível verificar ainda, que a fila de RED cresce lentamente, sendo praticamente constante para cargas entre 0.9 e 1.0, demonstrando, assim, a pouca flexibilidade desta política em se ajustar rapidamente diante de flutuações do nível de congestionamento.

A Figura 7.9 apresenta a percentagem média de banda passante obtida, ou vazão, por conexão ativa em função da carga. A vazão obtida pelo HSTCP-H2 é, consideravelmente, maior do que a de RED. Esta diferença torna-se mais acentuada a medida que o congestionamento intensifica-se. A vazão do HSTCP-H2 é, respectivamente, no mínimo 40% maior e no máximo 109% maior do que a de RED para cargas entre 0.4 e 1.0. Estes valores justificam o tamanho médio da fila mostrado anteriormente, ou seja, confirmam a maior ocupação do enlace pelo HSTCP-H2. A vazão reflete a quantidade de dados que a janela de congestionamento injeta na rede. A Figura 7.10 mostra o tamanho médio da janela de congestionamento por conexão ativa, dada em unidades de MSS. O tamanho

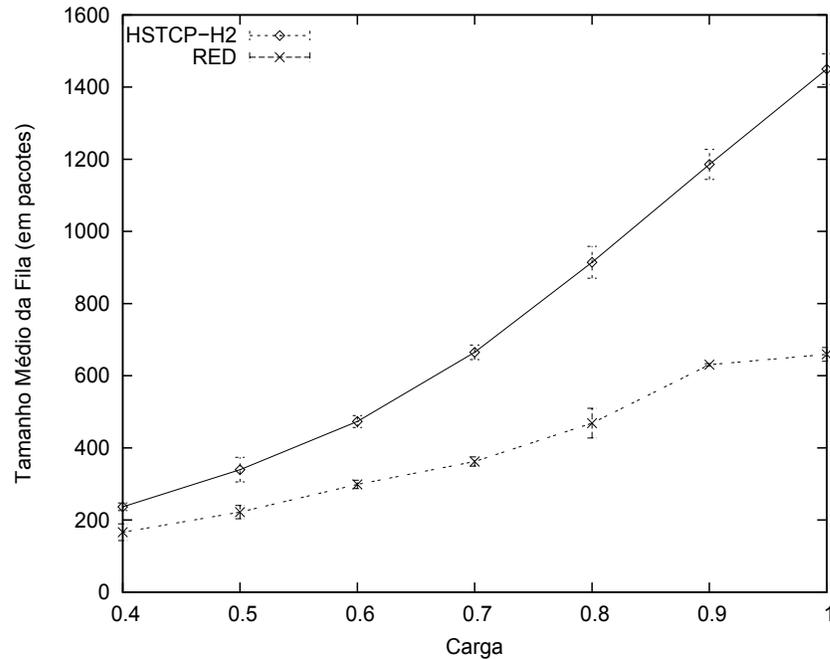


Figura 7.8: Tráfego FTP: tamanho médio da fila em função da carga

de janela obtida pelo HSTCP-H2 é, consideravelmente, maior do que o tamanho da janela de RED. Sob carga de 0.4, a janela do HSTCP-H2 é 107% maior que a de RED, chegando a ser três vezes maior sob carga de 0.9. A diferença entre os valores de janela do HSTCP-H2 e de RED são, novamente, justificados pelas características das políticas de AQM empregadas. Para alcançar o objetivo de “prevenir a subutilização do enlace”, o HSTCP-H2 deve sinalizar o descarte/marcação aos emissores HSTCP, de modo que estes possam aumentar a sua janela de congestionamento e, conseqüentemente, maximizar a sua taxa de transmissão. Por outro lado, o RED, com suas limitações de ajustes diante de variações das condições do meio, não consegue realizar uma sinalização eficiente aos emissores HSTCP e, conseqüentemente, as janelas não crescem tanto quanto poderiam.

É possível observar ainda os elevados valores de janela de congestionamento, que chegam a ter 2744 segmentos. Estes resultados evidenciam, a eficiência do HSTCP em ajustar adequadamente a janela congestionamento em ambientes com produto banda-atraso elevado.

Uma métrica de grande importância para avaliar os mecanismos de controle de congestionamento é o *goodput*, que mede o quanto a banda é utilizada para transmissões úteis. O *goodput* é função do tamanho da janela de congestionamento e da quantidade de perdas, ou seja, quanto maior a janela de congestionamento e quanto menor a quantidade de perdas, maior é o *goodput*. Dado o tamanho médio de janela de congestionamento obtido

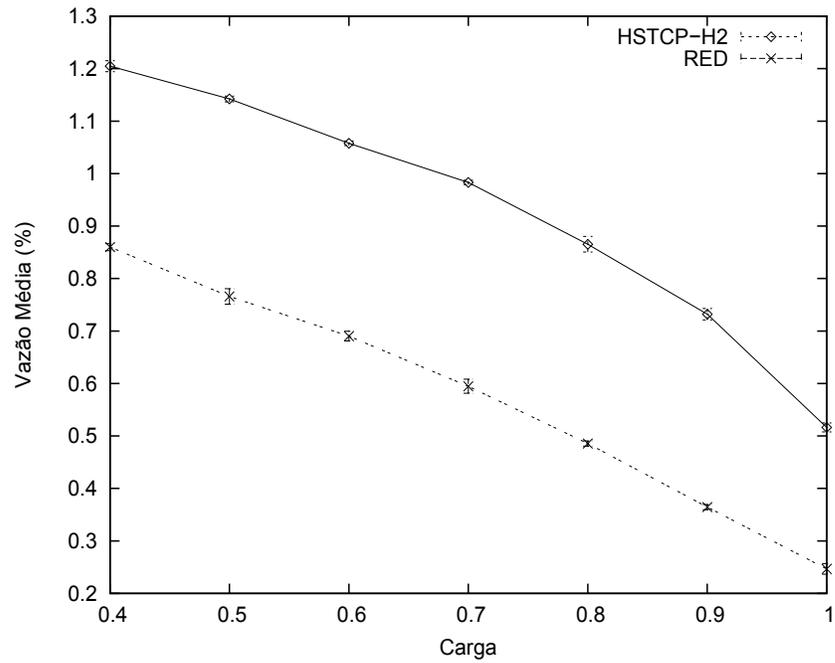


Figura 7.9: Tráfego FTP: vazão média por conexão ativa em função da carga

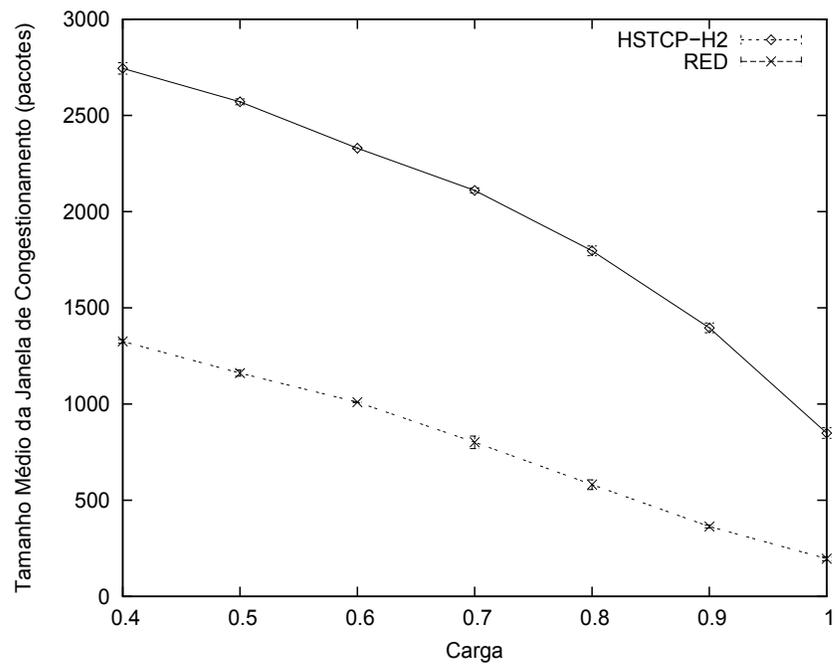


Figura 7.10: Tráfego FTP: tamanho médio da janela de congestionamento por conexão ativa em função da carga

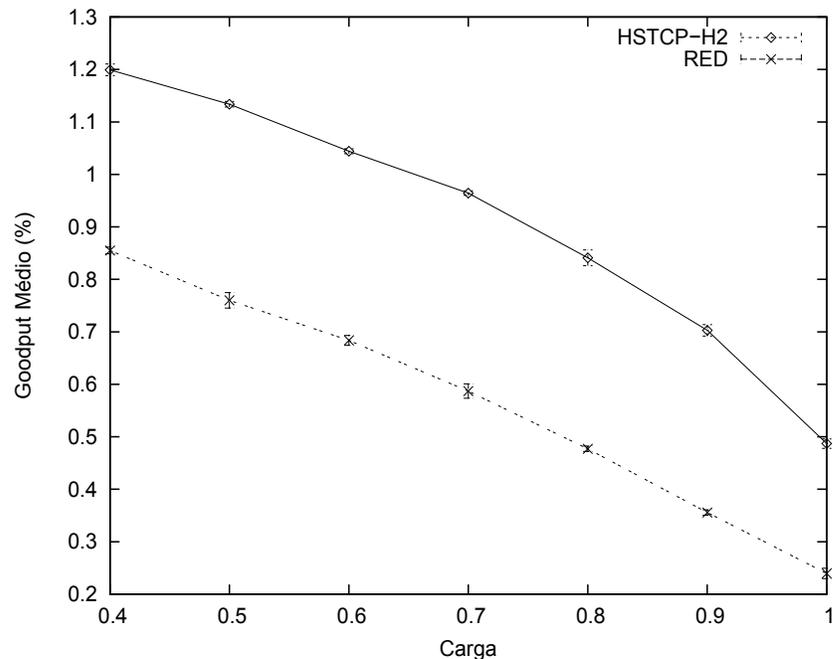


Figura 7.11: Tráfego FTP: goodput médio por conexão ativa em função da carga

e os baixos valores de perda (Figura 7.12), o *goodput* por conexão ativa (Figura 7.10) é muito maior quando o controlador HSTCP-H2 é usado do que quando RED é utilizado. O *goodput* por conexão obtido pelo controlador HSTCP-H2 é 40% maior do que o de RED sob carga de 0.4. Sob carga de 0.9 a diferença pode ser da ordem de 100%. Estes resultados tornam evidente a vantagem de empregar HSTCP-H2 para conexões individuais. Os valores de *goodput* evidenciam que o HSTCP-H2 é capaz de capitalizar uma maior utilização do enlace e maiores tamanhos de fila produzindo maiores valores de vazão útil por conexão.

A Figura 7.12 apresenta o número médio de retransmissões devido ao esgotamento do temporizador (RTO) por conexão ativa. Sob congestionamento leve a moderado, o HSTCP-H2 apresenta um número de RTO's menor do que o de RED. Sob carga de 0.4, o número de RTO's apresentado pelo HSTCP-H2 é 78% menor do que os valores de RED. Conforme o congestionamento se intensifica, a diferença entre o HSTCP-H2 e RED diminui devido a perda de um grande número de pacotes. Entretanto, com carga de 1.0, a longa fila produzida pelo HSTCP-H2 leva a um número de perdas maior do que os produzidos por RED. O tamanho da fila de RED (Figura 7.8) quando o congestionamento está na sua intensidade máxima é praticamente constante, enquanto que o tamanho da fila do HSTCP-H2 continua a crescer. Como se pode observar na Figura 7.11, devido aos valores de vazão, o HSTCP-H2 continua a obter valores de *goodput* mais elevados.

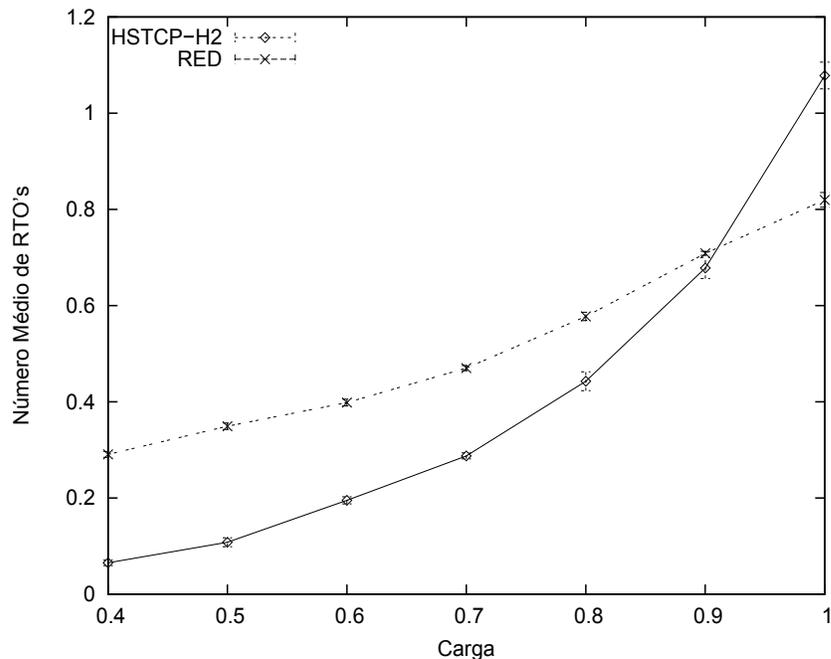


Figura 7.12: Tráfego FTP: número médio de RTO por conexão ativa em função da carga

Embora o HSTCP-H2 gere tamanhos de fila maiores que o RED, ele ainda mantém baixos valores para o RTT médio (Figura 7.13). Pode-se observar que, independentemente do nível de congestionamento na rede, o HSTCP-H2 mantém o RTT médio menor do que o de RED e com diferença praticamente constante. Tal resultado é uma consequência do critério de “minimização de *jitter*” adotado como um objetivo de projeto para o controlador HSTCP-H2. Este resultado reforça a importância da utilização de controladores ótimos, os quais permitem a definição de objetivos específicos no projeto de AQM.

Com redução do número de RTO's, aumenta-se a vazão de dados e o tempo de vida da conexão é reduzido, ou seja, a latência na transferência dos arquivos é reduzida. Na Figura 7.14, pode-se observar uma redução significativa da latência do HSTCP-H2, a qual é no mínimo 49% menor do que RED sob carga 0.4, e no máximo 67% menor sob carga 0.9. Com latência menor, a utilização dos recursos é otimizada, permitindo que um número maior de conexões ocupem o enlace quando HSTCP-H2 é a política de AQM empregada. Na Figura 7.15, apresenta-se o número médio de conexões em função da carga. Sob cargas leves, a diferença entre o número de conexões obtidas pelo HSTCP-H2 e por RED é pequena. No entanto, quando o nível de congestionamento eleva-se a diferença passa a ser mais acentuada. Sob carga 1.0, as simulações com HSTCP-H2 apresentam aproximadamente 630 conexões a mais do que quando RED está em uso.

Os resultados obtidos demonstram o desempenho do controlador ótimo HSTCP-H2 em

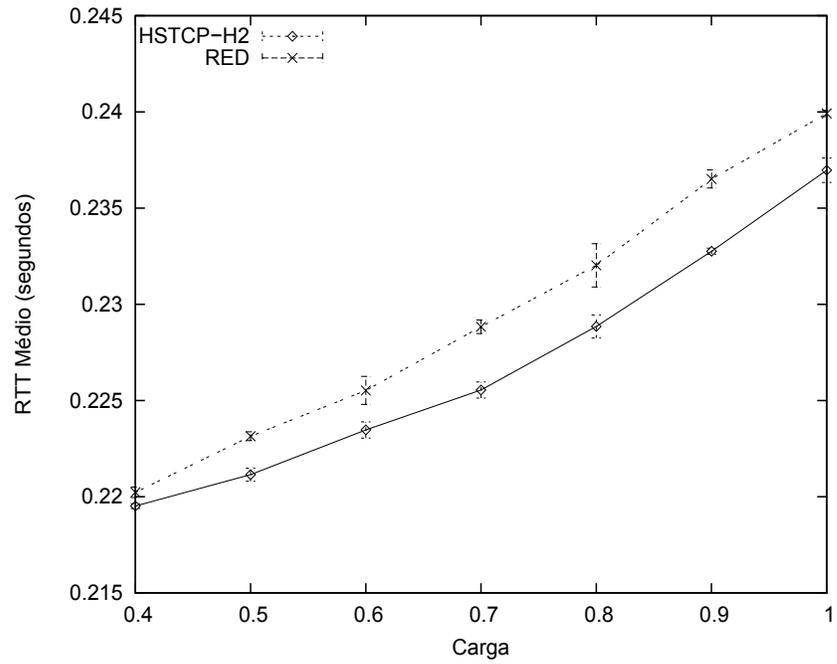


Figura 7.13: Tráfego FTP: RTT médio por conexão ativa em função da carga

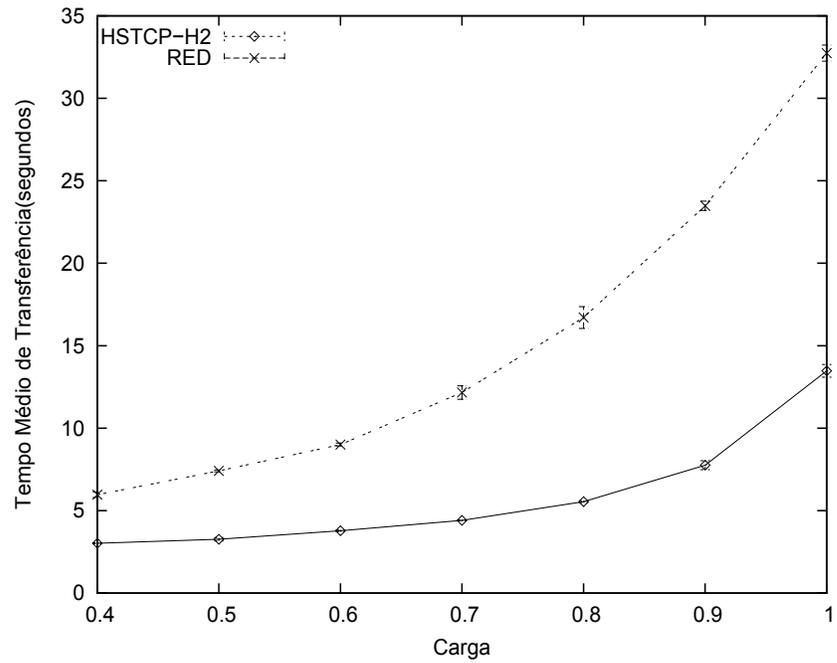


Figura 7.14: Tráfego FTP: tempo de transferência por conexão ativa em função da carga

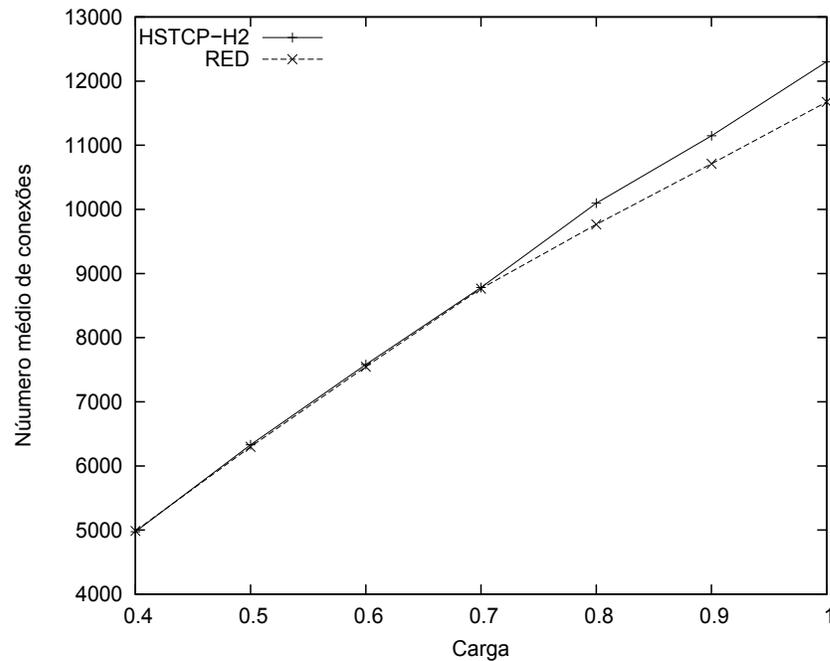


Figura 7.15: Tráfego FTP: média do número de conexões ativa em função da carga

relação a política de AQM padrão RED quando o tráfego é do tipo FTP. O HSTCP-H2 é mais eficiente tanto quando se considera o tamanho de fila, que considera a influência da política de AQM no tráfego agregado, bem como outras métricas que consideram a influência da política de AQM sobre as conexões ativas. Pode-se verificar, portanto, que o HSTCP-H2 é uma política de AQM que apresenta utilização de banda consideravelmente superior a produzida por RED; resultado este, que é consequência das altas taxas de vazão e elevados valores de janela de congestionamento obtidas pelo HSTCP-H2. Além disso, os ajustes do HSTCP-H2 levam a um número de RTO's inferior ao número de RTO's de RED, exceto quando o congestionamento está na sua intensidade máxima, esta é uma consequência dos valores elevados do tamanho de fila produzidas pelo HSTCP-H2 enquanto o RED mantém o tamanho da fila aproximadamente constante. O HSTCP-H2 apresenta, também, *goodput* significativamente maior do que RED, enquanto mantém baixos valores de RTT.

7.3.2 Experimentos Utilizando tráfego de Curta Duração

Nesta seção, são apresentados os resultados obtidos nos experimentos com tráfego de curta duração (WEB) em enlace com um único gargalo.

Na Figura 7.16, apresenta-se o tamanho médio da fila em função da carga. É possível

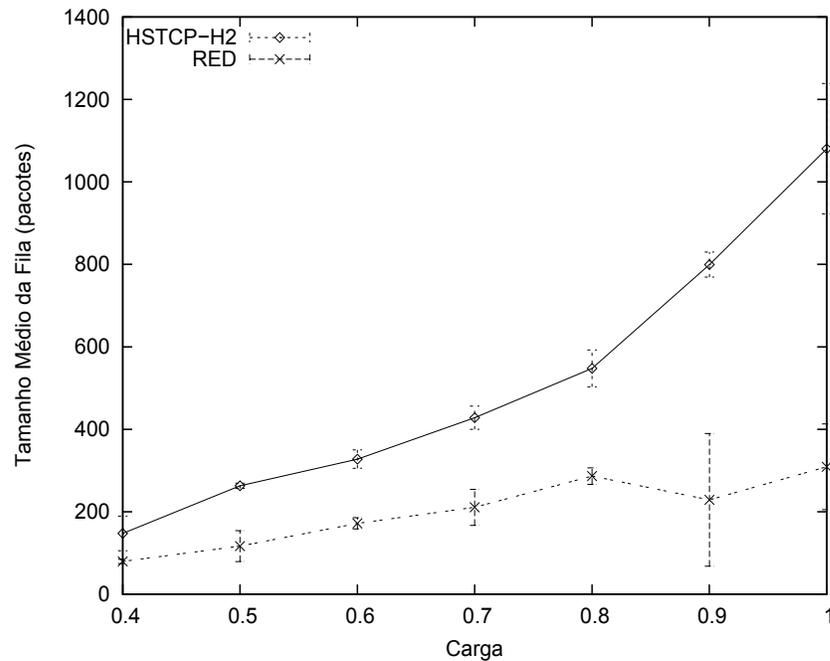


Figura 7.16: Tráfego WEB: tamanho médio da fila em função da carga

observar que quando HSTCP-H2 é a política de AQM utilizada o tamanho da fila aumenta consideravelmente com o aumento da carga, ao passo que a fila gerenciada por RED cresce lentamente. Isto demonstra que a capacidade do HSTCP-H2 em utilizar a banda passante é significativamente maior do que RED. Comparando-se o tamanho de fila do HSTCP-H2 sob tráfego WEB ao tamanho da fila sob tráfego FTP (Figura 7.8), constata-se que os valores de fila sob tráfego de longa duração são maiores que os produzidos quando sob tráfego de curta duração, o que era esperado dado as características do tráfego WEB em produzir rajadas menores e, portanto, apresentar menor ocupação da fila.

As Figuras 7.17 e 7.18 mostram, respectivamente, a vazão média e o tamanho médio da janela de congestionamento obtidos por conexão ativa em função da carga. Assim como acontece com tráfego FTP, para tráfego WEB o HSTCP-H2 apresenta maior percentagem de vazão do que RED. A vazão obtida pelo HSTCP-H2 sob carga de 0.4 e 0.8 é, respectivamente, no mínimo 5% maior e no máximo 19% maior do que RED. A vazão determina a utilização do enlace. Um dos objetivos do HSTCP-H2 é exatamente minimizar a subutilização do enlace, ficando comprovada a sua eficiência quando comparada a RED. Este resultado é consequência dos maiores valores de janela (Figura 7.18), obtidos quando o HSTCP-H2 é utilizado. O tamanho da janela para HSTCP-H2 é significativamente superior a produzida por RED. A janela para HSTCP-H2 é no mínimo 22% maior e no máximo 53% maior do que a produzida por RED sob cargas entre 0.4 e 1.0.

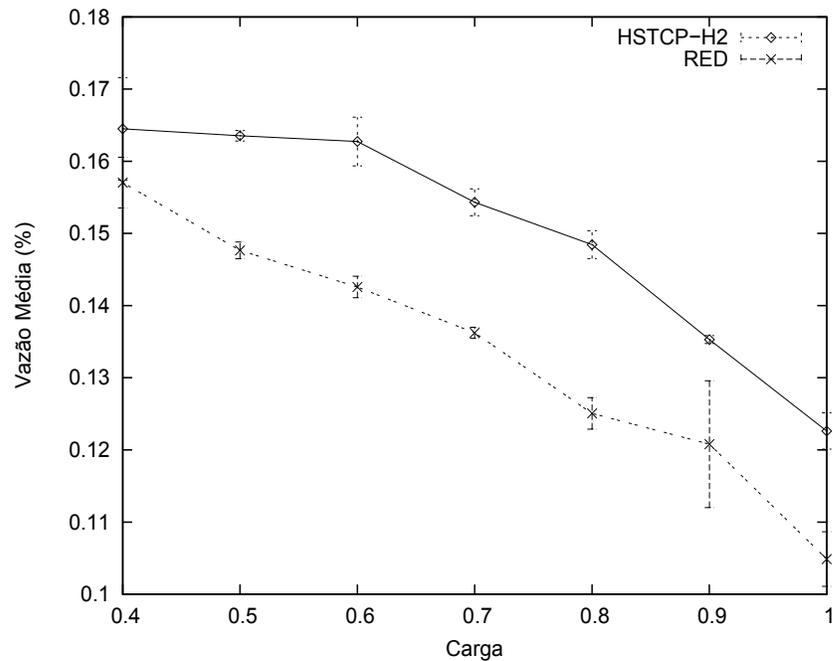


Figura 7.17: Tráfego WEB: vazão média por conexão ativa em função da carga

Comparando-se os valores de janela de congestionamento sob tráfego WEB e sob tráfego FTP, verifica-se que os valores para tráfego WEB são menores, o que acontece devido ao fato das conexões de curta duração terem uma quantidade menor de dados a transmitir.

Nas Figuras 7.20 e 7.21, são mostrados, respectivamente, o RTT e o número de RTOs médio por conexão ativa em função da carga. Verifica-se que os valores para RTT produzidos por RED são menores do que os valores produzidos por HSTCP-H2. No entanto, esta diferença é pequena, chegando a ser no máximo 4% menor do que HSTCP-H2, quando o congestionamento está na sua intensidade máxima, ou seja, sob carga de 1.0. O HSTCP-H2 produz valores de RTT maiores devido ao maior tamanho de fila. Como RED apresenta tamanho de fila praticamente constante, era de se esperar que apresentasse RTT com pouca variação. A quantidade de dados transmitido pelo tráfego WEB é menor e, portanto, diminui a probabilidade de um pacote ser retransmitido devido a sinalização por ACK, aumentando, assim, a necessidade de *timeouts* para que a perda seja reconhecida e o pacote retransmitido, ou seja, o número de RTO's tende a aumentar. Na Figura 7.21, é possível verificar que para as cargas iniciais de 0.4 e 0.5 o número de RTOs do HSTCP-H2 é cerca de 12% menor do que o número de RTOs gerados por RED. No entanto, a medida que o congestionamento intensifica-se o HSTCP-H2 apresenta maior número de RTOs do que RED, sendo no mínimo 3% maior para carga de 0.6 e no máximo 77% para carga 1.0. Isto acontece devido ao fato de HSTCP-H2 ter uma quantidade significativamente maior

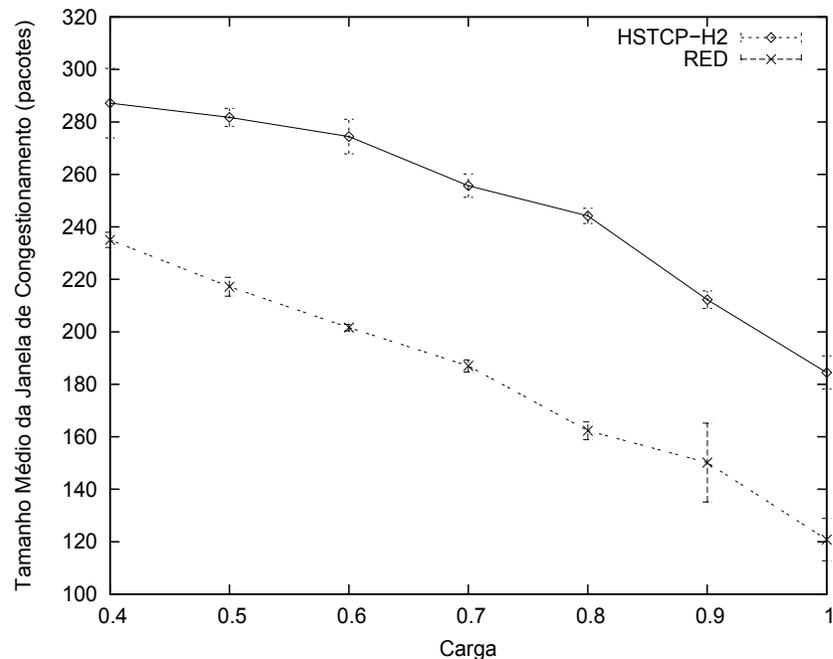


Figura 7.18: Tráfego WEB: tamanho médio da janela de congestionamento por conexão ativa em função da carga

de pacotes enfileirados. Como para tráfego WEB a quantidade de dados a transmitir é bem menor, os pacotes enfileirados por HSTCP-H2 possuem uma probabilidade maior de serem considerados perdidos, enquanto ainda permanecem na fila, do que os pacotes que estão enfileirados por RED em uma fila de tamanho praticamente constante.

Embora apresente número de RTO's maior do que RED, em decorrência dos valores elevados de janela de congestionamento, o *goodput* obtido por conexão ativa é consideravelmente maior quando HSTCP-H2 é empregado. Na Figura 7.19, pode-se verificar que o *goodput* obtido por HSTCP-H2 é no mínimo 7% maior do que RED sob cargas de 0.4 e no máximo 19% sob cargas de 0.8. Estes resultados reafirmam a superioridade do HSTCP-H2 em utilizar a banda passante de forma eficiência tanto sob tráfego FTP quanto sob tráfego WEB.

As Figura 7.22 e 7.23 apresentam o tempo de transferência e o número de conexões em função da carga. Mesmo tendo obtido um número maior de RTOs para congestionamento intenso, a vazão de dados apresentada pelo HSTCP-H2 é maior, levando a uma latência de transferência de arquivos menor. O HSTCP-H2 apresenta valores de tempo de transferência menores do que os gerados por RED para todo o intervalo de cargas testado, sendo no mínimo 5% menor do que RED sob carga de 0.4 e no máximo 13% menor sob carga de 0.8. Com menor latência, um número maior de conexões utilizam o enlace, ou seja, o

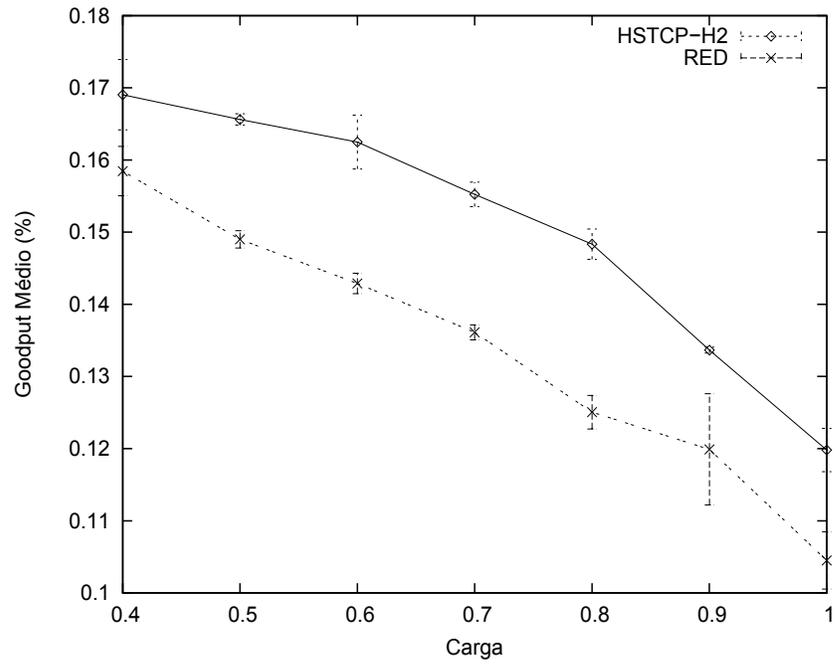


Figura 7.19: Tráfego WEB: goodput médio por conexão ativa em função da carga

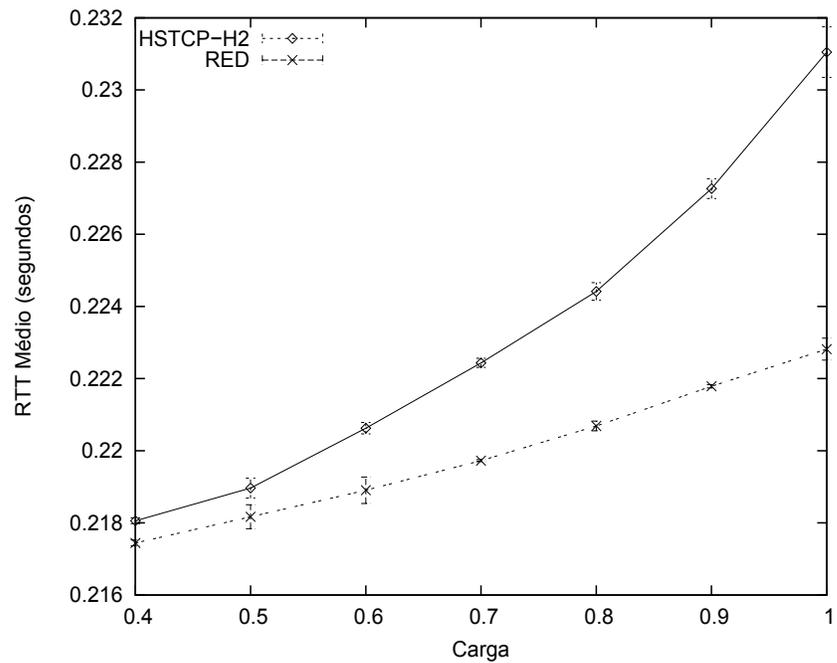


Figura 7.20: Tráfego WEB: RTT médio por conexão ativa em função da carga

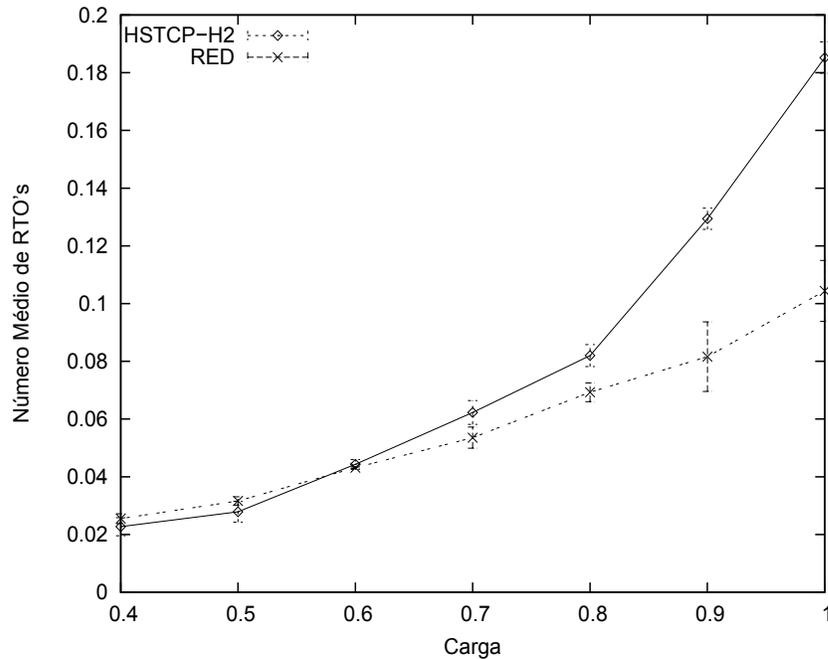


Figura 7.21: Tráfego WEB: número médio de RTO's por conexão ativa em função da carga

número de conexões ativas que podem utilizar a mesma quantidade de recursos durante um período de tempo é maior. Para todo o intervalo de cargas testado o HSTCP-H2 apresenta um número maior de conexões ativas (Figura 7.23), sendo que a maior diferença é de 12808 conexões sob carga 1.0.

Os resultados obtidos demonstram o desempenho do controlador HSTCP-H2 em relação a política AQM RED sob o tráfego WEB. Devido ao elevado tamanho de fila obtido pelo HSTCP-H2, e as características do tráfego WEB, o controlador apresenta desempenho inferior ao de RED para as métricas de RTT e RTO. Mesmo assim, o HSTCP-H2 utiliza mais eficientemente a banda passante. Estes resultados confirmam que o controlador ótimo HSTCP-H2 é, também, eficiente quando o tráfego é do tipo WEB em redes com produto banda-atraso elevado.

7.4 Cenários das Simulações com Múltiplos Enlaces Gargalo

Esta seção apresenta as configurações e o cenário utilizados nos experimentos para enlace com múltiplos gargalos.

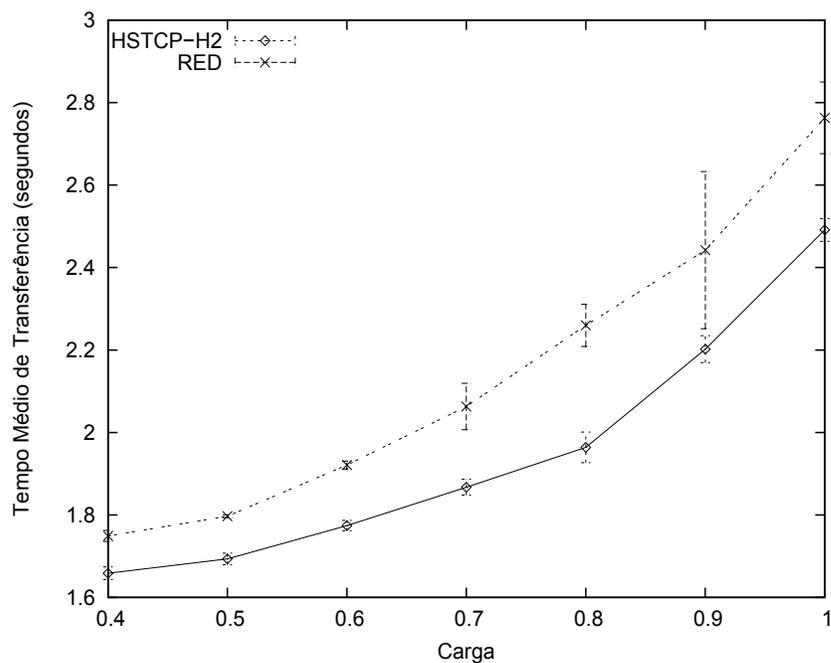


Figura 7.22: Tráfego WEB: tempo médio de transferência por conexão ativa em função da carga

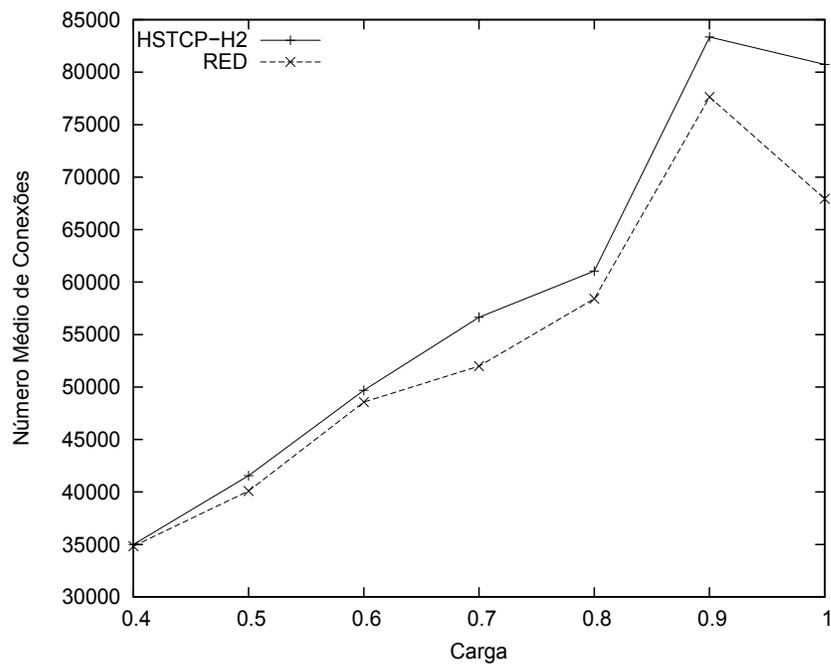


Figura 7.23: Tráfego WEB: número médio de conexões em função da carga

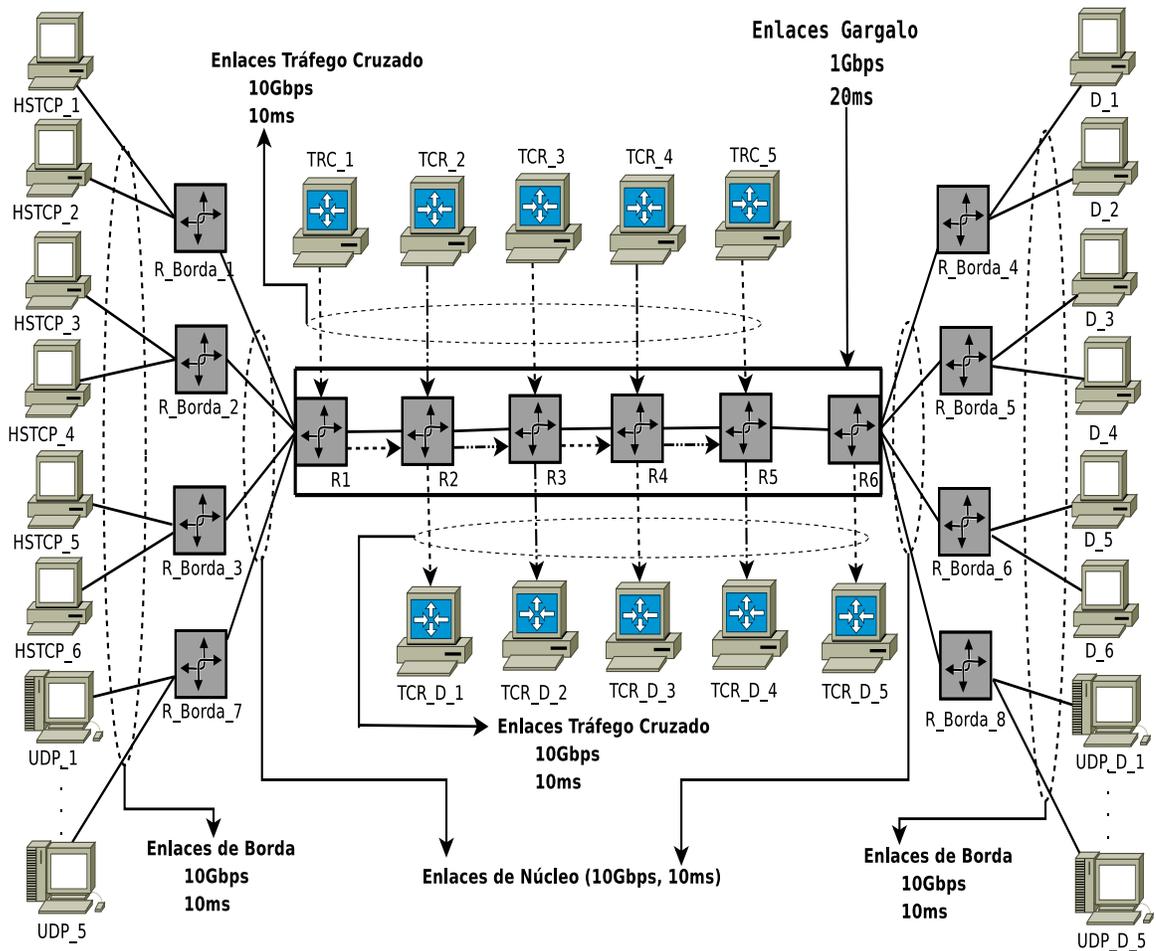


Figura 7.24: Topologia utilizada nos experimentos com múltiplos enlaces gargalo

Na Figura 7.24, representa-se a topologia, a capacidade dos enlaces e o tempo de propagação utilizados para os experimentos com múltiplos enlaces gargalo, conhecida como topologia *Parking-Lot*. Além de múltiplos enlaces gargalos, considerou-se, também, para estes experimentos a existência de tráfego cruzado. O intuito ao considerar tal cenário é verificar o desempenho e a eficácia do controlador diante de cenários mais elaborados, que apresentam maior proximidade aos de redes operacionais [Floyd and Kohler, 2002] [Ha et al., 2006]. Os enlaces entre os nós R_1 e R_2 , R_2 e R_3 , R_2 e R_4 , R_4 e R_5 , R_5 e R_6 formam os múltiplos enlaces gargalo da topologia. Cada um dos enlaces gargalo possui capacidade de 1Gbps e atraso de 20ms. Os demais enlaces presentes na topologia possuem uma capacidade de 10Gbps e atraso de 10ms. O tamanho do *buffer* utilizado é de 3333 pacotes o que corresponde a 20% do produto banda-atraso [Barman et al., 2004].

Para verificar a eficiência da política HSTCP-H2-AQM (Algoritmo 4), variou-se o grau de congestionamento na rede, utilizando cargas de 0.4 até 1.0 da capacidade do

enlace. Para gerar as cargas específicas foi empregado o gerador de tráfego Traffic-Gen [Cardoso and Rezende, 2000]. Foram considerados tráfego de longa duração (FTP) e curta duração (WEB). De modo a considerar um ambiente de rede realista, foram inseridos também fluxos não-adaptativos CBR/UDP para gerar o tráfego de ruído.

As simulações foram compostas por 80% [Fomenkov et al., 2004] de tráfego de longa duração ou de curta duração como tráfego principal, 10% de tráfego de ruído composto por tráfego não-adaptativos e 10% de tráfego cruzado composto, também, por tráfego não-adaptativo.

A variação TCP utilizada foi o *High Speed* TCP. O tamanho dos segmentos gerados foi de 1500 *bytes*, que corresponde ao tamanho máximo do quadro do padrão Ethernet.

O tráfego principal HSTCP foi gerado nos nós origem *HSTCP_1*, *HSTCP_2*, *HSTCP_3*, *HSTCP_4*, *HSTCP_5* e *HSTCP_6* para seus respectivos nós destino *D_1*, *D_2*, *D_3*, *D_4*, *D_5*, *D_6*. O tráfego ruído foi gerado nos nós origem *UDP_1*, *UDP_2*, *UDP_3*, *UDP_4* e *UDP_5* para os respectivos nós destinos *UDP_D_1*, *UDP_D_2*, *UDP_D_3*, *UDP_D_4* e *UDP_D_5*. O tráfego cruzado é gerado nos nós origem *TRC_1*, *TRC_2*, *TRC_3*, *TRC_4* e *TRC_5* para os nós destinos *TRC_D_1*, *TRC_D_2*, *TRC_D_3*, *TRC_D_4*, *TRC_D_5*, respectivamente.

O tráfego WEB é gerado usando uma distribuição híbrida Lognormal/Pareto. O corpo da distribuição corresponde a uma área de 0.88 e é modelado por uma distribuição Lognormal com média de 7247 *bytes*, já a cauda é modelada por uma distribuição Pareto com média de 10558 *bytes* [Barford et al., 1999]. O tráfego FTP é gerado, utilizando-se uma distribuição exponencial com média de 512 *KBytes*.

O controle de fluxo foi desabilitado ajustando-se a janela do receptor para um valor elevado (100000 segmentos) de forma que o crescimento da janela do transmissor fosse governada apenas pelo congestionamento na rede e não por limitações do controle de fluxo no receptor.

Os ajustes de RED foram realizados utilizando o limiar *max_th* de 20% do produto banda-atraso, ou seja, 3333 pacotes. Os ajustes dos demais parâmetros consideram as recomendações em [Floyd, 1997a]. Assim: $min_th = \frac{1}{3}max_th = 1111$ pacotes, $max_p = 0.1$ e $w_q = 0.002$. Seguindo as recomendações em [Floyd, 2000], a opção *gentle_* foi habilitada.

Para derivar os valores mostrados nos gráficos utilizou-se o método de replicação independente com intervalo de confiança de 95%

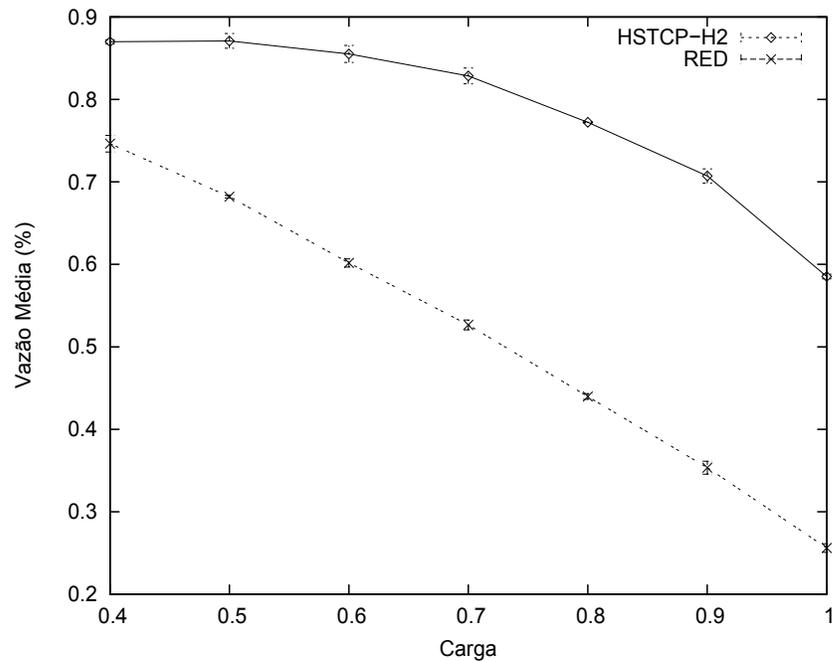


Figura 7.25: Tráfego FTP: vazão média por conexão ativa em função da carga

7.4.1 Experimentos Utilizando tráfego de Longa Duração

Nesta seção, são apresentados os resultados obtidos nos experimentos realizados com tráfego de longa duração (FTP) em enlaces com múltiplos gargalos.

Na Figura 7.25, apresenta-se a vazão obtida por conexão ativa em função da carga. Independentemente da intensidade da carga, o HSTCP-H2 produz vazões maiores do que RED, sendo que a diferença acentua-se a medida que o congestionamento torna-se intenso, o que demonstra a eficiência dos ajustes do HSTCP-H2 diante de variações das condições da carga. A vazão obtida pelo HSTCP-H2 é no mínimo 17% maior, e no máximo 129% maior do que a de RED sob cargas entre 0.4 e 1.0. Um ponto interessante a ser observado é que a diferença entre vazão de HSTCP-H2 e RED é maior quando o enlace com múltiplos gargalos é considerado quando comparado a cenários com enlace de gargalo único (Figura 7.9), isto demonstra a capacidade do HSTCP-H2 de ocupar o enlace em topologia com múltiplos enlaces gargalos e níveis de congestionamento distintos. Por outro lado, RED, com seus ajustes baseados em heurísticas, não consegue se adaptar de maneira rápida o suficiente às variações de carga, levando, assim, a redução drástica na transmissão de dados.

A Figura 7.26 apresenta o tamanho médio da janela de congestionamento por conexão ativa em função da carga. A janela de congestionamento é o mecanismo responsável por

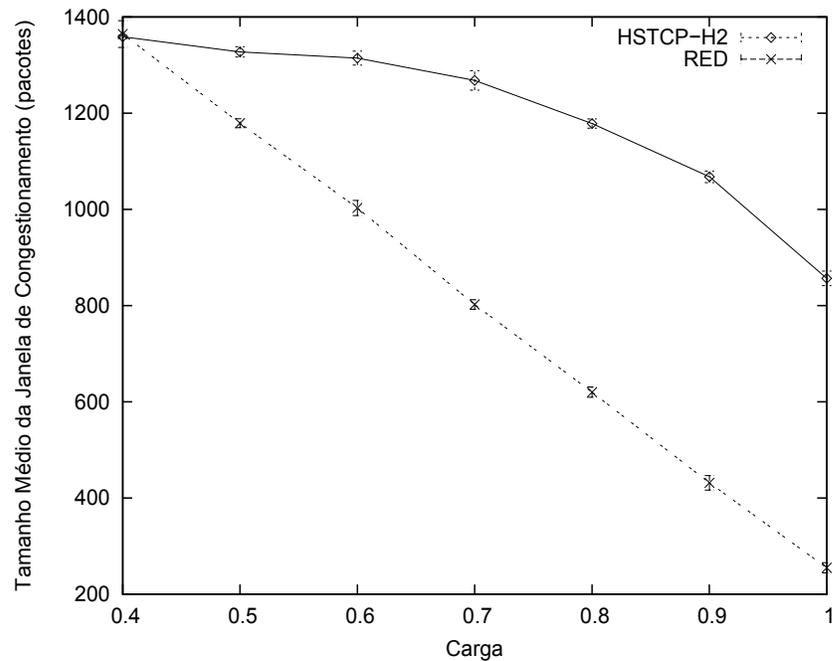


Figura 7.26: Tráfego FTP: tamanho médio da janela de congestionamento por conexão ativa em função da carga

inserir dados na rede, portanto, os altos valores de vazão obtidos resultam do tamanho da janela. O tamanho da janela para o HSTCP-H2 é no mínimo 13% maior do que o tamanho da janela para RED com carga de 0.5, chegando a ser 236% maior do que a de RED sob carga de 1.0. Comparando a diferença do tamanho da janela entre HSTCP-H2 e RED para redes com um único enlace gargalo (Figura 7.10), consta-se que a diferença entre HSTCP-H2 e RED aumenta a medida que o congestionamento se intensifica, em cenários com múltiplos gargalos. Os resultados obtidos para o tamanho da janela ratificam a capacidade do HSTCP-H2 em lidar com condições adversas, ao passo que o desempenho de RED degrada-se nas mesmas condições. Destacam-se, ainda, na Figura 7.10, os elevados valores da janela de congestionamento, o que é uma consequência direta da utilização do protocolo HSTCP, que é um protocolo para redes com produto banda-atraso elevado, e permite que a janela de congestionamento seja ajustada de forma a atingir valores elevados e, assim, utilizar a banda passante disponível.

Além de uma vazão elevada, espera-se que um mecanismo de controle de congestionamento utilize os recursos disponíveis de maneira eficiente, ou seja, que apresente valores de *goodput* elevado. Na Figura 7.27, apresenta-se o *goodput* médio obtido por conexão ativa em função da carga. Pode-se verificar que para todo o intervalo de cargas testado o HSTCP-H2 apresenta taxas de *goodput* superiores as apresentadas por RED, sendo no

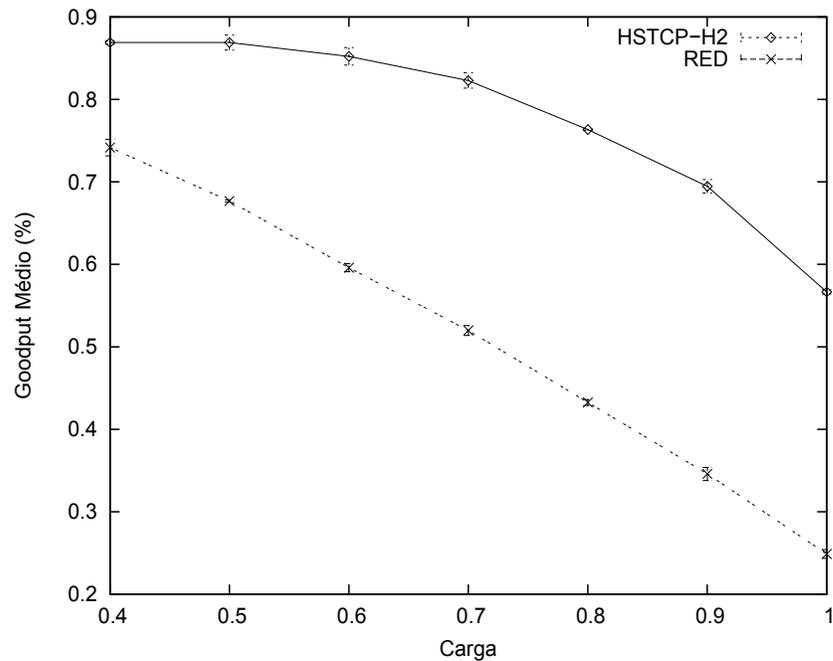


Figura 7.27: Tráfego FTP: goodput médio por conexão ativa em função da carga

mínimo 17% maior e no máximo 128% maior do que RED sob cargas de 0.4 e 1.0. Tanto para múltiplos enlaces gargalos quanto para enlace com único gargalo (Figura 7.11), a diferença entre o *goodput* produzido por HSTCP-H2 e o *goodput* produzido por RED se acentua conforme o congestionamento aumenta. O HSTCP-H2 é um mecanismo de controle de congestionamento que apresenta maior capacidade em utilizar de maneira eficiente os recursos disponíveis.

A Figura 7.28 apresenta o número médio de retransmissão devido ao esgotamento do temporizador (RTO) obtido por conexão ativa em função da carga. Sob diferentes cargas, o HSTCP-H2 apresenta um número menor de RTO's do que RED, sendo que a diferença minimui sob cargas maiores, o que era de se esperar, dado que congestionamento com maior intensidade leva a uma probabilidade de perdas também maior. O número de RTO's gerados por HSTCP-H2 é no máximo 70% menor e no mínimo 19% menor do que os gerados por RED para cargas entre 0.4 e 1.0. Uma menor quantidade de perdas, e valores elevados para a janela de congestionamento justificam os valores de *goodput* elevados obtidos pelo HSTCP-H2 (Figura 7.27).

Na Figura 7.29, apresenta-se o RTT médio por conexão ativa em função da carga. Pode-se verificar que os valores de RTT para HSTCP-H2 e RED são muito próximos. O HSTCP-H2 apresenta, para cargas baixas, RTT médio menor. No entanto, a medida que o congestionamento aumenta, o RTT gerado por HSTCP-H2 é ligeiramente maior, chegando

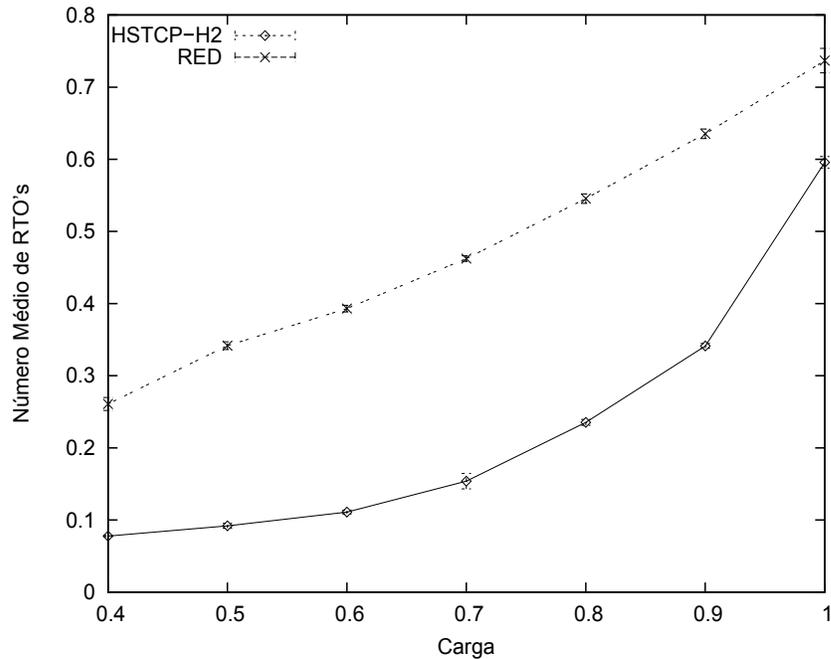


Figura 7.28: Tráfego FTP: número médio de RTO por conexão ativa em função da carga

a ser no máximo 2% maior do que a de RED sob carga 1.0. Esta diferença no valor de RTT produzido por HSTCP-H2 e produzido por RED decorre do fato do controlador HSTCP-H2 proporcionar maior ocupação do enlace, o que é constatado pelos valores elevados de vazão. A ocupação das filas nos enlaces gargalo também é, conseqüentemente, maior, o que eleva o atraso.

Nas Figuras 7.30 e 7.31, apresentam-se, respectivamente, o tempo de transferência e o número de conexões. Com o aumento de vazão e diminuição no número de RTO's, o tempo de vida das conexões diminui, ou seja, a latência na transferência dos arquivos é reduzida. O tempo de transferência por conexão ativa (Figura 7.30) obtido pelo HSTCP-H2 é no mínimo 37% menor e no máximo 73% menor do que RED, sob cargas entre 0.4 e 0.9. O HSTCP-H2 permite que um número maior de conexões ocupem o enlace (Figura 7.31) e, conseqüentemente, maximiza a utilização dos recursos. O número de conexões permitido sob HSTCP-H2 é no mínimo 70 e no máximo 822 conexões a mais do que sob RED, para cargas entre 0.5 e 1.0.

Os resultados obtidos demonstram o desempenho do controlador ótimo HSTCP-H2 e do controlador RED para enlace com múltiplos gargalos, sob tráfego FTP. Assim como para enlace com um único enlace gargalo, o desempenho do HSTCP-H2 foi superior ao desempenho de RED. Além disso, pode-se verificar que a diferença entre os valores produzidos por HSTCP-H2 e os valores produzidos por RED acentuam-se em cenários múltiplos

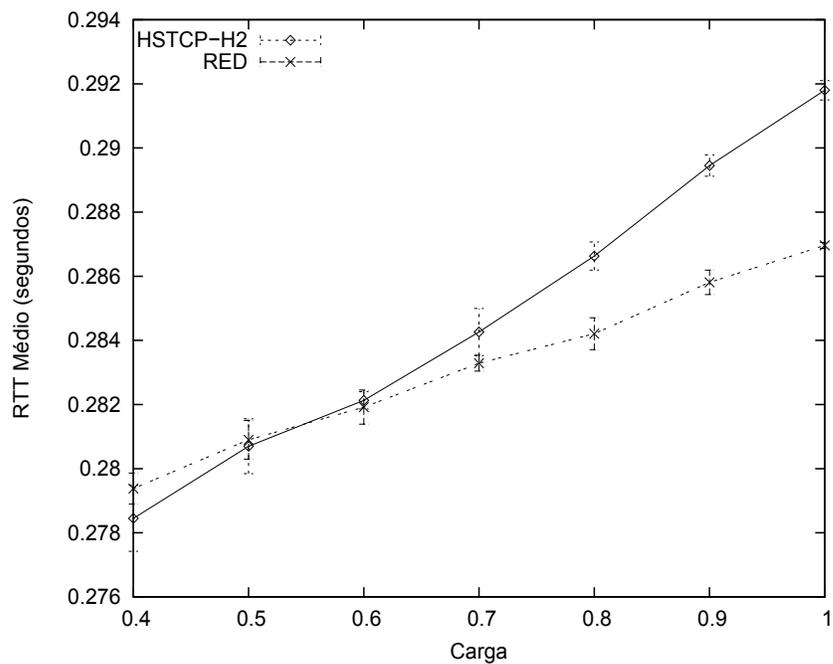


Figura 7.29: Tráfego FTP: RTT médio por conexão ativa em função da carga

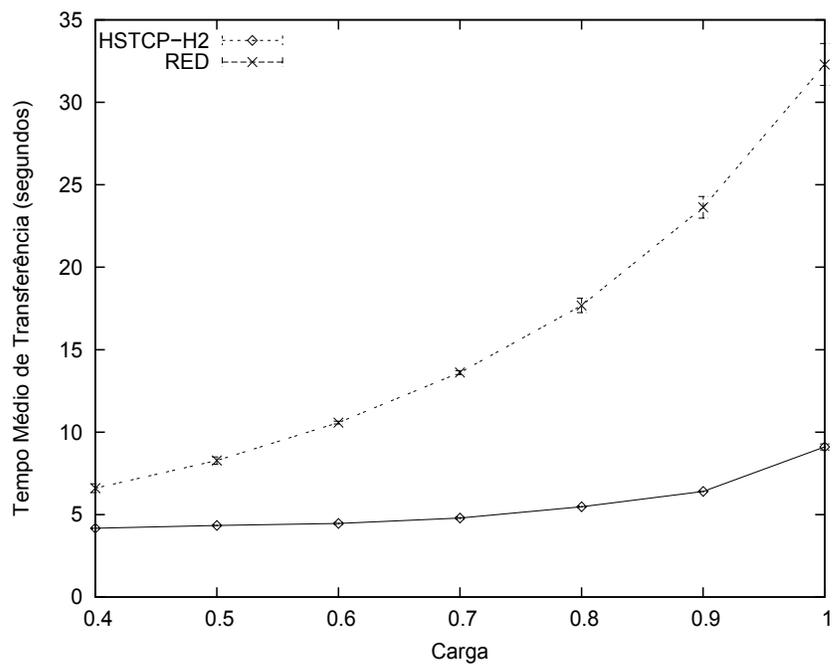


Figura 7.30: Tráfego FTP: tempo de transferência por conexão ativa em função da carga

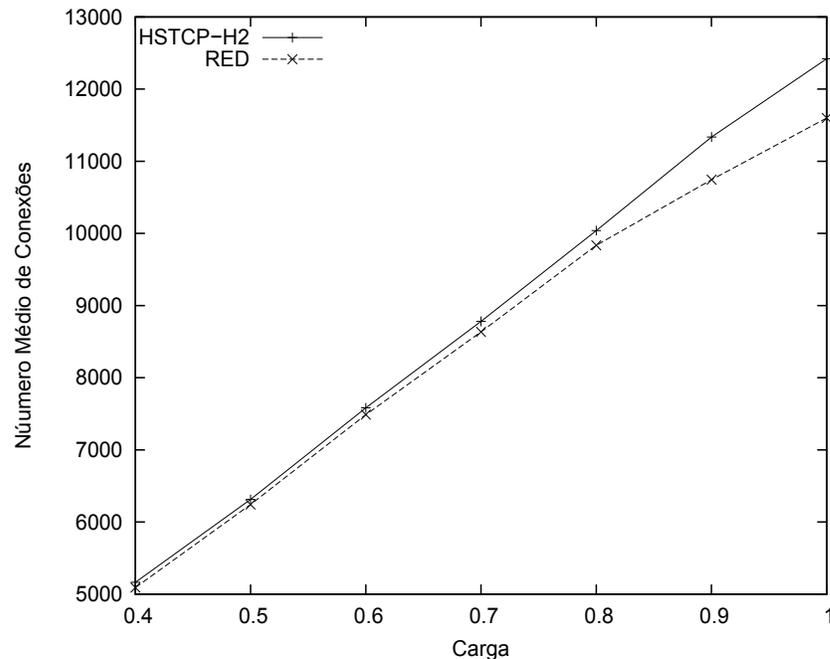


Figura 7.31: Tráfego FTP: média do número de conexões ativa em função da carga

gargalos, reforçando, assim, a capacidade do HSTCP-H2 em ajustar-se de maneira rápida e eficiente às variações do meio, enquanto que RED, devido aos seus ajustes tem seu desempenho degradado. A eficiência dos ajustes do HSTCP-H2, reforçam a importância em utilizar controladores ótimos para derivar políticas de AQM eficientes e robustas.

7.4.2 Experimentos Utilizando tráfego de Curta Duração

Nesta seção, apresentam-se os resultados obtidos nos experimentos realizados com tráfego de curta duração (WEB) em enlaces com múltiplos gargalos.

As Figuras 7.32 e 7.33 apresentam, respectivamente, a vazão obtida por conexão ativa em função da carga e, o tamanho médio da janela de congestionamento por conexão ativa em função da carga. Sob cargas de 0.4, a vazão obtida por RED é 1% maior do que a obtida por HSTCP. No entanto, com o aumento do congestionamento, RED apresenta maior queda na vazão, enquanto a vazão gerada pelo HSTCP-H2 varia pouco. Pode-se verificar a capacidade do HSTCP-H2 em se ajustar, de modo a garantir o desempenho das conexões em redes com topologias mais complexas. Sob carga de 0.6, a vazão obtida pelo HSTCP-H2 é 3% maior do que a vazão de RED. A diferença chega a ser no máximo 14% maior para carga de 0.9. A janela de congestionamento é o mecanismo responsável por inserir os dados na rede, ou seja, determina a vazão de dados. Na Figura 7.33, é possível verificar

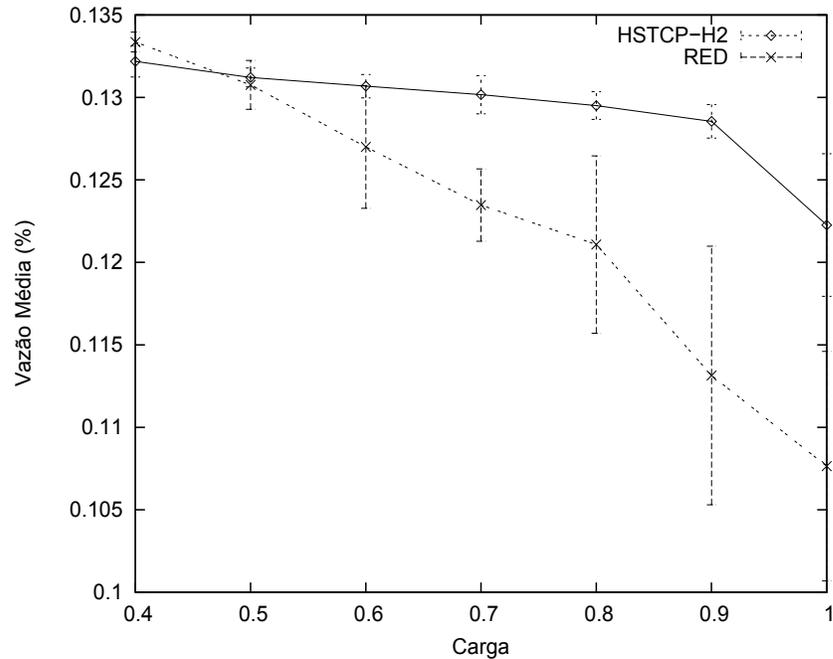


Figura 7.32: Tráfego WEB: vazão média por conexão ativa em função da carga

que diante de congestionamento brando o tamanho da janela do RED chega a ser 7% maior do que a de HSTCP-H2. A medida que o nível de congestionamento torna-se moderado a elevado, o HSTCP-H2 produz maiores valores de janela de congestionamento quando comparado a RED. Além disso, a diferença entre os tamanhos da janela gerados pelo HSTCP-H2 e por RED aumenta de maneira significativa a medida que a carga aumenta. Sob carga de 0.7, o janela do HSTCP-H2 é 4% maior do que a janela gerada por RED e, sob carga de 1.0, a diferença é máxima, sendo a janela produzida pelo HSTCP-H2 23% maior do que a produzida por RED. Conforme já constatado, sob tráfego FTP (Figuras 7.25 e 7.26), as vazões obtidas por HSTCP-H2 e por RED, demonstram a eficiência dos ajustes do HSTCP-H2, apesar do aumento da carga na rede, o desempenho produzido por HSTCP-H2 varia pouco, enquanto que o desempenho de RED degradou devido a sua capacidade limitada de ajustar a janela frente às variações das condições do meio.

Na Figura 7.34, apresenta-se o *goodput* por conexão ativa em função da carga. O *goodput* indica o quão eficiente é a utilização do enlace e é influenciado pelo tamanho da janela de congestionamento e pela quantidade de perdas na transmissão, ou seja, quanto maior a janela e quanto menor a quantidade de perdas, maior será o *goodput* obtido. Semelhante ao comportamento do tamanho da janela (Figura 7.33), o *goodput* obtido pelo RED é 2% maior sob cargas de 0.4. No entanto, a medida que a carga aumenta o HSTCP-H2 diferencia-se em relação a RED. Os valores obtidos pelo HSTCP-H2 são

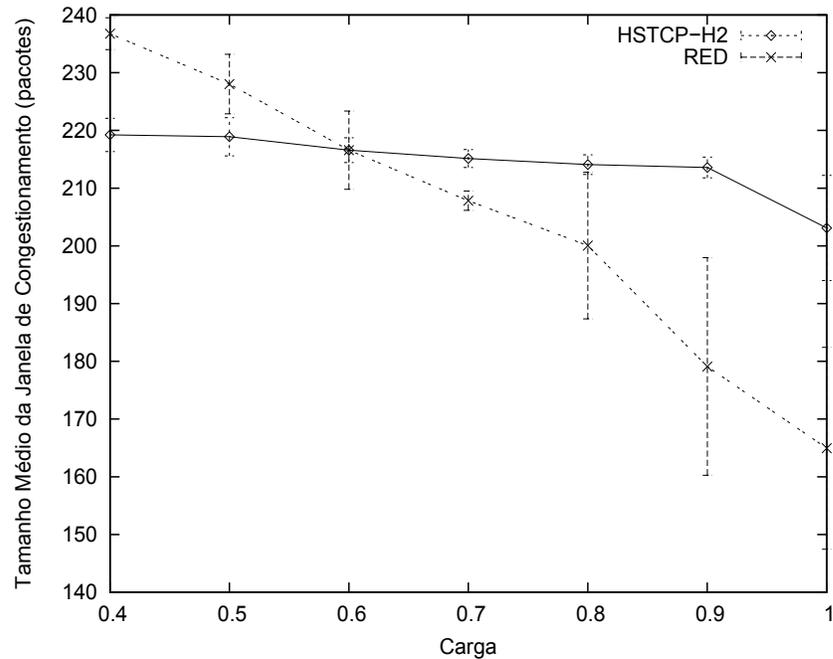


Figura 7.33: Tráfego WEB: tamanho médio da janela de congestionamento por conexão ativa em função da carga

no mínimo 3% maior e no máximo 13% maior do que os de RED, sob cargas de 0.6 e 0.9, respectivamente. Tanto sob tráfego FTP (Figura 7.27) quanto sob tráfego WEB, o HSTCP-H2 é o controlador que utiliza de forma mais eficiente os recursos de banda passante disponíveis.

A Figura 7.35 apresenta o número médio de retransmissões devido ao esgotamento do temporizador (RTO) por conexão ativa em função da carga. O número de RTOs apresentado pelo HSTCP-H2 é no mínimo 35% menor e no máximo 69% menor do que RED, sob cargas entre 0.9 e 1.0. Ficam evidentes, mais uma vez, as vantagens do HSTCP-H2 para utilizar a banda de forma eficiente.

Na Figura 7.36, o RTT médio por conexão ativa em função da carga é apresentado. É possível observar a diferença entre os valores apresentados pelo HSTCP-H2 e por RED são praticamente inexistentes. O RTT médio do HSTCP-H2 é 1% maior do que RED, e esta é a diferença máxima, quando os enlaces estão completamente congestionados, ou seja, 1.0 de carga. Levando em consideração a diferença na vazão obtida por HSTCP-H2 e por RED (Figura 7.32), constata-se que o HSTCP-H2 apresenta maior ocupação de enlace, e conseqüentemente de filas ao longo dos gargalos, consideravelmente maior do que RED e, portanto, era esperado uma variação maior no RTT do HSTCP-H2. Entretanto, como foi verificado na Figura 7.36, esta diferença é irrisória, o que reforça ainda mais as condições

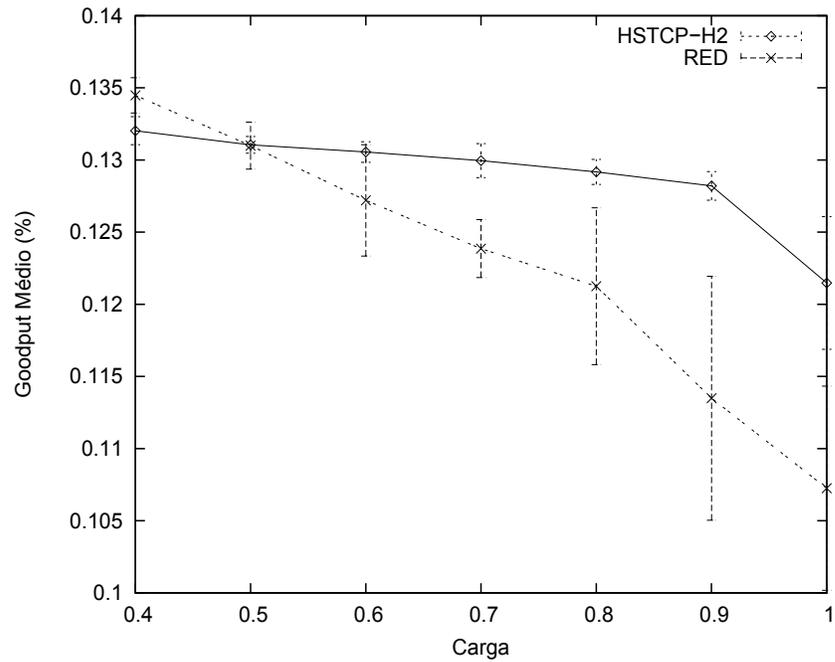


Figura 7.34: Tráfego WEB: goodput médio por conexão ativa em função da carga

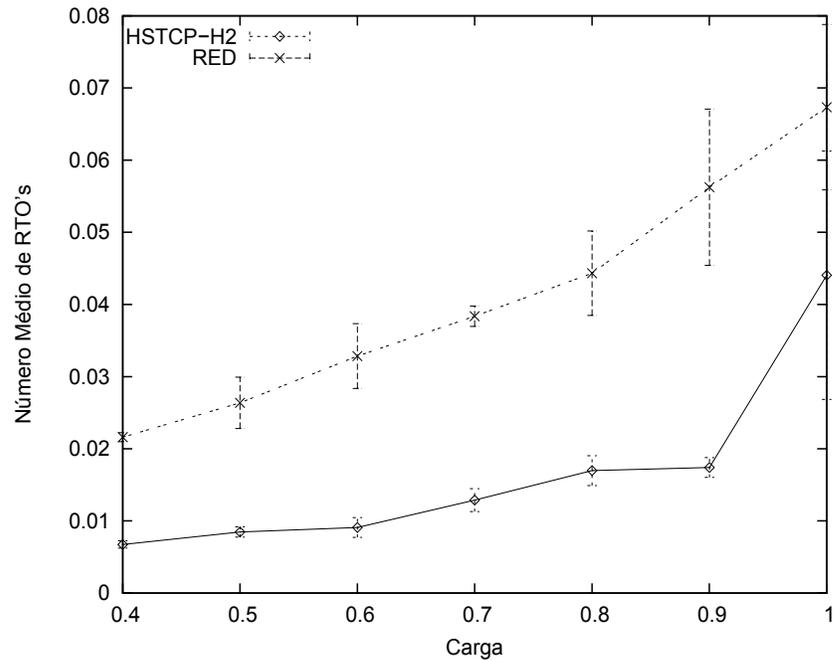


Figura 7.35: Tráfego WEB: número médio de RTO por conexão ativa em função da carga

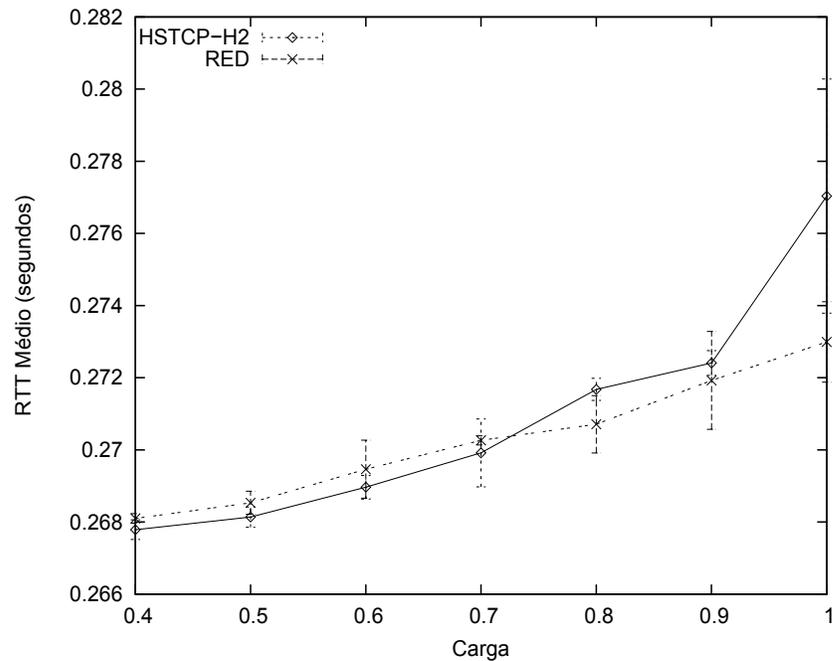


Figura 7.36: Tráfego WEB: RTT médio por conexão ativa em função da carga

do HSTCP-H2 como política de AQM.

Nas Figuras 7.37 e 7.38 são apresentados, respectivamente, a latência na transferência de arquivos e o número de conexões em função da carga. Assim como sob tráfego FTP (Figura 7.30), a latência na transferência de arquivos apresentada pelo HSTCP-H2 é menor do que as obtidas quando se emprega RED. A diferença mínima é de 3% menor sob carga de 0.4, e é no máximo 17% menor sob carga de 0.9. A redução no tempo de vida das conexões é uma consequência direta da redução do número de RTOs, o que maximiza a utilização dos recursos disponíveis.

Os resultados obtidos demonstram a capacidade dos controladores HSTCP-H2 e do controlador RED em se ajustar sob tráfego WEB, em redes com múltiplos enlaces gargalo. Assim como no experimento utilizando tráfego de longa-duração, o HSTCP-H2 destaca-se quando comparado ao RED. O HSTCP-H2 consegue ajustar-se para gerar maior utilização do enlace, enquanto maximiza o tamanho da janela de congestionamento e minimiza o número de perdas. Além disso, o RED tem seu desempenho seriamente degradado, pois seus ajustes são baseados em heurísticas, e acabam perdendo sua eficácia quando ocorrem variações no meio onde é empregado. Desta forma, pode-se verificar que o HSTCP-H2 é, também, um mecanismo eficiente sob tráfego de curta-duração.

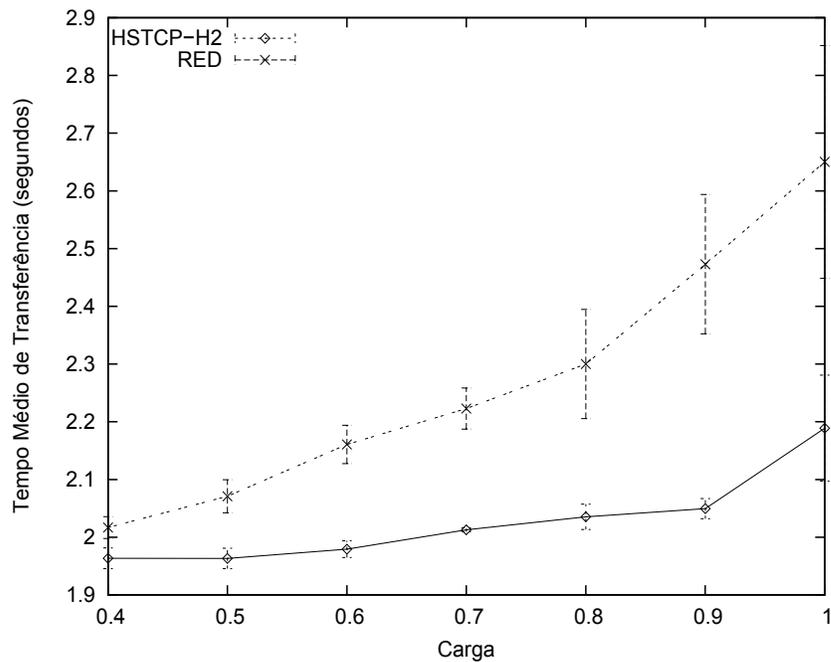


Figura 7.37: Tráfego WEB: tempo de transferência por conexão ativa em função da carga

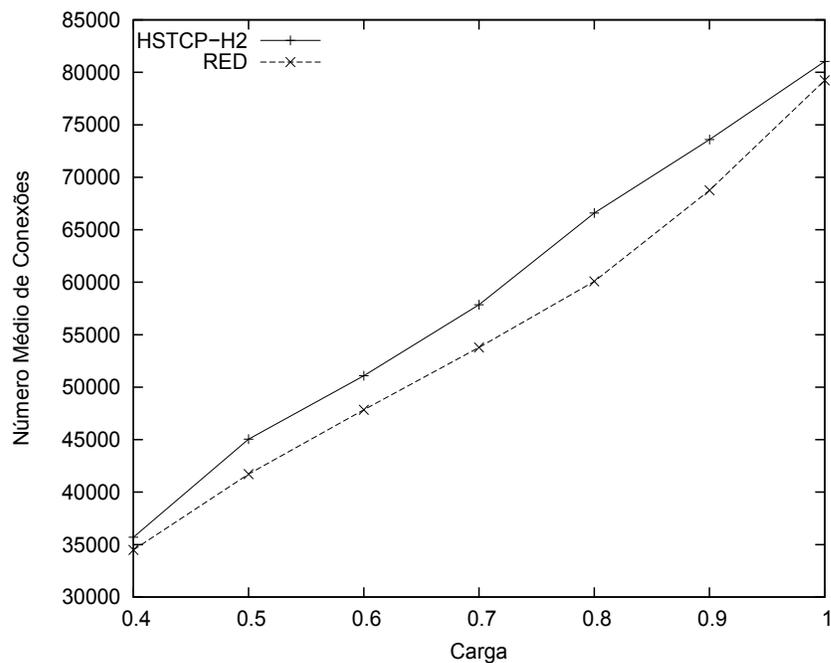


Figura 7.38: Tráfego WEB: média do número de conexões ativa em função da carga

Capítulo 8

Conclusões e Trabalhos Futuros

Neste capítulo, são resumidas as contribuições da presente dissertação. São sugeridos também alguns direcionamentos de trabalhos futuros.

8.1 Conclusões

O foco deste trabalho foi o desenvolvimento de um controlador ótimo para AQM em redes com produto banda-atraso (PBA) elevado, utilizando-se técnicas de Teoria de Controle Ótimo.

Inicialmente, abordou-se os problemas do TCP em utilizar de maneira eficiente os recursos disponíveis em redes com produto banda-atraso elevado. Para isso, foram apresentados os conceitos, os mecanismos e procedimentos do TCP Reno, para controlar o congestionamento na Internet. Da análise do funcionamento do TCP, foi possível verificar que o comportamento oscilatório na janela de transmissão, a penalização de conexões com RTTs longos, a dependência de perdas e a vazão do TCP inversamente proporcional a probabilidade de descarte/marcação das conexões são as principais deficiências do mecanismo de controle de congestionamento do TCP em redes com PBA elevado. Foram apresentadas algumas propostas de mecanismos TCP variantes para redes de alta velocidade e atrasos elevados. Os TCPs estudados foram: HSTCP, FAST TCP, BIC TCP, CUBIC e STCP.

Discorreu-se sobre o mecanismo de controle de congestionamento presente nos roteadores, o gerenciamento ativo de filas (AQM) e sobre a política RED recomendada pelo IETF para a Internet, suas vantagens e deficiências. Algumas políticas de AQM baseadas em Teoria de Controle foram também mencionadas.

Os conceitos e fundamentos básicos de modelagem matemática e Teoria de Controle, foram apresentados para dar o embasamento necessário para a compreensão do desenvolvimento deste trabalho.

Por ser uma das primeiras variantes do TCP para redes de alta velocidade, apresentar um modelo de funcionamento compatível com o TCP Reno e ter sua utilização recomendada pela *Internet Engineering Task Force* (IETF) [Floyd, 2003], o *High Speed* TCP foi escolhido para a derivação do controlador de AQM para redes com PBA elevado.

O sistema de controle de congestionamento HSTCP-H2/AQM foi tratado como um problema de controle com retroalimentação, no qual a taxa de transmissão é ajustada em função da disponibilidade da fila, de forma a evitar congestionamento. Um modelo simplificado da dinâmica deste sistema foi utilizado para a derivação dos controladores. Os controladores são responsáveis por determinar a probabilidade de descarte/marcação adequada, de forma a minimizar perdas e maximizar a vazão, enquanto garante a estabilidade do tamanho da fila independentemente das variações das condições da rede.

O diferencial do HSTCP-H2/AQM é a utilização de técnicas de Teoria de Controle Ótimo ao invés de técnicas de controle clássico. A utilização de uma abordagem não-racional faz com que os recursos da rede sejam utilizados de forma eficiente. Além disso, a planta utilizada para a derivação do sistema representa a dinâmica do sistema em detalhes, considerando a existência de fluxos não-adaptativos, o que garante a estabilidade do sistema independente das condições da rede.

O ponto de equilíbrio escolhido para o sistema, levou em consideração as situações de maior suscetibilidade, ou seja, com carga elevada. Resultados obtidos por meio de simulações, utilizando este ponto de equilíbrio, demonstraram a superioridade do HSTCP-H2 quando comparado com a política de AQM padrão, RED.

Foram discutidos os objetivos de desempenho almejados em redes com produto banda-atraso elevado, e com base nestes objetivos, foram derivados três controladores. Os controladores obtiveram desempenho equivalente quando comparadas as suas médias. Foi, então, utilizado uma medida de dispersão, o coeficiente de variação (CV), como medida estatística adequada para compará-los. O controlador com melhor desempenho para as métricas tamanho da fila, tamanho da janela de congestionamento e probabilidade de descarte/marcação, tanto para tráfego FTP quanto para WEB, foi o controlador com o objetivo de desempenho de “prevenir a subutilização do enlace e minimizar a ocorrência de *jitter*”. Para este controlador, foram, então, feitas análise de desempenho em relação ao RED.

Embora tenha sido utilizada uma abordagem não-racional para a derivação do controlador, foi possível o cancelamento do termo com atraso, e portanto, obtem-se um controlador racional, que garante a estratégia global ótima para solucionar o problema de minimização da norma H_2 . Além disso, o algoritmo obtido da equação diferencial é simples.

O modelo utilizado para a derivação do controlador HSTCP-H2/AQM considera o tráfego de longa duração. Desta forma, o desempenho dos controladores para tráfego de curta duração, pode não ser tão eficiente quanto para tráfego de longa duração.

Para a avaliação de desempenho do HSTCP-H2/AQM, foram consideradas a topologia *Dumb-bell*, com um único enlace gargalo, e a topologia *Parking-Lot*, com múltiplos enlaces gargalo, com tráfego cruzado e, portanto, com maior complexidade e maior proximidade às redes operacionais. Para os dois cenários, o desempenho do HSTCP-H2 foi superior ao RED, sendo que para o cenário com múltiplos enlaces gargalo, a diferença de desempenho entre HSTCP-H2 e RED é maior do que na topologia com um único enlace gargalo. Confirmando, assim, a viabilidade em se utilizar o controlador HSTCP-H2 como política de AQM. Para os dois cenários, foram realizados experimentos tanto para tráfego FTP quanto para tráfego WEB. Para tráfego FTP, o HSTCP-H2 foi superior em todas as métricas. Para tráfego WEB, as diferenças entre HSTCP-H2 e RED são menores, mas mesmo assim, o HSTCP-H2 utiliza a banda passante de maneira mais eficiente do que RED.

Desta forma, as principais contribuições destes trabalho, foram: o desenvolvimento de um controlador utilizando técnicas de Teoria de Controle Ótimo e a variante do TCP para redes com produto banda-atraso elevado HSTCP; a investigação do comportamento da interação entre uma política de AQM e um TCP variante, em redes com PBA elevado; a validação do controlador, por meio de resultados de experimentos de simulações, como candidato em potencial para política de AQM em redes que usam o protocolo HSTCP; simulações considerando cenários complexos, diferentes tipos de tráfego e variação de carga.

8.2 **Trabalhos Futuros**

Neste trabalho, comparou-se o controlador HSTCP-H2 com a política de AQM RED. Uma comparação com um conjunto de políticas de AQM em redes com PBA elevado, poderiam demonstrar as vantagens e vulnerabilidades destas políticas, e melhoramentos

no controle de congestionamento em redes de alta velocidade e atraso elevado.

O HSTCP realiza o controle de congestionamento baseado na sinalização de perdas de pacotes. O desenvolvimento de um controlador que considerasse um mecanismo TCP não dependente de perda como indicativo de congestionamento, como por exemplo o FAST TCP [Jin et al., 2003], seria uma investigação interessante, no sentido de entender qual a real influência do tipo de sinalização sobre o controlador ótimo, em redes com PBA elevado.

Caberia, também, o uso de um controlador ótimo em redes com PBA elevado no provimento de Qualidade de Serviço (QoS).

A utilização aplicações multimídia é cada vez maior, e, portanto, é necessária a utilização de redes de alta velocidade. Um mecanismo recente, específico para a transmissão de dados de aplicações multimídia é o *Datagram Congestion Control Protocol* (DCCP). Desta forma, o DCCP e sua interação com um controlador ótimo são áreas em aberto para estudos.

Um outro estudo, seria a investigação cuidadosa da influência de políticas de AQM na escalabilidade, na justiça, *Friendliness*, na convergência e na estabilidade dos mecanismos de controle de congestionamento TCP em redes com produto banda-atraso elevado.

Referências Bibliográficas

- [Agrawal and Granelli, 2004] Agrawal, D. and Granelli, F. (2004). Redesigning an active queue management system. In *Proc GLOBECOM '04. IEEE*, volume 2, pages 702–706.
- [Allman et al., 1999] Allman, M., Paxson, V., and Stevens, W. (1999). *TCP Congestion Control. RFC 2581*.
- [Augusto et al., 2007] Augusto, L. R., Fonseca, N. L. S., and Lima, M. M. A. E. (2007). Optimal aqm controler for the diffserv architecture. In *50th IEEE Globecom*, pages 1–6.
- [Barakat et al., 2000] Barakat, C., Altman, E., and Dabbous, W. (2000). On TCP performance in an heterogeneous network: A survey. In *IEEE Communication Magazine*, volume 38, pages 40–46.
- [Barford et al., 1999] Barford, P., Bestavros, A., Bradley, A., and Crovella, M. (1999). Changes in web client access patterns: Characteristics and caching implications. In *Proc of World Wide Web*, 2(1-2):15–28.
- [Barman et al., 2004] Barman, D., Smaragdakis, G., and Matta, I. (2004). The Effect of Router Buffer Size on HighSpeed TCP Performance. In *Proc of IEEE Globecom'04*, pages 1807–1810.
- [Bonald et al., 2000] Bonald, T., May, M., and Bolot, J.-C. (2000). Analytic evaluation of RED performance. In *INFOCOM (3)*, pages 1415–1424.
- [Boyd et al., 1981] Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V. (1981). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Reading, Massachusetts, second edition. This is a full BOOK entry.

- [Braden et al., 1998] Braden, B., Floyd, S., and et. al, V. J. (1998). Recommendations on queue management and congestion avoidance in the internet. *RFC 2309*, page 16.
- [Brakmo et al., 1994] Brakmo, L. S., O'Malley, S. W., and Peterson, L. L. (1994). Tcp vegas: new techniques for congestion detection and avoidance. *SIGCOMM Comput. Commun. Rev.*, 24(4):24–35.
- [Cardoso and Rezende, 2000] Cardoso, K. and Rezende, J. (2000). Http traffic modeling: Development and application.
- [Chen and Bensaou, 2007] Chen, S. and Bensaou, B. (2007). Can high-speed networks survive with droptail queues management? In *Computer Networks: The International Journal of Computer and Telecommunications Networking*, volume 51, pages 1763–1776.
- [Chiu and Jain, 1989] Chiu, D.-M. and Jain, R. (1989). Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. In *Computer Networks and ISDN Systems*, volume 17, pages 1 – 14.
- [Cottrell et al., 2003] Cottrell, R. L., Logg, C., and Mei, I.-H. (2003). Experiences and results from a new high performance network and application monitoring toolkit. *Passive and Active Measurement Workshop Proceedings*, pages 205–217.
- [Distefano et al., 1972] Distefano, J. J., Stubberud, A. R., and Williams, I. J. (1972). *Sistemas de Retroalimentação e Controle*. McGraw-Hill.
- [Feng, 1999] Feng, W. (1999). *Improving Internet congestion control and queue management algorithms*. PhD thesis, University of Michigan, USA.
- [Feng et al., 1999] Feng, W., Kandlur, D. D., Saha, D., and Shin, K. G. (1999). A self-configuring red gateway. In *IEEE INFOCOM'99, New York, NY*, volume 3, pages 1320–1328.
- [Firoiu and Borden, 2000] Firoiu, V. and Borden, M. (2000). A study of active queue management for congestion control. In *IEEE INFOCOM'00*, pages 1435–1444.
- [Floyd, 1994] Floyd, S. (1994). Tcp and explicit congestion notification. In *ACM SIGCOMM*, volume 24, pages 8 – 23.
- [Floyd, 1997a] Floyd, S. (1997a). Red: Discussions of setting parameters.

- [Floyd, 1997b] Floyd, S. (1997b). Red: Optimum functions for computing the drop probability?
- [Floyd, 2000] Floyd, S. (2000). Recommendation on using the gentle_ variant of red.
- [Floyd, 2003] Floyd, S. (2003). Highspeed tcp for large congestion windows. Technical report, RFC 3649, <http://www.ietf.org/rfc/rfc3649.txt>.
- [Floyd, 2007] Floyd, S. (2007). Metrics for the evaluation of congestion control mechanisms.
- [Floyd, 2008] Floyd, S. (2008). Ns: Network simulator. [Online]. Available:<http://www.isi.edu/nsnam/ns>.
- [Floyd and Jacobson, 1993] Floyd, S. and Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413.
- [Floyd and Jacobson, 1994] Floyd, S. and Jacobson, V. (1994). The synchronization of periodic routing messages. *IEEE/ACM Transactions on Networking*, 2(2):122–136.
- [Floyd and Kohler, 2002] Floyd, S. and Kohler, E. (2002). Internet research needs better models. In *Proceedings of HorNets-I*.
- [Floyd et al., 2002] Floyd, S., Ratnasamy, S., and Shenker, S. (2002). Modifying tcp’s congestion control for high speeds. *Preliminary Draft*.
- [Fomenkov et al., 2004] Fomenkov, M., Keys, K., Moore, D., and Claffy, K. (2004). Longitudinal study of internet traffic in 1998-2003. In *WISICT '04: Proceedings of the winter international symposium on Information and communication technologies*, pages 1–6. Trinity College Dublin.
- [Gunnarsson, 2000] Gunnarsson, F. (2000). Problems and improvements of internet traffic congestion control. Master’s thesis, Linkopings Universitet.
- [Ha et al., 2006] Ha, S., Kim, Y., Le, L., Rhee, I., and Xu, L. (2006). A step toward realistic performance evaluation of high-speed tcp variants. In *Fourth International Workshop on Protocols for Fast Long-Distance Networks*.
- [Hollot et al., 2001a] Hollot, C. V., Misra, V., Towsley, D. F., and Gong, W. (2001a). A control theoretic analysis of red. In *IEEE INFOCOM'01*, pages 1510–1519.

- [Hollot et al., 2001b] Hollot, C. V., Misra, V., Towsley, D. F., and Gong, W. (2001b). On designing improved controllers for aqm router supporting tcp flows. In *IEEE INFOCOM'01*, pages 1726–1734.
- [Hollot et al., 2002] Hollot, C. V., Misra, V., Towsley, D. F., and Gong, W. (2002). Analysis and design of controllers for aqm routers supporting tcp flows. *IEEE Trans. Automat. Contr.*, 47(6):945–959.
- [Huang et al., 2004] Huang, X., Lin, C., Ren, F., and Yin, H. (2004). Highspeed tcp modeling and analysis. *Broadnets - First Annual International Conference on Broadband Networks*.
- [Jacobson and Karels, 1988] Jacobson, V. and Karels, M. J. (1988). Congestion avoidance and control. *ACM Sigcomm'88*, 18, 4:314–329.
- [Jin et al., 2003] Jin, C., Wei, D. X., and Low, S. H. (2003). Fast tcp for high-speed long-distance networks.
- [Jin et al., 2005] Jin, C., Wei, D. X., Low, S. H., Buhrmaster, G., Bunn, J., Choe, D. H., Cottrel, R. L. A., Doyle, J. C., Feng, W., Martin, O., Newman, H., Paganini, F., Ravot, S., and Singh, S. (2005). Fast tcp: From theory to experiments. *IEEE Network*, 19:4–11.
- [Jinsheng et al., 2003] Jinsheng, S., Ko, K.-T., Guanrong, C., Chan, S., and Zukerman, M. (2003). Pd-red: to improve the performance of red. In *IEEE Communications Letters*, volume 7, pages 406–408.
- [Katabi et al., 2002] Katabi, D., Handley, M., and Rohrs, C. (2002). Congestion control for high bandwidth-delay product networks. *SIGCOMM'02*.
- [Kelly, 2003] Kelly, T. (2003). Scalable tcp: Improving performance in highspeed wide area network. Technical report, First International Workshop on protocols for Fast Long-Distance Networks.
- [Kim and Low, 2002a] Kim, K. B. and Low, S. H. (2002a). Analysis and design of aqm for stabilizing tcp.
- [Kim and Low, 2002b] Kim, K. B. and Low, S. H. (2002b). Cost of aqm in stabilizing tcp.

- [Kim et al., 2003a] Kim, K. B., Tang, A., and Low, S. (2003a). A stabilizing AQM based on virtual queue dynamics in supporting TCP with arbitrary delays. In *Proc 42nd IEEE Conference on Decision and Control*, volume 4, pages 3665–3670.
- [Kim et al., 2003b] Kim, K. B., Tang, A., and Low, S. (2003b). Design of aqm in supporting tcp based on the well-known aimd model. In *Proc GLOBECOM '03. IEEE*, volume 6, pages 3226–3230.
- [Kuusela et al., 2001] Kuusela, P., Lassila, P., and Virtamo, J. (2001). Stability of tcp-red congestion control. In *ITC-17, Salvador, Brazil*, pages 655–666.
- [Lakshman and Madhow, 1997] Lakshman, T. and Madhow, U. (1997). The performance of tcp/ip for networks with high bandwidth-delay products and random loss. In *IEEE/ACM Transactions on Networking*, volume 5, pages 336–350.
- [Li et al., 2007] Li, Y.-T., Leith, D., and Shorten, R. N. (2007). Experimental evaluation of tcp protocols for high-speed networks. In *IEEE/ACM Transactions on Networking*, volume 15, pages 1109 – 1122.
- [Lima, 2005] Lima, M. M. (2005). *Projeto de Controladores Ótimos para Gerenciamento Ativo de Filas*. PhD thesis, Universidade Estadual de Campinas.
- [Lima and Fonseca, 2002] Lima, M. M. A. E. and Fonseca, N. L. S. (2002). *Controle de Tráfego Internet*.
- [Lima et al., 2004] Lima, M. M. A. E., Fonseca, N. L. S., and Geromel, J. C. (2004). An optimal active queue management controller. In *Proc of IEEE ICC'04*, volume 4, pages 2261–2266.
- [Lin and Morris, 1997] Lin, D. and Morris, R. (1997). Dynamics of random early detection. In *SIGCOMM '97*, pages 127–137, Cannes, France.
- [Liu et al., 2003] Liu, S., Basar, T., and Srikant, R. (2003). Controlling the internet: A survey and some new results.
- [Long Chengnian and Xinping, 2006] Long Chengnian, Yang Huilong, L. Z. and Xinping, G. (2006). Fairness performance analysis: Hstcp/droptail vs. hstcp/aqm. In *Proc of 25th Chinese Control Conference*, pages 1807–1810.

- [Low and Srikant, 2004] Low, S. H. and Srikant, R. (2004). A mathematical framework for designing a low-loss, low-delay internet. *Networks and Spatial Economics*, 4(1):75–101.
- [Martin-Flatin and Ravot, 2002] Martin-Flatin, J.-P. and Ravot, S. (2002). Tcp congestion control in fast long-distance networks. *Technical Report CALT-68-2398, California Institute of Technology, USA*.
- [Mathis et al., 1997] Mathis, M., Semke, J., and Mahdavi, J. (1997). The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3).
- [May et al., 1999] May, M., Bolot, J., Diot, C., and Lyles, B. (1999). Reasons not to deploy RED. In *Proc. of 7th. International Workshop on Quality of Service (IWQoS'99), London*, pages 260–262.
- [Misra et al., 2000] Misra, V., Gong, W. B., and Towsley, D. F. (2000). Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In *ACM SIGCOMM'00, Stockholm, Sweden*, pages 151–160.
- [Ogata, 1998] Ogata, K. (1998). *Engenharia de Controle Moderno*. LTC.
- [Oliveira and Geromel, 2008] Oliveira, D. P. F. M. C. and Geromel, J. C. (2008). Lmi solver user's guide. [Online]. Available: <http://www.dt.fee.unicamp.br/mauricio/lmisol/userguide.ps.gz>.
- [Oliveira and Geromel, 2004] Oliveira, M. C. and Geromel, J. C. (2004). Synthesis of non-rational controllers for linear delay systems. *Automatica*, 40(2):171–188.
- [Ott et al., 1999] Ott, T., Kemperman, J., and Mathis, M. (1999). The stationary behavior of ideal TCP congestion avoidance. <ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps>.
- [Park, 1999] Park, P. (1999). A delay-dependent stability criterion for system with uncertain time-invariant delays. In *IEEE Trans. Automat. Contr.*, volume 44, pages 876–877.
- [Paxson and Allman, 2000] Paxson, V. and Allman, M. (2000). Rfc 2988 - computing tcp's retransmission timer.

- [Peng et al., 2005] Peng, Y., Yuan, G., and Ozbay, H. (2005). A variable structure control approach to active queue management for tcp with ecn. In *Proc IEEE Transactions on Control Systems Technology*, volume 13, pages 203–215.
- [Postel, 1981] Postel, J. (1981). Rfc 793 - transmission control protocol.
- [Ramakrishnan and Floyd, 1999] Ramakrishnan, K. and Floyd, S. (1999). Rfc 2481 - a proposal to add explicit congestion notification (ecn) to ip.
- [Ramakrishnan et al., 2001] Ramakrishnan, K., Floyd, S., and Black, D. (2001). Rfc 3168 - the addition of explicit congestion notification (ecn) to ip.
- [Rezende et al., 2002] Rezende, J. F., Lima, M. M., and Fonseca, N. L. (2002). *Mobility over transport control protocol/internet protocol (TCP/IP)*. CRC Press. in The HandBook of Ad Hoc Wireless Networks.
- [Rhee and Xu, 2005] Rhee, I. and Xu, L. (2005). Cubic: A new tcp-friendly high-speed tcp variant. *Third International Workshop on Protocols for Fast Long-Distance Networks*.
- [Santi et al., 2006] Santi, J., Fonseca, N. L. S., and Lima, M. M. A. E. (2006). Aqm controllers for high speed tcp networks. *III Workshop TIDIA*, pages 87–89.
- [Santi et al., 2007] Santi, J., Fonseca, N. L. S., and Lima, M. M. A. E. (2007). Projeto de controladores ótimos para redes com produto banda-atraso elevado. In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 899–912.
- [Santi et al., 2008a] Santi, J., Fonseca, N. L. S., and Lima, M. M. A. E. (2008a). Active queue management controller for the high speed tcp protocol. In: *IEEE International Conference on Communications (IEEE ICC)*, pages 6–11.
- [Santi et al., 2008b] Santi, J., Fonseca, N. L. S., and Lima, M. M. A. E. (2008b). Avaliação de desempenho de gerenciamento ativo de filas para redes de alta velocidade. In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 203–216.
- [Shakkottai et al., 2004] Shakkottai, S., Srikant, R., Brownlee, N., Broido, A., and kc claffy (2004). The rtt distribution of tcp flows on the internet and its impact on tcp based flow control. *CAIDA paper*.

- [Sonkoly et al., 2005] Sonkoly, B., Trinh, T. A., and Molnár, S. (2005). Understanding highspeed tcp: A control-theoretic perspective. *The Third IASTED International Conference on Communications and Computer Networks (CCN 2005)*.
- [Souza and Agarwal, 2003] Souza, E. and Agarwal, D. (2003). A highspeed tcp study: Characteristics and deployment issues. Technical report, LBNL Technical Report LBNL-53215. <http://www.icir.org/floyd/hstcp.html>.
- [Srikant, 2004] Srikant, R. (2004). *The Mathematics of Internet Congestion Control*. Systems & Control: Foundations & Applications. Birkhauser.
- [Subramanyan, 1995] Subramanyan, M. (1995). High speed networks and asynchronous transfer mode. Tutorial NPAC da Universidade Syracuse disponível em: http://www.npac.syr.edu/users/mahesh/homepage/atm_tutorial/p_slide.html.
- [Vinnicombe, 2002] Vinnicombe, G. (2002). Robust congestion control for the internet.
- [Wei et al., 2007] Wei, D. X., Jin, C., Low, S. H., and Hegde, S. (2007). Fast tpc: Motivation, architecture, algorithms, performance. *IEEE/ACM Transactions on Networking, to appear 2007*.
- [Willigenburg, 2003] Willigenburg, L. G. V. (2003). Systems and control theory part 2: Optimal control. Course material Systems and Control Theory SCO31304: Lecture Notes of lectures 7-12. <http://www.aenf.wau.nl/mrs/Education/sco31304/sco31304.html>.
- [Xidong et al., 2003] Xidong, D., Sungwon, Y., Kesidis, G., and Das, C. (2003). A control theoretic approach for designing adaptive aqm schemes. In *GLOBECOM '03. IEEE*, volume 5, pages 2947– 2951.
- [Xu et al., 2004] Xu, L., Harfoush, K., and Rhee, I. (2004). Binary increase congestion control (bic) for fast long-distance networks. *IEEE Infocom*.
- [Zhu et al., 2005] Zhu, L., Ansari, N., and Liu, J. (2005). Throughput of high-speed tcp in optical burst switching networks. In *IEE Proceedings Communications*, volume 152, pages 349– 352.