

SAMUEL BERNARDI

**CONTROLE DE CONGESTIONAMENTO EM
REDES TCP/IP: UMA ABORDAGEM VIA LMI**

**FLORIANÓPOLIS
2006**

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**CONTROLE DE CONGESTIONAMENTO EM
REDES TCP/IP: UMA ABORDAGEM VIA LMI**

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.

SAMUEL BERNARDI

Florianópolis, junho de 2006.

CONTROLE DE CONGESTIONAMENTO EM REDES TCP/IP: UMA ABORDAGEM VIA LMI

Samuel Bernardi

‘Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica, Área de Concentração em *Automação e Sistemas*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Alexandre Trofino Neto, Dr.
Orientador

Prof. Nelson Sadowski, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Alexandre Trofino Neto, Dr.
Presidente

Prof. Jean Marie Farines, Dr.
Co-Orientador

Prof. Carlos Barros Montez, Dr.

Prof. Ubirajara Franco Moreno, Dr.

Prof. Daniel Ferreira Coutinho, Dr.

Aos meus pais, Orfeu (in memoriam) e Beatriz.

AGRADECIMENTOS

Aos meus pais, Orfeu e Beatriz, pelo incentivo, carinho e amor que me dedicaram durante a realização deste trabalho.

À Universidade Federal de Santa Catarina através do Programa de Pós-Graduação em Engenharia Elétrica pela oportunidade de desenvolver esta pesquisa.

À minha namorada Carmeane pelas idéias, momentos de descontração e, também, pelo companheirismo e apoio nos momentos difíceis. Valeu mesmo.

Ao meu orientador Prof. Alexandre Trofino Neto pelo empenho, amizade e disponibilidade durante o desenvolvimento da dissertação.

À minha querida tia Orfélia pelo apoio e entusiasmo, mesmo estando longe.

Aos professores Jean Marie Farines e Carlos Barros Montez pelas valorosas discussões e sugestões que contribuíram muito na estrutura do trabalho.

Ao meu irmão Tiago pela força nesta importante etapa da minha vida.

Aos amigos Leomir, Carme e Amábile pelo encorajamento e pelas alegrias que me proporcionaram desde o início desta dissertação.

*Quando você partir, em direção a Ítaca,
que sua jornada seja longa,
repleta de aventuras, plena de conhecimento.*

*Não temas Laestrigones e Ciclopes nem o furioso Poseidon;
você não irá encontrá-los durante o caminho, se
o pensamento estiver elevado, se a emoção
jamais abandonar seu corpo e seu espírito.
Laestrigones e Ciclopes, e o furioso Poseidon
não estarão em seu caminho
se você não carregá-los em sua alma,
se sua alma não os colocar diante de seus passos.*

*Espero que sua estrada seja longa.
Que sejam muitas as manhãs de verão,
que o prazer de ver os primeiros portos
traga uma alegria nunca vista.
Procure visitar os empórios da Fenícia,
recolha o que há de melhor.
Vá às cidades do Egito,
aprenda com um povo que tem tanto a ensinar.*

*Não perca Ítaca de vista,
pois chegar lá é o seu destino.
Mas não apresse os seus passos;
é melhor que a jornada demore muitos anos
e seu barco só ancore na ilha
quando você já estiver enriquecido
com o que conheceu no caminho.*

*Não espere que Ítaca lhe dê mais riquezas.
Ítaca já lhe deu uma bela viagem;
sem Ítaca, você jamais teria partido.
Ela já lhe deu tudo, e nada mais pode lhe dar.*

*Se, no final, você achar que Ítaca é pobre,
não pense que ela o enganou.
Porque você tornou-se um sábio, viveu uma vida intensa,
e este é o significado de Ítaca.*

— KONSTANTINOS KAVAFIS (1863 – 1933)

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

CONTROLE DE CONGESTIONAMENTO EM REDES TCP/IP: UMA ABORDAGEM VIA LMI

Samuel Bernardi

Junho/2006

Orientador : Alexandre Trofino Neto, Dr.

Co-Orientador : Jean-Marie Farines, Dr.

Área de Concentração : Automação e Sistemas

Palavras-chave : Desigualdades Matriciais Lineares, Sistemas com Atraso, Gerenciamento Ativo de Fila

Número de Páginas : xiv + 111

Neste trabalho, é realizado o estudo e a implementação de Desigualdades Matriciais Lineares (LMIs) no projeto dos controladores Proporcional Integral (PI) e Detecção Preventiva Aleatória (RED), quando esses são aplicados no Gerenciamento Ativo de Fila (AQM) de uma rede de computadores. As LMIs são construídas observando-se a minimização dos critérios de desempenho \mathcal{H}^∞ para sistemas com atraso, os quais garantem a estabilidade assintótica do sistema estudado. Como o protocolo de transporte TCP é o que incorpora o AQM nos seus serviços de entrega confiável de dados, ele foi escolhido para interagir com os mecanismos de gerenciamento ativo de fila que estão sendo propostos. O modelo do TCP é desenvolvido baseado na análise de fluxo-fluído e na teoria de Equações Diferenciais Estocásticas (SDEs), já para o modelo do AQM são utilizadas as respectivas funções de transferência dos controladores PI e RED. Para comparar a eficácia da técnica de projeto através das LMIs, os controladores também foram projetados através de técnicas da Teoria de Controle Clássico e um estudo comparativo entre esses métodos é apresentado. As performances dos controladores, verificadas e comparadas usando-se o simulador de redes NS-2, mostram que os métodos são equivalentes, e que o PI é menos sensível do que o RED na presença de variações nos parâmetros da rede como número de sessões, atraso de propagação e capacidade do enlace.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

CONGESTION CONTROL IN TCP/IP NETWORKS: A LMI APPROACH

Samuel Bernardi

June/2006

Advisor: Alexandre Trofino Neto, Dr.

Co-Advisor: Jean-Marie Farines, Dr.

Area of Concentration: Automation and Systems

Keywords: Linear Matrix Inequality, Time-Delay Systems, Active Queue Management

Number of Pages: xiv + 111

In this work, Linear Matrix Inequalities (LMIs) are used to design the Proportional Integral (PI) and the Random Early Detection (RED) controllers when they are applied in the Active Queue Management (AQM) of networks. The LMIs are built observing the \mathcal{H}^∞ performance criteria, to systems with delay, which assure the asymptotically stability of the studied system. As the Transmission Control Protocol (TCP) incorporates the AQM policy in its services of trustworthy delivery of data, it was chosen to interact with the mechanisms of active queue management that are being considered. The TCP dynamic model is based on fluid-flow and Stochastic Differential Equations (SDEs) analysis, and to the AQM model is used the respective transfer functions of controllers PI and RED. To prove the effectiveness of the design technique with LMIs, the controllers were also configured through techniques of the Theory of Classic Control and a comparative study between these methods is presented. The performance of the controllers, verified and compared using the network simulator NS-2, show that the methods are equivalent, and that the PI is less sensitive than RED in the presence of network parameter variations such as load factor, round-trip time and link capacity.

Sumário

1	INTRODUÇÃO	1
2	OS PROTOCOLOS TCP/IP	5
2.1	ARQUITETURA DO TCP/IP	6
2.1.1	Camada de Aplicação	7
2.1.2	Camada de Transporte	7
2.1.3	Camada de Rede	10
2.1.4	Camada de Enlace e Física	13
2.1.5	Estrutura dos Cabeçalhos e Segmentos	16
2.2	ATRASOS NA INTERNET	16
2.2.1	Tipos de Atrasos	16
2.2.2	Atraso Nodal	19
2.3	CARACTERÍSTICAS DO TCP	19
2.3.1	Paralelismo	20
2.3.2	A Conexão TCP	22
2.3.3	Números de Seqüência e de Reconhecimento	23
2.3.4	Controle de Fluxo	25
2.3.5	Controle de Congestionamento	26
2.3.6	Variações do Controle de Congestionamento do TCP	27
3	GERENCIAMENTO DE FILA PARA O PROTOCOLO TCP	32
3.1	GERENCIAMENTO DE FILA	33
3.1.1	Gerenciamento Ativo de Fila	35
3.1.2	Eqüidade na Largura de Banda	36
3.1.3	Descrição do Controle RED	38
3.2	MODELO DINÂMICO DO PROTOCOLO TCP PARA APLICAÇÕES DE GERENCIAMENTO ATIVO DE FILA (AQM)	40
3.2.1	Comportamento da Janela	41

3.2.2	Comportamento da Fila	42
3.2.3	Tamanho Médio da Fila	43
3.3	LINEARIZAÇÃO	45
3.3.1	Derivadas Parciais	47
3.3.2	Linearização para Controle \mathcal{H}^∞	49
3.4	COMENTÁRIOS	51
4	PROJETO DOS CONTROLADORES PARA ROTEADORES AQM	52
4.1	MODELO DO PROTOCOLO TCP	52
4.2	PROJETO DE CONTROLADORES VIA BODE	55
4.2.1	Critérios de Estabilidade	56
4.2.2	Projeto do Controle RED no Domínio de Frequência	57
4.2.3	Cálculo do Controle C_{red}	59
4.2.4	Projeto do Controle PI no Domínio de Frequência	61
4.2.5	Cálculo do Controle C_{pi}	62
4.3	DESIGUALDADES MATRICIAIS LINEARES (LMIs) NA TEORIA DE CONTROLE	65
4.3.1	Sistemas Incertos	65
4.3.2	Representação de LMIs	67
4.3.3	Estabilidade Quadrática	68
4.3.4	Complemento de Schur	69
4.3.5	Lema de Finsler	71
4.3.6	Procedimento-S	72
4.4	PROJETO DE CONTROLADORES VIA LMI	73
4.4.1	Construção das LMIs para o Controle RED	73
4.4.2	Estabilidade	74
4.4.3	LMIs com Controle \mathcal{H}^∞	76
4.4.4	Cálculo do Controle C_{red}	79
4.4.5	Construção das LMIs para o Controle PI	81
4.4.6	Cálculo do Controle C_{pi}	84
4.5	COMENTÁRIOS	85
5	EXPERIMENTOS COM CONTROLADORES RED E PI	86
5.1	CENÁRIO DE SIMULAÇÃO	87
5.2	SIMULAÇÕES PARA O CONTROLE RED	88
5.2.1	Experimento 1: Evolução da Fila para Diferentes Valores de p_{max}	88

5.2.2	Experimento 2: Variações no RTT	91
5.2.3	Experimento 3: Variações na Carga e no RTT	92
5.2.4	Experimento 4: Evolução da Fila para um Baixo RTT	92
5.2.5	Experimento 5: Tráfego Intenso de Fluxos FTP e HTTP	93
5.2.6	Experimento 6: Mudança no Ponto de Equilíbrio	93
5.3	SIMULAÇÕES PARA O CONTROLE PI	94
5.3.1	Experimento 7: Comportamento do PI sem Variações na Rede	95
5.3.2	Experimento 8: Variações no RTT	95
5.3.3	Experimento 9: Variações na Carga e no RTT	96
5.3.4	Experimento 10: Tráfego Intenso de Fluxos com um Baixo RTT	96
5.3.5	Experimento 11: Variações na Largura de Banda	97
5.4	COMENTÁRIOS	98
6	CONCLUSÕES	100
A	IMPLEMENTAÇÃO DAS LMIs	102
A.1	Algoritmo para Resolução das LMIs do Controle RED	102
A.2	Algoritmo para Resolução das LMIs do Controle PI	104
	REFERÊNCIAS BIBLIOGRÁFICAS	106

Lista de Figuras

2.1	Topologia da Internet	6
2.2	Arquitetura do TCP/IP	7
2.3	Comunicação lógica entre sistemas finais.	8
2.4	Endereço IP	13
2.5	Formato das classes de endereço IP.	14
2.6	Estrutura de cabeçalhos e segmentos.	16
2.7	Tipos de atraso.	17
2.8	Comportamento da janela no protocolo Volta-N	20
2.9	Comportamento da janela no protocolo SR	22
2.10	Números de seqüência em segmentos TCP.	24
2.11	Buffer de recepção.	25
2.12	Comportamento da janela de congestionamento do TCP.	28
2.13	Exemplo do mecanismo de retransmissão Vegas	31
3.1	Inconvenientes provocados pelo gerenciamento de fila com Drop-Tail.	34
3.2	Gerenciamento de fila com RED.	37
3.3	RED — Filtro passa-baixa em cascata com função de descarte.	38
3.4	Diagrama de blocos da dinâmica do TCP.	46
3.5	Diagrama destacando a variável de perturbação δR	50
4.1	Diagrama de blocos da dinâmica linearizada do TCP.	52
4.2	Diagrama de blocos do sistema TCP com controle de realimentação.	53
4.3	Diagrama de Bode (amplitude) de $P(s)$ e $\Delta(s)$	54
4.4	Resposta ao degrau unitário de $P(s)$ (sem compensação).	56
4.5	Diagrama de Bode de $P(s)e^{-sR_0}$	58
4.6	Respostas de frequência com controle $C_{red}(s)$	60
4.7	Respostas de frequência com controle $C_{pi}(s)$	63
4.8	Função de transferência do PI em cascata com função de descarte.	82

5.1	Topologia de uma rede simples.	87
5.2	RED – Comportamento da fila sem variações na rede, $p_{max} = 0.05$	88
5.3	RED – Comportamento da fila sem variações na rede, $p_{max} = 0.10$	89
5.4	RED – Comportamento da fila sem variações na rede, $p_{max} = 0.15$	90
5.5	RED – Comportamento da fila para um aumento em K	90
5.6	RED – Comportamento da fila para variações no RTT.	91
5.7	RED – Comportamento da fila para variações no RTT e na carga N	92
5.8	RED – Comportamento da fila para uma diminuição no RTT.	93
5.9	RED – Comportamento da fila para um grande número de fluxos FTP e HTTP.	94
5.10	RED – Comportamento da fila para mudança do ponto de operação.	94
5.11	PI – Evolução do tamanho da fila sem variações na rede.	95
5.12	PI – Evolução do tamanho da fila para variações no RTT.	96
5.13	PI – Evolução do tamanho da fila para variações no RTT e na carga N	97
5.14	PI – Evolução do tamanho da fila para RTT baixo e carga N muito elevada.	97
5.15	PI – Evolução do tamanho da fila para larguras de banda diferentes.	98

Lista de Tabelas

4.1	Valores de $P(s)$ e $\Delta(s)$ para diferentes fatores de carga N	54
4.2	Parâmetros do RED (<i>via Bode</i>) para valores diferentes de p_{max}	61
4.3	Valores de a e b (<i>via Bode</i>) para diferentes frequências f_a	64
4.4	Parâmetros do RED (<i>via LMI</i>) para valores diferentes de p_{max}	80
4.5	Valores de a e b (<i>via LMI</i>) para diferentes frequências f_a	84

Lista de Abreviaturas e Símbolos

AQM	<i>Active Queue Management</i>
ARE	<i>Algebraic Riccati Equation</i>
EWMA	<i>Exponential Weight Moving Average</i>
IP	<i>Internet Protocol</i>
LMI	<i>Linear Matrix Inequalities</i>
LQG	<i>Linear Quadratic Gaussian</i>
ODE	<i>Ordinary Differential Equations</i>
PR	<i>Positive Real</i>
RED	<i>Random Early Detection</i>
RTT	<i>Round-Trip Time</i>
SDE	<i>Stochastic Differential Equations</i>
TCP	<i>Transmission Control Protocol</i>
\mathbb{R}	Conjunto dos números reais
\mathbb{R}^n	Conjunto de vetores reais de dimensão n
$\mathbb{R}^{n \times m}$	Conjunto das matrizes reais de ordem $n \times m$
$A \in \mathbb{R}^{n \times m}$	Matriz A de dimensão $n \times m$
A^T	Matriz transposta de A
$A = A^T$	Matriz A simétrica
$A > 0$	Matriz A simétrica e positiva definida
$\mathbf{E}[\cdot]$	Operador esperança matemática
$\begin{bmatrix} A_{11} & A_{12} \\ * & A_{22} \end{bmatrix}$	Matriz simétrica onde o elemento $*$ se deduz por simetria
$e^{\{\cdot\}}$	Indica função exponencial de $\{\cdot\}$
I_n	Matriz identidade de ordem n
$0_{n \times m}$	Matriz de zeros de ordem $n \times m$
$diag\{A_1, \dots, A_n\}$	Matriz bloco-diagonal formada pelas matrizes A_n na diagonal
$min \gamma$	Problema de otimização onde busca-se o mínimo de γ

Capítulo 1

INTRODUÇÃO

Devido ao rápido avanço tecnológico dos sistemas de comunicação via redes, muita pesquisa tem sido feita para melhorar a transmissão de dados através de redes de computadores, respeitando prioridades para o envio de informações e aspectos de confiabilidade e segurança. Destacam-se, entre outras redes, a Internet, que é uma rede que conecta milhões de equipamentos de computação em todo o mundo, e redes locais (LAN), que são concentradas em uma área específica como um prédio, uma indústria ou uma universidade. Em particular; cada vez mais equipamentos como pagers, telefones celulares, televisores com tecnologia digital e outros equipamentos incomuns na área de computação como torradeiras e geladeiras estão sendo conectados a redes de computadores, mais especificamente na Internet, o que permite o compartilhamento de recursos de forma simples e eficiente além de agilizar ou minimizar o tempo de execução de serviços como acessar os dados de uma conta bancária. Este constante crescimento de sistemas finais¹, cada vez mais começa a congestionar o núcleo da rede² podendo ocasionar vários contratempos como grandes atrasos para a entrega dos dados, perda de informação ou parte dela e retransmissões desnecessárias.

Imagine-se, por exemplo, que uma empresa de exploração de petróleo utilize a Internet para fazer o controle do sistema de posicionamento de uma plataforma que está em alto mar. Neste caso, toda troca de informação entre sensores, controladores e atuadores passa através da rede e está sujeita aos contratempos citados anteriormente, os quais podem degradar o desempenho e até instabilizar o sistema realimentado.

Suponha-se agora que se está em uma sala de cirurgia, no Brasil, onde o paciente sofrerá uma intervenção cirúrgica com o uso de uma videoconferência. O médico que fará a cirurgia

¹Os termos sistemas finais ou hospedeiros, referem-se a qualquer equipamento conectado a uma rede. Em uma rede que usa o modelo cliente-servidor, um programa cliente que roda em um sistema final pede e recebe informações de um servidor que roda em outro sistema final.

²O núcleo da rede é formado por uma malha de roteadores que conectam os sistemas finais.

está em um continente diferente, por exemplo, na Europa. A equipe médica brasileira apenas dará apoio. Nesta cirurgia, instrumentos como bisturis e tesouras serão comandados ou manipulados pelo médico europeu através de uma rede de computadores, como a Internet. Se houver uma pequena distorção no sinal de controle enviado da Europa para o Brasil, um erro irreparável pode ocorrer. Nesse sentido, a solução de problemas na área de controle de sistemas passa a ser muito dependente das estratégias de controle de congestionamento em redes, o que tem aproximado os interesses de pesquisadores nas áreas de controle e da informação como MISRA et al. [44, 47] e MANFREDI et al. [42]. Para PAGANINI et al. [53], o mecanismo de controle de congestionamento na Internet já representa o maior sistema realimentado, o que dá uma sólida dimensão de quanto vem se tornando crucial pesquisas em controle.

O projeto de políticas de controle de congestionamento em redes é bastante complexo pois depende de inúmeros fatores como: o tamanho da rede e sua complexidade; limitações de software, hardware e econômicas; distorções e atrasos da informação. Quando a vazão da informação é suficientemente rápida pode-se desprezar pequenos atrasos na entrega da informação e, nesse caso, o projeto de controle de fluxo é um problema estático. No entanto, com o constante carregamento das redes de comunicação, essa situação não é muito realista pois a quantidade de informação a ser processada introduz atrasos importantes, e o projeto da política de controle de congestionamento da rede torna-se um problema dinâmico. Neste caso, o estudo da estabilidade aliada ao desempenho do sistema dinâmico se torna indispensável como demonstrado em [4, 36, 54].³

Os sistemas finais, na Internet, utilizam dois protocolos⁴ de transporte: UDP (*User Datagram Protocol* — Protocolo de Datagrama do Usuário) e TCP (*Transmission Control Protocol* — Protocolo de Controle de Transmissão), os quais controlam o envio e o recebimento de informações. Porém apenas o protocolo TCP oferece um controle de fluxo e de congestionamento. O controle de fluxo garante que nenhum dos lados (cliente–servidor) sobrecarregue um ao outro enviando demasiadamente pacotes⁵. O controle de congestionamento ajuda a evitar que a Internet fique engarrafada. Quando um roteador (também conhecido como comutador de pacotes) fica muito congestionado, seus buffers (também chamados de filas)

³O atraso sofrido pelos dados é uma das principais características que deve ser analisada ao se projetar um eficiente controle de congestionamento, entretanto, existem outras variáveis que interferem de modo decisivo como, por exemplo, o carregamento da rede e os protocolos usados para a troca de dados. Ao longo do trabalho é feito um detalhamento dos principais aspectos que contribuem para a formação e prevenção do congestionamento.

⁴Para KUROSE et al. [39], um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou recebimento de uma mensagem ou outro evento.

⁵Em redes de comutação de pacotes, o emissor fragmenta as mensagens em ‘pacotes’ menores para só então enviá-los.

podem transbordar ocasionando a perda de pacotes. Entretanto, este mecanismo de controle de congestionamento do TCP usa um controle fim a fim [36, 26] em vez de um controle de congestionamento assistido pela rede, já que o protocolo IP⁶ da camada de rede da Internet não fornece aos sistemas finais realimentação explícita relativa ao congestionamento da rede [39]. Esta forma de controle de congestionamento atua na rede apenas depois que ocorre uma perda de pacotes no roteador, ou seja, quando o congestionamento já ocorreu. Um dos problemas que ocorrem neste caso inclui a sincronização de fluxo e a degradação do desempenho devido aos excessivos estouros do tempo de transmissão (*Timeouts*) e as constantes diminuições do tamanho da janela do emissor as quais são adotadas para diminuir o problema do congestionamento.⁷

Para permitir que o núcleo da rede consiga ajudar no gerenciamento de congestionamento do TCP, é introduzido no roteador o AQM (*Active Queue Management* — Gerenciamento Ativo de Fila) [44, 47, 48]. O exemplo clássico de um AQM é o controle RED (*Random Early Detection*) o qual foi introduzido por FLOYD et al. [27]. Ao invés de esperar o congestionamento ocorrer, o RED atua preventivamente medindo o tamanho da fila do roteador e ajustando a janela do emissor de acordo para permitir uma boa utilização da largura de banda da rede em questão. O grande inconveniente do RED provém das dificuldades de ajuste dos seus parâmetros [15, 43].

Recentemente, esquemas alternativos para o AQM foram propostos [47, 42, 46] utilizando-se técnicas de controle clássicas como os controles Proporcional (P) e Proporcional Integral (PI) e também metodologias de controle ótimo como LQG, \mathcal{H}^∞ e H_2 . Mais especificamente, MISRA et al. [48, 47, 46] ajustaram os parâmetros do controle RED e implementaram os controles P e PI utilizando recursos como Diagrama de Bode, Ganho de Margem e Ganho de Fase [52]. Entretanto, este método produz uma má performance destes controles na presença de variações nos parâmetros da rede como o RTT (*Round-Trip Time* — Tempo de Ida e Volta), no número de sessões e na capacidade do enlace as quais ocorrem em uma rede mais realista [41, 56]. Para melhorar o comportamento dos controles P, PI e RED quando da ocorrência dos fenômenos citados anteriormente; o uso de métodos de controle ótimo, mais especificamente o \mathcal{H}^∞ para sistema com atraso [30], no projeto destes controles pode conseguir uma maior robustez da estabilidade e do desempenho pois é possível levar em consideração as variações dos parâmetros da rede durante o desenvolvimento dos controladores e também minimizar a influência destas perturbações, fato que não ocorre com os recursos utilizados atualmente.

⁶O protocolo IP (*Internet Protocol*) da camada rede especifica o formato da informação que é enviada e recebida entre roteadores e os sistemas finais.

⁷Para uma maior compreensão dos termos aqui citados, veja [47, 39].

Uma vez definido o modelo matemático que represente de forma confiável a dinâmica do controle de congestionamento do protocolo TCP, o objetivo deste trabalho de mestrado será desenvolver um estudo comparativo entre as técnicas de projeto dos controladores RED, P e PI que são usadas atualmente e a metodologia de controle ótimo \mathcal{H}^∞ para sistemas com atraso também aplicada no projeto destes controladores.

Uma visão geral sobre redes de computadores, introduzindo conceitos e terminologias que serão importantes para a compreensão do restante da dissertação será apresentado no Capítulo 2. O modelo matemático, técnicas de controle e o projeto dos controladores serão desenvolvidas nos Capítulos 3 e 4. O Capítulo 5 faz a aplicação das técnicas de controle e as compara usando o simulador de redes NS-2 [51]. Finalmente, as conclusões são apresentadas no Capítulo 6.

Capítulo 2

OS PROTOCOLOS TCP/IP

A Internet é um conjunto interconectado de redes públicas e privadas, cada uma com gerenciamento próprio. Qualquer rede conectada à Internet precisa seguir os ‘padrões da Internet’, os quais são desenvolvidos pela IETF (*Internet Engineering Task Force* — Força de Trabalho de Engenharia de Internet). Os documentos que estabelecem os padrões da IETF são chamados de RFCs (*Request for Comments* — Pedido de Comentários). A estrutura de interconexão dos vários componentes da Internet segue uma hierarquia. Basicamente, os sistemas finais são conectados aos ISPs (*Internet Service Providers* — Provedores de Serviços de Internet) por meio de redes de acesso. Uma rede de acesso pode ser uma linha telefônica acoplada a um modem, uma rede local de uma empresa ou universidade ou uma rede de acesso de alta velocidade dedicada ou comutada. Conforme a figura 2.1, os ISPs locais estão conectados a ISPs regionais, que por sua vez estão ligados a ISPs nacionais e Internacionais.

Os sistemas finais estão interligados por meio de enlaces físicos. A tecnologia dos enlaces pode variar, sendo que os tipos mais conhecidos são: fibra ótica, cabos coaxiais, ondas de rádio e fio de cobre. A velocidade de transmissão de dados do enlace muda de acordo com o meio utilizado e recebe o nome de **largura de banda** a qual é medida em bits por segundo (bps). É lógico que, para redes como a Internet, entre dois sistemas finais existem vários roteadores os quais comutam ou redirecionam as informações que estão chegando no enlace de entrada para um enlace de saída conforme o destino da informação.

O conjunto de protocolos que os sistemas finais operam recebeu o nome TCP/IP em homenagem as dois principais protocolos da Internet, o TCP e o IP. O TCP/IP estabelece uma camada padrão de serviços de comunicações sobre a camada de hardware, de forma que o usuário não tome conhecimento da arquitetura física da rede. Dessa forma, esses serviços de comunicação são usados pelas aplicações¹, que rodam nos sistemas finais, de forma in-

¹Dentre as aplicações estão a transferência de arquivos, o correio eletrônico, a World Wide Web, a transmissão de áudio e vídeo, a vide-conferência em tempo real e muitas outras [3].

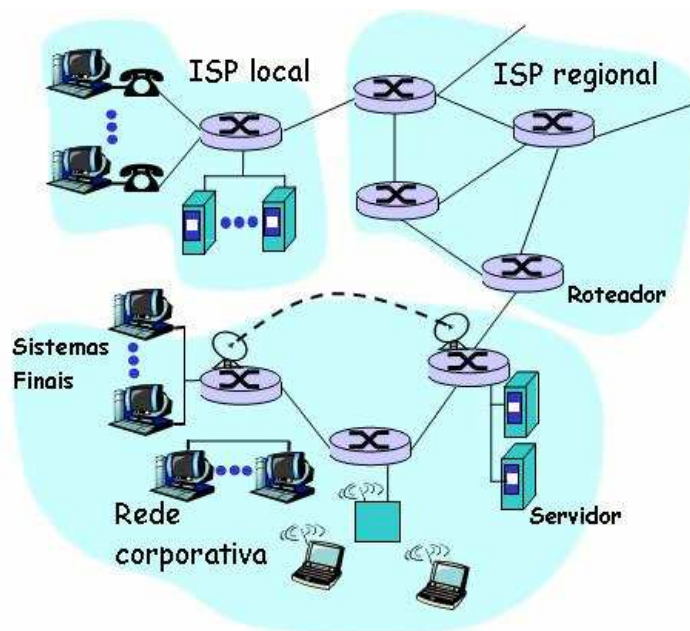


Figura 2.1: Topologia da Internet. Adaptado de [39].

dependente da tecnologia da rede subjacente. O que permite a interconexão de redes físicas diferentes.

2.1 ARQUITETURA DO TCP/IP

Os protocolos TCP/IP estão distribuídos em cinco camadas distintas de acordo com a funcionalidade: camada de aplicação, camada de transporte, camada de rede, camada de enlace e camada física. O conjunto destas camadas é chamado de **pilha de protocolos**. Os projetistas de rede organizam os protocolos desta maneira para reduzir a dificuldade de projeto o que permite lidar com sistemas mais complexos e facilitar a manutenção e atualização da rede. Este tipo de estrutura impõe que as camadas superiores se utilizem dos serviços das camadas inferiores, criando uma hierarquia. É importante entender que nem todas as entidades da rede (as entidades mais importantes são os sistemas finais e os roteadores) implementam integralmente a pilha de protocolos [60]. Desta maneira; conforme figura 2.3, as camadas de rede, enlace e física estão presentes na borda² e no núcleo da rede, enquanto que as camadas de aplicação e transporte ficam apenas nos sistemas finais. A figura 2.2 mostra as cinco camadas da família TCP/IP além de indicar alguns dos principais protocolos posicionados na camada correspondente aos serviços que executam. As camadas de enlace

²A borda ou periferia da rede é formada pelos mais diversos tipos de sistemas finais.

e física estão representadas juntas.

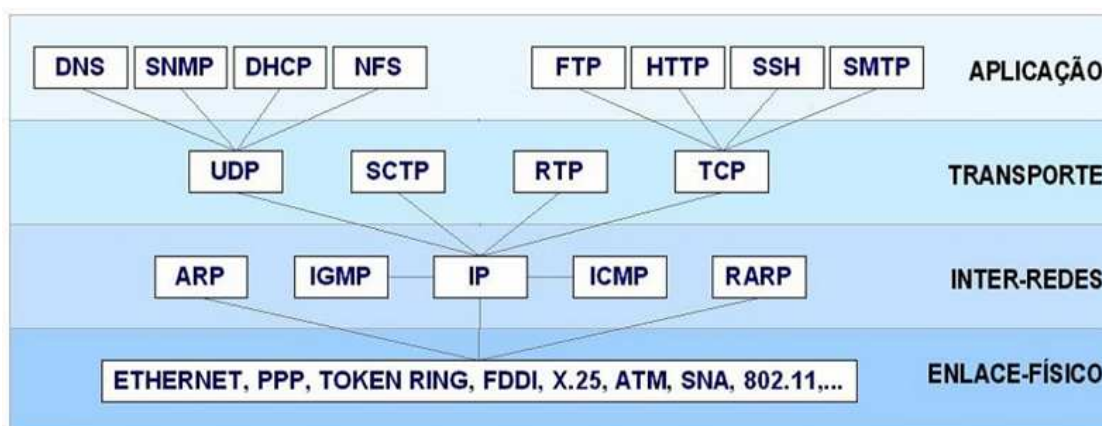


Figura 2.2: Arquitetura do TCP/IP. Adaptado de [37].

2.1.1 Camada de Aplicação

A camada de aplicação é responsável pelo suporte das aplicações de rede. Os protocolos desta camadas são chamados de alto nível, dentre eles o HTTP, para suportar a WEB; o SMTP e POP3, para suportar o correio eletrônico; e o FTP, para suportar o transporte de arquivos. Devido as aplicações de rede serem a essência da Internet ou de qualquer outra rede de computadores, os protocolos das várias camadas estão em constante desenvolvimento devido as necessidades impostas pelas mais variadas aplicações.

É importante notar que protocolos da camada de aplicação e aplicações de rede, apesar de trabalharem em conjunto, possuem funções diferentes. Por exemplo, para uma aplicação Web muitos componentes são necessários como: um padrão para formato de documentos (isto é, HTML); navegadores Web (por exemplo, Netscape Navigator e Microsoft Internet Explorer); servidores Web e um protocolo de camada de aplicação [39]. Portanto, o protocolo de camada de aplicação HTTP [19] é apenas uma parte da aplicação e, define o formato de como as informações entre navegador e o servidor Web são passadas.

2.1.2 Camada de Transporte

A camada de transporte provê comunicação lógica entre processos de aplicação em sistemas finais diferentes. Comunicação lógica entre processos de aplicação significa que, apesar dos clientes e servidores não estarem conectados 'fisicamente', o que normalmente ocorre na Internet, para aplicação é como se estivessem. A figura 2.3 fornece uma noção

melhor deste conceito. Na Internet existem dois protocolos de transporte: o TCP e o UDP. Estes protocolos fornecem serviços orientado à conexão e não orientado à conexão.

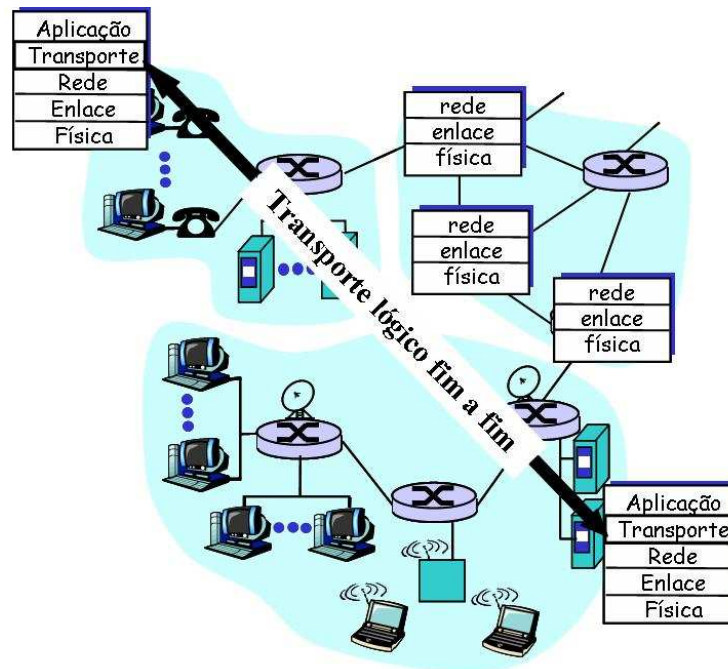


Figura 2.3: Comunicação lógica entre sistemas finais. Adaptado de [39].

Serviço orientado à conexão

O serviço orientado à conexão é fornecido pelo protocolo TCP. Quando uma aplicação usa este tipo de serviço, o cliente e o servidor trocam pacotes ou mensagens de controle antes de iniciar efetivamente a enviar os pacotes com os dados da informação (por exemplo, páginas da Internet e mensagens de e-mail). Esta apresentação que o TCP faz entre os sistemas finais, permite que ambos os lados do processo fiquem preparados para o início da comunicação reservando recurso e espaço em seus buffers (ou filas). Depois que esta apresentação termina, diz-se que se estabeleceu uma conexão entre os sistemas finais. Porém, apenas o cliente e o servidor sabem que existe a conexão; os roteadores do núcleo da rede apenas fazem a comutação dos pacotes, ficando alheios a qualquer estabelecimento de uma conexão TCP.

Os serviços da Internet que são orientados a conexão também fornecem um controle de fluxo e um controle de congestionamento de dados, além de usar métodos de reconhecimento (*acknowledgements*) e retransmissão de pacotes dando uma maior confiabilidade na transferência das informações. Aliás, a transferência confiável de dados é a principal característica que faz com que cerca de 90% dos pacotes na Internet pertençam a fluxos TCP [61].

Serviço não orientado à conexão

O protocolo UDP fornece o serviço não orientado à conexão na Internet. Ao contrário do protocolo TCP, o UDP não faz uma apresentação prévia entre o cliente e o servidor. Quando um sistema final quer enviar pacotes usando um serviço não orientado a conexão, ele simplesmente o faz; sem ter garantia nenhuma que o destinatário recebeu a informação. É um método não confiável de transmissão de pacotes de dados, mas que pode ter algumas vantagens em relação ao serviço orientado à conexão. Por exemplo, o UDP pode enviar pacotes utilizando toda a largura de banda disponível, pois, não utiliza os controles de fluxo nem de congestionamento. Este aspecto do protocolo UDP é essencial para aplicações que tenham uma exigência de largura de banda mínima como é o caso da Telefonia por Internet, que utiliza aplicações de áudio em tempo real.

Como o protocolo UDP não garante transmissão confiável, é possível conseguir tal transmissão adicionando-se controles na própria camada de aplicação. O programador pode, por exemplo, adicionar eventos como o de reconhecimento dos pacotes enviados, como ocorre no TCP, porém não com a mesma eficiência.

Multiplexação e demultiplexação

Uma das tarefas mais importantes realizada pelos protocolos TCP e UDP é a de ampliar o serviço de entrega oferecido pelo protocolo IP da camada de rede. Esta ampliação do serviço de entrega de dados entre sistemas finais é chamada de multiplexação/demultiplexação de aplicações. Apesar da multiplexação/demultiplexação não ser o serviço de maior destaque na camada de transporte, ele desempenha um papel fundamental. Isto fica evidente quando se nota que o protocolo IP entrega dados apenas entre sistemas finais; ficando a cargo da camada transporte, através do serviço de multiplexação/demultiplexação, entregar esses dados para os respectivos processos de aplicação. Considera-se, então, um exemplo simples.

Suponha-se que, na figura 2.3, o computador da parte superior esquerda seja o emissor e o computador da parte inferior direita seja o receptor, e que é desejado rodar dois processos de aplicação: enviar uma mensagem de e-mail (SMTP) e fazer transferência de arquivos (FTP). Desta forma, quando a camada de transporte no emissor reúne ou encapsula os dados provenientes dos processos SMTP e FTP da camada de aplicação, entregando posteriormente para a camada de rede, ocorre a multiplexação. Após o protocolo IP entregar os dados para a camada de transporte no receptor, é necessário direcioná-los para um dos dois processos. Esta tarefa de direcionamento das informações é chamada demultiplexação.

2.1.3 Camada de Rede

Em redes de comutação de pacotes existem dois principais modelos de serviço que o usuário pode definir: circuito virtual ou datagrama. Em redes de circuitos virtuais o caminho a ser percorrido pelos pacotes fica pré determinado de forma que cada roteador ao longo do caminho fica ciente de todos os circuitos virtuais (CVs) que passam por ele. Tecnologias como ATM (*Asynchronous Transfer Mode* — Modo de Transferência Assíncrono), Frame-relay e X.25 usam modelo de circuitos virtuais. Já em redes datagrama (modelo usado na Internet) os pacotes são roteados de acordo com o endereço de destino, os roteadores não guardam nenhum estado das conexões fim a fim e o caminho a ser percorrido pode variar de acordo com a disponibilidade dos enlaces. Desta maneira, a ordem dos pacotes que chegam no destinatário pode não ser a mesma ordem que foi enviada pelo emissor. PAXSON [55] faz um interessante estudo quantitativo sobre como os pacotes se comportam na Internet, discutindo entre outros assuntos a perda de pacotes, a largura de banda do enlace e o comportamento dos pacotes quando são roteados por caminhos diferentes.

Este modelo de serviço empregado na Internet, também chamado de serviço de melhor esforço, não fornece nenhuma garantia de que: não haverá perdas; os pacotes serão entregues; a temporização³ será preservada e que haverá indicação de um congestionamento da rede. Apesar de não apresentar estas garantias, este modelo funciona muito bem e está sendo ampliado com o intuito de incluir os serviços integrados (IntServ) e os serviços diferenciados (DiffServ) que atualmente são os serviços que implementam QoS (*Quality of Service* — Qualidade de Serviço).

Qualidade de Serviço — QoS

Com a tecnologia QoS é possível oferecer maior garantia e segurança nas aplicações da Internet, uma vez que o tráfego de aplicações multimídia (como voz sobre IP e transmissão de áudio e vídeo) passam a ter prioridade sobre as aplicações tradicionais (como o correio eletrônico) que continuam a utilizar o serviço de melhor esforço. Quando um congestionamento ocorre, apenas os pacotes que estão utilizando o serviço de melhor esforço são descartados. Para implementar esta aplicação, os pacotes são marcados de acordo com o tipo de aplicação, e os roteadores são desenvolvidos para criar filas distintas para cada aplicação considerando, inclusive, as prioridades das mesmas.

O modelo IntServ, que implementa QoS, caracteriza-se pela reserva de recursos antes de iniciar uma comunicação. Esta reserva de recursos é feita para aplicações que necessitam de

³Algumas aplicações, como telefonia e jogos pela Internet, exigem limitações de tempo na entrega dos dados para funcionarem perfeitamente.

um atraso constante, ou seja, que não haja variações no atraso (*jitter*) e para aplicações que requerem segurança, ou seja, que seus pacotes passem pelos roteadores sem serem descartados e sem sofrer atrasos nas filas. Na figura 2.4, suponha-se que o computador **A** deseje fazer uma comunicação de voz através da Internet com o computador **B**. A aplicação de voz sobre IP necessita de baixo atraso e baixa variação no atraso para que durante a transmissão dos dados sejam mantidos os requisitos de qualidade. Então, **A** envia uma mensagem para **B** especificando as características do tráfego; e quando a mensagem chega até **B**, inicia-se o processo de alocação de recursos (buffers e largura de banda) pelos roteadores ao longo do caminho (na figura 2.4, o núcleo da rede possui apenas um roteador). Caso nenhum roteador rejeite a solicitação dos recursos, a aplicação é iniciada.

No modelo IntServ, os roteadores fazem classificações dos diversos fluxos colocando-os em filas distintas. Isso exige uma grande capacidade de processamento, armazenamento e ótimos algoritmos de processamento de fila, aumentando os custos dos roteadores e seu grau de complexidade.

No DiffServ, são os próprios emissores que marcam o campo DS Field (Differentiated Service Field — Campo de Serviço Diferenciado) no cabeçalho de um pacote IP. O DS Field é uma nova arquitetura para o campo ToS (Type of Service — Tipo de Serviço) que já existia na definição do pacote IP, e é nele que são codificadas as classes para o DiffServ. Entretanto, como o DS Field é marcado nos emissores, não existe como se ter garantias de que existam recursos disponíveis para uma determinada aplicação; fazendo com que, por exemplo, um pacote que chega a um roteador que não provê QoS com DS Field, seja considerado como um pacote de serviço de melhor esforço.

Para melhorar a alocação de recursos no DiffServ foi inserido o BB (*Bandwidth Broker* — Gerenciador de Largura de Banda), o qual é responsável pela verificação de disponibilidade de recursos, pelas prioridades dentro e entre organizações e pela autorização para uma conexão dentro do domínio QoS.

O DiffServ exige menos processamento dos roteadores, o que implica em custos menores tanto no desenvolvimento quanto na atualização dos algoritmos de classificação de fluxos e, esta maior simplicidade na implementação dos serviços diferenciados seja a razão pela qual o DiffServ vem sendo preferido para implementar QoS.

O QoS se consolida cada vez mais como uma nova forma de comércio na Internet que, além de propor serviços diferenciados para as mais diversas aplicações, é uma proposta de melhorar a eficiência na transferência de dados, suprimindo as necessidades de confiabilidade e garantias para aplicações que necessitam destas características. Contudo, para XIAO et al. [64], os usuários da Internet que precisarem de qualidade de serviço, terão que pagar um

bom preço para que suas aplicações rodem utilizando QoS; dando direito, desta forma, a uma maior quantidade de largura de banda a quem puder pagar mais. Nesta linha de raciocínio, SHIN et al. [59] estudaram a influência das estratégias de preço do QoS no futuro da comercialização do QoS na Internet e, propuseram um ponto de equilíbrio entre o serviço prestado e o preço a ser cobrado.

Endereçamento IP

O protocolo IP é o responsável pelo roteamento dos pacotes entre sistemas finais. Ele define o endereçamento da camada de rede e as ações que devem ser tomadas pelos roteadores e os sistemas finais. O protocolo IP possui duas versões que são usadas atualmente: o IPv4; e o IPv6.

O endereçamento IP está diretamente ligado a uma interface. A interface é a fronteira entre o sistema final e o enlace físico e também entre o roteador e os vários enlaces conectados a ele. Assim, se um computador tem mais de uma interface, cada interface terá um endereço IP. Na versão IPv4, cada endereço IP possui 32 bits⁴ divididos em duas partes: os primeiros bits constituem a parcela da rede e os bits restantes são destinados ao endereço IP do sistema final. Considerando, por exemplo, o endereço IP 234. 12. 153. 6. Cada um dos quatro números decimais deste endereço equivale a um conjunto de oito bits do endereço. Conseqüentemente, o endereço 234. 12. 153. 6, da forma binária, é:

$$\underbrace{11101010}_{234} \quad \underbrace{00001100}_{12} \quad \underbrace{10011001}_{153} \quad \underbrace{00000110}_6$$

Na figura 2.4 um esquema de endereçamento IP é mostrado. Nela pode-se observar três redes IP: 234.1.1.0/24, 234.1.2.0/24 e 234.1.3.0/24.⁵ A notação '/24', também chamada de máscara de rede, indica que os 24 bits da esquerda definem o endereço da rede⁶. Desta forma, nota-se, que uma rede IP se forma quando as interfaces estão conectadas entre si por um único enlace físico, sem passar por um roteador.

Os endereços IP ainda são divididos em classes, onde cada classe é definida conforme o espaço a ser destinado para a parcela de rede e para a parcela do endereço da interface. Esta divisão do endereço IP em classes promove a flexibilidade exigida para suportar os diferentes tamanhos de redes e, uma das suas características fundamentais é que cada endereço contém um código que identifica a que classe ele pertence. Conforme a figura 2.5, este código é

⁴O IPv6 usa 128 bits para o endereço.

⁵A rede 234.1.1.0/24, por exemplo, consiste em três interfaces de hospedeiros (234.1.1.1, 234.1.1.2 e 234.1.1.3) e uma interface de roteador (234.1.1.4).

⁶O conjunto de bits que definem o endereço da rede também é chamado de **prefixo de rede**.

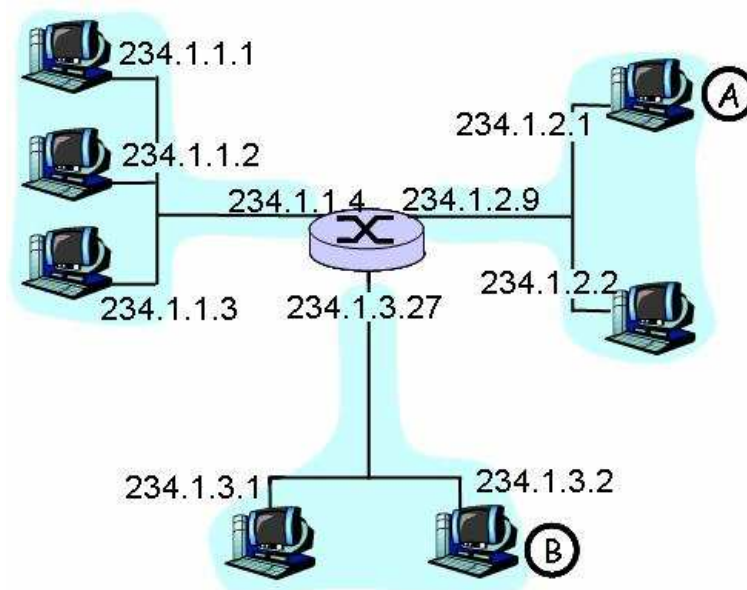


Figura 2.4: Endereço IP. Adaptado de [39].

implementado nos primeiros bits do endereço IP (chamados de bits de alta ordem). Por exemplo, se os primeiros dois bits de um endereço IP são 1-0, este endereço pertence a classe B.

A classe A destina 8 bits para o prefixo da rede, 24 bits para identificar a interface dentro da rede e o seu bit de alta ordem é 0 (zero). Assim, podemos ter 2^7 redes e, cada uma com 2^{24} interfaces. Cada endereço da classe B tem 2^{14} redes, com até 2^{16} interfaces em cada rede; e o seus bits de alta ordem são 1-0. Na classe C, os bits de alta ordem são 1-1-0 e para identificar a rede e as interfaces são usados 24 bits e 8 bits, respectivamente. Além destas três classes mais populares, existem mais duas classes: Os endereços da classe D tem como bits de alta ordem 1-1-1-0 e são destinados aos endereços multicast; os endereços da classe E, cujo os primeiros bits do endereço são 1-1-1-1-0, são reservados para uso experimental. SEMERIA [58] faz um ótimo estudo sobre o endereçamento IP detalhando ainda mais a versão IPv4.

2.1.4 Camada de Enlace e Física

Camada de enlace

Para levar a informação entre clientes e servidores⁷, a camada de rede comuta pacotes

⁷Na camada de enlace, os clientes, os servidores e os roteadores são simplesmente chamados de **nós**, pois, não faz diferença saber a diferença entre eles.

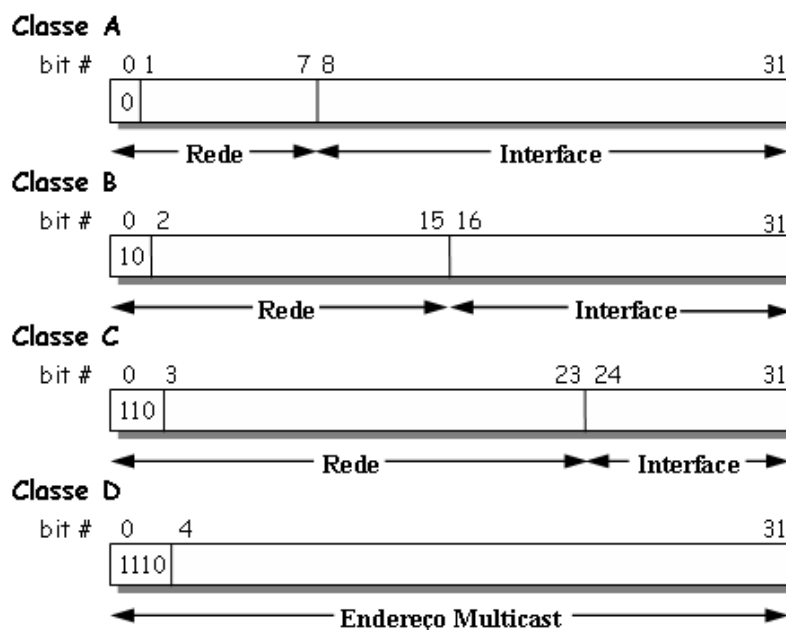


Figura 2.5: Formato das classes de endereço IP.

por meio de uma série de roteadores e, para fazer isto, depende dos serviços da camada de enlace, que é responsável pela transferência de um pacote sobre um enlace individual. Os principais protocolos desta camada, como mostrado na figura 2.2, são os protocolos Ethernet, Token Ring, PPP (*Point-to-Point Protocol* — Protocolo ponto a ponto), FDDI, ATM e o Frame Relay.⁸ As unidades de dados trocadas por estes protocolos são chamadas de **quadros** e cada quadro encapsula um pacote da camada de rede.

Uma característica interessante na camada de enlace é que um pacote pode ser transmitido por diferentes protocolos de enlace ao longo do caminho. Por exemplo, na figura 2.4, se o computador A envia dados para o computador B; os pacotes referentes a estes dados podem ser manipulados pelo protocolo PPP no primeiro enlace (do computador A até o roteador), e por Frame Relay no segundo enlace (do roteador até o computador B). Fato que não ocorre com os protocolos da camada de transporte.

A quantidade máxima de dados que um quadro pode carregar é chamada de MTU (*Maximum Transfer Unit* — Unidade Máxima de Transferência). Os protocolos Ethernet e PPP possuem uma MTU de 1500 bytes. Isso significa que, se o datagrama IP exceder 1500 bytes, o roteador terá que fragmentar os dados deste datagrama em datagramas IP menores. O problema que surge neste ponto é que cada enlace ao longo do caminho pode possuir uma MTU diferente, o que pode gerar um grande número de fragmentações, aumentando as ocor-

⁸Alguns autores preferem colocar a Ethernet, o ATM, o PPP, o Frame Relay e o Token Ring como padrões que fornecem a base para o desenvolvimento de tecnologias de redes, do que como protocolos.

rências de atrasos e perdas na transmissão de dados. Por simplicidade, a remontagem destes datagramas IP menores é feita nos sistemas finais e não no núcleo da rede e, logicamente, antes que cheguem à camada de transporte do destinatário.

Como na camada de transporte, os serviços oferecidos pelos protocolos da camada de enlace podem ser diferentes. A camada de rede pode, por exemplo, escolher utilizar ou não um serviço de entrega confiável de dados; utilizando, para este fim, as técnicas de retransmissão e reconhecimento de dados. Diferentemente da camada de transporte onde o serviço é feito fim a fim, a camada de enlace fornece seus serviços entre dois nós ligados por um único enlace. Em enlaces com baixa taxa de erro, como enlaces de fibra e de pares de fios de cobre trançados, o serviço de entrega confiável de dados da camada de enlace é raramente usado. Contudo, para enlaces com altas taxas de erros, como enlaces sem-fio (*wireless*), o serviço de entrega confiável da informação se torna imprescindível.

Entre outros serviços da camada de enlace, destacam-se: o controle de fluxo que, como na camada de transporte, limita a taxa de transmissão entre emissor e receptor evitando que um nó receptor receba quadros a uma velocidade maior do que sua capacidade de processá-los; a detecção de erros, onde o nó transmissor envia bits de detecção de erros no quadro; e a correção de erros, que é semelhante à detecção de erros, entretanto o nó receptor pode também determinar onde ocorreu o erro no quadro.

Camada física

Ao contrário da camada de enlace que possui a tarefa de mover quadros inteiros de um elemento da rede até o elemento adjacente, a camada física é responsável por movimentar bits de dados de um nó para o nó seguinte através dos meios físicos. Os dados, na forma de uns e zeros, são transformados em sinais elétricos, pulsos de luz ou sinais sem fio (*wireless*), através de uma placa de rede. Ao receber os dados, a placa de rede da interface receptora transforma os sinais elétricos, pulsos de luz ou sinais sem fio de volta a forma de uns e zeros; para depois seguir a hierarquia de camadas do TCP/IP.

Os protocolos dessa camada dependem diretamente do meio de transmissão. Por exemplo, a Ethernet tem muitos protocolos de camada física: um para par de fio de cobre trançado; outro para cabo coaxial; outro para fibra óptica; etc. Sendo que, o bit é movimentado de maneira diferente em cada caso. Alguns dispositivos que fazem parte da camada física são: repetidores, repetidores multiporta (chamados de *hubs*), MAUs (*Media Access Units* — Unidades de Acesso de Mídia) e transmissores/receptores (chamados de *transceivers*) que convertem um tipo de sinal em outro.

2.1.5 Estrutura dos Cabeçalhos e Segmentos

Para que haja uma perfeita comunicação entre as camadas da pilha de protocolos, a arquitetura TCP/IP se utiliza dos chamados **cabeçalhos**. Os cabeçalhos contêm informações adicionais referentes aos protocolos que operam em cada camada. Por exemplo, conforme figura 2.6, a camada de transporte do emissor coloca um cabeçalho C_t na frente da mensagem M proveniente da camada de aplicação para identificá-la e envia para a camada de rede. O cabeçalho inclui informações de controle, como números de seqüência, para permitir que a camada de transporte do receptor repasse os dados na ordem correta, para o caso das camadas de rede e enlace não conseguirem manter a a seqüência correta das informações. No receptor, as camadas inferiores removem seus respectivos cabeçalhos antes de repassar os dados pilha de protocolos acima.

Quando uma camada n do emissor adiciona um cabeçalho C_n na frente da mensagem

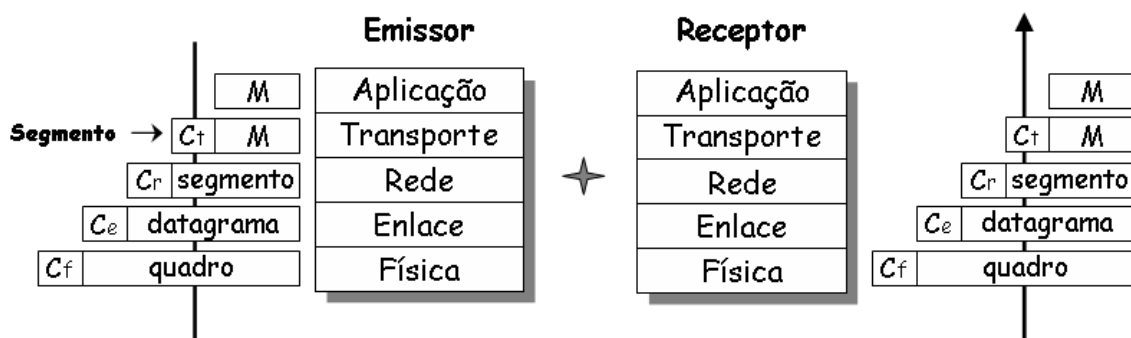


Figura 2.6: Estrutura de cabeçalhos e segmentos.

M , ele está criando uma unidade de dados da camada n ou mais comumente **n -PDU** (*Protocol Data Units* — Unidade de Dados de Protocolo da Camada n). A PDU recebe uma denominação diferente conforme a camada da pilha de protocolos. Na camada de transporte é chamada de **segmento**; na camada de rede é conhecida como **datagrama**; e para a camada de enlace é reservado o termo **quadro**. Geralmente, a PDU da camada física não recebe nenhum nome especial e, também, o seu cabeçalho C_f pode ser nulo.

2.2 ATRASOS NA INTERNET

2.2.1 Tipos de Atrasos

Na subseção 2.1.3, foi comentado que a ordem dos pacotes que chegam a um destinatário pode não ser a mesma ordem que foi enviada pelo emissor. Isto acontece porque os pacotes

enviados pelo emissor podem seguir caminhos diferentes conforme a disponibilidade dos enlaces e roteadores. Assim, o atraso sofrido por cada pacote também pode ser diferente. Os atrasos mais importantes, mostrados na figura 2.7, são o atraso de propagação, o atraso de fila, o atraso de processamento e o atraso de transmissão. Considerando que existam inúmeros nós entre o emissor e o receptor, os atrasos a que todas as informações que trafegam pela Internet estão sujeitas pode degradar bastante a eficiência entre processos de aplicações. Por isso, conhecer os tipos de atrasos e saber como tratá-los de forma a diminuir os seus impactos, se torna inevitável. Por conseguinte, alguns comentários a respeito da influência dos tipos de atraso no comportamento da Internet serão feitos baseados na figura 2.7.

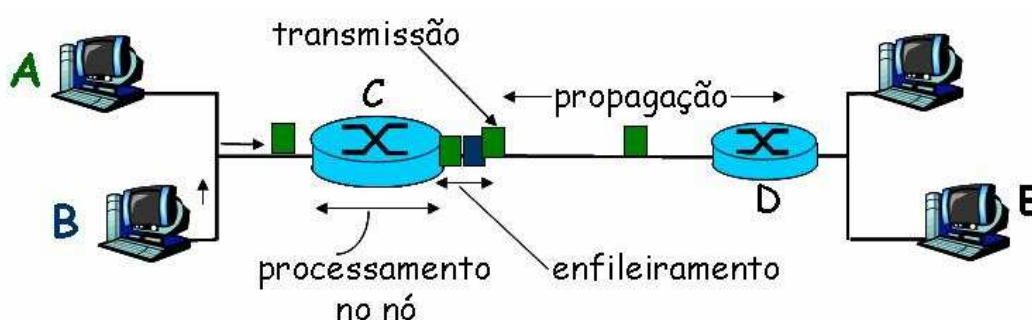


Figura 2.7: Tipos de atraso. Adaptado de [39].

Atraso de propagação

O atraso de propagação é o tempo necessário para um pacote, ou mais especificamente o conjunto dos bits que formam o pacote, percorra o enlace entre o roteador C e o roteador D. A velocidade com que cada bit é propagado é limitada pela velocidade de propagação de cada enlace, que muda conforme a tecnologia do meio físico do enlace. Por exemplo, a fibra óptica permite atingir 70% da velocidade da luz, ou seja, cerca de $2,0 \cdot 10^8$ m/s. Nos cabos de cobre, a velocidade situa-se entre 40% e 60% da velocidade da luz, ou seja, na faixa de $1,2 \cdot 10^8$ m/s a $1,8 \cdot 10^8$ m/s.

Ao percorrer uma distância d , a uma velocidade de propagação v , o atraso de propagação sofrido por um bit é d/v . Esta expressão é válida para dois nós que estão diretamente ligados ao mesmo meio físico, o que, na prática, nem sempre acontece. Desta forma, quando é necessário que um bit passe por diversos meios físicos, o atraso de propagação final é a soma dos atrasos de cada um dos meios físicos que estão entre os dois nós.

Atraso de transmissão

No roteador C, os pacotes são transmitidos conforme a ordem de chegada, isto é, o primeiro pacote a chegar é transmitido primeiro. É bom notar que, quando um pacote chega

ao roteador D, vindo dos computadores A ou B, ele verifica o cabeçalho do pacote para direcioná-lo para o buffer (fila) de saída do enlace mais apropriado. E, só é transmitido para o enlace quando não houver outros pacotes à frente da fila.

O atraso de transmissão de um pacote é L/R , onde L (em bits) é o comprimento do pacote e R (em bits/seg.) é a taxa de transmissão do enlace entre o roteador C e o roteador D. Para um enlace com tecnologia Ethernet de 10 Mbps, por exemplo, temos $R = 10Mbps$. E, se considerarmos que os pacotes que trafegam por este enlace tenham um tamanho de 64 bytes (512 bits); teremos, então, um atraso de transmissão de $51,2\mu s$ para cada pacote.

É importante entender a diferença entre o atraso de transmissão e o atraso de propagação. O primeiro é o tempo que o roteador leva para enviar o pacote para fora; é uma função do comprimento do pacote e da taxa de transmissão do enlace, não sendo dependente da distância entre roteadores. O segundo, por outro lado, é o tempo que um pacote leva para ir de um roteador para outro, não sendo dependente do comprimento do pacote e da taxa de transmissão do enlace.

Atraso de fila

O atraso de fila é o tempo que um pacote espera na fila para ser transmitido. Este atraso é influenciado pela quantidade de outros pacotes que chegaram antes no roteador e que também estão esperando para serem transmitidos. Se um pacote chegar no roteador C, e a capacidade de armazenamento da fila estiver esgotada, o pacote será imediatamente descartado. Como se verá adiante, pacotes também podem ser descartados mesmo que já estejam na fila; isto é feito pelo controle de congestionamento do protocolo TCP.

A carga de fluxos que está passando pelo roteador é fator determinante no atraso de fila. Com um tráfego muito intenso, o atraso de fila é longo. Por outro lado, se a fila estiver vazia, o tempo de atraso de fila praticamente não existe. Devido a esta variação no tempo de espera na fila entre um pacote que está no começo da fila para um pacote que acabou de chegar ao roteador, normalmente, o tempo de atraso de fila é medido estatisticamente através de técnicas como: atraso de fila médio; variância do atraso de fila e a probabilidade de atraso de fila.

Atraso de processamento

O atraso de processamento é constituído de várias partes. As que se destacam mais são: o tempo necessário para o roteador C verificar o cabeçalho de um pacote e determinar o melhor enlace de saída; e a soma de verificação (*checksum*) que auxilia o roteador na detecção de erros de bits. Outro fator que influencia o tempo de atraso de processamento é a eficiência do algoritmo de roteamento que é implementado no roteador.

2.2.2 Atraso Nodal

O atraso total no roteador C , também chamado de atraso nodal, é a soma dos vários tipos de atrasos mostrados na subseção 2.2.1. Desta forma, denominando t_{fila} , t_{proc} , t_{trans} , t_{prop} , respectivamente, como os atrasos de fila, de processamento, de transmissão e de propagação, chega-se que o atraso nodal é dado por :

$$t_{nodal} = t_{fila} + t_{proc} + t_{trans} + t_{prop}.$$

A influência de cada tipo de atraso varia conforme a configuração da rede. Para redes locais (LANs), por exemplo, t_{prop} não tem grande peso no atraso nodal, chegando em alguns casos a ser desprezível; entretanto, quando a comunicação entre dois roteadores é feita por um enlace de satélite, t_{prop} pode ser o termo dominante no t_{nodal} [39]. O mesmo raciocínio vale para o t_{trans} . Para uma LAN com velocidade de 100 Mbps, a sua influência praticamente desaparece; ao contrário do que acontece quando se transmite pacotes por enlaces com velocidade na ordem de 56 Kbps, onde passa a ter um peso maior. Já o t_{fila} ; como comentado na subseção 2.2.1, varia de pacote para pacote, sofrendo forte influência da intensidade de tráfego na rede e da capacidade de armazenamento do buffer. Sendo que, quanto maior a capacidade do buffer menor será t_{fila} .

2.3 CARACTERÍSTICAS DO TCP

Como comentado na subseção 2.1.2, para garantir que a cadeia de dados que o receptor lê a partir do seu buffer de recebimento TCP não está corrompida, não tem lacunas, não possui duplicações e está em seqüência, o protocolo TCP cria um serviço de transferência confiável de dados sobre o protocolo IP, o qual não fornece nenhuma garantia de que os pacotes serão entregues. Para tal implementação, são usados princípios como: temporizadores para cada segmento liberado para a camada de rede; detecção de erro (*checksum*), campos de cabeçalho para números de seqüência e de reconhecimento, retransmissão de pacotes perdidos e reconhecimento cumulativo.

Os princípios citados anteriormente, os quais são usados pelo TCP, são oriundos de vários outros protocolos. Dentre eles, é fundamental entender o funcionamento dos protocolos de transferência de dados com paralelismo⁹: Volta-N (*Go-Back-N* — GBN) e repetição seletiva (*Selective Repeat* — SR).

⁹O termo paralelismo é empregado em [39] como opção para a tradução da palavra inglesa *pipelining*; que também pode ser traduzida como dutagem. Desta forma, os protocolos com paralelismo também são conhecidos como protocolos dutados.

2.3.1 Transferência de dados com paralelismo

Os protocolos de transferência de dados com paralelismo utilizam a técnica de resposta de reconhecimentos (*acknowledgements* — ACKs). Este método é utilizado para que o remetente possa saber o que está acontecendo no destinatário e, é considerada uma realimentação explícita que o destinatário fornece ao remetente. Quando um pacote é entregue corretamente, o destinatário envia um ACK com o número de seqüência do segmento TCP para o remetente, o qual obtém a certeza de que o pacote chegou no seu destino.

Protocolo volta-N (GBN)

No protocolo volta-N, o remetente pode enviar vários pacotes sem esperar por reconhecimentos, porém fica limitado a um número N de pacotes transmitidos e ainda não reconhecidos. Este conjunto de números de seqüência permitidos para pacotes enviados mas ainda não reconhecidos é denominado como **janela** e, por conseguinte, N é chamado de **tamanho de janela**. A medida que os pacotes vão sendo reconhecidos, como fica implícito na figura 2.8, a janela vai se deslocando para o espaço de números de seqüência ainda não usados. Ou seja, os números de seqüência maiores ou iguais a $base+N$ só podem ser utilizados quando o remetente recebe o reconhecimento de um pacote não reconhecido. Devido a este modelo de janela, o protocolo volta-N é conhecido como **protocolo de janela deslizante** (*sliding window protocol*).

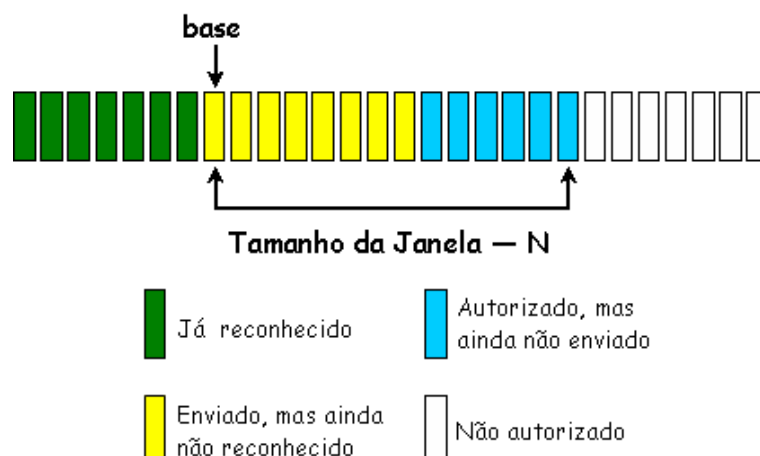


Figura 2.8: Comportamento da janela no protocolo Volta-N — visão do remetente [39].

Existe uma distinção dos eventos que são realizados na janela do remetente e na janela do destinatário. Esses eventos mostram como os pacotes são tratados antes de serem enviados para a camada de rede (no caso do remetente) ou para a camada de aplicação (no caso do destinatário). A janela do remetente que utiliza o protocolo volta-N, deve operar pelo menos

três eventos:

- *Chamada da camada de aplicação.* Quando a camada de transporte é comunicada pela camada de aplicação que existem dados para serem enviados, é feita uma verificação se existe espaço na janela do remetente. Se ela não estiver cheia, os devidos pacotes são criados e enviados. Se ela estiver cheia, ou os dados são devolvidos a camada superior, ou são armazenados em um buffer para serem enviados mais tarde.
- *Reconhecimento.* No protocolo volta-N, o recebimento de um ACK de um pacote com número de seqüência n é considerado um **reconhecimento cumulativo**, ou seja, mostra que todos os pacotes enviados até e inclusive n foram corretamente recebidos no destinatário.
- *Temporização.* Como o protocolo volta-N usa apenas um temporizador para o pacote mais antigo enviado mas que ainda não foi reconhecido, no caso de ocorrer um estouro deste tempo de transmissão, o lado remetente reenvia todos os pacotes que tinham sido transmitidos mas que ainda não tinham sido reconhecidos.

O lado do destinatário no protocolo volta-N possui a mesma simplicidade que no remetente. Se um pacote chegar ao destinatário com o número de seqüência n e estiver na ordem, isto é, é realmente este pacote que o destinatário está esperando e não um pacote com número de seqüência $n-1$, então é enviado um ACK para o remetente e o pacote será entregue para a camada de aplicação. Esta é a única maneira das informações serem corretamente recebidas, em todos os outros casos, o destinatário descarta os pacotes e envia um ACK para o remetente do último pacote corretamente recebido.

Protocolo de repetição seletiva (SR)

Um problema muito comum no protocolo volta-N são as retransmissões desnecessárias, as quais acontecem principalmente quando o tamanho da janela é grande e a taxa de erro dos enlaces é alta. Para reduzir estas retransmissões, o protocolo de repetição seletiva propõe retransmitir somente os pacotes que chegaram corrompidos no receptor; isto é implementado colocando-se um temporizador para cada pacote transmitido pelo emissor.

Esta forma de tratar os pacotes, impõe um reconhecimento individual para cada pacote no receptor. Este método permite que o receptor reconheça pacotes, que não estão corrompidos, mesmo que estejam fora de ordem, ao contrário do protocolo volta-N onde o reconhecimento é cumulativo.

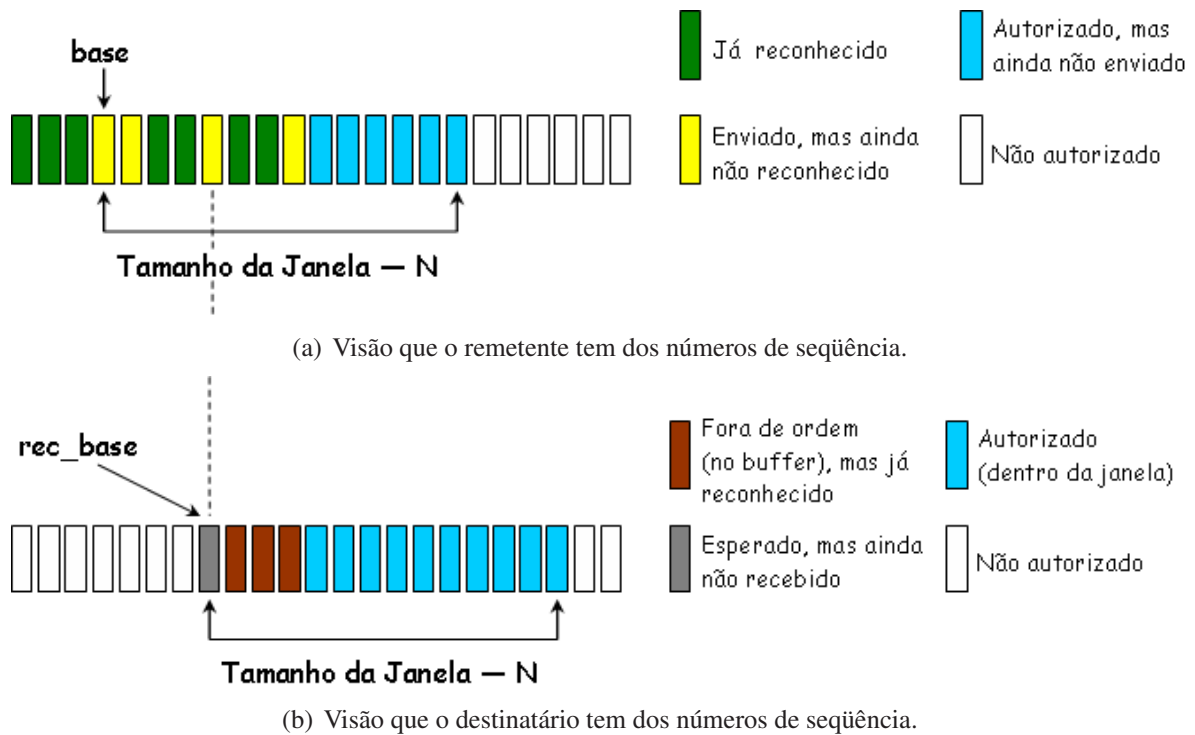


Figura 2.9: Comportamento da janela no protocolo SR. Adaptado de [39].

A figura 2.9 mostra que a complexidade na manipulação dos pacotes tanto no remetente como no destinatário aumenta consideravelmente. Como pode ser observado na janela do remetente, os pacotes enviados e não reconhecidos se misturam com os pacotes já reconhecidos, não existindo a ordem apresentada na figura 2.8. De forma semelhante, na janela do destinatário, os pacotes que chegam fora de ordem são armazenados no buffer com numeração consecutiva a partir do número de seqüência `rec_base`. Estes pacotes serão repassados para a camada de aplicação somente quando todos os pacotes com números de seqüência menores forem corretamente recebidos. Conseqüentemente, o destinatário entrega um conjunto de pacotes para a camada superior. É bom lembrar que como o protocolo volta-N, o protocolo de repetição seletiva também é um protocolo de janela deslizante e, desta maneira, à medida que o protocolo vai operando, a janela se desloca para para frente sobre o espaço de números de seqüência antes não autorizados. Outro ponto em comum são os eventos citados no protocolo GBN, evidentemente que adaptados para as características do protocolo SR.

2.3.2 A Conexão TCP

O protocolo TCP, como os protocolos GBN e SR, utiliza uma transferência do tipo **full-duplex**. Assim, é possível o destinatário enviar um ACK ou alguma informação no mesmo

instante que o remetente está transmitindo pacotes, promovendo uma melhor utilização da capacidade do enlace. Além disso, o TCP sempre estabelece uma conexão entre um único remetente e um único destinatário, não sendo possível, através de uma única conexão TCP, a emissão de dados de um remetente para vários destinatários.

Depois do estabelecimento da conexão, também denominada de apresentação de três vias (*3-way handshake*), os processos da camada de aplicação tanto do remetente quanto do destinatário ficam autorizados a trocarem informações. Quando a camada de aplicação passa para a camada de transporte os dados a serem transmitidos, o TCP os direciona para o buffer de envio da conexão, o qual foi reservado durante a apresentação de três vias. Como nos protocolos GBN e SR, o próximo passo é aguardar espaço na janela do remetente, para só então os dados serem enviados.

A quantidade máxima de dados que pode ser retirada do buffer de envio e colocada em um segmento é limitada pelo MSS (*Maximum Segment Size* — Tamanho Máximo do Segmento). Já se referiu na subseção 2.1.4, à quantidade máxima que um quadro da camada de enlace pode carregar, a MTU. Geralmente, estes limites máximos que cada camada impõe são escolhidos de forma a diminuir ao extremo a probabilidade de existir a fragmentação dos datagramas IP pelos roteadores do núcleo da rede.

Dentro do segmento TCP existem campos de cabeçalho e um campo de dados. O MSS limita o tamanho máximo do campo de dados, contudo, nem toda aplicação necessita utilizar todo o espaço do campo de dados. No cabeçalho estão incluídas várias informações necessárias para uma transmissão de dados; as principais estão distribuídas pelos seguintes campos: campo de soma de verificação, para detectar erros nos bits; campos dos números de porta da fonte e do destino, que são usados na multiplexação e demultiplexação de dados; campo do tamanho da janela, usado no controle de fluxo; campos do número de seqüência e do número de reconhecimento, usados na implementação do serviço confiável de dados.

2.3.3 Números de Seqüência e de Reconhecimento

Os números de seqüência e de reconhecimento constituem uma parte fundamental no serviço de entrega confiável de dados. Através deles é possível saber exatamente qual o segmento que está faltando dentre os vários que compõe uma determinada informação. Este serviço se torna imprescindível dentro de uma rede de comutação de pacotes (caso da Internet, através do protocolo IP), a qual pode rotear pacotes pelos mais diferentes caminhos conforme a disponibilidade dos enlaces, não garantindo a ordem dos pacotes que chegam a um destinatário.

O protocolo TCP, apesar de enviar pacotes ou segmentos para o protocolo IP, considera

os dados como um conjunto de bytes ordenados. Desta forma, o número de seqüência de um segmento é o número do primeiro byte do segmento. Um exemplo simples. Suponha-se que um computador A deseje enviar um arquivo de dados para um computador B, que o tamanho do arquivo seja de 100 000 bytes e que o MSS seja de 500 bytes. Conforme a figura 2.10, o TCP do computador A distribui este arquivo em 200 segmentos. Atribuindo um número de seqüência 0 (zero) para o primeiro segmento; um número de seqüência 500, para o segundo; e assim para todos os outros segmentos.

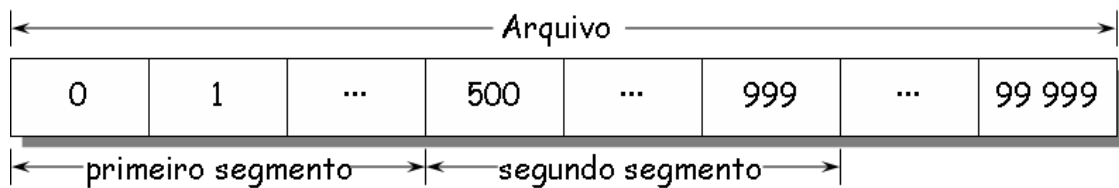


Figura 2.10: Números de seqüência em segmentos TCP.

De forma semelhante, o número de reconhecimento que o protocolo TCP do computador B atribui a um segmento, é o número de seqüência do próximo byte que ele estiver esperando do computador A. Como outro exemplo, suponha-se que o computador B receba três segmentos contendo bytes de 0 a 1499 bytes e outro segmento com os bytes de 2000 a 2499. Por algum motivo, os bytes 1500 a 1999 ainda não chegaram no computador B. Por conseguinte, ele coloca no campo de número de reconhecimento de seu próximo segmento, o número 1500, referente ao segmento que ainda não chegou.

Como no protocolo volta-N, o TCP usa reconhecimentos cumulativos. Conseqüentemente, no exemplo anterior, os bytes até e inclusive 1499 que foram corretamente recebidos pelo computador B, são reconhecidos através de um ACK enviado para o computador A. Já os bytes 2000 a 2499 que chegaram fora de ordem, são armazenados no buffer e reconhecidos somente após o recebimento dos bytes que estão faltando.

Retransmissão rápida

Quando o remetente recebe um ACK do destinatário, é essencial que ele determine se o ACK é um primeiro ACK, ou se é um ACK duplicado que reconhece novamente um segmento que já foi reconhecido. O envio de um ACK duplicado ao remetente acontece justamente quando o destinatário detecta o recebimento de segmentos com números de seqüência fora de ordem, ou seja, o segmento que estava sendo aguardado ou se perdeu ou está no núcleo da rede. O destinatário permanece enviando ACKs que correspondem ao último segmento que foi recebido na ordem esperada até que a lacuna na janela seja completada. Se

o remetente receber três ACKs duplicados referentes ao mesmo segmento, então, ele executa uma retransmissão rápida, onde é retransmitido o segmento com número de seqüência seguinte ao segmento reconhecido três vezes. Este procedimento é feito, logicamente, antes de ‘estourar’ o tempo do temporizador do segmento em questão.

2.3.4 Controle de Fluxo

Pelo fato do protocolo TCP estabelecer uma conexão full-duplex entre dois sistemas finais, cada lado da conexão tem um buffer de envio e um buffer de recepção. Mais especificamente, quando chegam pacotes que estão corretos e em seqüência no destinatário, o TCP os coloca no buffer de recepção, onde ficam à disposição do processo de aplicação da camada superior. Contudo, a leitura desses dados disponíveis pode não ser feita imediatamente, dependendo do quão ocupado está o processo de aplicação (Em um leitor de e-mail como o Outlook, por exemplo, é possível configurar a recepção de novos e-mails a cada n minutos). Se o repasse dos dados do buffer para os processos de aplicação se prolongar por um período de tempo considerável, o remetente pode facilmente saturar o buffer de recepção da conexão e, conseqüentemente, ocasionar o descarte de vários pacotes.

Para controlar a transmissão de dados entre sistemas finais, o TCP possui um serviço de controle de fluxo, com o intuito de não deixar o remetente saturar o buffer do destinatário, seja pelo envio de informações em grande quantidade ou pela alta velocidade nas transmissões. Como mostra a figura 2.11, este controle é implementado através de uma variável, chamada de janela de recepção, dentro do buffer de recepção. Ela é ajustada de acordo com o tamanho do buffer, o número de bytes lidos pelo processo de aplicação e o número de bytes recém-chegados no buffer. Conclui-se então que, a janela de recepção não é estática,

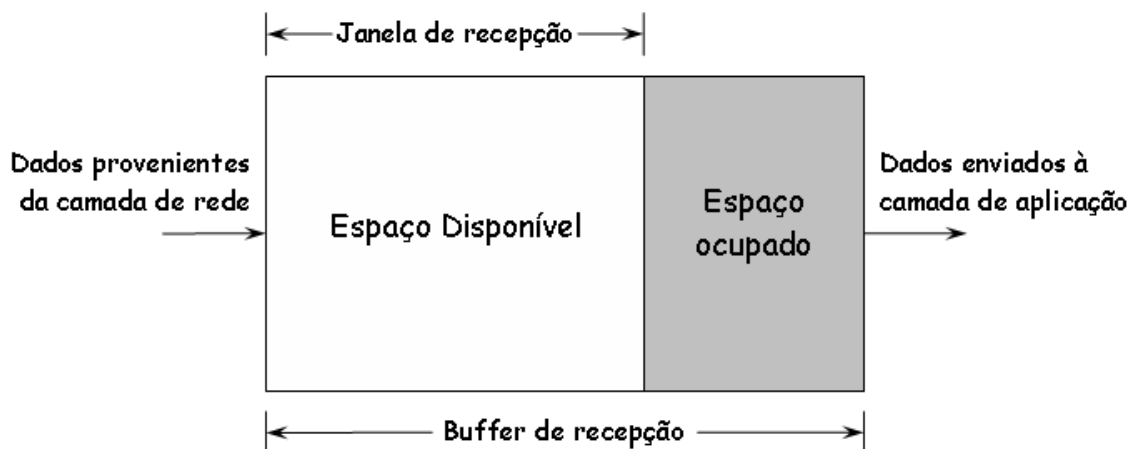


Figura 2.11: Buffer de recepção.

variando durante toda duração da conexão TCP. Por este motivo, o valor desta variável é constantemente enviada ao lado emissor da conexão, com o objetivo de informar a quantidade exata de espaço disponível no buffer do destinatário.

Porém, não é suficiente que o emissor apenas tome conhecimento do valor da variável em questão. Deste modo, somente mantendo-se a quantidade de pacotes enviados mas ainda não reconhecidos menor do que o valor mais recente da janela de recepção, é que se garantirá que o buffer do destinatário não ficará saturado.

2.3.5 Controle de Congestionamento

De modo semelhante ao controle de fluxo, o controle de congestionamento do TCP também controla a taxa de transmissão do remetente limitando o número de pacotes enviados mas ainda não reconhecidos. Entretanto, o motivo que leva o controle de congestionamento a agir na janela de emissão é o congestionamento no núcleo da rede e não a saturação do buffer do destinatário. Sendo que, este serviço oferecido pelo TCP está mais direcionado para o bom funcionamento da Internet do que para a melhoria direta da comunicação entre dois sistemas finais, tratando especificamente a causa do congestionamento da rede, isto é, os inúmeros computadores interligados em todo o mundo transmitindo dados na velocidade mais alta possível.

Para um usuário da Internet o desejável é que suas informações possam ser enviadas o quanto mais rápido; contudo, esta idéia não é a melhor quando aplicada na Internet como um todo. Por exemplo, quando a vazão (em bytes por segundo) na entrada dos roteadores de uma rede começa a aumentar muito, se aproximando da capacidade do enlace, a rede começa a ficar congestionada. Dentre outros problemas de uma rede congestionada pode-se destacar os seguintes: os pacotes passam a sofrer consideráveis atrasos nas filas dos roteadores; se o roteador possuir uma capacidade de buffer pequena, haverá pacotes descartados; se ocorrer o estouro do tempo de transmissão de um pacote que está atrasado nas filas dos roteadores, o remetente fará a retransmissão deste pacote, ‘desperdiçando’ os recursos¹⁰ da rede.

O controle de congestionamento do TCP utiliza o método fim a fim, isto é, os sistemas finais implementam o controle sem qualquer suporte da camada de rede. Deste modo, o congestionamento da rede deve ser detectado através do comportamento da rede. Por exemplo, eventos como o excesso de perda de pacotes, atrasos e retransmissões são tomados como indicação de congestionamento.

¹⁰Neste caso, a capacidade de transmissão do enlace, o roteador e o remetente são usados equivocadamente para retransmitir um pacote que não foi perdido, mas que apenas está atrasado.

Detalhamento do controle de congestionamento

Para implementar o mecanismo de controle de congestionamento, o TCP faz uso de duas variáveis: a janela de congestionamento e o patamar (*threshold*). Assim, dentro do buffer de emissão de um sistema final, teremos duas variáveis limitando a quantidade máxima de pacotes enviados mas ainda não reconhecidos que pode ser enviada dentro da conexão: a janela de recepção, referente ao controle de fluxo; e a janela de congestionamento, do método de controle de congestionamento. Sendo que, prevalecerá como limite, o menor valor destas duas variáveis.

Como comentado na subseção 2.3.2, após estabelecida a conexão, a camada de aplicação no remetente entrega seus dados para a camada de transporte, que os coloca no buffer de envio. Quando existe espaço na janela do remetente, o TCP divide estes dados em pedaços de tamanho MSS, os encapsula em segmentos TCP e envia para a camada de rede. Inicialmente o tamanho da janela de congestionamento é igual a um MSS. Se este segmento for enviado para o destinatário e reconhecido¹¹ sem que haja um estouro do tempo de retransmissão (*timeout*), o remetente aumentará a janela de congestionamento e passará a enviar dois segmentos de tamanho MSS. Desta maneira, a janela de congestionamento vai aumentando de um MSS para cada um dos segmentos reconhecidos, o que constitui um crescimento exponencial.

Esta fase de crescimento da janela de congestionamento é chamada de partida lenta e continua até atingir o valor de patamar. Neste ponto, a janela começa a crescer linearmente, é a fase de prevenção de congestionamento. Ou seja, considerando W o tamanho da janela, então, a cada W segmentos reconhecidos o remetente mudará a janela de congestionamento para $W + 1$.

Em ambas as fases do controle de congestionamento, quando ocorre uma perda, o valor do patamar é reduzido para a metade do valor atual da janela de congestionamento e a janela de congestionamento é drasticamente ajustada para um MSS; iniciando-se então novamente a evolução da janela de congestionamento como mostra a figura 2.12.

2.3.6 Variações do Controle de Congestionamento do TCP

Tahoe TCP

O algoritmo de controle de congestionamento detalhado anteriormente foi proposto por JACOBSON [33] e é chamado de Tahoe. Além das fases de partida lenta e de prevenção

¹¹O tempo transcorrido desde o instante em que um segmento é enviado até o instante que ele é reconhecido no remetente é conhecido como RTT (*Round-Trip Time* — Tempo de Ida e Volta), o qual varia de segmento para segmento.

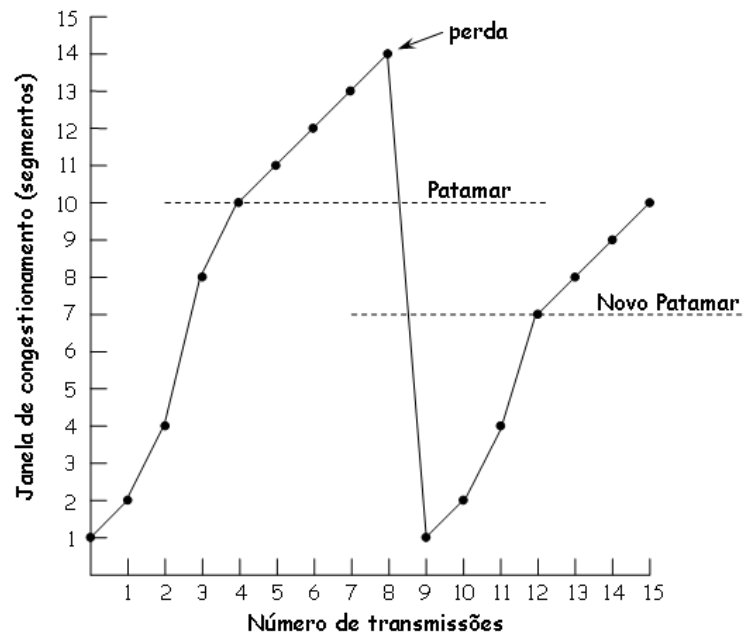


Figura 2.12: Comportamento da janela de congestionamento do TCP.

de congestionamento, o Tahoe também inclui o mecanismo de retransmissão rápida (*fast retransmit*) que foi estudada na subseção 2.3.3 e modificações no cálculo da estimativa do tempo de ida e volta (RTT) que é usado para definir o tempo de retransmissão dos segmentos TCP.

Reno TCP

Talvez, o principal problema do algoritmo Tahoe seja os longos períodos de tempo que o remetente pode ter de esperar pelo esgotamento de temporização, quando um segmento for perdido. Para resolver este problema foi proposto [34], no algoritmo chamado Reno, o mecanismo de recuperação rápida (*fast recovery*) que cancela a fase de partida lenta após uma retransmissão rápida. Desta forma, o Reno reduz a janela de congestionamento para um MSS, somente se a perda acontecer por um estouro do tempo de retransmissão. Quando a perda é detectada através de ACKs duplicados, então a janela de congestionamento é reduzida apenas pela metade do seu valor corrente e o remetente permanece na fase de prevenção de congestionamento, deixando de fora a fase de partida lenta. Com este procedimento, o Reno evita que a conexão fique praticamente ociosa logo após a retransmissão rápida de um segmento.

New-Reno TCP

O Reno melhora o comportamento do algoritmo Tahoe de forma significativa quando

um único pacote é perdido. Entretanto, como demonstrado por FALL et al. [20], se vários pacotes forem perdidos a performance do Reno decai ficando parecida com a do Tahoe. Para suprir esta deficiência, o algoritmo New-Reno faz uma pequena mudança no Reno que elimina a espera pelo tempo de retransmissão quando múltiplos pacotes são descartados. Esta mudança diz respeito ao comportamento do emissor durante o processo de recuperação rápida quando um *ACK parcial* é recebido reconhecendo alguns mas não todos os pacotes que estão faltando no buffer do receptor. No Reno, este ACK parcial faz com que o TCP saia do processo de recuperação rápida; ao contrário do que acontece no New-Reno onde um ACK parcial é tratado como uma indicação de que o pacote que está na seqüência foi perdido, e deve ser retransmitido. Por isso, quando vários pacotes são perdidos, o New-Reno consegue recuperá-los sem esperar pelo tempo de retransmissão, retransmitindo um pacote a cada RTT. O emissor sai da fase de recuperação rápida quando forem reconhecidos todos os pacotes que estavam pendentes no início deste processo.

SACK TCP

Como o algoritmo New-Reno, o SACK (*Selective Acknowledgements*) tem como objetivo evitar os problemas de desempenho do Reno quando múltiplos pacotes são descartados. Porém, como analisado por FALL et al. [20], o New-Reno não tem um desempenho tão bom como o SACK quando um enorme número de pacotes são perdidos. O algoritmo SACK incorpora os benefícios dos reconhecimentos seletivos estudados na subseção 2.3.1 mantendo as propriedades do Reno e do New-Reno, sendo robusto na presença de pacotes fora de ordem e utilizando a retransmissão de pacotes devido ao estouro do tempo de retransmissão, como último recurso de recuperação de dados.

Durante a recuperação rápida o SACK mantém uma variável chamada *pipe* a qual representa o número estimado de segmentos pendentes na conexão. (Este método é diferente do implementado no Reno.) Quando o valor desta variável é menor do que o tamanho da janela, então, o emissor envia novos pacotes ou os retransmite. Neste caso, quando o emissor envia novos ou velhos pacotes, a variável *pipe* é incrementada em um pacote. Contudo, quando um ACK com a opção SACK em seu cabeçalho é recebido informando que foi recebido algum pacote fora de ordem, a variável *pipe* é decrementada em um pacote.¹² Como no New-Reno, o emissor sai da fase de recuperação rápida quando forem reconhecidos todos os pacotes que estavam pendentes no início deste processo.

O uso da variável *pipe* desacopla a decisão de *qual* pacote dever ser enviado da decisão de *quando* enviar um pacote, permitindo que mais de um pacote perdido possa ser retrans-

¹²Em algumas implementações a variável *pipe* pode ser decrementada em mais de um pacote quando o emissor recebe um ACK informando que o receptor recebeu mais de um pacote fora de ordem.

mitido a cada RTT e garantindo que o TCP se recupere rapidamente de eventuais perdas. Na ausência do SACK, os emissores da conexão precisam escolher entre as seguintes estratégias para recuperar dados: (1) retransmitir um pacote perdido por RTT ou (2) retransmitir pacotes que poderiam ter sido entregues com sucesso. O Reno e New-Reno usam a primeira estratégia enquanto o Tahoe usa a segunda.

Um dos inconvenientes do SACK é que a implementação do mecanismo de reconhecimento seletivo nos sistemas finais não é uma tarefa muito fácil além de aumentar o tempo de processamento.

Vegas TCP

O algoritmo Vegas, proposto por BRAKMO et al. [10, 11], também é um algoritmo que busca melhorar o desempenho do Reno e do Tahoe. Enquanto estes reagem ao congestionamento (ou seja, aos estouros dos buffers dos roteadores), o Vegas tenta evitá-lo, alcançando entre 37% e 71% melhor vazão do que o Reno, além de proporcionar melhor eficiência na utilização da largura de banda disponível e diminuição na perdas de dados [11]. Uma comparação mais detalhada entre os algoritmos Reno e Vegas pode ser encontrada em [2, 49]. Para isso, são implementadas consideráveis mudanças no mecanismo de retransmissão rápida e nas fases de partida lenta e prevenção de congestionamento.

O Vegas estende o mecanismo de retransmissão rápida do Reno. Sempre que um segmento é enviado, o Vegas lê o seu relógio interno e armazena o instante de tempo. Quando o ACK referente ao segmento é recebido, ele novamente lê o relógio e calcula o tempo estimado do RTT usando o valor atual e o valor armazenado anteriormente (*timestamp*). Este valor estimado do RTT é usado para decidir pela retransmissão de segmentos nas seguintes situações, as quais são mostradas na figura 2.13:

- No caso de um ACK duplicado chegar, o Vegas checa se a diferença entre o tempo atual e o tempo armazenado é maior que o valor do tempo de retransmissão; se for, o segmento em questão é retransmitido sem esperar pelo terceiro ACK duplicado ou pelo vencimento do tempo de retransmissão.
- Quando um ACK não duplicado é recebido, se ele for o primeiro ou o segundo após uma retransmissão, o Vegas novamente checa para ver se o intervalo de tempo desde que ele foi enviado é maior que o tempo de retransmissão. Se for, o segmento é retransmitido. Desta maneira é possível detectar múltiplos pacotes descartados antes da retransmissão sem precisar esperar por um ACK duplicado.

O comportamento do algoritmo Vegas durante a fase de prevenção de congestionamento é diferente de todas as outras variações do controle de congestionamento do TCP. Ao invés de

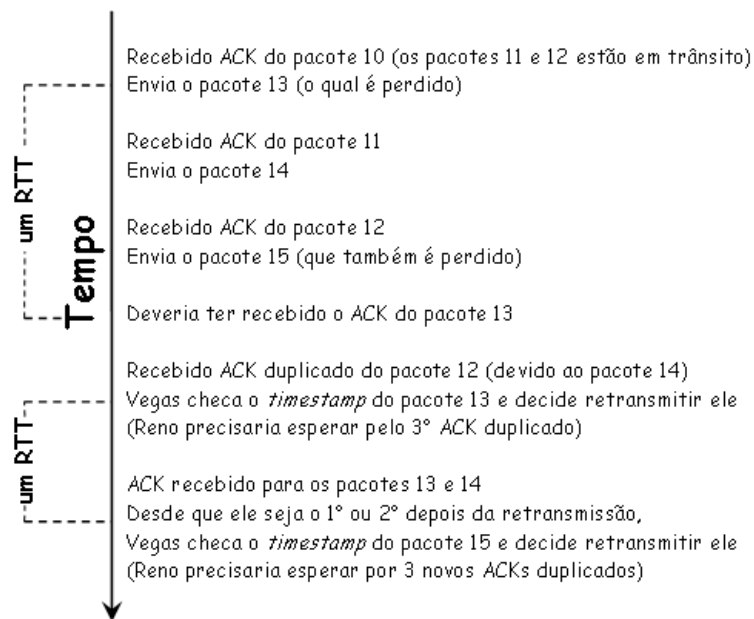


Figura 2.13: Exemplo do mecanismo de retransmissão Vegas [10].

usar a perda de pacotes como sinal de congestionamento, o Vegas detecta o congestionamento com base no decremento da taxa de envio de pacotes em relação a taxa esperada, o qual é devido ao aumento das filas nos roteadores. Para fazer a detecção do congestionamento é usado o esquema de controle de congestionamento Tri-S, proposto por Crowcroft e Wang, cujo detalhes podem ser encontrados em [63].

Sempre que a taxa de vazão calculada é menor que a taxa de vazão esperada, o Vegas aumenta as transmissões para utilizar o máximo da largura de banda. E, sempre que a taxa de vazão calculada se aproxima da taxa esperada, as transmissões são diminuídas prevenindo a saturação dos buffers dos roteadores.

Quando a conexão inicia, o TCP não tem idéia da largura de banda que está disponível e, por isso, é possível que na fase de partida lenta ele possa induzir um congestionamento na rede. Com a finalidade de prevenir que isto aconteça, o Vegas incorpora o mecanismo de detecção de congestionamento também nesta fase, fazendo o aumento exponencial do envio de dados de forma intercalada entre um e outro RTT; ao contrário do Reno, onde o aumento é feito a cada RTT. Durante o RTT em que não ocorre o aumento exponencial, o tamanho da janela de congestionamento permanece fixa e uma comparação entre a vazão calculada e a esperada pode ser feita.

Capítulo 3

GERENCIAMENTO DE FILA PARA O PROTOCOLO TCP

Na subsecção 2.3.5 estudou-se o controle de congestionamento do TCP, o qual utiliza o método fim a fim. Como foi discutido, este método deve ser necessariamente implementado pelo TCP nos hospedeiros, uma vez que a camada de rede não fornece nenhum suporte sobre o comportamento do congestionamento; sendo que a sua detecção é feita através da análise de eventos como: perda pacotes, retransmissão de pacotes e atrasos na transmissão de pacotes. Entretanto, esta maneira de detecção do congestionamento através da análise do comportamento da rede, muitas vezes leva ao descarte prematuro dos pacotes além de subutilizar a capacidade dos enlaces.

Para melhorar a troca de informações entre hospedeiros e núcleo da rede, o método de controle de congestionamento assistido pela rede é implementado. Neste método, também conhecido como gerenciamento ativo de fila (AQM), a camada de rede (isto é, os roteadores) fornecem realimentação específica de informação ao remetente a respeito do estado de congestionamento da rede. No esquema AQM a comunicação de congestionamento dos roteadores para os emissores da conexão TCP é feita de duas formas: pelo descarte de pacotes; ou pela marcação de pacotes.

Na primeira forma, que ainda não é uma comunicação explícita por parte da rede, o emissor TCP detecta a perda de pacotes no roteador ou pelo recebimento de três reconhecimentos duplicados (ACKs) ou pelo estouro do tempo de retransmissão, e responde a perda de pacotes reduzindo a sua janela de transmissão. Já no sistema de marcação de pacotes, o roteador detecta um princípio de congestionamento através, por exemplo, do cálculo do tamanho médio da fila e envia uma notificação explícita de congestionamento ao emissor que, como anteriormente, reduz a sua janela de transmissão. Neste trabalho, o processo de

marcação de pacotes será feito usando-se o **ECN** (*Explicit Congestion Notification* — Notificação Explícita de Congestionamento), assim, quando o valor do tamanho médio da fila ultrapassar um determinado patamar, os roteadores enviam um bit **ECN** nos cabeçalhos dos pacotes informando os emissores TCP sobre o congestionamento [57]. Esta abordagem é proposta e discutida por FLOYD [25] para redes TCP/IP. Mais especificamente, ela introduz o ECN no controle RED (*Random Early Detection*) mostrando as vantagens deste método em relação a outros esquemas AQM (*Active Queue Management* — Gerenciamento Ativo de Fila) sem ECN.

Neste capítulo, na seção 3.1 introduz-se os conceitos de gerenciamento de fila, gerenciamento ativo de fila, equidade na largura de banda e, em seguida, é feita uma análise detalhada do funcionamento do controle RED que é o exemplo mais difundido de um AQM. Nas seção 3.2 o protocolo TCP é modelado através de Equações Diferenciais Estocásticas (EDEs) as quais são transformadas em Equações Diferenciais Ordinárias (ODEs) para que se efetue a linearização do modelo na seção 3.3.

3.1 GERENCIAMENTO DE FILA

A mais tradicional técnica de gerenciar o tamanho da fila nos roteadores, chamada de Drop-Tail, consiste em determinar um tamanho máximo (em pacotes) para cada fila, aceitar pacotes dos vários emissores até que este valor máximo seja alcançado, e então rejeitar novos pacotes até que o tamanho da fila diminua devido ao envio de algum pacote que estava esperando na fila. De forma mais resumida, quando a capacidade da fila do roteador está esgotada, os pacotes que chegam ao roteador são imediatamente descartados. Este método possui dois importantes inconvenientes:

1. Em algumas situações o Drop-Tail permite que uma única conexão ou poucos fluxos monopolizem o espaço disponível na fila, privando outras conexões de usar o roteador. Este fenômeno é chamado de *lock-out*.
2. O mecanismo do Drop-Tail permite que os roteadores mantenham o espaço de suas filas totalmente (ou quase totalmente) preenchido durante longos períodos de tempo, enviando um sinal de congestionamento (através da perda de pacotes) somente quando a fila torna-se cheia.

Uma primeira análise do comportamento do Drop-Tail, levaria a conclusão de que mantendo a fila com um status “não-cheio”, estaria se dando mais importância a um baixo atraso fim-a-fim sofrido pelos pacotes do que para uma alta vazão de dados.

Entretanto, esta análise não leva em consideração o papel crítico que as rajadas de pacotes¹ possuem na Internet. Segundo LELAND [40], embora o TCP restrinja o fluxo de dados pelo controle do tamanho da janela do emissor, os pacotes frequentemente chegam na forma de rajadas nos roteadores. Por conseguinte, se a fila do roteador está cheia ou quase cheia, a chegada de uma rajada de pacotes causará o descarte de vários pacotes ao mesmo tempo. Resultando em uma sincronização global entre os fluxos dos emissores, seguido de uma grande oscilação do tamanho das filas nos roteadores e um período de baixa utilização do enlace e uma redução total da vazão.

Dessa forma, o grande interesse do processo de enfileiramento na Internet é absorver as rajadas de dados e as retransmitir durante o chamado período de silêncio entre uma e outra rajada de dados. Em resumo, o protocolo TCP detecta o congestionamento após um pacote ter sido descartado. Entretanto, a utilização de filas muito grandes aumenta o atraso médio na rede. Por isso, é importante ter mecanismos os quais mantenham uma vazão de informação elevada com tamanhos médios de filas baixos [27].

Ambos fenômenos, sincronização e oscilação, podem ser observados claramente nos gráficos da figura 3.1.² Na figura 3.1(b), destaca-se o alto nível de sincronização entre as janelas; situação que faz todos os emissores perderem pacotes ao mesmo tempo. Além disso, na figura 3.1(a), tem-se as grandes flutuações do tamanho da fila que correspondem exatamente às flutuações das janelas.

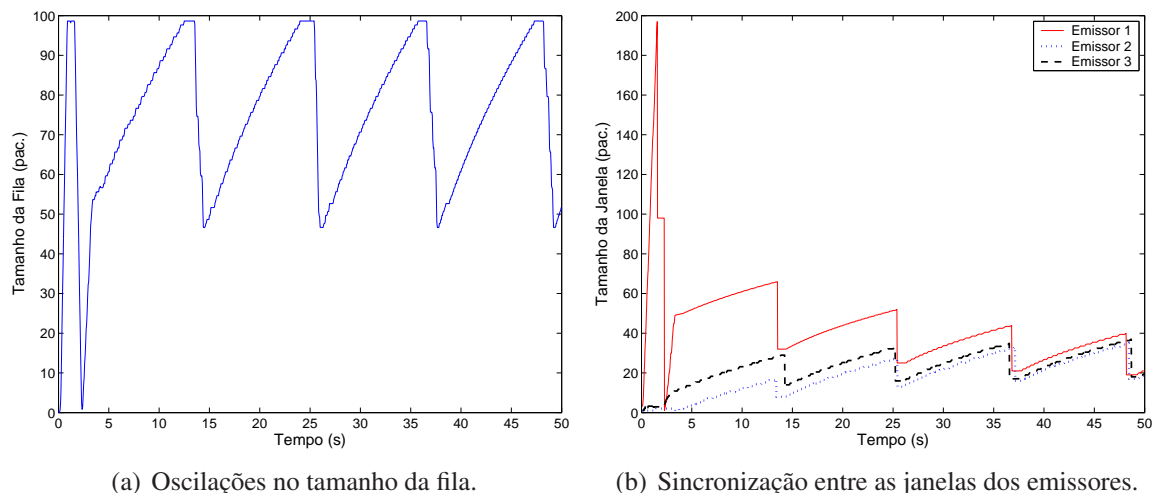


Figura 3.1: Inconvenientes provocados pelo gerenciamento de fila com Drop-Tail.

¹O termo rajada de pacotes é usado como opção de tradução do termo inglês *packet bursts*.

²Para obter esta figura, no simulador NS-2, foi considerado um cenário com três emissores, um roteador e um destino. Assumiu-se, por simplificação, que as janelas dos emissores reagem instantaneamente quando há descarte de pacotes.

Duas técnicas podem ser aplicadas para resolver o problema de *lock-out* do Drop-Tail quando a fila está cheia: o descarte casual de pacotes (*random drop*) e o descarte do primeiro pacote da fila (*drop front*). Usando o descarte casual de pacotes, o roteador descarta um pacote selecionado ao acaso da fila quando a fila está cheia e um novo pacote chega. Já com a técnica de descarte do primeiro pacote, o roteador descarta o primeiro (ou o mais antigo) pacote da fila quando um novo pacote chega e não existe espaço no buffer.

3.1.1 Gerenciamento Ativo de Fila

Apesar das técnicas apresentadas anteriormente resolverem o problema de lock-out, elas não melhoram em nada o fenômeno da sincronização de fluxos que ocorre no Drop-Tail quando as filas dos roteadores estão totalmente ocupadas. Isto acontece porque o descarte de pacotes quando a fila está no limite de sua capacidade é um recurso pouco eficiente para o controle de congestionamento da rede. A solução, então, seria descartar pacotes antes que a fila fique próxima de sua capacidade máxima, permitindo que os sistemas finais possam regular o envio de dados antes que ocorra o estouro dos buffers dos roteadores. Este método é chamado de gerenciamento ativo de fila (AQM).

O AQM é um processo de sinalização que o núcleo da rede fornece aos emissores TCPs com o objetivo de prover uma eficiente utilização da capacidade dos enlaces e do tamanho da fila nos roteadores, e uma diminuição dos atrasos sofridos pelos pacotes. O exemplo clássico e mais difundido de um AQM é o controle RED (*Random Early Detection*) [27], porém, recentemente técnicas de controle clássicas como os controles Proporcional (P) e Proporcional-Integral (PI) [47, 46] além de metodologias de controle ótimo como \mathcal{H}^∞ [42], vem sendo aplicadas como alternativa para melhorar a performance obtida com o RED. As seguintes vantagens são obtidas pelo AQM no tratamento dos fluxos de dados nos roteadores:

1. Mantendo o tamanho médio da fila pequeno, o AQM provê uma maior capacidade de absorver naturalmente a ocorrência de rajadas de pacotes sem descartar pacotes, ou seja, diminui a perda de dados na rede. Além do mais, o TCP tem maior dificuldade de se recuperar de uma rajada de pacotes descartados do que de um único pacote descartado como ocorre no AQM.
2. Com o tamanho médio da fila pequeno, o atraso que os pacotes sofrem ao longo do caminho são menores. E isto é de fundamental importância para aplicações que são sensíveis ao tempo como sessões de áudio e vídeo, e videoconferências.
3. O AQM previne o comportamento de lock-out assegurando que quase sempre haverá espaço disponível no buffer para pacotes novos. É importante prevenir o lock-out

porque ele causa uma grande injustiça entre vários tipos de fluxos. Por exemplo, em um roteador que não usa o mecanismo AQM, dois fluxos podem receber largura de bandas muito diferentes simplesmente porque eles possuem diferentes RTTs [22], e um fluxo que não usa controle de congestionamento pode receber mais largura de banda do que um fluxo que o faz. É o caso de aplicações que usam o protocolo de transporte UDP. Essas aplicações não querem que sua taxa de transmissão seja limitada, mesmo que a rede esteja congestionada; preferindo enviar seus dados a uma taxa constante e, ocasionalmente, perder pacotes a reduzir as suas taxas a níveis “justos” quando da eminência ou ocorrência de um congestionamento.

Desta forma, quando fluxos TCP e UDP dividem o mesmo enlace, na ocorrência de um congestionamento o TCP irá reduzir a sua taxa de envio, deixando mais largura de banda disponível para o “não cooperativo” fluxo UDP. Nesta direção, FLOYD et al. [26] prevêm que um importante desafio será o desenvolvimento de mecanismos de controle de congestionamento para a Internet que evitem que o tráfego de UDP leve a vazão da Internet a níveis muito baixos, por efeito cumulativo.

Para o mesmo cenário da figura 3.1, porém, agora, com o roteador executando o RED, chega-se aos gráficos da figura 3.2. Ao contrário do Drop-Tail, o RED garante que as janelas não fiquem sincronizadas e que o tamanho médio do tamanho da fila seja bem menor; cerca de 10 pacotes, ao invés de 75 pacotes no caso do Drop-Tail.

3.1.2 Eqüidade na Largura de Banda

Os mecanismos de controle de congestionamento implementados pelo protocolo TCP cada vez mais se consolidam como um dos principais fatores para o bom desempenho da Internet. Por causa dos “recuos” do TCP durante um congestionamento, um grande número de conexões TCP podem ser compartilhadas em um único e congestionado enlace, no qual a largura de banda é distribuída razoavelmente equitativamente entre os fluxos. CHIU et al. [14] mostram de forma intuitiva e precisa por que o controle de congestionamento consegue compartilhar a largura de banda de uma maneira justa entre as várias conexões TCP. Entretanto, uma melhor eqüidade no compartilhamento da largura de banda entre as conexões concorrentes só será alcançada se a maioria dos fluxos executarem basicamente o mesmo algoritmo de prevenção de congestionamento [9].

Para compreender melhor a eqüidade na distribuição de largura de banda, é interessante dividir os fluxos em três classes: fluxos compatíveis com o TCP, fluxos não responsivos, e fluxos que são responsivos mas não compatíveis com o TCP. A classe de fluxos compatíveis, diz respeito aos fluxos que são responsivos a uma notificação de congestionamento

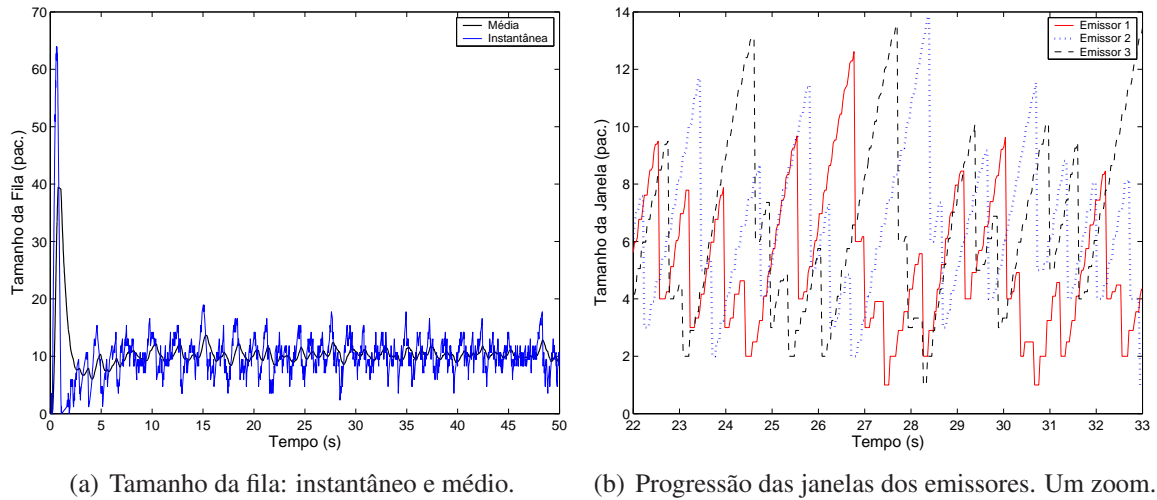


Figura 3.2: Gerenciamento de fila com RED.

(abaixando a sua vazão) e que não usam mais largura de banda do enlace do que os padrões TCP, quando a rede está operando sem congestionamento.

A segunda classe, a de fluxos não responsivos, são as aplicações baseadas no sistema de entrega de dados do protocolo UDP, nas quais algoritmos de controle de congestionamento ou são implementados de forma inadequada ou simplesmente não existem. Ou seja, o fluxo não diminui quando recebe uma notificação de congestionamento na rede. Tais aplicações, que crescem constantemente na internet, incluem aplicações multimídia como transmissão de áudio, vídeo e imagem, e também aplicações multicast como a transferência de dados em grandes volumes (por exemplo, a transferência de uma atualização de software do desenvolvedor do software para os usuários que necessitam da atualização).

Apesar de certas aplicações serem sensíveis às limitações na largura de banda (isto é, necessitam de uma “generosa” largura de banda para não perderem eficiência) é necessário ou a introdução de mecanismos de controle nos fluxos não responsivos, como propõe BOLOT et al. [7], ou que a rede seja capaz de se proteger contra este tipo de fluxo, a fim de evitar um caos no congestionamento da Internet.

Os fluxos responsivos mas não compatíveis também podem ser um problema para o bom compartilhamento do enlace. Algumas implementações que usam os padrões TCP não conseguem desenvolver os algoritmos para o mecanismo de controle de congestionamento corretamente, provocando injustiças na divisão da largura de banda. Outras, no entanto, são desenvolvidas justamente para permitir um maior uso da largura de banda. A consequência inevitável das aplicações pertencentes a esta terceira classe é o grande aumento da agressividade dos fluxos TCP com relação ao uso da largura de banda disponível o que leva a uma subutilização dos recursos da rede.

3.1.3 Descrição do Controle RED

Em contraste com o tradicional algoritmo Droptail de gerenciamento de fila o qual descarta pacotes somente quando a fila do roteador está cheia, o algoritmo RED (*Random Early Detection*) descarta os novos pacotes probabilisticamente; sendo que, a probabilidade de descarte é uma função do tamanho médio da fila. Antes de esperar o congestionamento ocorrer, o RED atua preventivamente obtendo a *média* do tamanho da fila do roteador. Se este tamanho de fila médio exceder um valor limite predefinido, o roteador *marca* cada novo pacote que chega com uma certa probabilidade. Assim, o roteador notifica cada conexão a respeito de um congestionamento ou pelo descarte de pacotes ou pelo envio de um bit no cabeçalho dos pacotes.

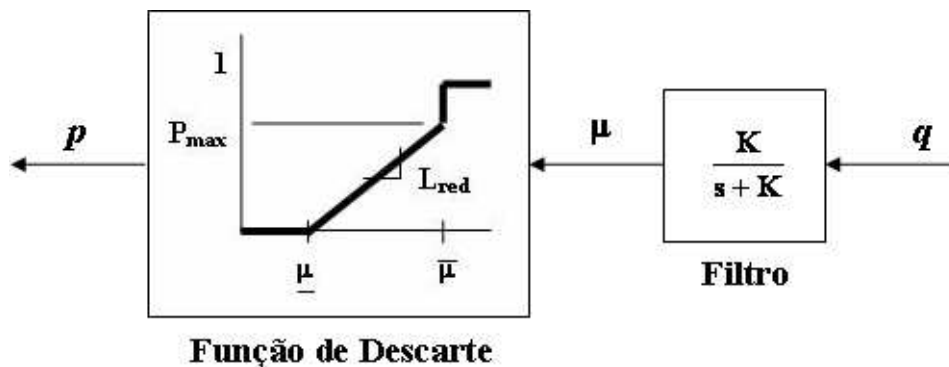


Figura 3.3: RED – Filtro passa-baixa em cascata com função de descarte. Adaptado de [47].

O algoritmo RED

Conforme a figura 3.3, o RED pode ser visto como um filtro passa-baixa e uma função de descarte de pacotes [36]. O descarte aleatório de pacotes provocado por esta função não linear tem como objetivo eliminar a sincronização dos fluxos de dados, enquanto o filtro (através do cálculo da fila média) é introduzido para atenuar os efeitos das rajadas de pacotes.

Nesse sentido, com o algoritmo RED, o roteador calcula o tamanho médio da fila μ usando o bem conhecido filtro passa-baixa EWMA (*Exponential Weighted Moving Average* — Média Móvel Exponencial Ponderada), sendo que, q é o tamanho instantâneo da fila. Os detalhes deste cálculo serão analisados na subseção 3.2.3. O tamanho médio da fila é, então, comparado com dois limites: um limite mínimo $\underline{\mu}$ e um limite máximo $\bar{\mu}$. O limite mínimo especifica o tamanho médio da fila *abaixo* do qual nenhum pacote é marcado, enquanto que o limite máximo especifica o tamanho médio da fila *acima* do qual todos os pacotes são marcados. Quando o tamanho médio da fila varia do limite mínimo ao máximo, cada pacote

será marcado com probabilidade p que varia entre 0 e p_{max} . É importante notar que o RED pode marcar um pacote descartando ele no roteador ou colocando um bit no cabeçalho do pacote.

Como o RED trabalha com a média da fila do roteador, ele não toma decisões levando em conta o comportamento instantâneo da fila. Assim, se a fila permaneceu quase sempre vazia no “passado recente”, o RED tende a não descartar pacotes (a menos que ocorra um estouro da capacidade do buffer). Por outro lado, se a fila esteve relativamente cheia, indicando congestionamento, os pacotes que chegarem ao roteador serão descartados com uma probabilidade bem maior. A decisão de descartar ou não os novos pacotes que chegam ao roteador pode ser feita no “modo pacote”, ignorando o tamanho do pacote, ou no “modo byte”, levando em consideração o tamanho de cada pacote. Para FLOYD [23] o modo byte de descarte é mais útil, porque geralmente o recurso que é escasso é a largura de banda em bytes por segundo. Por outro lado, se o recurso escasso é a capacidade de processamento da CPU (do sistema final) em pacotes por segundo, então o descarte no modo pacotes é melhor. Considerando que um dos propósitos do gerenciamento de fila do RED é dar indicações de congestionamento para os emissores *antes* que o engarrafamento fique suficientemente grande para causar estouro na capacidade do buffer, o RED deve medir o tamanho médio da fila em pacotes para uma fila cuja capacidade é em unidades de pacotes, e em bytes para uma fila cuja capacidade é em unidades de bytes. Por exemplo, se o RED calcular o tamanho médio da fila em bytes e a capacidade da fila é em unidades de pacotes, e considerando que todos os pacotes que estão na fila são pequenos, então o cálculo do RED em bytes pode não ser suficiente para distinguir entre um fila cheia (com muitos pacotes pequenos) ou uma fila pequena (com poucos pacotes grandes). Em particular, é possível ter a capacidade da fila e o tamanho médio da fila em unidades de pacotes, enquanto que ao mesmo tempo é usado o descarte de pacotes em bytes; onde o tamanho do pacote que chega ao roteador em bytes é um fator de decisão para descartar ou não o pacote em questão.

Se um roteador, que executa o algoritmo RED, marca pacotes quando o tamanho médio da fila excede o limite máximo, então, este roteador pode controlar o tamanho médio da fila mesmo na ausência de um protocolo de transporte cooperativo com o tráfego na Internet, como o TCP. Uma das vantagens de se implementar um roteador com este mecanismo de controle de congestionamento que trabalha tanto com o protocolo de transporte TCP como com o protocolo de transporte UDP, e que não exige que todos os outros roteadores da Internet implementem o mesmo mecanismo de controle de congestionamento, é que ele pode ser gradualmente implementado pela rede. Entretanto, apesar do RED trabalhar com ambos os protocolos de transporte TCP e UDP, ele é projetado para redes onde o protocolo de

transporte responda as indicações de congestionamento do núcleo da rede, como é o caso do protocolo TCP.

3.2 MODELO DINÂMICO DO PROTOCOLO TCP PARA APLICAÇÕES DE GERENCIAMENTO ATIVO DE FILA (AQM)

O modelo do Protocolo de Controle de Transmissão (TCP) é baseado no trabalho de MISRA et al. [44, 45] o qual utilizou a análise de fluxo-fluído e a teoria de Equações Diferenciais Estocásticas (EDEs) para modelar as interações entre os fluxos TCP e o gerenciamento ativo de fila (AQM) em uma rede, mostrando, inclusive, como as EDEs podem ser transformadas em um conjunto de Equações Diferenciais Ordinárias (ODEs) as quais podem ser facilmente resolvidas numericamente. Apesar do modelo proposto em [44, 45] prever os mecanismos de estouro do tempo de retransmissão de pacotes (*timeout*) e de partida lenta (*slow start*), ignora-se estes dois eventos para o modelo a ser utilizado no projeto dos controladores; sem o custo de perder a precisão na captura da dinâmica do TCP, como foi amplamente demonstrado e discutido em [42, 46, 47, 48].

No desenvolvimento do modelo é considerado um único roteador congestionado com uma capacidade de transmissão C , para o qual são enviados N fluxos TCP. Associado com esta rota está uma política de gerenciamento ativo de fila (AQM) que é caracterizada por uma função de descarte de pacotes $p(\mu)$ que tem como argumento uma *estimativa*³ do tamanho médio da fila no roteador, obtida através de um filtro passa-baixa. A política AQM que nós usaremos será o RED (Random Early Detection)[27] onde $p(\mu)$ obedece a função de descarte descrita na figura 3.3.

O tráfego de dados é modelado como um fluido e é baseado na suposição que as perdas de pacotes sofridas por um fluxo i são descritas pelo Processo de Poisson [12], com o número de perdas sofridas pelo fluxo i sendo denotado por $N_i(t)$ ⁴. O contador de Poisson conta o número de eventos que ocorrem durante um intervalo de tempo, sendo que, a taxa média na qual esses eventos ocorrem por unidade de tempo é $\lambda_i(t)$. Assim, em um modelo de perdas que utiliza o contador de Poisson, as indicações de perdas de pacotes chegam ao remetente da conexão a uma certa taxa (que pode ser interpretado como ACKs duplicados).

³A média do tamanho da fila no RED é diferente da noção de média das variáveis W e q que serão apresentadas nas subseções 3.2.1 e 3.2.2. A primeira é uma média sobre o tempo, enquanto W e q são um conjunto de médias. Para que não haja equívocos, quando a média se referir ao valor obtido pelo filtro do RED, ela será chamada de estimativa.

⁴É bom notar que t , aqui, refere-se ao ponto no tempo onde o fluxo detectou a perda, o qual é diferente de quando o descarte atual ocorre. Dessa maneira, doravante quando não existir ambigüidade, o argumento t será omitido.

3.2.1 Comportamento da Janela

Ignorando-se a fase de partida lenta do mecanismo de controle de congestionamento proposto na subseção 2.3.5, constata-se que o TCP essencialmente aumenta o tamanho de sua janela de 1 MSS para cada RTT quando o caminho pela rede não está congestionado e diminui o tamanho de sua janela de um fator de 2 para cada RTT quando o caminho está congestionado. Por essa razão, o TCP é freqüentemente chamado de algoritmo de **aumento-aditivo e diminuição-multiplicativa** (*Additive-Increase and Multiplicative-Decrease* — AIMD), e a equação diferencial estocástica (EDE) que descreve o comportamento do tamanho de sua janela $W_i(t)$ no remetente é

$$dW_i(t) = \frac{dt}{R_i(q(t))} - \frac{W_i(t)}{2} dN_i(t) \quad (3.1)$$

onde $R_i(t)$ representa o RTT (*round-trip time*) e é definido como a soma do atrasos de propagação T_p e de fila $q(t)/C$, produzindo

$$R_i(t) = \frac{q(t)}{C} + T_p, \quad i = 1, \dots, N \quad (3.2)$$

O primeiro termo da equação (3.1) corresponde a parte do aumento-aditivo, a qual mostra que o tamanho da janela aumenta de um a cada RTT. O segundo termo corresponde a parte da diminuição-multiplicativa, a qual diminui o tamanho da janela de um fator de 2 quando chega uma comunicação de perda de pacotes ($dN_i(t) = 1$). Note que não foi considerado o comportamento de fase lenta do TCP, uma vez que em implementações mais realísticas, esta fase do processo demora somente poucos RTTs e, portanto, pode ser negligenciada no processo de modelagem. Tomando a esperança matemática⁵ em ambos os lados da equação (3.1) produz-se

$$E[dW_i(t)] = E\left[\frac{dt}{R_i(q(t))}\right] - \frac{E[W_i(t)dN_i(t)]}{2} \quad (3.3)$$

$$dE[W_i] \approx E\left[\frac{dt}{R_i(q)}\right] - \frac{E[W_i]\lambda_i(t)}{2} dt \quad (3.4)$$

A equação (3.4) é aproximada porque o termo $E[W_i(t)dN_i(t)]$ foi considerado como $E[W_i(t)]E[dN_i(t)]$, fato que corresponde à independência entre $W_i(t)$ e $dN_i(t)$, o que não é

⁵Ao longo do documento vamos representar a esperança matemática de uma variável ω qualquer por $\bar{\omega}$.

verdade. Entretanto, esta aproximação não muda a característica da diminuição-multiplicativa e, portanto, captura muito bem a dinâmica do TCP. Note também, que foi assumido $E[dN(t)] = \lambda dt$; igualdade que provém das propriedades do contador de Poisson, onde o termo $\lambda(t)$ é a indicação de perda que o emissor recebe. Ela alcança o emissor aproximadamente com um atraso igual o RTT após o evento ter ocorrido, ou seja, do pacote ter sido marcado e descartado na fila.

Em um esquema AQM as perdas são distribuídas proporcionalmente à largura de banda compartilhada por uma determinada conexão no roteador. Por isso, sendo a vazão do fluxo $B(t-R)$ igual à $W(t-R)/R(q(t-R))$, a taxa de indicação de perdas recebida no instante t é $p(\mu(t-R))B(t-R)$. A esperança matemática desta expressão é $p(\bar{\mu}(t-R)E[W(t-R)/R(q(t-R))])$. Reescrevendo a equação (3.4), tem-se

$$dE[W_i] \approx \frac{dt}{R_i(\bar{q})} - \frac{\bar{W}_i}{2} p(\bar{\mu}(t-R)) \frac{\bar{W}_i(t-R)}{R_i(\bar{q}(t-R))} dt \quad (3.5)$$

$$\frac{d\bar{W}_i}{dt} = \frac{1}{R_i(\bar{q})} - \frac{\bar{W}_i \bar{W}_i(t-R)}{2R_i(\bar{q}(t-R))} p(\bar{\mu}(t-R)) \quad (3.6)$$

Observa-se que a equação (3.6) é dependente da função de descarte do AQM (neste caso representado pelo RED), do tempo de ida e volta (RTT) de cada pacote e indiretamente pelo comportamento da fila no roteador.

3.2.2 Comportamento da Fila

A equação que descreve o comportamento da fila $q(t)$ em um roteador, a qual foi analisada por CASSANDRAS [12] em seus estudos sobre a teoria do enfileiramento, é a versão diferencial da equação de Lindley, que possui a seguinte forma:

$$\frac{dq(t)}{dt} = -1_{q(t)}C + \sum_{i=1}^N \frac{W_i}{R_i(q)} \quad (3.7)$$

onde a variável $-1_{q(t)}$ é nula se $q(t) = 0$ e unitária se $q(t) > 0$. O primeiro termo modela a diminuição da fila, quando ela é maior do que zero, devido ao processamento dos pacotes que chegam ao roteador. O segundo termo representa o aumento no tamanho da fila devido a chegada de pacotes dos N fluxos do TCP. Tomando a esperança matemática obtém-se:

$$\frac{d\bar{q}(t)}{dt} = E[-1_{q(t)}]C + \sum_{i=1}^N E \left[\frac{W_i}{R_i(q)} \right] \quad (3.8)$$

$$\frac{d\bar{q}(t)}{dt} \approx E[-1_{q(t)}]C + \sum_{i=1}^N \frac{\bar{W}_i}{R_i(\bar{q})}. \quad (3.9)$$

Admitindo que para uma fila congestionada (engarrafada) a probabilidade de ter-se $q(t) > 0$ seja grande para $\forall t$, pode-se aproximar $E[1_{q(t)}] \approx 1$ e a equação (3.9) toma a seguinte forma

$$\frac{d\bar{q}(t)}{dt} \approx -C + \sum_{i=1}^N \frac{\bar{W}_i}{R_i(\bar{q})}. \quad (3.10)$$

Fica evidente, na equação (3.10), que o emissor numa conexão TCP influencia diretamente o comportamento da fila no roteador através de sua vazão $W/R(q)$, além, é claro, da própria capacidade C do roteador.

3.2.3 Estimativa do Tamanho Médio da Fila

O filtro passa-baixa média móvel exponencial ponderada (EWMA), o qual estima o tamanho médio da fila no roteador, está representado em sua forma discreta na equação (3.11). Como pode ser analisado, o EWMA calcula a fila média tirando amostras a cada ζ segundos e usando uma ponderação α , $0 < \alpha < 1$. Por conseguinte, o EWMA se constitui da soma ponderada do valor atual do tamanho da fila $q(k)$ e do valor médio da fila μ que foi obtido na amostra anterior, produzindo

$$\mu((k+1)\zeta) = (1-\alpha)\mu(k\zeta) + (\alpha)q(k\zeta) \quad (3.11)$$

Quando $\alpha = 1$ nenhum sinal é filtrado, ou seja, ao invés de se calcular o tamanho médio da fila, obtém-se apenas o valor instantâneo dela. Já quando $\alpha = 0$ a ponderação é totalmente ignorada. Para converter a equação (3.11) em uma equação diferencial, é natural considerar-se que

$$\frac{d\mu}{dt} = A\mu(t) + Bq(t) \quad (3.12)$$

sendo que, em um sistema discreto [1], $\mu(t_{k+1})$ é dado por

$$\mu(t_{k+1}) = e^{A(t_{k+1}-t_k)}\mu(t_k) - \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bd\tau q(t_k). \quad (3.13)$$

Veja que a equação diferencial segue exatamente os pontos de amostragem do sistema de tempo discreto, não havendo um acúmulo de erro conforme k aumenta. Comparando os

coeficientes das equações (3.13) e (3.11), obtém-se

$$1 - \alpha = e^{A\zeta}$$

ou

$$A = \frac{\ln(1 - \alpha)}{\zeta} = -B .$$

Substituindo os coeficientes na equação (3.12), obtém-se a seguinte equação que descreve o comportamento de μ ,

$$\frac{d\mu}{dt} = \frac{\ln(1 - \alpha)}{\zeta} \mu(t) - \frac{\ln(1 - \alpha)}{\zeta} q(t)$$

Tomando a esperança matemática em ambos os lados, produz-se

$$\frac{d\bar{\mu}}{dt} = \frac{\ln(1 - \alpha)}{\zeta} \bar{\mu}(t) - \frac{\ln(1 - \alpha)}{\zeta} \bar{q}(t) . \quad (3.14)$$

Com esta última equação, são três as equações acopladas, (3.6), (3.10) e (3.14), e três variáveis desconhecidas ($\bar{\mu}$, \bar{q} , \bar{W}) as quais podem ser resolvidas numericamente. Através da solução destas equações é possível conseguir o tamanho médio da fila, a estimativa do tamanho médio da fila e a evolução do tamanho médio da janela diretamente, além do tempo de ida e volta (RTT).

Considerando agora, $\ln(1 - \alpha)/\zeta = -K$ (para $\ln(1 - \alpha)$ negativo), pode-se reescrever a equação (3.14), obtendo

$$\frac{d\mu}{dt} = -K\mu(t) + Kq(t) . \quad (3.15)$$

Aplicando a transformada de Laplace em ambos os lados da equação, consegue-se a função de transferência do filtro passa-baixa do RED, a qual é descrita da seguinte forma

$$\frac{\mu}{q} = \frac{K}{s + K} . \quad (3.16)$$

Esta função de transferência é exatamente o filtro passa-baixa de primeira ordem que foi mostrado na figura 3.3, o qual tem como objetivo calcular o tamanho médio da fila (nas baixas frequências), e filtrar as rajadas de pacotes (para sinais de alta frequência). O sinal de entrada deste filtro é o tamanho instantâneo da fila q , e o sinal de saída é o valor estimado médio da fila μ . Ele permite que sinais de frequência menores que K passem

através dele, enquanto que sinais de entrada como frequências maiores que K são cortados. Quanto maior o valor de K , mais rápido o filtro irá responder a mudanças súbitas da fila. Se um valor alto de K for mantido, então a função AQM começa a calcular um valor quase instantâneo do tamanho da fila resultando em oscilações. As oscilações na fila do roteador não são desejáveis em uma rede porque elas podem gerar filas de grande tamanho, o que é inaceitável, e conseqüentemente adicionar consideráveis alterações no atraso (*jitter*) sofrido pelos fluxos que passam pelo roteador [44].

3.3 LINEARIZAÇÃO

As dinâmicas do tamanho da janela e do tamanho da fila do protocolo TCP, modeladas na seção 3.2, serão linearizadas sobre um ponto de operação com o propósito de projetar os controladores de malha fechada. Como já mencionado, o modelo não trata os mecanismos de *timeout* e de *slow start*, o que não afeta a captura precisa do comportamento qualitativo do tráfego de fluxos TCP. Este modelo relata o valor médio das variáveis chaves de uma rede e é descrito pelas seguintes equações diferenciais ordinárias (ODEs)⁶ não lineares, as quais são redefinições das equações (3.6) e (3.10):

$$\begin{aligned}\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R((t-R(t)))}p(t-R(t)) \\ \dot{q}(t) &= -C + \frac{W(t)}{R(t)}N(t)\end{aligned}\tag{3.17}$$

onde

- W = tamanho médio do tamanho da janela do TCP (pacotes);
- q = tamanho médio da fila (pacotes);
- $R(t)$ = tempo de viagem de ida e volta (*round-trip time*), $R = \frac{q}{C} + T_p$ (segundos);
- C = capacidade do enlace (pacotes/segundo);
- T_p = atraso de propagação (segundos);
- N = fator de carga (número de sessões TCP);
- p = probabilidade de pacotes marcados.

⁶Para uma variável x qualquer usaremos a notação $\frac{dx}{dt} = \dot{x}$.

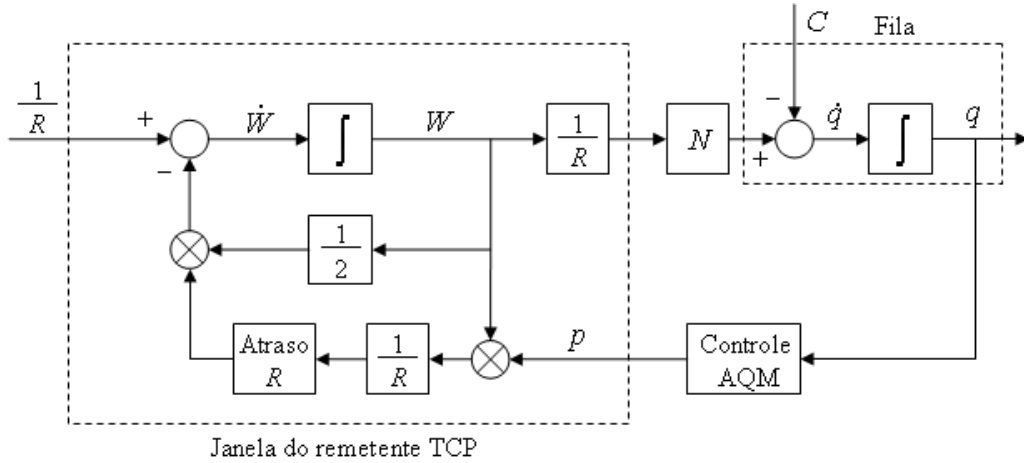


Figura 3.4: Diagrama de blocos da dinâmica do TCP. Adaptado de [47].

O tamanho da fila q e o tamanho da janela W são quantias positivas e limitadas, isto é, $q \in [0, \hat{q}]$ e $W \in [0, \hat{W}]$ onde \hat{q} é a capacidade máxima do buffer do roteador e \hat{W} é o tamanho máximo da janela. De forma semelhante, a probabilidade de pacotes marcados p é limitada ao conjunto $[0, 1]$. A figura 3.4 mostra o diagrama de blocos referente as equações diferenciais não lineares descritas na equação (3.17), destacando a dinâmica da janela do remetente, a fila do roteador e o controle por realimentação AQM. Considerando que, na equação (3.17), o número de sessões TCP e a capacidade do link são constantes, ou seja, $N(t) \equiv N$ e $C(t) \equiv C$; e assumindo (W, q) como os estados e p como a entrada, o ponto de operação do processo (W_0, q_0, p_0) é definido quando se faz $\dot{W} = 0$ e $\dot{q} = 0$, de modo que

$$\dot{W} = 0 \Rightarrow W_0^2 p_0 = 2 \quad (3.18)$$

$$\dot{q} = 0 \Rightarrow W_0 = \frac{R_0 C}{N}; \quad R_0 = \frac{q_0}{C} + T_p .$$

Na continuação da linearização será desprezado o efeito de variações do RTT nas variáveis $W(t - R(t))$, $q(t - R(t))$ e $p(t - R(t))$, isto é

$$\begin{aligned} W(t - R(t)) &= W(t - R_0) \\ q(t - R(t)) &= q(t - R_0) \quad , \\ p(t - R(t)) &= p(t - R_0) \end{aligned}$$

onde R_0 é o RTT no equilíbrio desejado. Esta hipótese é razoável quando as oscilações das variáveis são pequenas num intervalo de tempo de 1 (um) RTT. Entretanto, a dependência

do RTT sobre o tamanho da fila a cada instante é mantida, o que produz :

$$\begin{aligned}\dot{W}(t) &= \frac{1}{\frac{q(t)}{C} + T_p} - \frac{W(t)W(t-R_0)}{2\left(\frac{q(t-R_0)}{C} + T_p\right)} p(t-R_0) \\ \dot{q}(t) &= -C + \frac{W(t)}{R(t)} N(t)\end{aligned}\tag{3.19}$$

3.3.1 Derivadas Parciais

Conforme o método de linearização apresentado por CHEN [13], o sistema não linear da equação (3.19) pode ser aproximado por um sistema linear através do uso da técnica de derivadas parciais, também conhecidas como Jacobianos. Nesse sentido, antes de efetuar o cálculo dos Jacobianos é conveniente reescrever (3.19), definindo as funções não lineares f e h , ou seja

$$\begin{aligned}f(W, W_R, q_R, q, p_R) &= \frac{1}{\frac{q}{C} + T_p} - \frac{WW_R}{2\left(\frac{q_R}{C} + T_p\right)} p_R \\ h(W, q) &= \frac{N}{\frac{q}{C} + T_p} W - C\end{aligned}\tag{3.20}$$

onde $W_R = W(t - R_0)$, $q_R = q(t - R_0)$ e $p_R = p(t - R_0)$.

Calculando as derivadas parciais (Jacobianos) de f e h no ponto de operação (W_0, q_0, p_0) produz-se:

Para a função f :

$$\begin{aligned}\frac{\partial f}{\partial W} = \frac{\partial f}{\partial W_R} &= -\frac{W_0}{2R_0} p_0 \\ &= -\frac{W_0}{2R_0} \frac{2}{W_0^2} \\ &= -\frac{1}{R_0 W_0} \\ &= -\frac{N}{R_0^2 C} \\ \frac{\partial f}{\partial q} &= \frac{\partial}{\partial q} \left(\frac{1}{\frac{q}{C} + T_p} - \frac{WW_R}{2\left(\frac{q_R}{C} + T_p\right)} p_R \right) \\ &= -\frac{1}{R_0^2 C}\end{aligned}$$

$$\begin{aligned}
\frac{\partial f}{\partial q_R} &= \frac{\partial}{\partial q} \left(\frac{1}{\frac{q}{C} + T_p} - \frac{WW_R}{2\left(\frac{q_R}{C} + T_p\right)P_R} \right) \\
&= \frac{W_0^2 P_0}{2R_0^2 C} \\
&= \frac{1}{R_0^2 C}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f}{\partial p_R} &= -\frac{W_0^2}{2R_0} \\
&= -\frac{\frac{R_0^2 C^2}{N^2}}{2R_0} \\
&= -\frac{R_0 C^2}{2N^2}
\end{aligned}$$

Para a função h :

$$\begin{aligned}
\frac{\partial h}{\partial q} &= \frac{\partial}{\partial q} \frac{NW}{\left(\frac{q}{C} + T_p\right)} \\
&= -\frac{NW_0}{C\left(\frac{q_0}{C} + T_p\right)^2} \\
&= -\frac{1}{R_0}
\end{aligned}$$

$$\frac{\partial h}{\partial W} = \frac{N}{R_0}$$

Agrupando as derivadas parciais levando em consideração as respectivas variáveis de estado (W, q), o sistema linearizado da equação (3.19) é representado pelas seguintes equações lineares

$$\begin{aligned}
\delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t - R_0)) - \frac{1}{R_0^2 C}(\delta q(t) - \delta q(t - R_0)) \\
&\quad - \frac{R_0 C^2}{2N^2} \delta p(t - R_0)
\end{aligned}$$

$$\delta\dot{q}(t) = \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \tag{3.21}$$

onde

$$\delta W(t) = W - W_0;$$

$$\delta q(t) = q - q_0;$$

$$\delta p(t) = p - p_0.$$

representam a perturbação das variáveis no ponto de operação.

3.3.2 Linearização para Controle \mathcal{H}^∞

Como o controle \mathcal{H}^∞ tem como objetivo minimizar a influência do maior ganho que o sistema em malha fechada oferece a um sinal de perturbação qualquer [8, 62], deve-se linearizar o sistema a ser controlado considerando-se uma variável de perturbação. Dessa forma, especialmente para projetar o controle \mathcal{H}^∞ que é objeto deste trabalho, a variação de R em torno do seu valor nominal R_0 será a variável de perturbação do sistema a qual desejamos minimizar a sua influência sobre o comportamento do tamanho da fila $q(t)$, no roteador; como mostra a figura 3.5. Para isso, além das derivadas parciais feitas anteriormente será necessário derivar a equação (3.17) em relação a R e $R(t-R(t))$. Calculando os jacobianos de f e h para esta situação, obtém-se:

Para a função f :

$$\frac{\partial f}{\partial R} = -\frac{1}{R_0^2}$$

$$\begin{aligned} \frac{\partial f}{\partial R_R} &= \frac{W_0^2}{2} \frac{1}{R_0^2} p_0 \\ &= \frac{1}{R_0^2} \end{aligned}$$

Para a função h :

$$\begin{aligned} \frac{\partial h}{\partial R} &= -\frac{NW_0}{R_0^2} \\ &= -\frac{N}{R_0^2} \frac{R_0 C}{N} = -\frac{C}{R_0} \end{aligned}$$

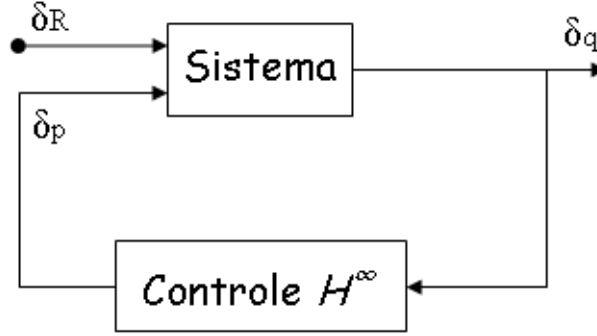


Figura 3.5: Diagrama destacando a variável de perturbação δR .

Incorporando estas derivadas parciais na equação (3.21) produz-se as seguintes equações lineares

$$\begin{aligned}\delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t - R_0)) - \frac{1}{R_0^2 C}(\delta q(t) + \delta q(t - R_0)) \\ &\quad - \frac{R_0^2 C^2}{2N^2} \delta p(t - R_0) - \frac{1}{R_0^2}(\delta R(t) - \delta R(t - R_0)) \\ \delta\dot{q}(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) - \frac{C}{R_0} \delta R(t)\end{aligned}\quad (3.22)$$

Considerando que o RTT varia pouco num intervalo de tempo dado por R_0 , se tem $\delta R(t) \cong \delta R(t - R_0)$. Conseqüentemente, redefinindo (3.22) obtém-se a equação linear (3.23) que será usada para projetar o controle \mathcal{H}^∞ através do uso de LMIs na seção 4.4.

$$\begin{aligned}\delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t - R_0)) - \frac{1}{R_0^2 C}(\delta q(t) + \delta q(t - R_0)) - \frac{R_0^2 C^2}{2N^2} \delta p(t - R_0) \\ \delta\dot{q}(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) - \frac{C}{R_0} \delta R(t)\end{aligned}\quad (3.23)$$

Ao contrário da equação linear (3.21), na equação (3.23) o tamanho da fila $q(t)$ do roteador é influenciada pela variável de perturbação $\delta R(t)$. Assim, o controle \mathcal{H}^∞ pode ser aplicado para configurar os parâmetros de desempenho dos controles RED e PI. Aliás, foi justamente para melhor determinar estes parâmetros que a variação do atraso R foi escolhida como variável de perturbação, uma vez que, em uma rede real o RTT é fator fundamental para se tenha um bom fluxo de dados.

3.4 COMENTÁRIOS

Neste capítulo, foram examinados os vários aspectos do gerenciamento de fila e do gerenciamento ativo de fila (AQM). Usando o simulador de redes NS-2, foi feita uma pequena comparação entre: o Drop-Tail, que é a técnica mais usada no gerenciamento de fila sem a sinalização do núcleo da rede; e o RED, que é a tradicional técnica de gerenciamento ativo de fila (AQM). Através dos gráficos obtidos pelas simulações pode-se ter uma precisa noção de como o AQM melhora sensivelmente a relação entre os fluxos dos vários emissores TCPs e o tamanho da fila no roteador. Comentou-se, também, o quão importante são os mecanismos de controle de congestionamento na garantia de uma melhor distribuição da largura de banda disponível. Em seguida, o modelo dinâmico do protocolo TCP para aplicações AQM foi desenvolvido destacando: o comportamento da janela dos emissores TCP; o comportamento da fila nos roteadores; e a estimativa do tamanho médio da fila a qual é feita pelo filtro passa-baixa do RED. Neste modelo, todos os fluxos TCP foram considerados iguais, isto é, eles possuem o mesmo tamanho e duração. Fato que não é muito realista, pois, por exemplo, na internet tem-se entre outros, fluxos FTP, HTTP e SMTP cada qual com características de tempo e tamanho bem diferentes. Entretanto, como será visto nos próximos capítulos, isso não será empecilho no projeto dos controladores. Por fim, na última parte deste capítulo, foi efetuada duas linearizações do modelo do protocolo TCP: uma para o projeto do AQM através dos métodos de controle clássicos; e a outra para o projeto AQM com LMIs.

Capítulo 4

PROJETO DOS CONTROLADORES PARA ROTEADORES AQM

4.1 MODELO DO PROTOCOLO TCP

A representação em diagrama de blocos das equações linearizadas descritas em (3.21) é mostrada na figura 4.1 a qual destaca os comportamentos do tamanho da janela e do tamanho da fila. Manipulando este diagrama e acrescentando o gerenciamento ativo de fila, chega-se ao sistema de controle realimentado mostrado da figura 4.2 o qual foi proposto e também utilizado em [47]. A função de transferência da planta $P(s)$, para o sistema TCP, demonstra de que forma a probabilidade de descarte influencia ou afeta o tamanho da fila no roteador,

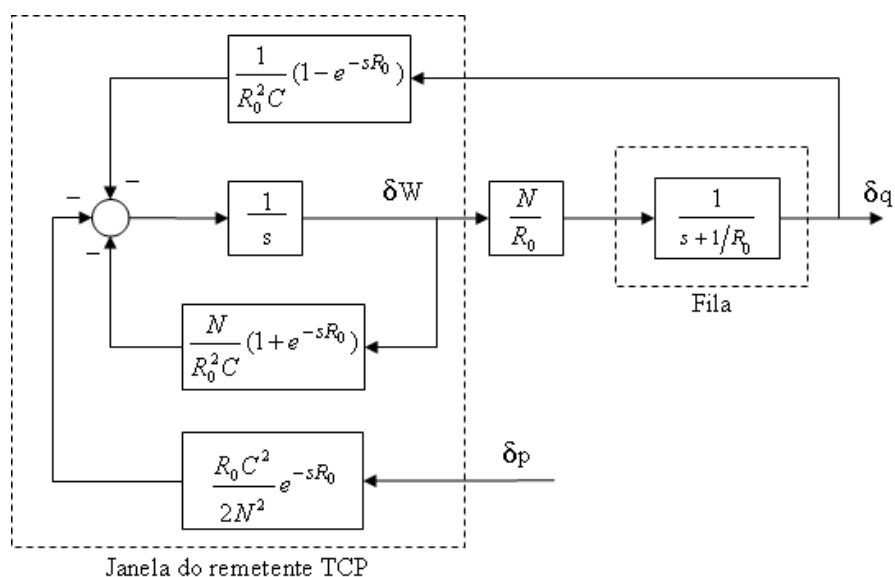


Figura 4.1: Diagrama de blocos da dinâmica linearizada do TCP [47].

sendo dada por

$$P(s) = \frac{\frac{C^2}{2N}}{(s + \frac{2N}{R_0^2 C})(s + \frac{1}{R_0})}. \quad (4.1)$$

A função de transferência $\Delta(s)$ representa a parte de alta frequência residual da dinâmica da janela do emissor, comumente chamada de “parasita” de alta frequência. Será visto adiante, que esta parcela provoca uma má performance do modelo do TCP quando este opera em frequências elevadas. Da manipulação do diagrama da figura 4.1 tem-se ;

$$\Delta(s) = \frac{2N^2 s}{R_0^2 C^3} (1 - e^{-sR_0}). \quad (4.2)$$

Analisando o sistema de controle realimentado da figura 4.2, é fácil concluir que a função de transferência $C(s)$ representa a lei de controle AQM linear e, como será mostrado ao longo das próximas seções, $C(s)$ será modificada de acordo com o método a ser empregado para estabilizar a dinâmica do protocolo TCP.

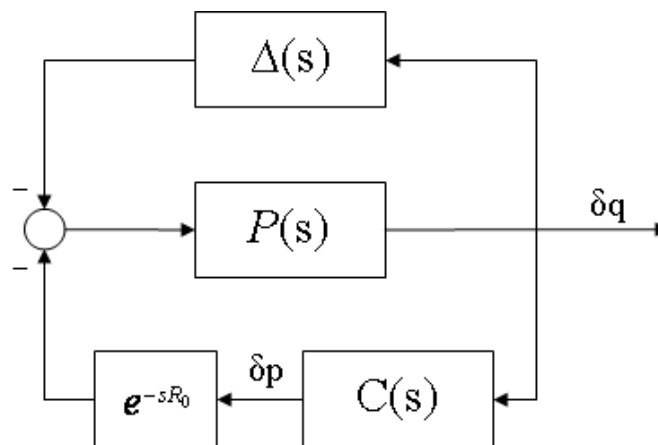


Figura 4.2: Diagrama de blocos do sistema TCP com controle de realimentação.

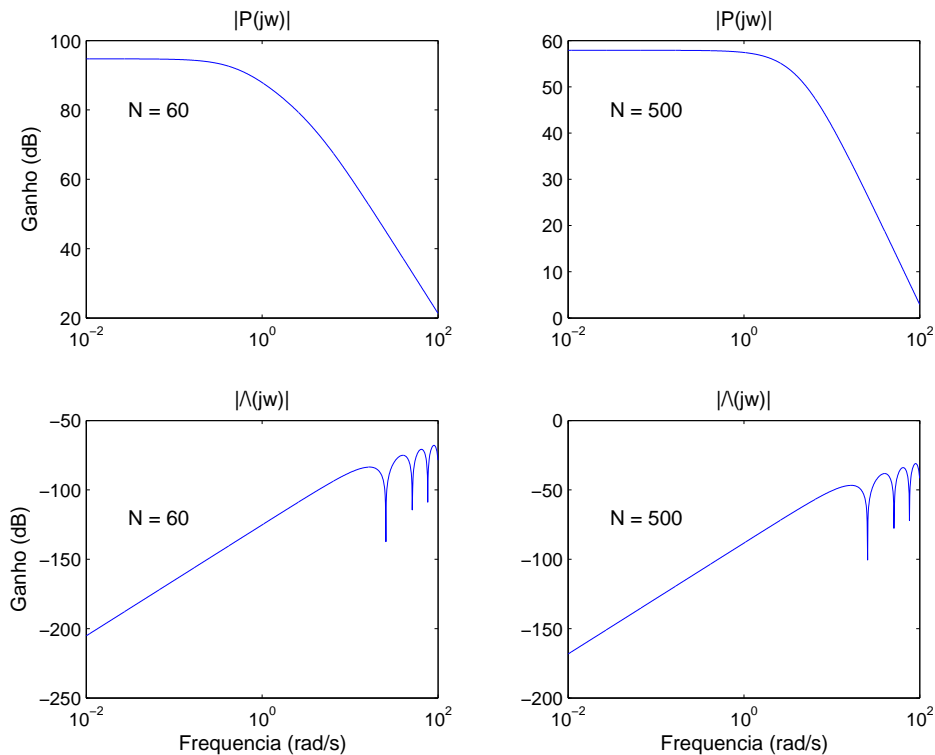
Como um exemplo numérico, será considerado o caso onde $q_0 = 175$ pacotes, $T_p = 0.2$ segundos e $C = 3750$ pacotes/seg. Então, de acordo com o valor do fator de carga N , define-se os outros parâmetros da rede assim como a planta $P(s)$ e o resíduo de alta frequência $\Delta(s)$ para cada situação. Na tabela 4.1 encontra-se o detalhamento dos referidos valores, sendo que, os diagramas de Bode referentes as funções de transferência desta tabela são mostrados na figura 4.3.

A resposta de frequência de $P(j\omega)$ revela que o sistema do protocolo TCP possui a essência de um filtro passa-baixa bem como a dependência inversa do ganho do sistema sobre

Tabela 4.1: Valores de $P(s)$ e $\Delta(s)$ para diferentes fatores de carga N .

N (sessões)	W_0 (pacotes)	p_0	R_0 (seg.)	$P(s)$	$\Delta(s)$
60	15	0.008	0.246	$\frac{117126}{(s+0.53)(s+4.1)}$	$0.00000224s(1 - e^{-sR_0})$
500	1.85	0.584	0.246	$\frac{14062}{(s+4.41)(s+4.1)}$	$0.00015600s(1 - e^{-sR_0})$

o número de sessões TCP. No traçado de Bode da parte residual de alta frequência pode-se ver, claramente, como $\Delta(j\omega)$ influencia negativamente o desempenho do protocolo TCP em altas frequências.

Figura 4.3: Diagrama de Bode (amplitude) de $P(s)$ e $\Delta(s)$.

Observando que o ganho de baixa-frequência de $P(s)$ em (4.1) é $\frac{(R_0 C)^3}{4N^2}$ enquanto que o ganho de alta-frequência é $\frac{C^2}{2N}$, pode-se verificar e comprovar a grande influência do fator de carga N e da capacidade do enlace C sobre a estabilidade, o transitório e o regime permanente do sistema. Ambos, baixo fator de carga N ou grande capacidade de enlace C , aumentam os referidos ganhos levando a uma diminuição da margem de estabilidade do sistema TCP.

Para melhorar a robustez do modelo TCP diante de incertezas dos parâmetros da rede

como fator de carga N , atraso de propagação T_p e também a capacidade de enlace C , serão projetados os compensadores ou controladores RED e PI utilizando o método no domínio de frequência (Diagrama de Bode) e o método das desigualdades matriciais lineares (LMI). Além da robustez, outro objetivo dos controladores é permitir um uso eficiente da fila no roteador. Para alcançar tal objetivo é necessário evitar que a fila fique ou vazia ou muito cheia. No primeiro caso existe uma subutilização do enlace, enquanto que a fila cheia provoca perda e retransmissões desnecessárias de pacotes, o que não é desejado.

Outra relação que deve ser respeitada é entre o tamanho da fila e o tempo de processamento de fila, também chamado de atraso de fila, que é igual a $\frac{q}{C}$. Como mostrado anteriormente, este atraso junto com o atraso de propagação T_p formam tempo de viagem de ida e volta (RTT) o qual é recomendado que seja o menor possível. Contudo, manter um baixo RTT também significa um pequeno atraso de fila e por consequência um baixo tamanho de fila, o que provoca a subutilização da rede.

4.2 PROJETO DE CONTROLADORES VIA BODE

No diagrama de Bode, a margem de estabilidade é medida através da análise da **margem de ganho** (MG) e da **margem de fase** (MF) [39]. A margem de ganho para um sistema de fase mínima, como é o caso da planta $P(s)$, representa o quanto o ganho de malha aberta pode ser aumentado antes que o sistema se torne instável. Dessa forma, voltando ao esquema da figura 3.4, verificamos que a margem de ganho é grosseiramente a quantidade N de sessões TCP que a rede pode tolerar. A margem de fase é o atraso de fase adicional na frequência de cruzamento do ganho, necessário para levar o sistema ao limiar de instabilidade. Assim, no contexto do controle AQM, podemos interpretar a margem de fase como a quantidade de incerteza no atraso RTT o modelo TCP sustenta sem tornar-se instável [46]. Ambas as margens de ganho ou de fase não fornecem uma indicação suficiente da estabilidade relativa de um sistema se forem usadas isoladamente. Então, para que um sistema de fase mínima seja estável, tanto a margem de ganho como a margem de fase devem ser positivas; para um desempenho satisfatório, a margem de ganho deve ser maior do que 6dB, e a margem de fase deve estar entre 30° e 60° [50].

No exemplo da figura 4.3, os diagramas de Bode de malha aberta demonstram de forma simplificada mas muito clara como a quantidade N de sessões TCP pode afetar a instabilidade de forma benéfica ou prejudicial. A partir destes diagramas obtemos: para $N=60$, $MG = +\infty$ dB e $MF = 0.77^\circ$; enquanto que para $N=500$, $MG = +\infty$ dB e $MF = 4.09^\circ$. Como pode-se concluir, quando foi aumentado o fator de carga de 60 para 500 sessões TCP, a

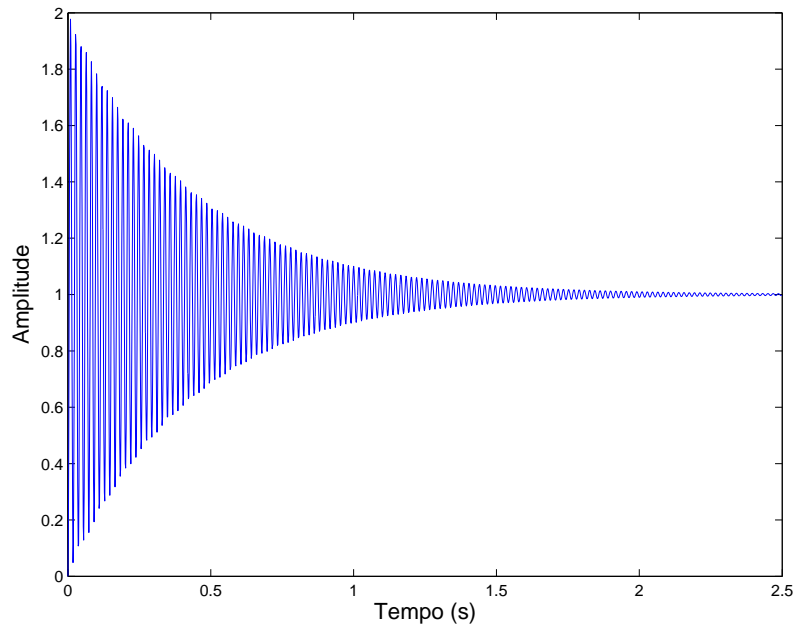


Figura 4.4: Resposta ao degrau unitário de $P(s)$ (sem compensação).

margem de estabilidade do sistema também aumentou apesar de não estar dentro do valor aceitável para uma boa performance do TCP. A margem de fase de 0.77° implica em um sistema bastante oscilatório, como pode ser observado no gráfico da resposta ao degrau unitário da planta $P(s)$, mostrado na figura 4.4.

4.2.1 Critérios de Estabilidade

Para estabilizar o sistema de malha fechada da figura 4.2 e por conseguinte alcançar os principais objetivos do controle AQM, robustez e uma eficiente utilização do enlace e da fila do roteador, o controlador $C(s)$ deve estabilizar a planta com atraso $P(s)e^{-sR_0}$ e também a dinâmica da janela de alta frequência $\Delta(s)$. Neste propósito, se faz necessária a seguinte função de transferência:

$$V(s) = \frac{P(s)}{1 + P(s)C(s)e^{-sR_0}}. \quad (4.3)$$

Então, se $C(s)$ estabiliza $P(s)e^{-sR_0}$, a função de transferência $V(s)$ é estável. E, como será comprovado nos resultados adiante, para que $\Delta(s)V(s)$ seja estável, a condição de ganho $|\Delta(j\omega)V(j\omega)| < 1$ deve ser obedecida, pois, conforme o critério de estabilidade de Nyquist [50, 52], isto implica na estabilidade de malha fechada da dinâmica do TCP.

4.2.2 Projeto do Controle RED no Domínio de Frequência

Na subseção 3.1.3 foi descrito a dinâmica do controle RED, o qual é o mais difundido método de implementar o gerenciamento ativo de fila (AQM). Como discutido, ele é constituído por um filtro passa-baixa e uma função de descarte como mostrado na figura 3.3. Seguindo este modelo, produz-se a função de transferência de $C(s)$ aplicada ao RED; portanto:

$$C_{red}(s) = \frac{KL_{red}}{(s+K)} \quad (4.4)$$

onde

$$L_{red} = \frac{P_{max}}{\bar{\mu} - \underline{\mu}}.$$

A função de transferência $G(s) = P(s)C_{red}(s)e^{-sR_0}$, proveniente da equação característica de (4.3), possui a forma

$$G(s) = \frac{\frac{KL_{red}C^2}{2N}e^{-sR_0}}{\left(s + \frac{2N}{R_0^2C}\right)\left(s + \frac{1}{R_0}\right)(s+K)}. \quad (4.5)$$

O objetivo da introdução do filtro passa-baixa no RED é atenuar os efeitos das rajadas de pacotes que eventualmente chegam ao roteador, todavia, a incorporação deste filtro ignora seu efeito sobre a estabilidade de malha fechada o qual introduz um atraso de fase adicional em um sistema que já possui duas dinâmicas de filtro passa-baixa e um atraso de tempo, como é verificado na equação (4.5). Seguindo a mesma linha de raciocínio proposta em [46, 47], projeta-se o pólo $-K$ do filtro passa-baixa para que o mesmo tenha uma frequência de corte (ou de cruzamento) menor do que as frequências de corte da planta $P(s)$.¹ Conseqüentemente, o tempo de malha fechada será grande, e este é o inconveniente que o atraso de fase adicional do RED gera no modelo do protocolo TCP. Para que K seja o pólo dominante na função de transferência $G(s)$, então a frequência de ganho unitário ω_c deve ser

$$\omega_c \ll \min \left\{ \frac{2N}{R_0^2C}, \frac{1}{R_0} \right\}. \quad (4.6)$$

A tendência do traçado do $|G(j\omega)|$ no diagrama de Bode é decair gradualmente e, portanto, possuir uma frequência de cruzamento ω_c com ganho unitário; isto é, $|G(j\omega_c)| = 1$.

¹Para a configuração do RED este raciocínio é o que melhor reflete a proposta do filtro passa-baixa, uma vez que, se o contrário for feito, ou seja, projetar o pólo $-K$ para que ele não seja o pólo dominante, o filtro passa-baixa apenas estaria medindo o valor instantâneo da fila.

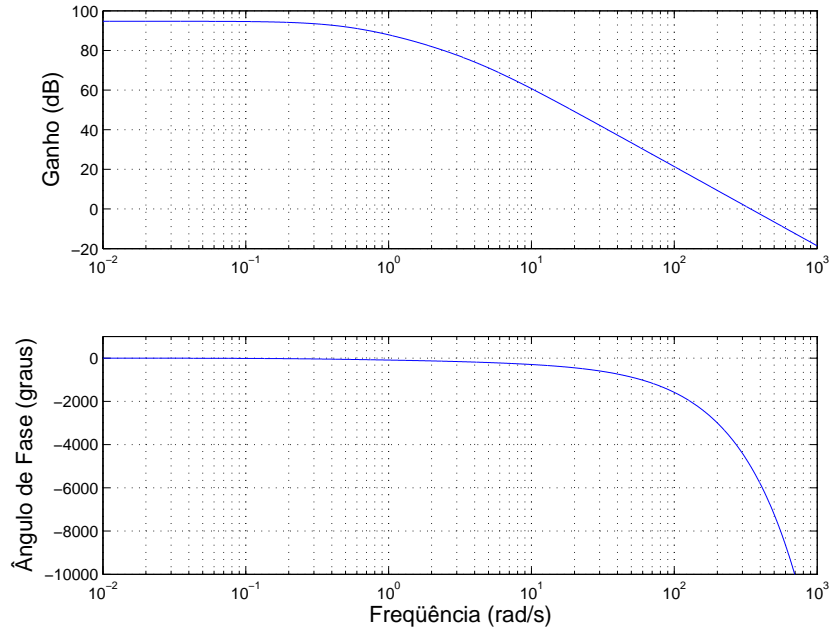


Figura 4.5: Diagrama de Bode de $P(s)e^{-sR_0}$.

Para conseguir que a margem de fase de cruzamento de $G(s)$ seja positiva, então, é necessário que $\angle G(j\omega_c) > -\pi$. Calculando o módulo e a fase de $G(j\omega_c)$ para a consideração feita em (4.6), é produzido:

$$|G(j\omega_c)| \approx \left| \frac{\frac{KL_{red}C^2}{2N}}{\frac{2N}{R_0^3C}(j\omega_c + K)} \right| \quad (4.7)$$

e

$$\angle G(j\omega_c) \approx -\omega_c R_0 - \text{atan} \frac{\omega_c}{K}. \quad (4.8)$$

Para que $G(s)$ seja estável é necessário adequar estas últimas equações à proposta do parágrafo anterior, ou seja, $|G(j\omega_c)| = 1$ e $\angle G(j\omega_c) > -\pi$, o que produz

$$\left| \frac{\frac{KL_{red}C^2}{2N}}{\frac{2N}{R_0^3C}(j\omega_c + K)} \right| = 1, \quad (4.9)$$

$$\omega_c R_0 - \text{atan} \frac{\omega_c}{K} + \pi > 0.$$

Através da equação (4.9) os parâmetros ω_c , K e L_{red} são ajustados de forma a produzir tanto a margem de ganho como a margem de fase positivos, sendo que, esta deve ser maior

que 30° . É importante salientar que a introdução do atraso e^{-sR_0} na Planta $P(s)$ tem por consequência margens de ganho e fase negativas como pode ser analisado na figura 4.5.

4.2.3 Cálculo do Controle C_{red}

Os parâmetros da rede para o cálculo do controle serão os mesmos utilizados na tabela 4.1: $C = 3750$ pacotes/seg, $R_0 = 0.246$ segundos e $N = 60$ sessões TCP. Fazendo a frequência de corte ω_c em (4.6) equivalente a um décimo do menor valor dos pólos de $P(s)$ [46], temos:

$$\omega_c = 0.1 \cdot \min \{0.53, 4.1\} = 0.053 \text{ rad/seg.} \quad (4.10)$$

O fator multiplicativo 0.1 usado para determinar ω_c é um dos quais sustenta a margem de estabilidade do modelo TCP. Se for escolhido um valor maior que 0.1, o controle $C(s)$ é projetado com uma margem de estabilidade menor; contudo, este perfil mais arrojado, provê uma maior velocidade de resposta.

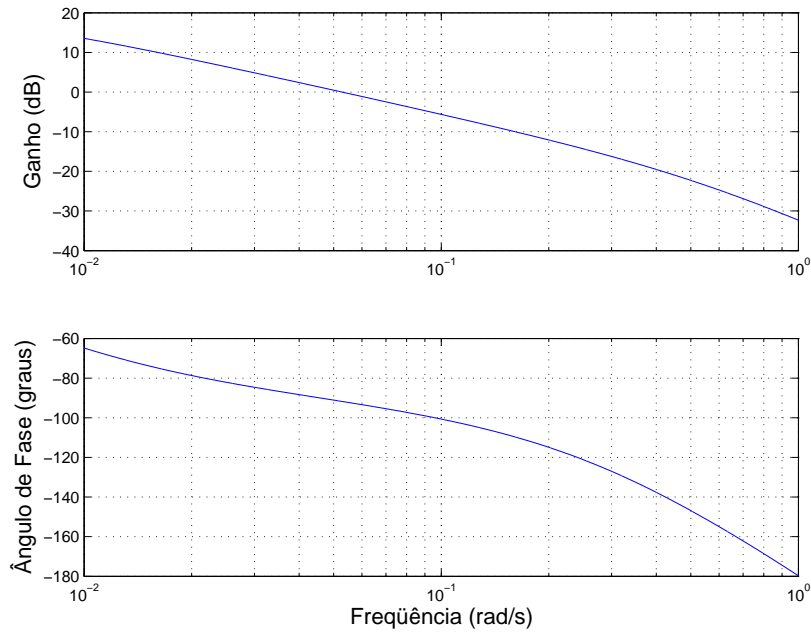
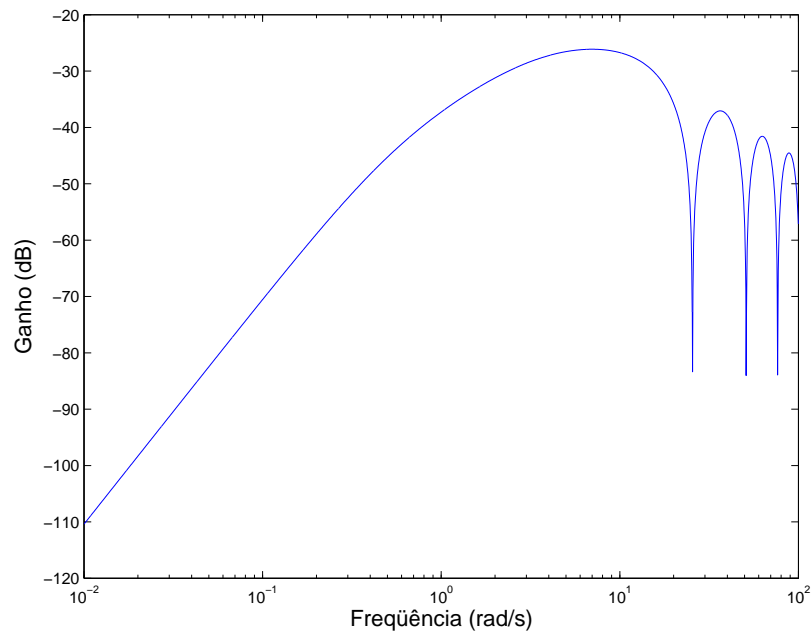
Como em [47], escolhemos $K = 0.005$ e, conseqüentemente, satisfazendo a equação (4.9) obtemos $L_{red} = 1.9530 \times 10^{-4}$; o que fornece o controle RED

$$C_{red}(s) = \frac{0.005 (1.95 \times 10^{-4})}{(s + 0.005)}. \quad (4.11)$$

As margens de ganho e de fase positivas (respectivamente, 32.6 dB e $88,7^\circ$) do diagrama de Bode de $G(s)$, na figura 4.6 (a), revelam que $C_{red}(s)$ realmente estabiliza $P(s)e^{-sR_0}$. Já na figura 4.6 (b), confirmamos que $|\Delta(j\omega)V(j\omega)| < 1$ e, assim, que o modelo TCP é estável em malha fechada contra parasitas de alta frequência.

Na subseção 3.2.3 considerou-se que o pólo K é igual a $\ln(1 - \alpha)/\zeta$, onde α é a ponderação e ζ é o tempo de amostragem do filtro passa-baixa média móvel exponencial ponderada (EWMA). Normalmente o algoritmo RED atualiza o EWMA a cada chegada de pacote, e portanto $\zeta = 1/C$. Esta expressão mostra a grande influência que a capacidade C do enlace tem sobre o tamanho da fila. A medida que a capacidade vai aumentando, o pólo K também aumenta, e o filtro passa-baixa do RED começa a calcular valores médios da fila mais próximos dos valores instantâneos, o que resulta em oscilações.

Com os mesmos valores de K e L_{red} usados no cálculo da equação (4.11), a tabela 4.2 mostra os valores obtidos para a ponderação α , o limite de mínimo $\underline{\mu}$ e o limite máximo $\bar{\mu}$ para diferentes valores da probabilidade máxima de descarte p_{max} e do tamanho máximo da fila do roteador (*Buffer*).

(a) Diagrama de Bode de $G(s)$.(b) Diagrama de Bode (amplitude) de $\Delta(j\omega)V(j\omega)$.Figura 4.6: Respostas de frequência com controle $C_{red}(s)$.

Observação 4.1: A restrição para a obtenção da margem de fase, imposta pela equação (4.9), é fundamentalmente uma relação entre a frequência de corte ω_c e o fator de ponderação K . Aquela é responsável pela constante de tempo do modelo TCP, enquanto que a ponderação do filtro passa-baixa indica o peso que é dado no cálculo da fila média. Pode-se, dessa maneira, alterar os valores destes parâmetros de forma a projetar outros controles

Tabela 4.2: Parâmetros do RED (*via Bode*) para valores diferentes de p_{max} .

p_{max}	α	$\underline{\mu}$	$\bar{\mu}$	Buffer	$\bar{\mu} - \underline{\mu}$
0.1	0.00000133	150	700	900	550
0.05	0.00000133	102	370	900	268
0.15	0.00000133	200	1006	1300	806

que ou privilegiem a estabilidade do sistema TCP ou que tenham uma constante de tempo de malha fechada menor; neste caso, o modelo TCP responderá mais rápido a possíveis flutuações da rede.

4.2.4 Projeto do Controle PI no Domínio de Frequência

A associação do modelo TCP com o controle RED, na equação (4.3), resulta em um sistema compensado tendo o erro de regime permanente com grande dependência dos parâmetros da rede. Por conseqüência, a variação da fila δq na figura 4.2 nunca converge para zero o que não é uma situação adequada. Além disso, para diminuir este erro, é desejável que o valor de L_{red} em (4.9) seja tão alto quanto possível; fato que não é viável nas circunstâncias do controle RED.

Os problemas de erro de regime permanente e do tempo de resposta à malha fechada relativamente lenta proporcionados pelo RED ao modelo TCP podem ser aperfeiçoados com a introdução do clássico compensador proporcional-integral (PI) cuja função de transferência é

$$C_{pi}(s) = \frac{K_{pi}(\frac{s}{z} + 1)}{s}, \quad (4.12)$$

a qual representa a lei de controle AQM linear $C(s)$ aplicada ao controle PI. A função de transferência de malha aberta $G(s)$ usando o controle PI é:

$$G(s) = \frac{\frac{K_{pi}C^2}{2N}(\frac{s}{z} + 1)e^{-sR_0}}{s(s + \frac{2N}{R_0^2C})(s + \frac{1}{R_0})}. \quad (4.13)$$

O controlador PI é um compensador por atraso de fase. Assim, sua característica é ter um ganho alto nas baixas frequências, mais especificamente ganho infinito na frequência zero; o que melhora o comportamento de regime estacionário. Assumindo que o zero z do PI coincida com a menor frequência de corte da planta $P(s)$ [48], evitam-se constantes de

tempo muito grandes para o compensador. Portanto, define-se:

$$z = \frac{2N}{R_0^2 C}. \quad (4.14)$$

Definindo a frequência de cruzamento com ganho unitário de $G(s)$, também como em [48], faz-se

$$\omega_c = \min \left\{ \frac{2N}{R_0^2 C}, \frac{1}{R_0} \right\},$$

ou seja

$$\omega_c = \frac{2N}{R_0^2 C}. \quad (4.15)$$

Calculando o módulo e a fase de $G(j\omega_c)$ para as considerações feitas em (4.14) e (4.15), produz-se:

$$|G(j\omega_c)| \approx \left| \frac{\frac{K_{pi} C^2}{2N}}{j\omega_c(j\omega_c + \frac{1}{R_0})z} \right| \quad (4.16)$$

e

$$\angle G(j\omega_c) \approx -90^\circ - \frac{180}{\pi} \omega_c R_0 - \text{atan}(\omega_c R_0). \quad (4.17)$$

Encontrando as condições de cruzamento, $|G(j\omega_c)| = 1$ e $\angle G(j\omega_c) > -\pi$, tem-se

$$\left| \frac{\frac{K_{pi} C^2}{2N}}{j\omega_c(j\omega_c + \frac{1}{R_0})z} \right| = 1, \quad (4.18)$$

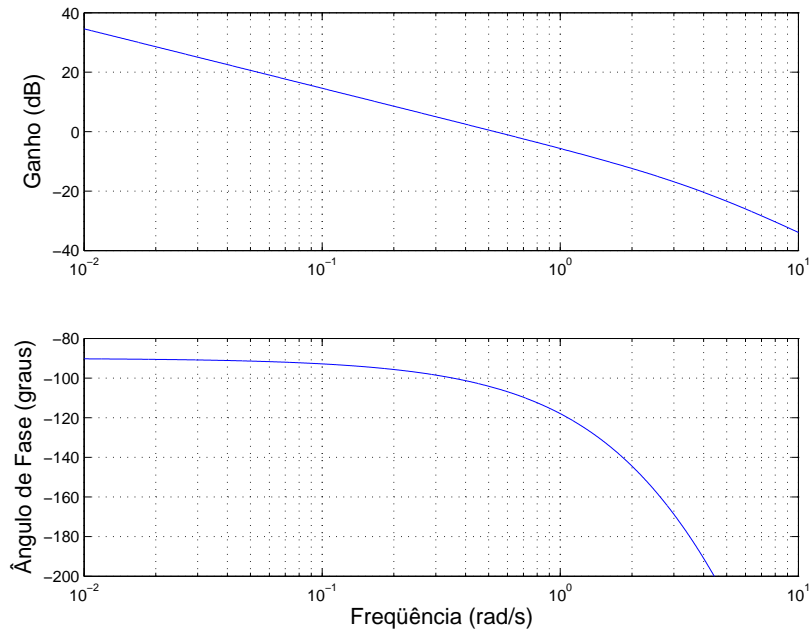
$$-90^\circ - \frac{180}{\pi} \omega_c R_0 - \text{atan}(\omega_c R_0) + \pi > 0.$$

4.2.5 Cálculo do Controle C_{pi}

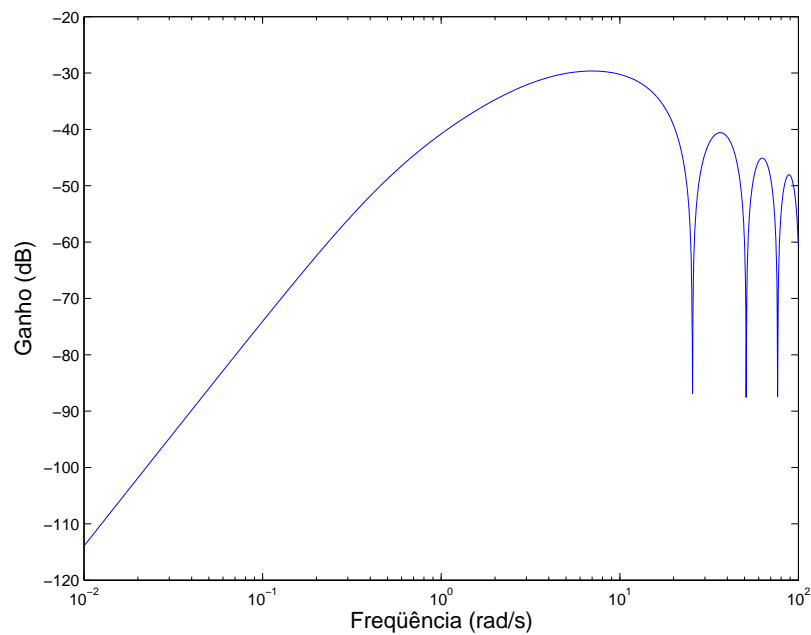
Considerando os parâmetros de rede da subseção 4.2.3: $C = 3750$ pacotes/seg, $R_0 = 0.246$ segundos e $N = 60$ sessões TCP. De (4.15), ajusta-se a frequência de corte para $\omega_c = z = 0.53$ rad/seg. (Note que os valores de ω_c e z foram escolhidos de forma a satisfazer as restrições de margem de fase da equação (4.18)). Também de (4.18) obtém-se a constante

$K_{pi} = 9.8264 \times 10^{-6}$. Daí,

$$C_{pi}(s) = 9.83 \times 10^{-6} \frac{(0.53s + 1)}{s} . \quad (4.19)$$



(a) Diagrama de Bode de $G(s)$.



(b) Diagrama de Bode (amplitude) de $\Delta(j\omega)V(j\omega)$.

Figura 4.7: Respostas de frequência com controle $C_{pi}(s)$.

Na figura 4.7 (a), tem-se o diagrama de Bode de $G(s)$ com o controle PI. Como no RED, as margens de ganho e de fase são positivas (respectivamente, 18.7 dB e 75°) o que comprova a estabilidade do modelo TCP; apesar da margem de estabilidade ter diminuído em relação a do controle RED, fato que já era esperado.

A largura de banda de 0.53 rad/seg. indica que a resposta do sistema ficou mais rápida do que a do controle RED. Isto é de grande valia quando o roteador precisa recuperar o valor de fila média pré-estabelecido, que foi “perdido”, por exemplo, devido a um estouro de buffer.

A função de transferência do controle PI na equação (4.19), é descrita no Domínio de S; para tal propósito utilizou-se a transformada de Laplace. Entretanto, para a implementação no simulador de redes NS-2 é necessário que se tenha o controle discreto. Desta maneira, sendo f_a a frequência de amostragem, e utilizando-se a aproximação bilinear de Tustin [1], produz-se a seguinte função de transferência de (4.19) no Domínio Z:

$$C_{pi}(z) = \frac{az - b}{z - 1}. \quad (4.20)$$

Como $\omega_c = 0.53$ rad/seg, a frequência de amostragem f_a precisa ser no mínimo maior que 10 vezes $\frac{\omega_c}{2\pi}$, o que é aproximadamente 3.4 Hz. Como a capacidade do roteador é de 3750 pacotes/seg, é possível se amostrar com uma frequência de 3750 Hz, ou seja, a cada pacote que chega, uma amostra é retirada. No entanto, isto não traz benefícios perceptíveis, como confirma a tabela 4.3.

Tabela 4.3: Valores de a e b (via Bode) para diferentes frequências f_a .

f_a (Hz)	a	b
160	0.000018571	0.000018510
3750	0.000018542	0.000018539

Observação 4.2: As regiões de estabilidade para os controles RED e PI são encontradas, respectivamente, através das equações (4.9) e (4.18). Mantendo-se K e L_{red} (do controle C_{red}) e K_{pi} (do controle C_{pi}) constantes, estas duas regiões dependem basicamente dos parâmetros da rede C , N e R_0 . Nesse sentido, uma diminuição do número de sessões, um aumento da capacidade do enlace e um aumento do RTT fazem com que a margem de estabilidade diminua.

4.3 DESIGUALDADES MATRICIAIS LINEARES (LMIs) NA TEORIA DE CONTROLE

Desigualdades matriciais Lineares (LMIs) surgiram na teoria de controle no final do século XIX, quando Lyapunov formulou a condição $A^T P + P A < 0$ com $P > 0$, a qual é uma forma especial de uma LMI, para verificar a estabilidade de sistemas dinâmicos do tipo $\dot{x}(t) = Ax(t)$. No entanto, foi somente a partir da década de 1940, através dos trabalhos de Lur'e, Postnikov e outros pesquisadores da antiga União Soviética, que os métodos de Lyapunov foram aplicados para solucionar problemas reais da engenharia de controle, especialmente, o problema da estabilidade de um sistema de controle na presença das não-linearidades dos atuadores. Os critérios de estabilidade por eles empregados tinham a forma de LMIs. Estas desigualdades eram reduzidas a desigualdades polinomiais e posteriormente resolvidas “à mão”; o que limitava as aplicações das LMIs a sistemas pequenos de no máximo terceira ordem.

Outros avanços significativos só vieram a acontecer na década de 1960, quando Yakubovich, Popov, Kalman e outros estudiosos foram bem sucedidos ao reduzir a solução das LMIs propostas por Lur'e e outros em simples critérios gráficos, usando o lema positivo-real (PR) [8], os quais podiam ser usados em sistemas de alta ordem porém que não continham mais de uma não-linearidade. Ainda na década de 1960 foram feitos estudos os quais comprovaram que a família de LMIs que eram atendidas pelo lema PR podiam não somente serem resolvida por meios gráficos como também através da resolução das equações algébricas de Riccati (ARE).

Foi a partir do final da década de 80, com a criação e aperfeiçoamento de algoritmos de otimização convexa, como pontos interiores, que os pesquisadores voltaram a se interessar pelo uso de LMIs na Teoria de Controle. Desde então muitos dos resultados usuais da teoria de controle e sistemas, estão sendo reescritos como LMIs. A vantagem é que com o uso de LMIs é possível tratar com mais facilidades sistemas mais complexos, como sistemas incertos e até mesmo sistemas não-lineares.

4.3.1 Sistemas Incertos

Em muitas situações práticas não é possível e/ou conveniente representar com precisão todas as dinâmicas de um sistema. Flutuações em parâmetros e dinâmicas muito rápidas são com frequência representadas como incertezas do modelo. No primeiro caso, a idéia é representar o parâmetro físico por um parâmetro incerto cujo valor não se conhece precisamente, mas se sabe que encontra-se em um intervalo conhecido. No segundo caso, as

dinâmicas rápidas negligenciadas no modelo do sistema são representadas por um ou mais operadores não conhecidos precisamente, mas com ganho limitado por uma constante com valor conhecido. Em ambos os casos, parâmetros e operadores, os elementos parcialmente conhecidos recebem o nome de incerteza e o modelo do sistema que contém algum desses elementos é chamado de sistema incerto o qual pode ser variante ou invariante no tempo.

Dessa forma, todo modelo matemático pode ser considerado na prática como uma aproximação da dinâmica do processo físico real. Conseqüentemente, o modelo matemático obtido pode apresentar diferentes tipos de incertezas, decorrentes das dinâmicas não modeladas, variações paramétricas, presença de ruídos, ou até mesmo erros decorrentes da etapa de linearização, entre outros fatores. Dependendo da sua origem e de como elas são descritas matematicamente, as incertezas podem ser classificadas como estruturadas, não estruturadas, paramétricas ou não paramétricas.

Obviamente, é de grande importância que as incertezas sejam levadas em conta tanto na análise como síntese de controladores para sistema sujeitos a variações paramétricas. Para tal, é conveniente representar o modelo físico por um sistema incerto, constituído do modelo matemático (sistema nominal) mais incertezas em torno deste, sendo a análise ou projeto feita em torno do sistema incerto. Várias dificuldades surgem ao se trabalhar com sistemas incertos. Uma das principais é como modelar e descrever as incertezas no problema. Incertezas descritas de forma genérica podem acarretar restrições na busca de soluções.

Descrição das incertezas

Assumindo que as incertezas ou parâmetros incertos admissíveis são pertencentes a um polítopo², considera-se o seguinte sistema linear

$$\dot{x}(t) = A(\theta)x(t), \quad x(0) = 0. \quad (4.21)$$

onde $x \in \mathbb{R}^n$ é o vetor de estados e $\theta \in \mathbb{R}^p$ é o vetor de parâmetros incertos que satisfaz a condição

$$\theta \in \mathfrak{B}_\theta,$$

onde $\theta = [\theta_1, \dots, \theta_p] \in \mathbb{R}^p$ e \mathfrak{B}_θ representa um polítopo com $\ell_\theta = 2^p$ vértices. Em algumas situações, assume-se também que θ_i é um parâmetro variante no tempo com taxa de variação $\dot{\theta}_i$, e que $(\theta, \dot{\theta}) \in \mathfrak{B}$, um polítopo com ℓ vértices. Por exemplo, supondo $\theta \in \mathfrak{B}_\theta$ e $A(\theta)$ afim

²Polítopo é um conjunto convexo fechado, que pode ser representado pela combinação convexa dos vértices, ou por inequações matriciais. Para mais detalhes sobre polítopo veja [8, 6, 62].

em θ , uma representação possível para a matriz $A(\theta)$ é dada por

$$\mathcal{A} = \left\{ A(\theta) = \sum_{i=1}^{\ell} q_i A_i, \quad \sum_{i=1}^{\ell} q_i = 1, q_i \geq 0 \right\}, \quad (4.22)$$

onde o conjunto \mathcal{A} é convexo fechado, as matrizes A_i são conhecidas como vértices e os escalares q_i são os elementos de combinação convexa. Uma das grandes vantagens em descrever os parâmetros incertos na forma politópica é a convexidade do conjunto resultante. Desta forma, é possível garantir que, se um dado conjunto de condições estiver satisfeito nos vértices do politopo, então, estas mesmas condições também estão satisfeitas no interior desta região. O inconveniente desta abordagem é o problema da explosão exponencial, já que o número de LMIs é dado por uma função do tipo 2^p , onde p é o número de parâmetros incertos no sistema. Por exemplo, para testar as condições de sistema com 3 elementos incertos, deve-se verificar as mesmas condições nos 2^3 vértices, ou seja, deve-se testar as condições para 8 valores diferentes da matriz $A(\theta)$.

4.3.2 Representação de LMIs

A idéia básica da LMI consiste em reformular problemas de controle em termos de inequações matriciais lineares, isto é, restrições de desigualdade envolvendo matrizes simétricas afins nas variáveis de decisão. Dessa forma, passa-se a trabalhar com problemas convexos que podem ser resolvidos de forma eficiente.

Definição 4.1 (LMI, [62]) *Uma LMI (Linear Matrix Inequality) é uma desigualdade matricial do tipo $F(g) > 0$, no qual $F(g) : \mathbb{R}^m \rightarrow \mathbb{R}^{q \times q}$ é simétrica e afim nas variáveis de busca que são representadas pelo vetor g . Assim uma LMI pode ser genericamente representada na forma*

$$F(g) = F_0 + \sum_{i=1}^m g_i F_i > 0 \quad \mathbf{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_m \end{bmatrix} \quad (4.23)$$

onde $F_i = F_i^T \in \mathbb{R}^{q \times q}$ são matrizes dadas e g_i são variáveis escalares a serem determinadas de forma a satisfazer a desigualdade (se possível). Quando existe uma solução g para $F(g) > 0$ diz-se que a LMI é factível. □□□

É importante enfatizar que uma LMI pode ser representada de várias formas e dificilmente aparece num problema na forma genérica afim (4.23). Por exemplo, dado uma matriz A e uma matriz $Q > 0$, a função matricial $F(P) = A^T P + PA + Q$, que aparece em vários problemas de estabilidade, é afim na variável P e portanto a desigualdade $F(P) < 0$ é uma LMI que pode ser facilmente reescrita na forma (4.23) onde g é o vetor contendo os elementos da matriz P a ser determinada. A vantagem da representação genérica afim (4.23) é que toda LMI pode ser reescrita dessa forma e por isso todos os algoritmos de resolução de LMIs são desenvolvidos para essa representação. No entanto a conversão de uma LMI para a forma afim é feita internamente nos pacotes e o usuário não precisa se preocupar com isso. Inúmeros problemas em diferentes áreas do conhecimento podem ser reformulados e numericamente resolvidos através de LMIs [8], é o caso por exemplo dos resultados apresentados em [5, 6, 17, 31, 42, 62].

4.3.3 Estabilidade Quadrática

Funções de Lyapunov quadráticas³ do tipo $v(x) = x^T P x$ são facilmente determinadas através de $A^T P + PA + Q = 0$ para o caso de sistemas lineares invariantes no tempo e além disso fornecem condições necessárias e suficientes para a estabilidade. Para sistemas lineares variantes o método de Lyapunov requer a utilização de uma função $v(t, x) = x^T P(t) x$ que também varia no tempo e que satisfaz a equação diferencial matricial $\dot{P}(t) + A(t)^T P(t) + P(t) A(t) + Q(t) = 0$ cuja solução é mais complicada de ser obtida numericamente. Devido à sua simplicidade, é muito comum o estudo da estabilidade de um sistema através da função $v(x) = x^T P x$, com P constante. Para sistemas variantes no tempo ou ainda não-lineares este tipo de função pode conduzir a resultados conservadores pois, sendo P constante, a função de Lyapunov não consegue levar em conta a rapidez de variação da matriz $A(t)$. Já com a função $v(t, x) = x^T P(t) x$ essa taxa de variação é levada em consideração através do termo $\dot{P}(t)$ presente nas condições de estabilidade $\dot{P}(t) + A(t)^T P(t) + P(t) A(t) + Q(t) = 0$. Assim, quando se consegue mostrar a estabilidade de um sistema com uma função do tipo $v(x) = x^T P x$ em que P é constante, apenas os valores que a matriz $A(t)$ assume com o passar do tempo é que são levados em consideração mas não a rapidez com que $A(t)$ muda de valores. Logo, se está afirmando que o sistema é exponencialmente estável [8, 62] independentemente da taxa de variação de $A(t)$. Essa forma de mostrar a estabilidade de um sistema é conhecida na literatura como estabilidade quadrática e é bastante útil no estudo de estabilidade de sistemas incertos quando nenhuma informação sobre a taxa de variação do

³Uma função quadrática do tipo $v(x) = x^T P x$ é negativa definida ($v(x) < 0$) se todos os autovalores de P forem negativos para $\forall x \neq 0$, ou seja, se a matrix P for negativa definida ($P < 0$).

parâmetro incerto está disponível.

Aplicando a função $v(x) = x^T P x$ ao sistema nominal $\dot{x} = Ax$, tem-se a noção de estabilidade quadrática apresentada na seqüência.

Definição 4.2 (Estabilidade Quadrática, [62, 5]) *o sistema $\dot{x} = Ax$ é assintoticamente estável se e somente se existir uma matriz $P > 0$ que satisfaça*

$$A^T P + P A < 0. \quad (4.24)$$

□□□

É fácil verificar da definição acima que se o sistema é quadraticamente estável, então, $v(x) = x^T P x$ é uma função de Lyapunov para o sistema, pois $v(x) = x^T P x > 0$ e $\dot{V}(x) = x^T (A^T P + P A)x < 0$ para $\forall x \neq 0$. A idéia central da teoria de Lyapunov, portanto, é buscar uma função positiva definida que dependa dos estados do sistemas de forma que a sua energia decresça, ou seja, que a derivada temporal da função de Lyapunov seja estritamente negativa para todas as trajetórias do sistema. O resultado da equação (4.24) fornece condições necessárias e suficientes para a análise da estabilidade do sistema nominal. Nas últimas décadas funções de Lyapunov quadráticas vem sendo também aplicadas a sistemas incertos. No entanto, neste caso, é necessário verificar se o sistema é estável para todos os valores possíveis dos parâmetros incertos. Para o caso de um sistema com incertezas na forma politópica a estabilidade quadrática é verificada de forma semelhante à apresentada anteriormente, sendo que maiores detalhes podem ser encontrados em [5, 8, 62]. A título de exemplo, o sistema (4.21) onde θ é um parâmetro incerto que pertence ao intervalo $\mathfrak{B}_\theta = \{\theta : \underline{\theta} \leq \theta \leq \bar{\theta}\}$ é quadraticamente estável se a condição (4.24) está satisfeita para os extremos do intervalo que correspondem aos vértices do politopo \mathfrak{B}_θ , ou seja,

$$P > 0, \quad A(\underline{\theta})^T P + P A(\underline{\theta}) < 0 \quad \text{e} \quad A(\bar{\theta})^T P + P A(\bar{\theta}) < 0.$$

A seguir apresenta-se as principais propriedades e conceitos envolvendo operações com LMIs.

4.3.4 Complemento de Schur

Tipicamente, o problema de controle que se quer resolver não aparece naturalmente na forma de uma LMI. Algumas ferramentas matemáticas são bastantes úteis na tarefa de

tentar reescrever o problema em questão através de LMIs. Se a natureza do problema original for convexa, muito provavelmente se encontrará uma forma de reescrevê-lo através de LMIs. O complemento de Schur é uma ferramenta básica na manipulação de desigualdades matriciais.

Lema 4.1 (Complemento de Schur, [8]) *Seja $g \in \mathbb{R}^m$ o vetor de variáveis de decisão e $M_1(g)$, $M_2(g)$, $M_3(g)$ funções afins em g com $M_1(g)$ e $M_2(g)$ simétricas. Então as seguintes afirmações são equivalentes*

$$a) \quad M_1(g) - M_3(g)M_2(g)^{-1}M_3(g)^T > 0 \quad \text{com} \quad M_2(g) > 0 \quad (4.25)$$

$$b) \quad \begin{bmatrix} M_1(g) & M_3(g) \\ M_3(g)^T & M_2(g) \end{bmatrix} > 0 \quad (4.26)$$

□□□

Note que (4.25) não é uma LMI pois $M(g) = M_1(g) - M_3(g)M_2(g)^{-1}M_3(g)^T$ não é uma função afim em g . No entanto, o conjunto de inequações não lineares em (4.25) é equivalente à LMI em (4.26). E que, para satisfazer (4.25) e (4.26) deve-se ter $M_1(g) > 0$ e $M_2(g) > 0$ como condições necessárias, porém não suficientes.

Exemplo 4.1 *Considere a seguinte inequação matricial não linear associada a equação de Riccati:*

$$A^T P + PA + PBR^{-1}B^T P + Q < 0 \quad (4.27)$$

onde A , B , $Q = Q^T$, $R = R^T > 0$ são matrizes dadas com dimensões apropriadas, e $P = P^T$ é a variável de decisão. Note que (4.27) é uma desigualdade matricial quadrática na variável P . Então, aplicando o complemento de Schur na expressão acima leva à seguinte LMI:

$$\begin{bmatrix} -A^T P - PA - Q & PB \\ B^T P & R \end{bmatrix} > 0 \quad (4.28)$$

A representação (4.28) mostra claramente que a desigualdade matricial quadrática (4.27) é convexa em P , o que não é óbvio.

4.3.5 Lema de Finsler

Utilizando o lema de Finsler é possível transformar uma LMI associada a uma restrição de igualdade em uma única LMI adicionando multiplicadores ou novos graus de liberdade. Estes graus de liberdades são inseridos a partir da inclusão de novas variáveis de decisões oriundas da aplicação deste lema.

Lema 4.2 (Lema de Finsler, [17, 62]) *Seja $C \in \mathbb{R}^{n \times m}$ uma matriz dada e C_0 uma base para o espaço nulo de C e, $F(g)$ uma função afim em $g \in \mathbb{R}^q$ com $F(g) = F(g)^T \in \mathbb{R}^{m \times m}$, então as seguintes condições são equivalentes*

$$\begin{aligned}
 a) \quad & \exists g : x^T F(g)x < 0 \quad \forall x \in \mathbb{R}^m : Cx = 0, x \neq 0 \\
 b) \quad & \exists g, L : F(g) + LC + C^T L^T < 0 \quad L \in \mathbb{R}^{m \times n} \\
 c) \quad & \exists g : C_0^T F(g)C_0 < 0 \\
 d) \quad & \exists g, \alpha : F(g) - \alpha C^T C < 0 \quad \alpha \in \mathbb{R}
 \end{aligned} \tag{4.29}$$

□□□

A prova de que as relações (4.29) são realmente equivalentes pode ser encontrada em [18]. Para mostrar a aplicabilidade do lema de Finsler, considere o seguinte exemplo.

Exemplo 4.2 *Considere o seguinte sistema linear invariante no tempo $\dot{x} = Ax$ onde $x \in \mathbb{R}^n$ é o vetor de estados e A é uma matriz Hurwitz. Pela teoria de Lyapunov existe uma função de Lyapunov $v(x) = x^T P x$, onde P é uma matriz simétrica definida positiva tal que:*

$$\dot{v}(x) = \dot{x}^T P x + x^T P \dot{x} = x^T (A^T P + P A)x < 0. \tag{4.30}$$

A condição acima é satisfeita se e somente $A^T P + P A < 0$. Alternativamente, a condição (4.30) pode ser representada em um espaço aumentado associada a uma restrição algébrica, em outras palavras:

$$\begin{bmatrix} \dot{x} \\ x \end{bmatrix}^T \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \end{bmatrix} < 0, \quad \forall \begin{bmatrix} \dot{x} \\ x \end{bmatrix} : \begin{bmatrix} -I & A \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \end{bmatrix} = 0. \tag{4.31}$$

Utilizando o lema de Finsler obtém-se a seguinte condição necessária e suficiente:

$$\begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} + L \begin{bmatrix} -I & A \end{bmatrix} + \begin{bmatrix} -I \\ A^T \end{bmatrix} L^T < 0. \quad (4.32)$$

onde P e L são as variáveis de decisão.

Observe que na condição acima não existe um acoplamento direto entre as matrizes de Lyapunov e do sistema. Tal característica pode ser explorada em sistemas incertos e não lineares na busca de soluções convexas ou menos conservadoras.

4.3.6 Procedimento-S

Freqüentemente, na teoria de controle, encontram-se problemas no qual alguma função quadrática (ou forma quadrática) precisa ser positiva (ou negativa) definida sempre que outro conjunto de formas quadráticas forem positivas (ou negativas) definidas. Tais problemas podem ser solucionados construindo-se uma única desigualdade através da técnica que ficou conhecida como Procedimento-S (*S-Procedure*) a qual permite concatenar várias restrições escalares de desigualdade em uma única, sendo que, para reduzir o conservadorismo, esta técnica introduz multiplicadores como fatores de ponderação a serem determinados.

Lema 4.3 (Procedimento-S, [8]) *Sejam F_0, \dots, F_p funções quadráticas da variável $\zeta \in \mathbb{R}^n$, com a seguinte estrutura:*

$$F_i(\zeta) = \zeta^T T_i \zeta + 2u_i^T \zeta + v_i \quad i = 0, \dots, p \quad (4.33)$$

onde $T_i = T_i^T \in \mathbb{R}^{n \times n}$, $u_i \in \mathbb{R}^n$ e $v_i \in \mathbb{R}$. Então, a condição

$$F_0(\zeta) \geq 0 \quad \text{para todo } \zeta \text{ tal que } F_i(\zeta) \geq 0 \quad i = 1, \dots, p \quad (4.34)$$

é satisfeita se existem escalares $\tau_1, \dots, \tau_p \geq 0$ tais que:

$$F_0(\zeta) - \sum_{i=1}^p \tau_i F_i(\zeta) \geq 0, \quad \text{para } \forall \zeta. \quad (4.35)$$

□□□

Exemplos com aplicações do lema 4.3 podem ser encontrados em [8, 17, 31, 62].

4.4 PROJETO DE CONTROLADORES VIA LMI

4.4.1 Construção das LMIs para o Controle RED

Para a construção das Lmis⁴ foi considerado o método proposto por FRIDMAN et al. [30] o qual descreve um eficiente critério de estabilidade para sistemas com variação no atraso (*jitter*) e é aplicado para resolver o problema de estabilidade robusta desses sistemas na presença de incertezas politópicas por meio de um controle de realimentação de estados.

É válido lembrar que o protocolo de transporte TCP é por essência um sistema que possui um atraso que sofre variações conforme evolui o comportamento da Internet. Assim, com o objetivo de usar o resultado de [30] como ponto de partida para o projeto dos parâmetros do RED, deve-se notar que o modelo do sistema em [30], já aplicado as características do TCP, é da forma

$$\begin{aligned}\dot{x}(t) &= \sum_{i=0}^1 A_i x(t - \tau_i(t)) + Bu(t), \quad 0 \leq \tau_i(t) \leq h_i, \quad \dot{\tau}_i(t) \leq d_i, \\ z(t) &= Lx(t),\end{aligned}\tag{4.36}$$

onde $x(t) \in \mathbb{R}^n$ é o vetor de estado do sistema, $u(t) \in \mathbb{R}^q$ é a entrada de perturbação, $A_i \in \mathbb{R}^{n \times n}$ são as matrizes do sistema, $B \in \mathbb{R}^{n \times 1}$ é o vetor de perturbação, h_i é o limite superior do atraso $\tau_i(t)$, d_i é a variação máxima do atraso $\tau_i(t)$ e $z(t)$ é a variável de saída desejada com $L \in \mathbb{R}^{1 \times 2n}$.

Considerando o TCP linearizado em (3.23) cuja forma é

$$\begin{aligned}\delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t - R_0)) - \frac{1}{R_0^2 C}(\delta q(t) + \delta q(t - R_0)) - \frac{R_0^2 C^2}{2N^2} \delta p(t - R_0) \\ \delta\dot{q}(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) - \frac{C}{R_0} \delta R(t)\end{aligned}\tag{4.37}$$

juntamente com as equações do RED indicadas na figura 3.3, temos

$$KL_{red} \delta q = \delta \dot{p} + K \delta p\tag{4.38}$$

$$p(t - R_0) = L_{red} \mu(t - R_0).\tag{4.39}$$

⁴Na construção das LMIs, o expoente ‘T’ significa a matriz transposta, o elemento ‘*’ se deduz por simetria, $\mathbb{R}^{n \times m}$ é o espaço das matrizes reais de dimensão $n \times m$ e a notação $P > 0$, para $P \in \mathbb{R}^{n \times n}$ significa que P é simétrico e positivo definido.

Reescrevendo as equações anteriores, (4.37), (4.38) e (4.39), de forma mais compacta produz-se:

$$\delta\dot{x}(t) = A_0 \delta x(t) + A_1 \delta x(t - R_0) + B \delta R \quad (4.40)$$

onde

$$\mathbf{A}_0 = \begin{pmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} & 0 \\ 0 & KL_{red} & -K \end{pmatrix} \quad \mathbf{A}_1 = \begin{pmatrix} -\frac{N}{R_0^2 C} & \frac{1}{R_0^2 C} & -\frac{R_0 C^2}{2N^2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 0 \\ -\frac{C}{R_0} \\ 0 \end{pmatrix} \quad \mathbf{x}(t) = \begin{pmatrix} \delta w(t) \\ \delta q(t) \\ \delta p(t) \end{pmatrix}$$

Como se pode notar, as matrizes do sistema A_i e o vetor de perturbação B são exatamente conhecidos, ou seja, não possuem incertezas pois trabalha-se com o modelo linearizado do TCP.

A lei de controle que fará com que o sistema fique assintoticamente estável, para o caso do protocolo de transporte TCP, é o gerenciamento ativo da fila (AQM), representado nesta subsecção pelo controle RED. Como a função de descarte do RED é uma função não linear, é necessário considerar que o funcionamento do RED ocorre na região linear onde $p = \delta p + p_0$, pois as equações do modelo do TCP foram linearizadas. Utilizando esta definição e analisando novamente a figura 3.3, são produzidas as seguintes inequações:

$$\begin{aligned} \underline{\mu} &< \frac{p}{L_{red}} < \bar{\mu} \\ \underline{\mu}L_{red} - p_0 &< \delta p < \bar{\mu}L_{red} - p_0 \end{aligned} \quad (4.41)$$

as quais serão a base para a construção, posteriormente, das LMIs.

Finalmente é importante notar que (4.40) pode ser reescrito como em (4.36) com $\tau_0 = 0$ e $\tau_1 = R_0$. Assim, considera-se $d_1 = 0$.

4.4.2 Estabilidade

Para a análise da estabilidade considera-se $B = 0$, e representa-se (4.36) em uma forma equivalente como proposto em [30], o que produz:

$$\dot{x}(t) = y(t), \quad 0 = -y(t) + \left\{ \sum_{i=0}^1 A_i \right\} x(t) - \sum_{i=0}^1 A_i \int_{t-\tau_i(t)}^t y(s) ds \quad (4.42)$$

ou

$$E\dot{\bar{x}}(t) = \begin{bmatrix} \dot{x}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & I \\ \sum_{i=0}^1 A_i & -I \end{bmatrix} \bar{x}(t) - \sum_{i=0}^1 \begin{bmatrix} 0 \\ A_i \end{bmatrix} \int_{t-\tau_i(t)}^t y(s) ds, \quad (4.43)$$

com $\bar{x}(t) = \text{col}\{x(t), y(t)\}$ e $E = \text{diag}\{I, 0\}$, o seguinte funcional de Lyapunov-Krasovskii [38, 28] é aplicado :

$$V(t) = \bar{x}^T(t) E P \bar{x}(t) + V_2 + V_3 \quad (4.44)$$

onde

$$\mathbf{P} = \begin{bmatrix} P_1 & 0 \\ P_3 & P_4 \end{bmatrix}, \quad \mathbf{P}_1 > 0, \quad EP = P^T E \geq 0, \quad (4.45a-e)$$

$$V_2 = \sum_{i=1}^n \int_{-h_i}^0 \int_{t+\theta}^t y^T(s) R_i y(s) ds d\theta, \quad V_3 = \sum_{i=1}^n \int_{t-\tau_i(t)}^t x^T(\tau) S_i x(\tau) d\tau$$

Lema 4.4 (Estabilidade, [30]) *O sistema (4.36) será assintoticamente estável, com $B = 0$, se existirem matrizes $n \times n$ $0 < P, S_1, Y_1, Z_1$ e $R_1 > 0$, que satisfaçam as seguintes LMIs:*

$$\Gamma = \begin{bmatrix} \Psi & P^T \begin{bmatrix} 0 \\ A_1 \end{bmatrix} - Y_1^T \\ * & -S_1(1-d_1) \end{bmatrix} < 0 \quad e \quad \begin{bmatrix} R_1 & Y_1 \\ * & Z_1 \end{bmatrix} \geq 0, \quad (4.46a-b)$$

onde $R \in \mathbb{R}^{n \times n}$, $Z \in \mathbb{R}^{2n \times 2n}$, $Y \in \mathbb{R}^{n \times 2n}$, $S \in \mathbb{R}^{n \times n}$ e

$$\mathbf{Y}_1 = \begin{bmatrix} Y_{11} & Y_{12} \end{bmatrix}, \quad \mathbf{Z}_1 = \begin{bmatrix} Z_{11} & Z_{12} \\ * & Z_{13} \end{bmatrix} \quad (4.47a-c)$$

$$\begin{aligned} \Psi &= P^T \begin{bmatrix} 0 & I \\ A_0 & -I \end{bmatrix} + \begin{bmatrix} 0 & I \\ A_0 & -I \end{bmatrix}^T P + h_1 Z_1 + \begin{bmatrix} S_1 & 0 \\ 0 & h_1 R_1 \end{bmatrix} \\ &+ \begin{bmatrix} Y_1 \\ 0 \end{bmatrix} + \begin{bmatrix} Y_1 \\ 0 \end{bmatrix}^T \end{aligned}$$

□□□

Na primeira e segunda parcelas da equação (4.47c), encontra-se uma restrição na matriz P , a qual é mais detalhada na equação (4.48) a seguir :

$$P^T \begin{bmatrix} 0 & I \\ A_0 & -I \end{bmatrix} = \begin{bmatrix} P_1^T & P_3^T \\ 0 & P_4^T \end{bmatrix} \cdot \begin{bmatrix} 0 & I \\ A_0 & -I \end{bmatrix} = \begin{bmatrix} P_3^T A_0 & P_1^T - P_3^T \\ P_4^T A_0 & -P_4^T \end{bmatrix} \quad (4.48)$$

A multiplicação de P_3^T e P_4^T por A_0 causa restrições nas terceiras colunas das matrizes P_3^T e P_4^T . Isso acontece devido, conforme equação (4.40), a introdução do controle de malha fechada na matriz A_0 . Detalhando as multiplicações citadas acima, tem-se:

$$\begin{aligned} P_3^T A_0 &= \left[\alpha_1 \mid \alpha_2 \mid \alpha_3 \right] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \\ &= \alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 \begin{bmatrix} 0 & KL_{red} & -K \end{bmatrix} \end{aligned} \quad (4.49)$$

onde α_i e a_i , para $i = 1, 2, 3$, são as colunas P_3^T e as linhas de A_0 , respectivamente. Como a_1 e a_2 são constantes, α_1 e α_2 podem ser colunas livres de P_3^T . No entanto, a_3 possui os parâmetros K e L_{red} do RED a serem determinados e, por esse motivo, deve-se impor restrições em α_3 para que o produto $\alpha_3 a_3$ possa ser linearizado com mudanças de variáveis adequadamente escolhidas. Assim, restringi-se α_3 da forma

$$\alpha_3 = \begin{bmatrix} 0 & 0 & K_1 \end{bmatrix}^T, \quad (4.50)$$

com K_1 a determinar e escolhe-se as mudanças de variáveis

$$K_2 = KK_1 L_{red}, \quad K_3 = KK_1. \quad (4.51)$$

Dessa forma, o produto $\alpha_3 a_3$ fica linear em K_2 e K_3 . Além disso, uma vez calculados os valores de K_1, K_2 e K_3 obtém-se os valores de K e L_{red} pelas transformações da equação (4.51). Pelos mesmos motivos acima, a terceira coluna de P_4^T também deve ser igual a α_3 .

4.4.3 LMIs com Controle \mathcal{H}^∞

A norma \mathcal{H}^∞ do sistema (4.36) representa o maior ganho da sua resposta em frequência e pode também ser vista como o maior ganho em termos de energia que esse sistema consegue oferecer a um sinal de entrada de perturbação, qualquer que seja ele [62]. O sinal

de perturbação escolhido para o nosso modelo do TCP, conforme a subseção 3.3.2, foram as variações no atraso $R(t)$ em relação ao seu valor nominal R_0 . A razão de se considerar o atraso como variável de perturbação provém do fato que o *round-trip time* varia de pacote para pacote exercendo um fator determinante para o bom fluxo de dados. Desta forma, o controle \mathcal{H}^∞ é um problema de otimização que busca determinar uma lei de controle que minimize a influência da perturbação sobre a variável de saída $q(t)$, isto é, $y(t) = \delta q(t)$ em (4.40).

Uma maneira comumente usada para encontrar o valor da norma \mathcal{H}^∞ , consiste em se determinar numericamente um limitante superior γ para $\|G(s)\|_\infty$, isto é, busca-se um escalar positivo tal que:⁵

$$\min \gamma : \|G(s)\|_\infty . \quad (4.52)$$

O problema de encontrar esse limitante superior também é conhecido como problema \mathcal{H}^∞ sub-ótimo; o qual pode ser resolvido, por exemplo, pela equação de Riccati, pela matriz Hamiltoniana ou por LMI.⁶ Seguindo a derivação do BRL em [29, 30] produz-se:

Lema 4.5 (Condição \mathcal{H}^∞ , [30]) *A norma \mathcal{H}^∞ dos sistemas (4.36) é menor que γ se existirem matrizes $n \times n$ $0 < P, S_1, Y_1, Z_1$ e $R_1 > 0$, que satisfaçam (4.46b) e*

$$\mathbf{M}_1 = \begin{bmatrix} \Psi & P^T \begin{bmatrix} 0 \\ B \end{bmatrix} & P^T \begin{bmatrix} 0 \\ A_1 \end{bmatrix} & -Y_1^T & [L^T] \\ * & -\gamma & 0 & 0 & \\ * & * & -S_1(1-d_1) & 0 & \\ * & * & * & * & -\gamma \end{bmatrix} < 0 \quad (4.53)$$

onde Ψ é dado por (4.47c). □□□

Para incluir a relação $q(t) = z(t)$, se define

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 & -1 \end{bmatrix}}_{C_0} \xi_a = 0 , \quad (4.54)$$

onde $\xi_a = \text{col}\{\delta x(t), \delta y(t), \delta R_0(t), \delta x(t - R_0), \delta z(t)\}$. Aplicando o Lema de Finsler, o qual permite que restrições de igualdade sejam inseridas em (4.53), reduzindo assim seu

⁵ $G(s)$ representa a função de transferência do sistema no domínio da frequência.

⁶BOYD et al. [8] e TROFINO et al. [62] abordam esse assunto da determinação da norma \mathcal{H}^∞ com riqueza de detalhes. Portanto, para uma melhor compreensão da eficiência e das vantagens entre esses métodos usados no cálculo do limitante γ , a leitura dessas duas referências é de grande valia.

conservadorismo, pode-se transformar a condição

$$\xi_a^T M_1 \xi_a < 0 \quad \forall \xi_a \quad (4.55)$$

para

$$\xi_a^T \underbrace{\left[M_1 + HC_0 + C_0^T H^T \right]}_M \xi_a < 0 \quad \forall \xi_a, \quad (4.56)$$

onde H é uma matriz de ponderação a ser determinada. Com isto, ao invés de $M_1 < 0$ pode-se utilizar

$$M_1 + HC_0 + C_0^T H^T < 0, \quad (4.57)$$

que nunca será mais conservador que $M_1 < 0$.

Incorporando as inequações obtidas em (4.41), defini-se as constantes K_4 e K_5 a fim de calcular os valores das variáveis $\underline{\mu}$ e $\bar{\mu}$ da função de descarte do RED, de forma que:

$$\begin{aligned} \delta p < (\bar{\mu} L_{RED} - p_0) &= 1 / \sqrt{K_4} \\ \delta p > (\underline{\mu} L_{RED} - p_0) &= 1 / \sqrt{K_5} \end{aligned} \quad (4.58)$$

Representando a equação (4.58) na forma quadrática, tem-se:

$$K_4 \delta p^2 < 1 = \xi^T \left[\begin{array}{c|c} \bar{M} & 0 \\ \hline 0 & -1 \end{array} \right] \xi < 0 \quad (4.59)$$

com $\bar{M} = \text{diag}\{0_{2 \times 2}, K_4, 0_{8 \times 8}\}$, e

$$K_5 \delta p^2 < 1 = \xi^T \left[\begin{array}{c|c} \bar{\bar{M}} & 0 \\ \hline 0 & -1 \end{array} \right] \xi > 0 \quad (4.60)$$

com $\bar{\bar{M}} = \text{diag}\{0_{2 \times 2}, K_5, 0_{8 \times 8}\}$; sendo que $\xi = \text{col}\{\delta x(t), \delta y(t), \delta R_0(t), \delta x(t - R_0), \delta z(t), 1\} \in \mathbb{R}^{12 \times 12}$. Para obter uma única LMI, será usado o Procedimento-S o qual permite agrupar (4.56), (4.59) e (4.60), produzindo

$$\xi^T \left[\begin{array}{c|c} M & 0 \\ \hline 0 & 0 \end{array} \right] \xi - \xi^T \left[\begin{array}{c|c} \bar{M} & 0 \\ \hline 0 & -1 \end{array} \right] \xi + \xi^T \left[\begin{array}{c|c} \bar{\bar{M}} & 0 \\ \hline 0 & -1 \end{array} \right] \xi \leq 0 \quad (4.61)$$

Reescrevendo a soma da equação acima, obtém-se :

$$\xi^T \left[\begin{array}{c|c} M - \bar{M} + \bar{\bar{M}} & 0 \\ \hline 0 & 0 \end{array} \right] \xi \leq 0 \quad ; \quad K_5 > K_4 . \quad (4.62)$$

Desta forma, as LMIs para o modelo TCP aplicadas para o controle RED, ficam:

$$\left[M + \tilde{M} \right] \leq 0 \quad ; \quad \left[\begin{array}{cc} R_1 & Y_1 \\ * & Z_1 \end{array} \right] \geq 0 \quad ; \quad K_5 - K_4 > 0 . \quad (4.63)$$

onde $\tilde{M} = \text{diag}\{0_{2 \times 2}, K_5 - K_4, 0_{8 \times 8}\}$.

4.4.4 Cálculo do Controle C_{red}

Os detalhes da construção das LMIs da equação (4.63), implementadas no pacote LMI-TOOL do programa SCILAB, podem ser analisados no programa apresentado na seção A.1 do Apêndice A. Como já comentado no Lema 4.5, através desse conjunto de LMIs, busca-se minimizar o limitante superior γ da norma \mathcal{H}^∞ . Por conseguinte, o programa encontra iterativamente as matrizes: P , S_1 , Y_1 , Z_1 e R_1 , de forma a cumprir as condições impostas pelo Lema 4.5; K_1 , K_2 e K_3 , para satisfazer as equações (4.50) e (4.51); K_4 e K_5 , para atender as restrições da equação (4.58); e H , para a consideração feita na equação (4.56).

Para encontrar valores adequados de K e L_{red} , os seguintes limites extras são impostos às constantes K_1 , K_2 e K_3 :

$$\begin{aligned} K_{min} &< K_3 / K_1 < K_{max} \\ L_{red/min} &< K_2 / K_3 < L_{red/max} \end{aligned} \quad (4.64)$$

Da mesma forma, para que diferença entre o limite mínimo $\underline{\mu}$ e o limite máximo $\bar{\mu}$ esteja dentro de um valor desejado em relação ao tamanho da fila do roteador (*Buffer*) disponível, restringe-se K_4 e K_5 , produzindo

$$\begin{aligned} K_5 &> C_1 K_4 \\ K_5 &< C_2 K_4 \end{aligned} \quad ; \quad \text{para } C_1 > C_2 \text{ dados.} \quad (4.65)$$

Ainda com a mesma intenção, é necessário que K_5 sofra uma outra restrição; agora, para regular o valor da probabilidade máxima de descarte p_{max} . Nesta direção, com as

relações das equações (4.4) e (4.58) chega-se a:

$$\frac{1}{\sqrt{K_4}} - \frac{1}{\sqrt{K_5}} = p_{max} , \quad (4.66)$$

e como, em (4.65), $C_2 > 1$, então,

$$\sqrt{\frac{K_5}{K_4}} < \sqrt{C_2} . \quad (4.67)$$

Reestruturando a equação (4.66) com a consideração feita em (4.67), tem-se

$$p_{max} = \frac{1}{\sqrt{K_4}} - \frac{1}{\sqrt{K_5}} < \frac{\sqrt{C_2}}{\sqrt{K_5}} - \frac{1}{\sqrt{K_5}} < p_{sup} , \quad (4.68)$$

e, conseqüentemente,

$$\frac{\sqrt{C_2} - 1}{p_{sup}} < \sqrt{K_5} . \quad (4.69)$$

Desde que, a primeira parcela da equação acima seja maior que 1 (um), então, além dos limites da equação (4.65), K_5 também deve ser encontrado, respeitando a relação:

$$K_5 > \left(\frac{\sqrt{C_2} - 1}{p_{sup}} \right)^2 = C_3 , \quad \text{com } C_3 \text{ dado} . \quad (4.70)$$

Executando o programa com as LMIs aplicadas ao controle RED, obtém-se a configuração desejada dos parâmetros deste controle, para que o modelo TCP seja robusto e eficiente diante de possíveis perturbações da rede. A tabela 4.4 mostra os valores desses parâmetros, considerando os mesmos valores de p_{max} apresentados na tabela 4.2.

Tabela 4.4: Parâmetros do RED (via LMI) para valores diferentes de p_{max} .

p_{max}	α	K	L_{red}	$\underline{\mu}$	$\bar{\mu}$	Buffer	$\bar{\mu} - \underline{\mu}$
0.10	0.00000148	0.0053519	0.0001533	130	795	900	665
0.05	0.00000140	0.0053593	0.0001535	102	478	900	376
0.15	0.00000150	0.0055034	0.0001542	239	1239	1300	1000

Comparação 4.1: Comparando as tabelas 4.2 e 4.4 nota-se que os valores de α , K e L_{red} não são exatamente os mesmos. Isso acontece porque no cálculo através do diagrama

de bode, α , K e L_{red} são obtidos da equação de estabilidade (4.9), sendo que, o valor de $K=0.005$ é escolhido de acordo com os requisitos de desempenho desejados. Já quando o cálculo é feito com LMIs, os valores das variáveis α , K e L_{red} são obtidos dentro dos limites (4.64) os quais são escolhidos para que os valores das referidas sejam equivalentes aos conseguidos com o outro método. Assim, como o método das LMIs encontra as variáveis citadas anteriormente respeitando limites ao contrário do método de Bode onde são utilizadas fórmulas, é natural que não se encontrem exatamente os mesmos valores para α , K e L_{red} e, sim, valores equivalentes.

Os valores diferentes de $\bar{\mu}$ e $\underline{\mu}$ são uma conseqüência do que já foi comentado. Contudo, é importante focar sobre a configuração destes valores. O ajuste ótimo para $\underline{\mu}$ e $\bar{\mu}$ depende da relação que se deseja entre um baixo atraso médio de fila e uma alta utilização do enlace disponível. Por exemplo, quanto mais congestionado está o tráfego de dados, maior tem que ser o valor de $\underline{\mu}$ para alcançar o valor desejado de utilização do enlace [24].

4.4.5 Construção das LMIs para o Controle PI

A estrutura mostrada na figura 4.8 será utilizada para substituir o controle RED pelo controle PI na execução do gerenciamento ativo de fila (AQM). Diferentemente do RED, a estrutura em cascata do PI possui um elemento de ganho não linear apenas para que o valor da probabilidade, calculado pela função de transferência, fique limitado entre 0 e 1. Contudo, esta saturação não é considerada durante o projeto das LMIs para o controle PI, sendo implementada somente no simulador de redes NS-2 onde a performance de cada controlador será avaliada. Conseqüentemente, analisando a figura 4.8 produz-se :

$$\delta\dot{p} = \frac{K_{PI}}{z_c} \delta\dot{q} + K_{PI} \delta q. \quad (4.71)$$

Reescrevendo a equação (3.23), que descreve o comportamento linearizado do tamanho da fila, tem-se

$$\delta\dot{q}(t) = \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) + \Theta \quad (4.72)$$

onde a entrada de perturbação Θ é escolhida de forma que

$$\Theta = \begin{cases} 0 & \rightarrow \text{para } \delta W \text{ como perturbação,} \\ -\frac{C}{R_0} \delta R(t) & \rightarrow \text{para } \delta R \text{ como perturbação.} \end{cases}$$

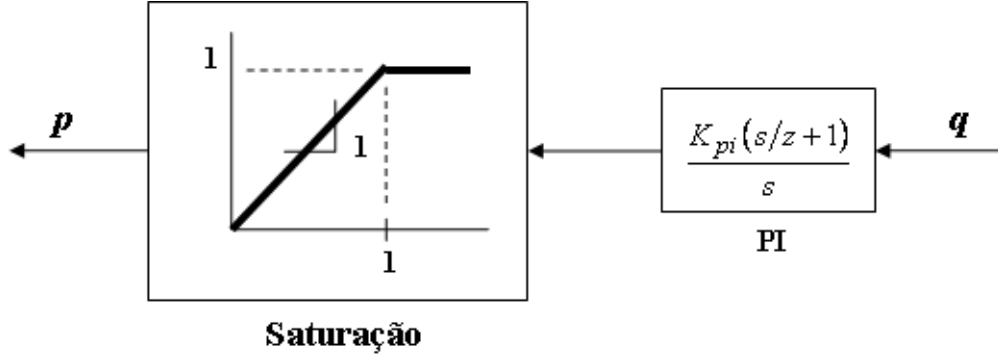


Figura 4.8: Função de transferência do PI em cascata com função de descarte.

Substituindo (4.71) em (4.72) chega-se a equação (4.73), a qual descreve o comportamento da probabilidade de descarte com o controle PI.

$$\delta \dot{p}(t) = \frac{K_{PI}}{z_c} \left(\frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) + \Theta \right) + K_{PI} \delta q(t) \quad (4.73)$$

ou

$$\delta \dot{p}(t) = \frac{NK_{PI}}{z_c R_0} \delta W(t) + \left(K_{PI} - \frac{K_{PI}}{z_c R_0} \right) \delta q(t) + \Theta \quad (4.74)$$

Como no projeto das LMIs para o RED, serão consideradas as equações linearizadas na subseção 3.3.2 e o sistema linear com atraso descrito na equação (4.40) juntamente com a dinâmica do controlador PI indicada acima, obtém-se :

$$\delta \dot{x}(t) = A_0 \delta x(t) + A_1 \delta x(t - R_0) + B \delta R$$

onde

$$\mathbf{A}_0 = \begin{pmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} & 0 \\ \frac{NK_{PI}}{z_c R_0} & K_{PI} - \frac{K_{PI}}{z_c R_0} & 0 \end{pmatrix} \quad \mathbf{A}_1 = \begin{pmatrix} -\frac{N}{R_0^2 C} & \frac{1}{R_0^2 C} & -\frac{R_0 C^2}{2N^2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \mathbf{B}_W = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{B} = \mathbf{B}_R = \begin{pmatrix} 0 \\ -\frac{C}{R_0} \\ -\frac{CK_{PI}}{z_c R_0} \end{pmatrix}$$

O vetor B_W é usado quando se considera δW como variável de perturbação⁷, enquanto que o vetor B_R é referente a perturbações provenientes de δR . Note que o controle de malha fechada da equação 4.74 já está inserido nas matrizes do sistema.

A estabilidade é verificada novamente através do Lema 4.4 e, como anteriormente, a inserção do controle de malha fechada na matriz A_0 causa restrições na matriz P ; mais especificamente nas terceiras colunas das matrizes P_3^T e P_4^T , quando estas são multiplicadas por A_0 . Reescrevendo a equação 4.49, agora para o controle PI, produz-se:

$$\begin{aligned} P_3^T A_0 &= \left[\alpha_1 \mid \alpha_2 \mid \alpha_3 \right] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \\ &= \alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 \left[\frac{NK_{PI}}{z_c R_0} \quad K_{PI} - \frac{K_{PI}}{z_c R_0} \quad 0 \right] \end{aligned} \quad (4.75)$$

onde α_i e a_i , para $i = 1, 2, 3$, são as colunas de P_3^T e as linhas de A_0 , respectivamente. Como a_1 e a_2 são constantes, α_1 e α_2 podem ser colunas livres de P_3^T . No entanto, a_3 possui os parâmetros K_{PI} e z_c do PI a serem determinados e, por esse motivo, deve-se impor restrições em α_3 para que o produto $\alpha_3 a_3$ possa ser linearizado com mudanças de variáveis adequadamente escolhidas. Assim, restringi-se α_3 da forma

$$\alpha_3 = \left[0 \quad 0 \quad K_1 \right]^T, \quad (4.76)$$

com K_1 a determinar e escolhe-se as mudanças de variáveis

$$K_2 = K_1 K_{PI}, \quad K_3 = \frac{K_1 K_{PI}}{z_c}. \quad (4.77)$$

Dessa forma, como para o RED, o produto $\alpha_3 a_3$ fica linear em K_2 e K_3 . Além disso, uma vez calculados os valores de K_1 , K_2 e K_3 obtém-se os valores de K_{PI} e z_c pelas transformações da equação (4.77). Pelos mesmos motivos acima, a terceira coluna de P_4^T deve também ser escolhida igual a α_3 .

Como a saturação do controle PI não é considerada no projeto, somente as LMIs provenientes do Lema 4.5, serão utilizadas para alcançar a norma \mathcal{H}^∞ menor que γ ; incluindo, é claro, a consideração $q(t) = z(t)$ feita na equação 4.54. Assim, seguindo o raciocínio da subseção 4.4.3 as LMIs para o modelo TCP aplicadas ao controle PI, e que serão implemen-

⁷A opção de usar δW como variável de perturbação é considerada apenas aqui na construção das LMIs, pois, como já comentado, tanto as simulações como o modelo do protocolo TCP são desenvolvidos tendo somente δR como sinal de perturbação.

tadas no pacote LMITOOL do programa SCILAB, ficam:

$$\underbrace{\begin{bmatrix} M_1 + HC_0 + C_0^T H^T \end{bmatrix}}_M < 0 \quad ; \quad \begin{bmatrix} R_1 & Y_1 \\ * & Z_1 \end{bmatrix} \geq 0. \quad (4.78)$$

4.4.6 Cálculo do Controle C_{pi}

De forma semelhante à seção 4.4.4, as LMIs da equação (4.78) cuja implementação no LMITOOL está detalhada na seção A.2, do Apêndice A, buscam minimizar o limitante γ da norma \mathcal{H}^∞ . Para o controle PI, devem ser determinadas as matrizes: P , S_1 , Y_1 , Z_1 e R_1 , para novamente cumprir as condições impostas pelo Lema 4.5; , para satisfazer as equações (4.76) e (4.77); e H , para a consideração feita na equação (4.56). Observe que, as definições das constantes K_2 e K_3 para as LMIs do controle PI obedecem as relações apresentadas pela equação (4.77), as quais são diferentes das relações usadas por essas constantes nas LMIs do controle RED. Tornando a impor limites extras em K_1 , K_2 e K_3 , com o objetivo de que o zero z_c e a constante K_{pi} do controle PI possuam valores viáveis, chega-se a

$$\begin{aligned} z_{c/min} &< K_2/K_1 < z_{c/max} \\ K_{pi/min} &< K_2/K_3 < K_{pi/max} \end{aligned} \quad (4.79)$$

Executando o programa com as LMIs aplicadas ao controle PI, obtém-se as configurações dos parâmetros desse controlador mostradas na tabela 4.5, considerando os mesmos valores da frequência de amostragem f_a usados na tabela 4.3.

Tabela 4.5: Valores de a e b (via LMI) para diferentes frequências f_a .

f_a (Hz)	z_c	K_{pi}	a	b
160	0.31	0.0000081	0.000025683	0.000025632
3750	0.31	0.0000081	0.000025658	0.000025656
160	0.71	0.0000083	0.000011726	0.000011674
3750	0.71	0.0000083	0.000011701	0.000011699

Comparação 4.1: Na tabela 4.3, definiu-se o valor do zero z_c igual a menor frequência de corte da planta $P(s)$. Já na tabela 4.5, os dois valores este zero foram obtidos através

da manipulação dos limites (4.79). Perceba que, com LMIs, o zero z_c não é imposto ou definido diretamente, mas calculado de forma a obter o valor ótimo entre os limites citados anteriormente. Observe, também, que quanto maior for z_c e a constante K_{pi} , mais rápida será a resposta do sistema, sendo que, uma constante de tempo muito alta pode resultar em indesejáveis oscilações no tamanho médio da fila.

4.5 COMENTÁRIOS

Tanto no cálculo do controle RED (subseção 4.4.4) como no cálculo do controle PI (subseção 4.4.6) foram impostos limites às constantes K_1 , K_2 e K_3 . Contudo, estes limites não foram desenvolvidos para que as LMIs das equações 4.63 e 4.78 tivessem solução, mas sim, para que os valores dos parâmetros dos controladores RED e PI calculados através do método de desigualdades matriciais lineares fossem equivalentes ou próximos aos valores obtidos na seção 4.2 com o uso do método clássico. Como pode ser analisado nos resultados das tabelas 4.2, 4.3, 4.4 e 4.5.

Entretanto, o critério de estabilidade proposto por FRIDMAN et. al. [30] pode sofrer outras alterações que objetivem melhorar o seu desempenho em relação ao método clássico quando se está trabalhando com sistemas com atraso, uma vez que, com o uso desse critério pode-se considerar o comportamento das variações do atraso já no projeto dos controladores, fato que não é possível com o diagrama de Bode. Outro fator favorável das LMIs é a maior quantidade de graus de liberdade os quais permitem uma maior flexibilidade na configuração dos controladores.

Capítulo 5

EXPERIMENTOS COM CONTROLADORES RED E PI

Para analisar o desempenho dos métodos utilizados no projeto dos controladores RED e PI utilizou-se o simulador de redes NS-2 [51] o qual é baseado em eventos discretos e orientado a objetos, sendo que, as linguagens de programação usadas são: o C++, para o núcleo do simulador; e o Otcl (*Object Tool Control Language* — Linguagem de Scripts Orientada a Objetos) para a construção dos *scripts* e modelagem da simulação. O NS-2 simula as condições estocásticas e não lineares de uma rede real qualquer ; o que torna a análise dos resultados obtidos, com os métodos desenvolvidos nas seções anteriores, muito mais confiável, mesmo não os implementando em computadores e roteadores reais .

Através deste simulador é possível reproduzir com riqueza de detalhes o funcionamento dos mais variados tipos de aplicações, de protocolos, de redes, de elementos da rede e de modelos de fluxo de dados. A biblioteca do NS-2 provê, por exemplo a implementação : dos protocolos TCP e UDP, da camada de transporte; do protocolo IP, da camada de rede; dos protocolos HTTP, SMTP e FTP, da camada de aplicação; de algoritmos de roteamento multicast como o PIM (*Protocol Independent Multicast* — Multicast independente de protocolo) e o DVMRP (*Distance Vector Multicast Routing Protocol* — Protocolo de Roteamento Multicast por Vetor de Distância) para fluxos do tipo CBR e VBR. Tecnologias de rede sem fio como o GPRS (*General Packet Radio Service* — Padrão de Transmissão de Dados por Pacotes) e Bluetooth, assim como, protocolos para redes móveis (como o IP móvel) e mecanismos de gerenciamento de fila (como o RED e o Drop-Tail), também são suportados pelo NS-2. Outras informações sobre este simulador de redes podem ser encontradas em [16, 21, 32, 35].

5.1 CENÁRIO DE SIMULAÇÃO

A topologia de rede mostrada na figura 5.1 será usada para as simulações. Ela consiste de três fontes (F_1 , F_2 e F_3) de envio de dados, um roteador e somente um destino. A política de gerenciamento de fila do roteador, naturalmente, alterna entre o RED e o PI conforme o experimento a ser realizado. Definindo que o tamanho de cada pacote é igual a 500 bytes, então, a capacidade de processamento do roteador $C = 3750$ pacotes/seg, a qual foi usada para a determinação dos controladores, é equivalente a 15Mbps; valor configurado no NS-2. Da mesma forma, com base no tempo de viagem de ida e volta $R_0 = 0.246$ segundos, o atraso de propagação T_p de cada enlace foi configurado de forma que: para os enlaces entre as fontes (F_1 , F_2 e F_3) e roteador, $T_{pf} = 20ms$; e para o enlace de engarrafamento entre o roteador e o destino, $T_{pe} = 100ms$.

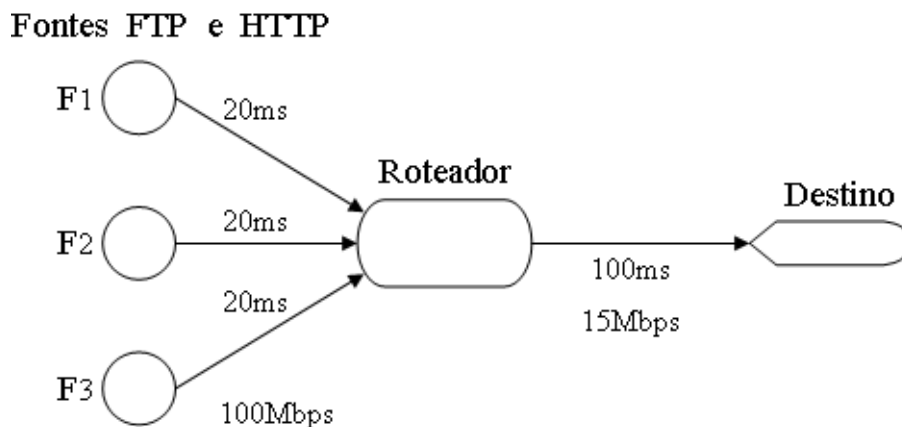


Figura 5.1: Topologia de uma rede simples.

O fator de carga N , modelado no capítulo 3, relata fluxos de dados TCP de longa duração tais como transferências de dados utilizando o protocolo FTP. Todavia, levando em consideração que a maioria dos fluxos que passam pela Internet são HTTP, no cenário de simulação, fluxos de dados FTP e HTTP são misturados, garantindo que a simulação seja a mais real possível. Os fluxos HTTP possuem a característica de terem curta duração e um grande tamanho; por isso, provocam “ruídos” na fila do roteador. Apesar de que, os controladores não foram projetados considerando as “rajadas” de dados dos fluxos de curta duração, eles também são capazes de manter o tamanho da fila do roteador e uma boa utilização da capacidade do enlace mesmo na presença deste tipo de fluxo.

As simulações serão realizadas através de vários experimentos os quais avaliarão o comportamento dos controladores em relação a diferentes aspectos como variações no RTT e aumento no fator de carga.

5.2 SIMULAÇÕES PARA O CONTROLE RED

As simulações para verificar a performance dos dois métodos de projeto, LMI e Bode, desenvolvidos nos capítulos 4, são divididas em duas etapas: a primeira, que se estende desde o experimento 1 até o experimento 6, avalia apenas o controle RED; e a segunda, que se prolonga do experimento 7 ao experimento 11, preocupa-se apenas com o controle PI.

Para as simulações com o RED, o tamanho do buffer do roteador alterna entre 900 e 1300 pacotes, conforme os parâmetros do controle RED obtidos nas tabelas 4.2 e 4.4, respectivamente, para o projeto através do diagrama de Bode e com o uso de LMIs.

5.2.1 Experimento 1: Evolução da Fila para Diferentes Valores de p_{max}

Nesta primeira simulação, o fluxo de dados no roteador é composto por 60 sessões FTP, mais 180 sessões HTTP. Não há nenhuma variação nos parâmetros da rede durante a simulação, que tem um tempo de duração de 200s. Os resultados são traçados nas figuras 5.2, 5.3 e 5.4, sendo que, nas duas primeiras figuras o buffer tem capacidade para 900 pacotes e, na última, a capacidade é aumentada para 1300 pacotes. Nas três simulações o sistema é estável, com flutuações em torno do ponto de operação da fila, para ambos os métodos de projeto do RED. Estas flutuações estão relacionadas a presença dos fluxos HTTP que e-

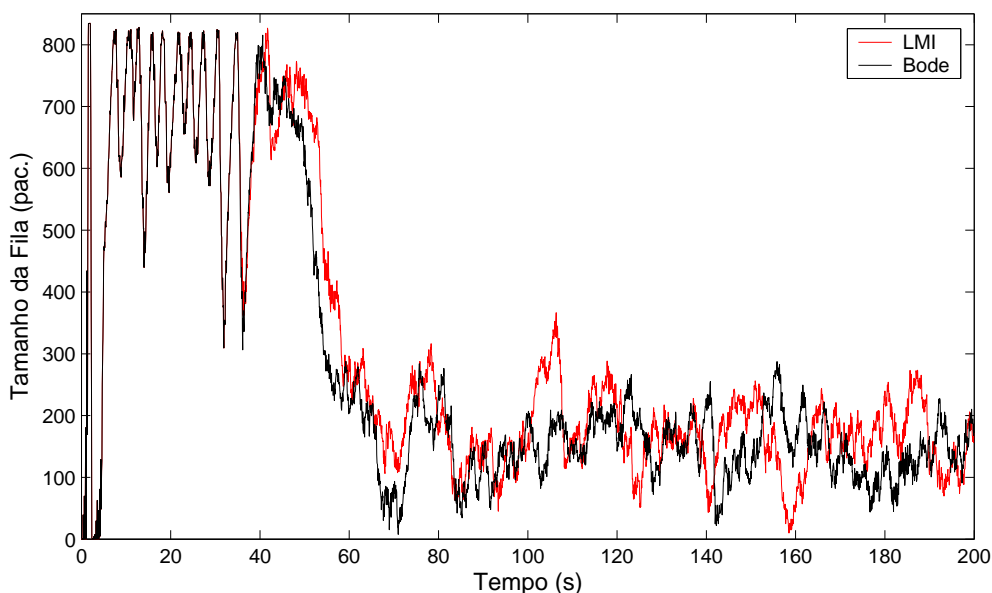


Figura 5.2: Comportamento da fila sem variações na rede, $p_{max} = 0.05$ e buffer = 900.

xercem maior influência quando o buffer não é grande o suficiente; fato que é comprovado comparando-se os gráficos da figura 5.4 com os gráficos das figuras 5.2 e 5.3.

Na figura 5.2, onde é considerado uma probabilidade máxima de descarte de 5%, o tempo de resposta e o tamanho de fila médio das duas curvas são, respectivamente, cerca de 60 segundos e 170 pacotes. Este valor médio significa que o atraso médio de fila sofrido pelos pacotes é igual a 49ms. Durante a simulação foram descartados 6433 pacotes para o RED/LMI e 6265 pacotes para o RED/Bode. Quando a probabilidade de descarte é au-

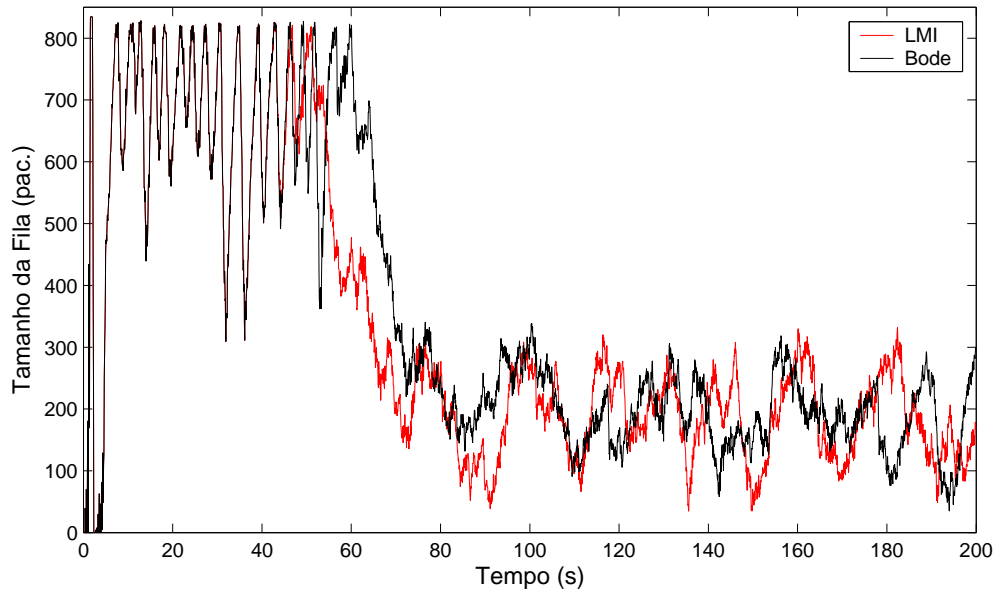


Figura 5.3: Comportamento da fila sem variações na rede, $p_{max} = 0.10$ e $buffer = 900$.

mentada para 10%, curvas da figura 5.3, o número de pacotes descartados pelo RED/LMI aumenta para 6492 e pelo RED/Bode para 6330. O ponto de operação passa para 200 pacotes, o que era esperado devido ao aumento do limite mínimo $\underline{\mu}$ nas configurações de ambos os métodos de projeto, LMI e Bode. Com isso o tempo médio de permanência dos pacotes na fila também aumenta para 58ms. A resposta do sistema fica entre 70s e 80s.

Os gráficos da figura 5.4 servem para comprovar o quão decisivo é o tamanho do buffer do roteador no gerenciamento ativo de fila. Além de influir nas oscilações, como comentado anteriormente, o tamanho do buffer também interfere no número de pacotes descartados. Nessa simulação, onde a probabilidade de descarte é de 15%¹ e o buffer possui capacidade para 1300 pacotes, o RED/LMI descartou 6076 pacotes enquanto que o RED/Bode descartou 6198 pacotes; valores ligeiramente menores do que os obtidos quando a probabilidade de descarte foi menor, porém com um buffer de 900 pacotes.

¹Para FLOYD [24], não há necessidade de se configurar a probabilidade máxima de descarte p_{max} maior que 10%. Ela ainda complementa relatando que, se um roteador está operando com uma taxa de descarte entre 20% e 30%, então, existe algo de errado na engenharia da rede, e não é esta região que deve ser otimizada. Outros trabalhos [42, 43, 44] também cultuam essa consideração, sendo que, inclusive o simulador de redes NS-2 utiliza $p_{max}=0.1$ como patamar padrão.

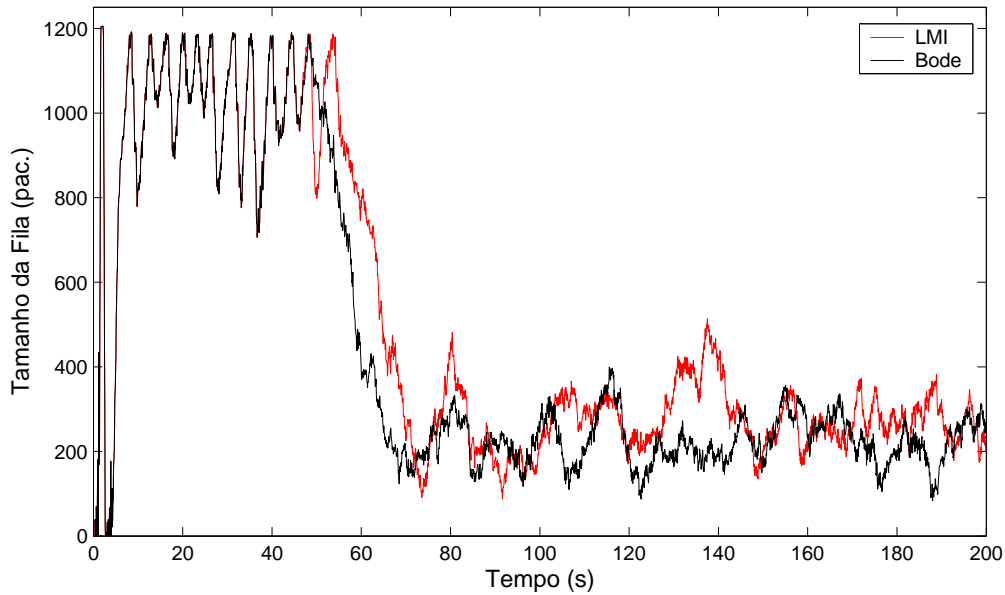


Figura 5.4: Comportamento da fila sem variações na rede, $p_{max} = 0.15$ e $buffer = 1300$.

Observação 5.1: A resposta lenta obtida pelo RED está diretamente relacionada com a inclinação L_{red} e o pólo K . O tempo de resposta diminui se: K for aumentado (o que resulta em um aumento no fator de ponderação α) e L_{red} mantido o mesmo; ou se o valor de L_{red} for aumentado e valor de K permanecer inalterado.

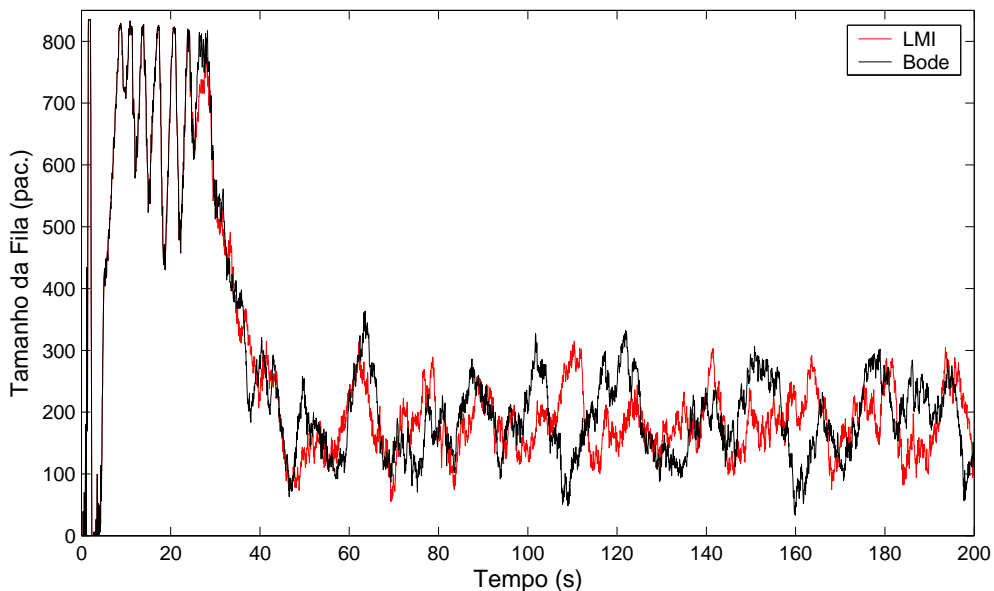


Figura 5.5: Comportamento da fila para $K = 0.011$.

O mesmo cenário utilizado para obter a figura 5.3 é usado para traçar as curvas da

figura 5.5, porém, agora, o valor do pólo K do filtro é alterado de 0.005 para 0.011. Como observa-se, o tempo de resposta de 45 segundos, tanto para o RED/LMI quanto para o RED/Bode, é significativamente menor que o tempo de 60 segundos alcançado na figura anterior. Entretanto, deve-se cuidar ao projetar o filtro passa-baixa do RED com um pólo de valor muito elevado, pois, desta maneira, o RED calcula valores praticamente instantâneos da fila ao invés de valores médios.

5.2.2 Experimento 2: Variações no RTT

Neste experimento, a configuração para $p_{max}=0.1$ apresentada anteriormente é avaliada quando a dinâmica da rede apresenta variações no atraso, ou seja, quando o tempo de propagação ou o RTT não são fixos. Dessa forma, durante a simulação, o atraso de propagação varia uniformemente entre 90ms e 120ms, o que é equivalente ao RTT variar entre 180ms e 240ms. O comportamento do tamanho da fila é mostrado na figura 5.6. Para essa situação de rede o RED/LMI descartou 6550 pacotes enquanto que o RED/Bode descartou 6669 pacotes. Comparando-se estes valores com os obtidos na figura 5.3, conclui-se que na presença de variações no RTT o RED/LMI possui um melhor desempenho, contudo, o número de descartes aumenta em ambas configurações.

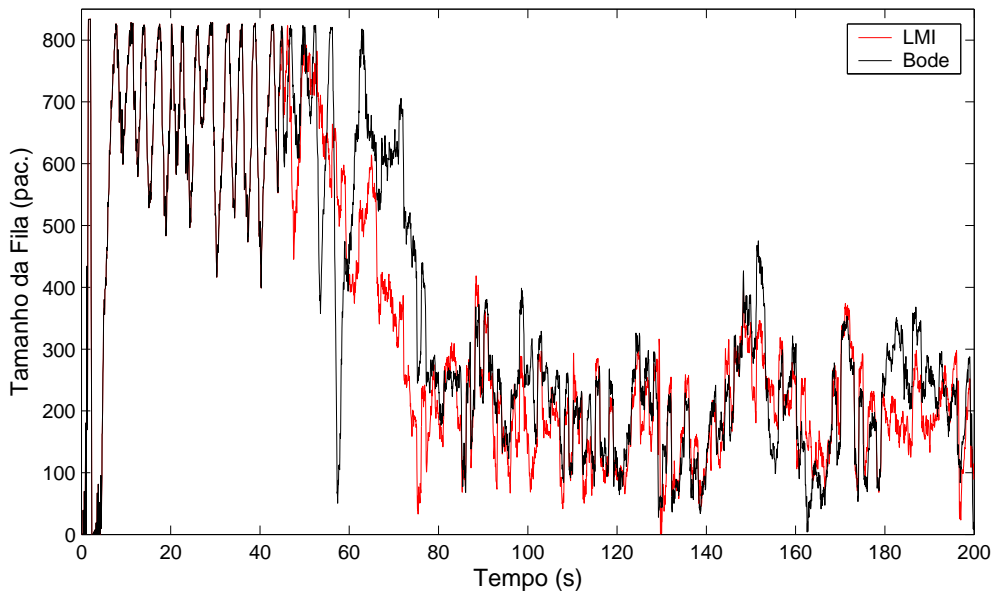


Figura 5.6: Comportamento da fila para variações no RTT, $p_{max} = 0.10$ e $buffer = 900$.

5.2.3 Experimento 3: Variações na Carga e no RTT

Para verificar a velocidade de resposta dos controladores diante de uma interrupção brusca de uma quantidade volumosa de conexões, 30 fluxos FTP são adicionados ao cenário da figura 5.6. Em $t = 90s$, esses 30 fluxos são cortados, e em $t = 120s$ eles são retomados. Como mostra a figura 5.7, com o incremento da quantidade de fluxos, o ponto de operação da fila naturalmente aumenta; estabilizando-se entre 250 e 300 pacotes, após $t = 170s$. O RED/LMI e o RED/Bode, aumentaram o número de descarte de pacotes, respectivamente, para 9886 e 10099.

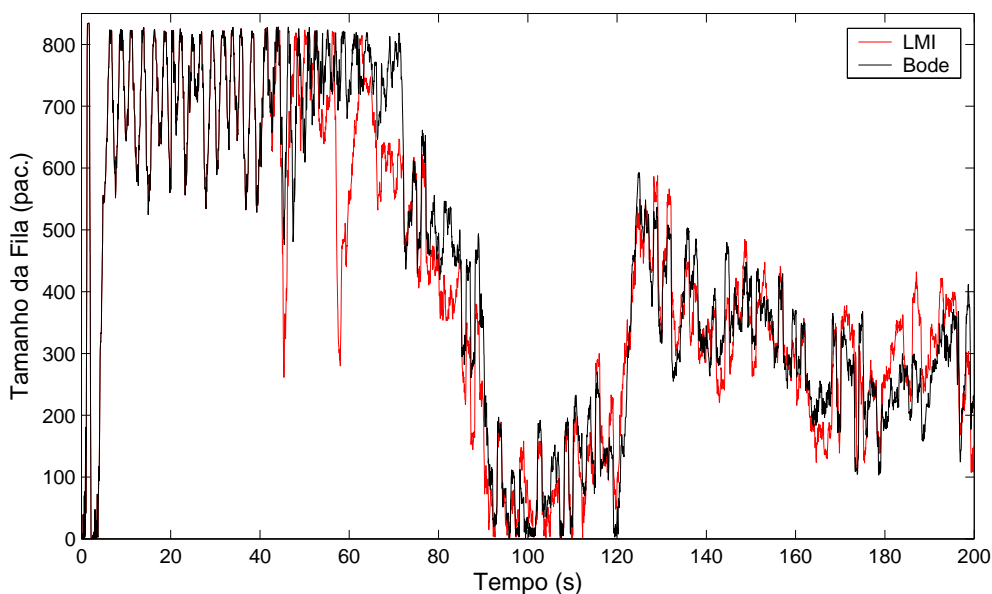


Figura 5.7: Comportamento da fila para variações no RTT e na carga N, buffer = 900.

5.2.4 Experimento 4: Evolução da Fila para um Baixo RTT

Com o mesmo cenário utilizado para traçar as curvas da figura 5.3 no experimento 1, porém, agora, com o RTT fixo e igual a 100ms, consegue-se analisar o desempenho para um baixo tempo de viagem de ida e volta dos pacotes, uma vez que, na figura 5.3 o RTT foi igual a 240ms. A figura 5.8 mostra o resultado, e não deixa dúvidas sobre a lentidão no tempo de resposta de ambos os controladores.

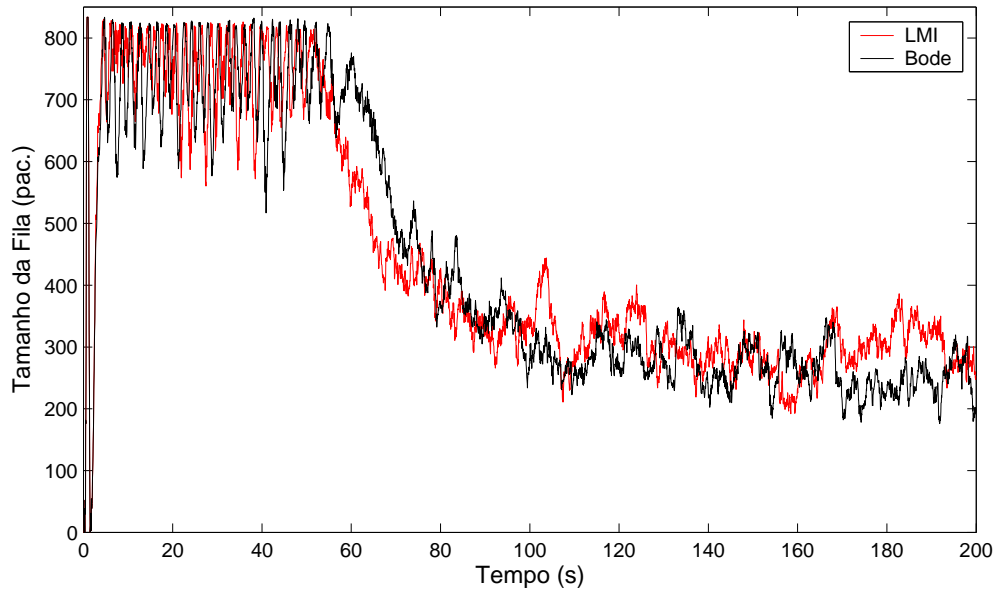


Figura 5.8: Comportamento da fila para $p_{max} = 0.10$, $buffer = 900$ e $RTT = 100ms$.

5.2.5 Experimento 5: Tráfego Intenso de Fluxos FTP e HTTP

Aumentando o número dos fluxos FTP para 150 e dos fluxos HTTP para 300, obtêm-se os gráficos da figura 5.9. Nota-se, claramente, que o desempenho do RED piora muito nestas condições; o que confirma a grande influência que o fator de carga N exerce sobre o RED. O número de pacotes descartados pelo RED/LMI foi de 20008 e pelo RED/Bode foi de 20750.

5.2.6 Experimento 6: Mudança no Ponto de Equilíbrio

Este experimento submete os controles RED/LMI e RED/Bode a uma mudança no ponto de equilíbrio da fila. Para isto, em $t = 120s$, o número de fluxos FTP é aumentado para 120. Por consequência, o ponto de equilíbrio passa de 200 pacotes para 300 pacotes e o atraso de fila é aumentado de 58ms para 86ms. O filtro passa-baixa é configurado com o pólo $K = 0.011$ para proporcionar uma melhor visualização das curvas referentes ao tamanhos da fila. Na figura 5.10 pode-se observar a evolução do ponto de operação da fila, sendo que, o roteador descartou 10739 pacotes com o controle RED/LMI e 10952 com o controle RED/Bode. Já ficou evidente nos outros experimentos que conforme o número de fluxos, tanto o RED/LMI como o RED/Bode mudam o seu ponto de operação. Este experimento, portanto, serve para deixar mais claro como isso acontece.

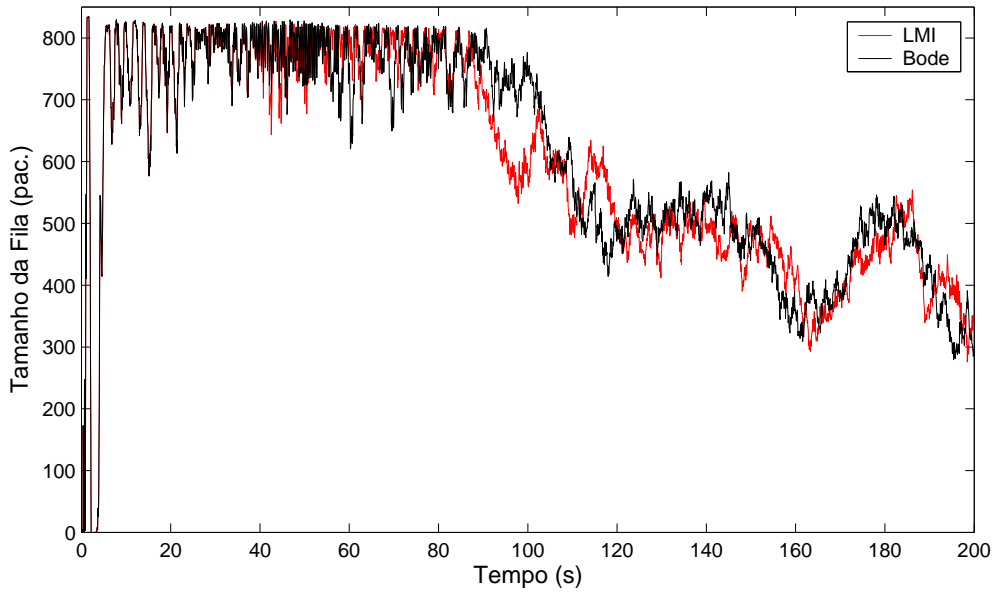


Figura 5.9: Comportamento da fila para $N = 450$, $p_{max} = 0.10$ e $buffer = 900$.

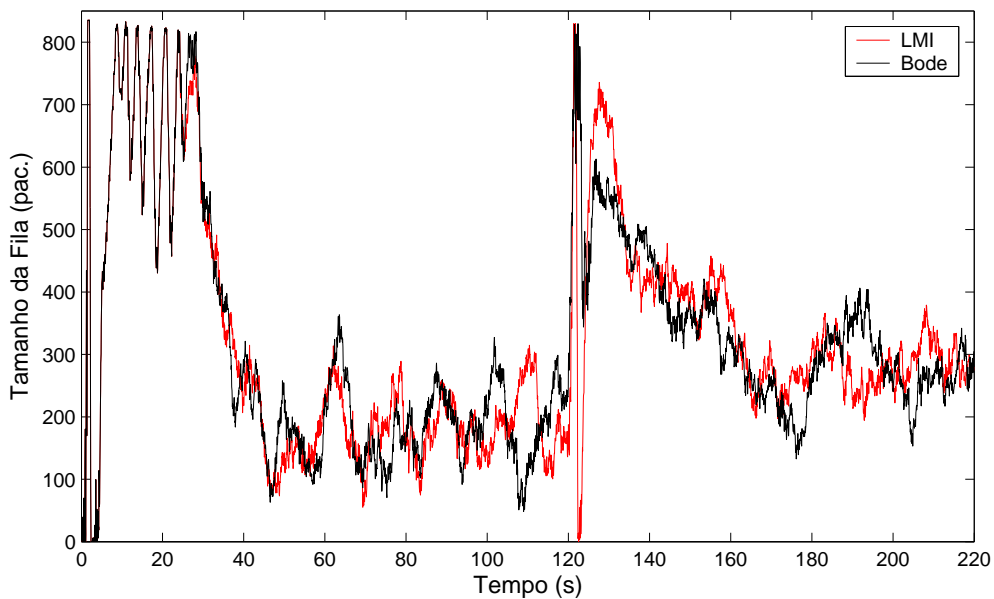


Figura 5.10: Comportamento da fila para mudança do ponto de operação.

5.3 SIMULAÇÕES PARA O CONTROLE PI

Basicamente, os mesmos cenários de simulação utilizados durante os experimentos do RED, na seção 5.2, serão simulados para o controle PI. Todas as simulações são feitas com frequência de amostragem $f_a = 160\text{Hz}$, como um buffer de 900 pacotes e com o tamanho de fila desejado q_{ref} (ponto de operação) regulado para 200 pacotes. Se ao longo das simu-

lações algum desses parâmetros sofrer alteração, a mesma será indicada no experimento em questão.

5.3.1 Experimento 7: Comportamento do PI sem Variações na Rede

Para esta primeira simulação como o controle PI, o mesmo cenário do experimento 1, da seção 5.2, é usada. Assim, a rede não tem variações e a carga de fluxos que passa através do roteador é composta de 60 sessões FTP, mais 180 sessões HTTP. O resultado pode ser conferido na figura 5.11, a qual revela como o PI possui uma resposta mais rápida, assim como, um ponto de operação mais facilmente ajustado do que o RED, entretanto, como nas simulações com o RED, os desempenhos do PI/LMI e do PI/Bode são equivalentes. Os descartes de pacotes diminuem em relação a mesma simulação feita no experimento 1: 5898 pacotes, para o PI/LMI; e 6011 pacotes, para o PI/Bode.

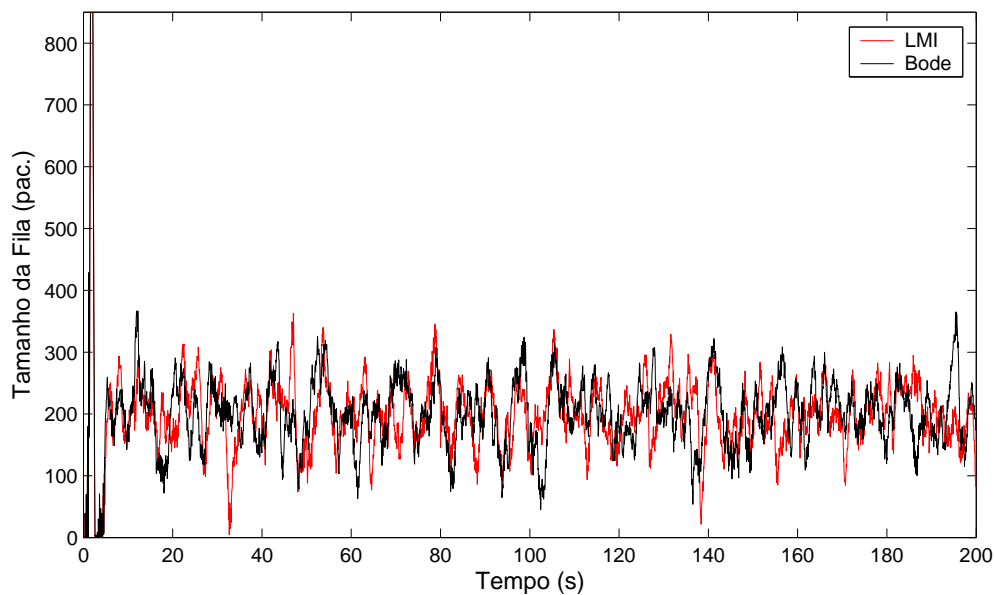


Figura 5.11: Evolução do tamanho da fila sem variações na rede, $q_{ref} = 200$ e $buffer = 900$.

5.3.2 Experimento 8: Variações no RTT

Neste experimento, o cenário a ser reutilizado é o do experimento 2. Note, nos gráficos da figura 5.12, que as variações no atraso de propagação entre 90ms e 120ms deixam o comportamento da fila mais oscilatório, o que faz tanto o RED/LMI quanto o RED/Bode descartarem mais pacotes; respectivamente, 6643 pacotes e 6769 pacotes.

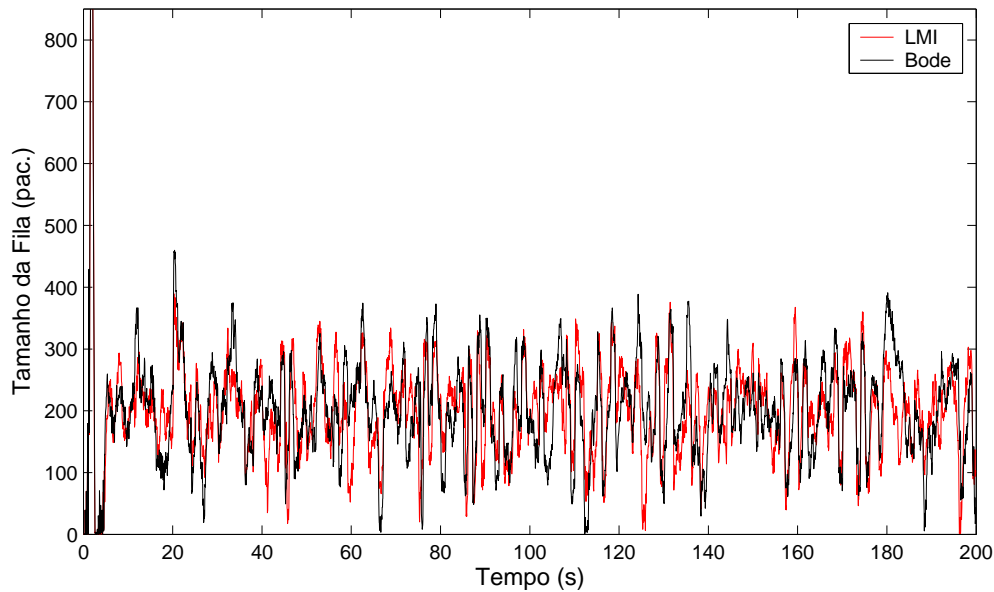


Figura 5.12: Evolução do tamanho da fila para variações no RTT entre 180ms e 240ms.

5.3.3 Experimento 9: Variações na Carga e no RTT

Seguindo o mesmo roteiro do experimento 3, eleva-se o número de fluxos FTP para 120 para verificar a velocidade de resposta do controle PI quando ocorrer quedas bruscas de grandes quantidades de sessões TCP. Assim, em $t=90s$, metade desses 120 fluxos FTP são cortados, e em $t=120s$ eles são retomados. As variações no atraso de propagação da simulação anterior são mantidos, isto é, variam entre 90ms e 120ms. Para uma melhor visualização o ponto de operação é ajustado para 300 pacotes. Como mostra a figura 5.13, ambos os controles PI/LMI e PI/Bode se recuperam de forma muito mais satisfatória do que o RED para essa possível situação de rede.

5.3.4 Experimento 10: Tráfego Intenso de Fluxos com um Baixo RTT

Neste experimento, avalia-se os controles PI em relação a combinação de dois extremos: fluxos FTP muito elevados; e um baixo valor de RTT. Para tal propósito, os fluxos FTP são fixados em 400 e o RTT em 50ms. Em $t=80s$, são retiradas 60 sessões FTP para, também, melhor observar o tempo de resposta. As curvas da figura 5.14 não deixam dúvidas sobre o quanto a performance dos controladores é afetada com essa situação extrema dos parâmetros da rede. O PI/LMI e o PI/Bode descartam, respectivamente, 92375 pacotes e 94843 pacotes; o que é um patamar inaceitável de descarte.

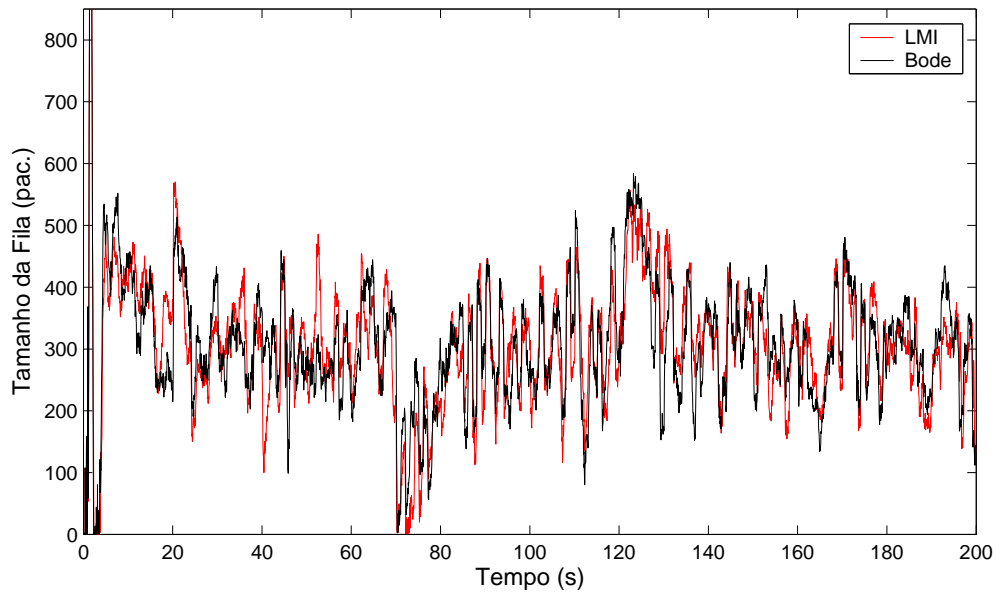


Figura 5.13: Evolução do tamanho da fila para variações no RTT e na carga N, buffer = 900.

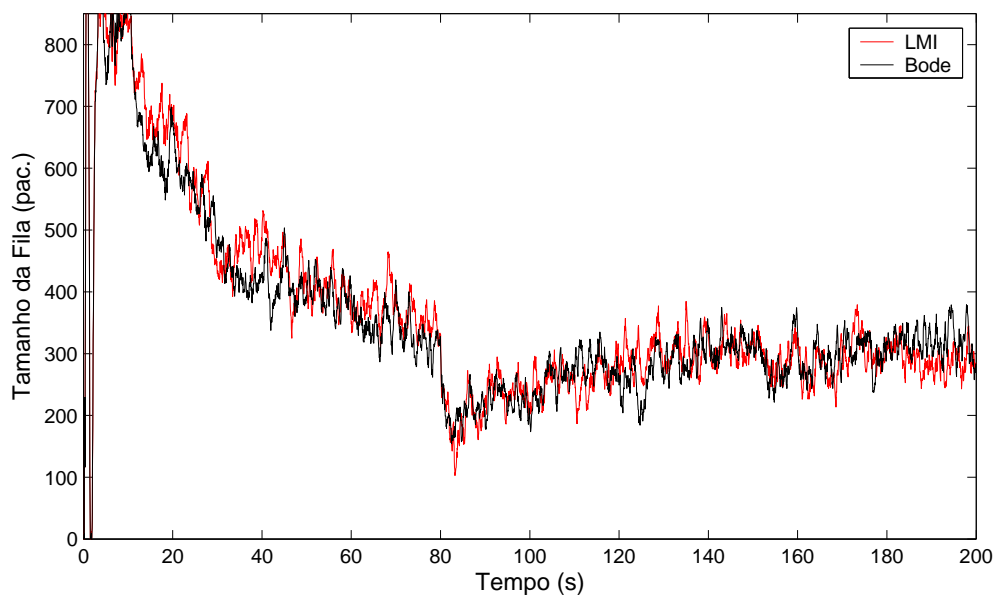


Figura 5.14: Evolução do tamanho da fila para N = 580 e RTT = 50ms.

5.3.5 Experimento 11: Variações na Largura de Banda

A largura de banda do enlace no qual os dados estão sendo transmitidos exerce fator determinante para o bom fluxo da comunicação. Para demonstrar isso, na figura 5.15, a da largura de banda do enlace de engarrafamento foi variada, de forma que, ela seja: de 15Mb, entre 0 e 79s; de 30Mb, entre 80 e 139s; e de 5Mb, entre 140 e 200s. Fica claro que, o melhor comportamento da fila acontece na largura de banda para a qual os controladores

foram projetados, isto é, 15Mb.

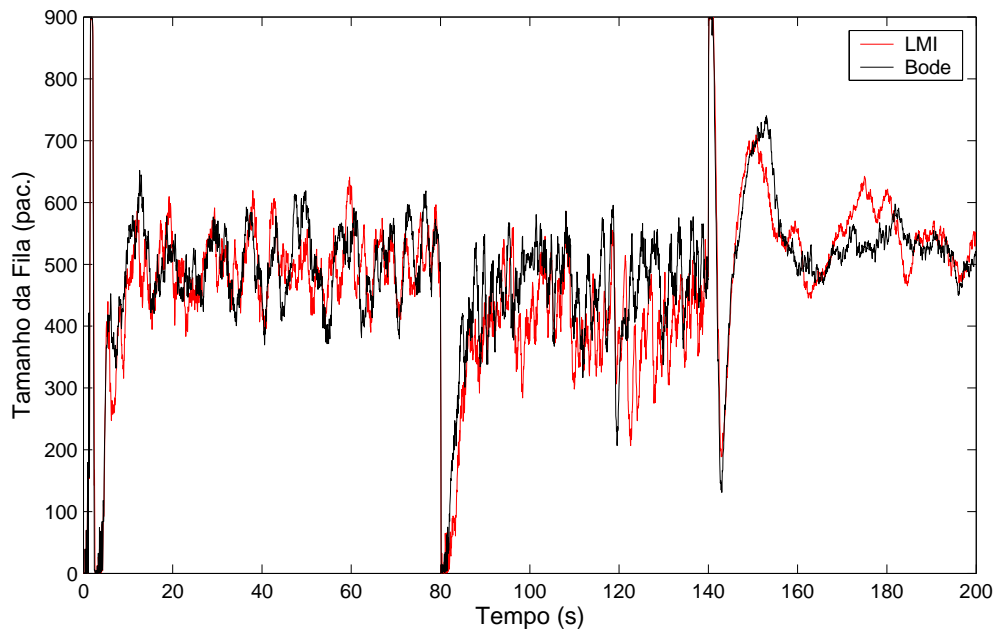


Figura 5.15: Evolução do tamanho da fila para: $C = 15\text{Mb}$, entre 0 e 79s; $C = 30\text{Mb}$, entre 80 e 139s; e $C = 5\text{Mb}$, entre 140 e 200s.

5.4 COMENTÁRIOS

Em todos os experimentos realizados os controladores projetados com LMIs tiveram desempenhos semelhantes aos controladores projetados com o método clássico (via diagrama de Bode). É válido lembrar que esses experimentos foram realizados tendo como cenário situações reais de uma rede de computadores. Os desempenhos demonstram que os critérios de estabilidade para sistemas com atraso propostos pelo projeto com desigualdade matriciais lineares são realmente eficazes e podem ser destinados para um projeto de uma rede real. Para que os controladores projetados com desigualdades matriciais tenham um melhor desempenho é necessário, como comentado na seção 4.5, que as LMIs propostas FRIDMAN et. al. [30] sejam melhor trabalhadas ou desenvolvidas.

Uma importante questão deve ser levada em consideração quando se desenvolve sistemas para o gerenciamento ativo de fila (AQM) é a relação entre o ponto de operação da fila e a utilização do enlace de engarrafamento. Nessa relação, a análise do ponto de operação da fila sempre nos conduz a outro relevante fato; o atraso de fila. Dependendo do valor médio da fila, tem-se diferentes valores para o tempo que os pacotes terão que esperar na fila até serem processados; é o caso, por exemplo, das figuras 5.2 e 5.3.

Nos experimentos, realizados anteriormente tendo como ponto de operação o valor de 200 pacotes, a utilização do enlace de engarrafamento variou: entre 75% a 85%, quando foram utilizados fluxos FTP e HTTP; e entre 90% e 97%, quando foram utilizados somente fluxos FTP. Estes dados abrangem as simulações dos controles PI e RED. Para o controle PI, grandes valores de q_{ref} produzem um atraso de fila maior e, também, uma maior utilização do enlace. Já no controle RED, como já foi demonstrado, o nível da fila depende diretamente de fatores como: probabilidade de descarte, intensidade do tráfego da rede, limite mínimo $\underline{\mu}$ e limite máximo $\bar{\mu}$.

Com respeito, a $\underline{\mu}$ e $\bar{\mu}$, a utilização do enlace tende a ficar mais penalizada a medida que a diferença entre esses dois limites decai. Outro fator importante é o tamanho da janela do emissor. Verificando-se um exemplo específico, observa-se que a utilização conseguida pelo experimento 7 foi de 77% da capacidade do enlace. Ajustando-se a janela dos três emissores para um terço do valor anterior, a utilização cai para 35%, sensivelmente menor. MISRA et al. [48] fazem um pequeno estudo o qual detalha como o RED e o PI mantêm o relacionamento entre o atraso de fila e a utilização do enlace de engarrafamento.

Capítulo 6

CONCLUSÕES

O enfoque principal desta dissertação foi buscar uma alternativa para se determinar os parâmetros dos controles RED e PI, quando estes são aplicados para executarem o gerenciamento ativo de fila (AQM) de um rede qualquer. Para conseguir esse objetivo foram utilizados o modelo linearizado do protocolo transporte TCP e a metodologia de controle ótimo \mathcal{H}^∞ para sistemas com atraso [30]. As condições necessárias para a determinação das configurações de ambos controles foram expressas em termos de LMIs, que, quando satisfeitas, garantem a estabilidade assintótica do sistema.

O estudo comparativo, realizado no capítulo 5, entre esse novo método e a técnica de projeto através da resposta em frequência, a qual provém da Teoria de Controle Clássico [50, 52] e vem sendo estudada e aplicada com muito sucesso na engenharia de redes de computadores, mostra resultados equivalentes. Ou seja, os métodos de projeto possuem desempenho muito semelhante mesmo na presença de variações no tempo de viagem de ida e volta (RTT), no número de fluxos que passam pelo roteador e na capacidade de transmissão do enlace de engarrafamento. É válido ressaltar que, ambos controladores (RED e PI) foram eficazes no controle do tamanho médio da fila; porém, como ficou evidente, o controle PI possui um melhor desempenho do que o RED.

O modelo do protocolo TCP da camada de transporte da Internet mostrou-se apropriado, pois possibilitou capturar a dinâmica desse protocolo de forma satisfatória, contribuindo para um bom projeto dos sistemas de controle. A adição dos mecanismos de *timeout* e *slow start* nas equações do capítulo 3 podem tornar o modelo mais realista. Embora, através do projeto com LMIs, seja possível considerar as possíveis variações do atraso (jitter) da rede durante o cálculo do controlador, essa vantagem não pôde ser confirmada com clareza nas simulações; mesmo com os controladores desenvolvidos com LMIs descartando menos pacotes do que os controladores projetados com o diagrama de Bode.

Todos os cálculos com LMIs foram feitos usando-se a ferramenta LMITOOL do SCI-LAB. Este *software* demonstrou ser bastante sensível aos dos limites extras C_1 , C_2 e C_3 , os quais foram impostos às constantes K_1 , K_2 e K_3 , no cálculo tanto do controle RED como do controle PI. Pequenas variações nesses limites, em certos casos, levaram a resultados bem diferentes dos anteriores, o que dificultou o ajuste. Este fato motiva a pensar em outras formas de introduzir esses limites .

Conforme já salientado durante este trabalho, diversas pesquisas e o próprio mercado têm evidenciado a importância do controle de congestionamento dos diversos fluxos de dados; os quais crescem com enorme velocidade. No entanto, verificam-se poucos estudos cujo problema do gerenciamento ativo de fila (AQM) seja tratado com desigualdades matriciais lineares (LMIs) que lidam com sistemas em atraso e esse, no entender do autor, foi o grande mérito deste trabalho de Mestrado. A sua contribuição, portanto, está na proposição, através de LMIs, de mecanismos os quais atenuam os efeitos negativos que as constantes variações dos parâmetros de uma rede causam para a própria rede, como um todo, e especificamente entre os roteadores e os sistemas finais.

Discussões e trabalhos interessantes podem ser obtidos com a continuação da linha pesquisa apresentada nesta dissertação, dentre eles estão:

- ✓ Aplicar os controles propostos para o gerenciamento ativo de fila em uma rede real, ou seja, criar uma rede de computadores locais (LAN) para analisar o fluxo de dados, o tamanho da fila e a vazão de dados nos roteadores ;
- ✓ Estender as leis de controle para um cenário com múltiplos enlaces de engarrafamento ;
- ✓ Implementar o controle \mathcal{H}^∞ , para sistemas com atraso, diretamente no gerenciamento ativo de fila. Para isso, deve-se elaborar, no simulador de redes NS-2, um algoritmo específico para esse controle ;
- ✓ Considerar algumas incertezas nos parâmetros do sistema, por exemplo, através da abordagem politópica e não apenas com o valor nominal; o fluxo N seria um bom ponto de partida ;
- ✓ Considerar o buffer como um sistema chaveado, isto é, conforme o nível de congestionamento e o tipo de fluxo ao qual o roteador está sendo submetido, a lei de controle AQM muda para se adequar a nova situação de rede ;
- ✓ Aplicar outra estrutura de controle .

Apêndice A

IMPLEMENTAÇÃO DAS LMIs

A.1 Algoritmo para Resolução das LMIs do Controle RED

```
function[H0,gama,K1,K2,K3,K4,K5,P1,P31,P41,S1,Y1,Z1,R1] = tcp(A01,A1,B)
//////////Generated by lmitool on
Mbound = 1e3;
abstol = 1e-10;
nu = 10;
maxiters = 100;
reitol = 1e-10;
options = [Mbound, abstol, nu, maxiters, reitol];
//////////DEFINE INITIAL GUESS AND PRELIMINARY CALCULATIONS BELOW
C0 = [0 1 0 0 0 0 0 0 0 0 -1];
H0_init = eye(11,1);
P32 = [0 0 1];
P42 = [0 0 1];
P31_init = eye(2,3);
P41_init = eye(2,3);
gama_init = 1;
K1_init = 1;
K2_init = 1;
K3_init = 1;
K4_init = 1;
K5_init = 1;
P1_init = eye(3,3);
S1_init = eye(3,3);
Y1_init = eye(3,6);
Z1_init = eye(6,6);
R1_init = eye(3,3);
//////////
XLIST0 = list(H0_init, gama_init, K1_init, K2_init, K3_init, K4_init, K5_init, P1_init, P31_init,
P41_init, S1_init, Y1_init, Z1_init, R1_init)
XLIST = lmsolver(XLIST0, tcp_eval, options)
[H0, gama, K1, K2, K3, K4, K5, P1, P31, P41, S1, Y1, Z1, R1] = XLIST(:)
```

```

endfunction
//////////EVALUATION FUNCTION//////////
function [LME,LMI,OBJ] = tcp_eval(XLIST)
[H0, gama, K1, K2, K3, K4, K5, P1, P31, P41, S1, Y1, Z1, R1] = XLIST(:)
//////////DEFINE LME, LMI and OBJ BELOW
P3 = [P31;K1*P32];
P4 = [P41;K1*P42];
P = [P1 zeros(3,3);P3 P4];
LME = list()
LME(1) = P1-P1';
LME(2) = R1-R1';
LME(3) = S1-S1';
LME(4) = Z1-Z1';
LMI = list()
LMI(1) = K1;
LMI(2) = K2;
LMI(3) = K3;
LMI(4) = K4;
LMI(5) = K5-c2;
LMI(6) = K3-K1*K_min;
LMI(7) = -K3+K1*K_max;
LMI(8) = K2-K3*Lred_min;
LMI(9) = -K2+K3*Lred_max;
LMI(10) = K5-c0*K4;
LMI(11) = -K5+c1*K4;
LMI(11) = [R1 Y1;Y1' Z1];
LMI(12) = -[[P32'*[0 K2 -K3]+P31'*A01 P1'-P3';P42'*[0 K2 -K3]+P41'*A01 -P4'] + [[0 K2 -
K3]'*P32+A01'*P31 [0 K2 -K3]'*P42+A01'*P41;P1-P3 -P4] + h1*Z1 + [S1 zeros(3,3);zeros(3,3)
h1*R1] + [Y1;zeros(3,6)] + [Y1;zeros(3,6)]' P'*[zeros(3,1);B] (P'*[zeros(3,3);A1]-Y1)' [0 1 0 0 0
0]';(P'*[zeros(3,1);B])' -gama*eye(1,1) zeros(1,3) zeros(1,1);(P'*[zeros(3,3);A1]-Y1)' zeros(3,1) -
S1*(1-d) zeros(3,1);[0 1 0 0 0 0] zeros(1,1) zeros(1,3) -gama*eye(1,1)] + [H0*C0] +[C0'*H0]' -
[zeros(2,11);0 0 K5-K4 zeros(1,8);zeros(8,11)];
OBJ = gama
endfunction

```

A.2 Algoritmo para Resolução das LMIs do Controle PI

```

function[H0, gama, K1, K2, K3, P1, P31, P41, S1, Y1, Z1, R1] = tcpPI(A01, A1, N, R0, d)
//////////Generated by lmitool on
Mbound = 1e3;
abstol = 1e-10;
nu = 10;
maxiters = 100;
reitol = 1e-10;
options=[Mbound, abstol, nu, maxiters, reitol];
//////////DEFINE INITIAL GUESS AND PRELIMINARY CALCULATIONS BELOW
C0 = [0 1 0 0 0 0 0 0 0 0 -1];
H0_init = eye(11,1);
P31_init = eye(2,3);
P41_init = eye(2,3);
gama_init = 10;
K1_init = 1000;
K2_init = 500;
K3_init = 10;
P1_init = 20*ones(3,3);
S1_init = 2*ones(3,3);
Y1_init = 7*ones(3,6);
Z1_init = ones(6,6);
R1_init = ones(3,3);
//////////
XLIST0 = list(H0_init, gama_init, K1_init, K2_init, K3_init, P1_init, P31_init, P41_init, S1_init,
Y1_init, Z1_init, R1_init)
XLIST = lmsolver(XLIST0, tcpPI_eval, options)
[H0, gama, K1, K2, K3, P1, P31, P41, S1, Y1, Z1, R1] = XLIST(:)
endfunction
//////////EVALUATION FUNCTION//////////
function [LME,LMI,OBJ] = tcpPI_eval(XLIST)
[H0, gama, K1, K2, K3, P1, P31, P41, S1, Y1, Z1, R1] = XLIST(:)
//////////DEFINE LME, LMI and OBJ BELOW
P32 = [0 0 1];
P42 = [0 0 1];
B = [0;-C*R0;1];
P3 = [P31;K1*P32];
P4 = [P41;K1*P42];
P = [P1 zeros(3,3);P3 P4];
LME = list()
LME(1) = P1-P1';
LME(2) = R1-R1';
LME(3) = S1-S1';
LME(4) = Z1-Z1';
LMI = list()

```

```

LMI(1) = K1;
LMI(2) = K2;
LMI(3) = K3;
LMI(4) = K2 - K1*Kpi_min;
LMI(5) = Kpi_max*K1 - K2;
LMI(6) = K2 - K3*zc_min;
LMI(7) = zc_max*K3 - K2;
LMI(8) = [R1 Y1; Y1 Z1];
LMI(9) = - [[P32'*[K3*N/R0 K2-K3/R0 0] + P31'*A01 P1'-P3'; P42'*[K3*N/R0 K2-K3/R0 0] +
P41'*A01 -P4'] + [[K3*N/R0 K2-K3/R0 0]'*P32 + A01'*P31 [K3*N/R0 K2-K3/R0 0]'*P42 +
A01'*P41; P1-P3 -P4] + h1*Z1 + [S1 zeros(3,3); zeros(3,3) h1*R1] + [Y1; zeros(3,6)] +
[Y1; zeros(3,6)]' P'*[zeros(3,1); B] P'*[zeros(3,3); A1]-Y1' [0 1 0 0 0 0]'; (P'*[zeros(3,1); B])'
-gama*eye(1,1) zeros(1,3) zeros(1,1); (P'*[zeros(3,3); A1]-Y1)' zeros(3,1) -S1*(1-d) zeros(3,1);
[0 1 0 0 0 0] zeros(1,1) zeros(1,3) -gama*eye(1,1)] + [H0*C0] + [C0'*H0'];
OBJ = gama;
endfunction

```


Referências Bibliográficas

- [1] ÅSTRÖM, K. J.; WITTENMARK, B. *Computer Controlled Systems: Theory and Design*. Prentice-Hall, 1984.
- [2] ALTMAN, E.; HELLAL, O. A. Analysis of TCP Vegas and TCP Reno. In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS (June 1997 : Montreal, Canadá). *Proceedings*. Montreal: 1988. p. 495–499.
- [3] AT&T. Killer Apps. Disponível em <<http://www.att.com>>. Acesso em: Agosto de 2005.
- [4] AZIMI-SADJADI, B. Stability of networked control systems in the presense of packet losses. In: IEEE / CONFERENCE ON DECISION AND CONTROL (42. : Dec. 2003: Maui, Hawaii, USA). *Proceedings*. Hawaii: 2003. p. 676–681.
- [5] BARBOSA, K. A. *Projeto de filtros robustos para sistemas lineares e não lineares*. Florianópolis, 2003. Tese (Doutorado em Engenharia Elétrica) - Centro Tecnológico, Universidade Federal de Santa Catarina.
- [6] BEAN, S. E. P. *Abordagens LMI para análise de uma classe de sistemas híbridos*. Florianópolis, 2002. Tese (Doutorado em Engenharia Elétrica) - Centro Tecnológico, Universidade Federal de Santa Catarina.
- [7] BOLOT, J. C.; TURLETTI, T.; WAKEMAN, I. Scalable Feedback Control for Multicast Video Distribution in the Internet. In: ACM/SIGCOMM (Aug. 1994 : London, UK). *Proceedings*. London: 1994. p. 58–67.
- [8] BOYD, S.; GHAOUI, L. E.; BALAKRISHNAN, V. *Linear Matrix Inequalities in Systems and Control Theory*. SIAM Frontier Series, 1994.
- [9] BRADEN, B.; CLARK, D. et al. Recommendations on Queue Management and Congestion Avoidance in the Internet: RFC 2309. Disponível em <<http://www.ietf.cnri.reston.va.us/rfc/rfc2309.txt>>. Acesso em: Abril de 2005.

- [10] BRAKMO, L.; O'MALLEY, S.; PETERSON, L. TCP Vegas: New techniques for congestion detection and avoidance. In: SIGCOMM (Aug. 1994 : London, UK). *Proceedings*. London: 1994. p. 24–35.
- [11] BRAKMO, L.; PETERSON, L. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communication*, v. **25**, n. 5, p. 69–86, Oct. 1995.
- [12] CASSANDRAS, C. G. The Poisson Counting Process. In:_____. *Discrete Event Systems: Modeling and Performance Analysis*. Massachusetts: IRWIN, 1993. p. 207–218.
- [13] CHEN, C. T. *Linear System Theory and Design*. 3. ed. New York : Oxford University, 1999.
- [14] CHIU, D.; JAIN, R. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, v. **17**, n. 1, p. 1–14.
- [15] CHRISTIANSEN, M.; JEFFAY, D. O.; SMITH, F. Tuning Red for Web Traffic. In: ACM/SIGCOMM (Aug. 2000 : Stockholm, Sweden). *Proceedings*. Stockholm: 2000. p. 139–150.
- [16] CHUNG, J.; CLAYPOOL, M. *NS by Example*. Disponível em <<http://nile.wpi.edu/NS>>. Acesso em: Março de 2005.
- [17] COUTINHO, D. F. *Análise de estabilidade e desempenho, e síntese de controle para sistemas não lineares incertos*. Florianópolis, 2003. Tese (Doutorado em Engenharia Elétrica) - Centro Tecnológico, Universidade Federal de Santa Catarina.
- [18] de OLIVEIRA, M. C.; SKELTON, R. Stability Tests for Constrained Linear Systems. In: *S. O. Reza Moheimani (.ed), Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2001. p. 241–257.
- [19] FIELDING, R; GETTYS, J.; MOGUL, J. et al. Hypertext Transfer Protocol: HTTP/1.1. RFC 2616. Disponível em <<http://www.rfc-editor.org/rfc/rfc2616.txt>>. Acesso em: Outubro de 2005.
- [20] FALL, K.; FLOYD, S. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *Computer Communication Review*, v. 26, n. 3, p. 5–21, June 1996.

- [21] FALL, K.; VARADHAN K. *The NS Manual*. Disponível em <<http://www.isi.edu/nsnam/ns/doc/index.html>>. Acesso em: Março de 2005.
- [22] FLOYD, S. Conexões com Múltiplos Gateways Congestionados em Redes de Pacote-Comutadas : Tráfego Unidirecional. *Computer Communication Review*, v. **21**, n. 5, p. 30–47, Oct.1991.
- [23] FLOYD, S. RED: Discussões de Modos de Byte e Pacote. Março, 1997. Disponível em <[www://icir.org/floyd/REDaveraging.txt](http://www.icir.org/floyd/REDaveraging.txt)>. Acesso em: Maio de 2005.
- [24] FLOYD, S. RED: Discussões de Parâmetros de Configuração. Novembro, 1997. Disponível em <[www://icir.org/floyd/REDparameters.txt](http://www.icir.org/floyd/REDparameters.txt)>. Acesso em: Maio de 2005.
- [25] FLOYD, S. TCP e Notificação Explícita de Congestão. *ACM Computer Communication Review*, v. **24**, n. 5, p. 10–23, Oct.1994.
- [26] FLOYD, S.; FALL, K. Promovendo o Uso de Controle de Congestão de Ponta-a-Ponta para a Internet. *IEEE/ACM Transactions on Networking*, v. **7**, n. 4, p. 458–472, Aug. 1999.
- [27] FLOYD, S.; JACOBSON, V. Detecção Antecipada de Gateways para Evitar Congestão. *IEEE/ACM Transactions on Networking*, v. **1**, n. 4, p. 397–413, Aug. 1993.
- [28] FRIDMAN, E. Novos funcionais de Liapunov-Krasovskii para estabilidade de sistemas lineares retardados e neutros. *Systems & Control Letters*, v. **43**, n. 4, p. 309–319, 2001.
- [29] FRIDMAN, E.; SHAKED, U. Estabilidade dependente de atraso e controle H_∞ : atrasos constantes e variáveis no tempo. *International Journal of Control*, v. **76**, n. 1, p. 48–60, 2003.
- [30] FRIDMAN, E.; SHAKED, U. Um método melhorado de estabilização para sistemas de atraso de tempo. *IEEE/Transactions on Automatic Control*, v. **47**, n. 11, p. 1931–1937, Nov. 2002.
- [31] GUAOUI, L. El; NICULESCU, S. *Advances in Linear Matrix Inequalities Methods in Control*. SIAM, 2000.
- [32] GREIS, M. *Tutorial for the Network Simulator NS*. Disponível em <<http://www.isi.edu/nsnam/ns/tutorial/index.html>>. Acesso em: Março de 2005.
- [33] JACOBSON, V. Congestion Avoidance and Control. In: ACM/SIGCOMM (Aug. 1988 : Stanford, USA). *Proceedings*. 1988. p. 314–329.

- [34] JACOBSON, V. Modified TCP Congestion Avoidance Algorithm. Technical report, 30 Apr. 1990. Disponível em <ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt. Acesso em: Abril de 2005.
- [35] JIMÉNEZ, T.; ALTMAN, E. *NS Simulator for Beginners*. Disponível em <http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf. Acesso em: Março de 2005.
- [36] JOHARI, R; TAN, D. K. H. End-to-End Congestion Control for the Internet: Delays and Stability. *IEEE/ACM Transactions on Networking*, v. **9**, n. 6, p. 818–832, Dec. 2001.
- [37] KAMIENSKI, C. A.; SADOK, D.; CAVALCANTI, D. A. T. et al. Simulando a Internet: Aplicações na pesquisa e no ensino. In: JAI/CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (21. : Jun. 2002 : Florianópolis). *Anais*. Florianópolis: 2002.
- [38] KOLMANOVSKII, V.; NICULESCU, S. I.; RICHARD, J. P. On the Liapunov-Krasovskii functionals for stability analysis of linear delay systems. *International Journal of Control*, v. **72**, n. 4, p. 374–384, Mar. 1999.
- [39] KUROSE, J. F.; ROSS, K. W. *Redes de Computadores: uma nova abordagem*. 1. ed. São Paulo : Addison-Wesley, 2003.
- [40] LELAND, W.; TAQQU, M.; WILLINGER, W. et al. On the Self-Similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking*, v. **2**, n. 1, p. 1–15, Feb. 1994.
- [41] MANFREDI, S.; BERNARDO, M. di.; GAROFALO, F. A Robust Approach to Active Queue Management Control in Networks. In: IEEE/INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (Nov. 2004 : Vancouver). *Proceedings*. Vancouver : 2004. p. 485–488.
- [42] MANFREDI, S.; BERNARDO, M. di.; GAROFALO, F. Robust Output Feedback Active Queue Management Control in TCP Networks. In: IEEE/CONFERENCE ON DECISION AND CONTROL (43. : Dec. 2004 : Bahamas). *Proceedings*. Paradise Island, Bahamas: 2004. p. 1004–1009.
- [43] MAY, M.; BONALD, T.; BOLOT, J. C. Analytic Evaluation of RED Performance. In: IEEE/INFOCOM (19. : Mar. 2000 : Tel-Aviv, Israel). *Proceedings*. Tel-Aviv: 2000. p. 1415–1424.

- [44] MISRA, V.; GONG, W. B.; TOWSLEY, D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In: ACM/SIGCOMM (Aug. 2000 : Stockholm, Sweden). *Proceedings*. Stockholm: 2000. p. 151–160.
- [45] MISRA, V.; GONG, W. B.; TOWSLEY, D. Stochastic Differential Equation Modeling and Analysis of TCP Window size Behavior. Technical Report ECE-TR-CCS-99-10-01, Istanbul, Oct. 1999. Disponível em <<ftp://gaia.cs.umass.edu/pub/Misra99-TCP-Stochastic.ps.gz>>. Acesso em: Março de 2005.
- [46] MISRA, V.; GONG, W. B.; TOWSLEY, D. et al. A Control Theoretic Analysis of RED. In: IEEE/INFOCOM (20. : Apr. 2001 : Anchorage, Alaska, USA). *Proceedings*. Anchorage: 2001. p. 1510–1519.
- [47] MISRA, V.; GONG, W. B.; TOWSLEY, D. et al. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows. *IEEE Transactions on Automatic Control*, v. **47**, n. 6, p. 945 – 959, June 2002.
- [48] MISRA, V.; GONG, W. B.; TOWSLEY, D. et al. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In: IEEE/INFOCOM (20. : Apr. 2001 : Anchorage, Alaska, USA). *Proceedings*. Anchorage: 2001. p. 1726–1734.
- [49] MO, J.; LA, R. J.; ANANTHARAM, V. et al. Analysis and Comparison of TCP Reno and Vegas. In: IEEE/INFOCOM (18. : Mar. 1999 : New York, USA). *Proceedings*. New York, 1999. p. 1556–1563.
- [50] NORMEY-RICO, J. E. *Apostila de Controle Clássico*. Florianópolis: UFSC, 2003.
- [51] NS-2 NETWORK SIMULATOR. Disponível em <<http://www.isi.edu/nsnam/ns/>>. Acesso em: Junho de 2005.
- [52] OGATA, K. *Engenharia de Controle Moderno*. 3. ed. Rio de Janeiro : LTC — Livros Técnicos e Científicos, 1998.
- [53] PAGANINI, F.; LOW, S. H.; DOYLE, J. C. Internet Congestion Control. *IEEE Control Systems Magazine*, v. **22**, n. 1, p. 28–43, Feb. 2002.
- [54] PAGANINI, F.; LOW, S. H.; DOYLE, J. C. et al. Congestion control for high performance, stability and fairness in general networks. *IEEE/ACM Transactions on Networking*, v. **13**, n. 1, p. 43– 56, Feb. 2005.

- [55] PAXSON, V. End-to-End Internet Packet Dynamics. In: ACM/SIGCOMM (Set. 1997: Cannes, France). *Proceedings*. Cannes, 1997. p. 139 – 152.
- [56] QUET, P. F.; OZBAY, H. On the Design of AQM Supporting TCP Flows Using Robust Control Theory. *IEEE Transactions on Automatic Control*, v. **49**, n. 6, p. 1031– 1036, June 2004.
- [57] RAMAKRISHNAN, K. K.; JAIN, R. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks. *ACM Transactions on Computer Systems*, v. **8**, n. 2, p. 158–181, 1990.
- [58] SEMERIA, C. Understanding IP Addressing: Everything You Ever Wanted to Know. Disponível em <http://www.3com.com/other/pdfs/infra/corpinfo/en_US/501302.pdf>. Acesso em: Setembro de 2005.
- [59] SHIN, S. J.; WEISS, M. B. H. Simulation Analysis of QoS Enabled Internet Pricing Strategies: Flat Rate Vs Two-Part Tarrif. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (36. : 2002 : Hawaii). *Proceedings*. Hawaii: 2002.
- [60] TANENBAUM, A. S. *Redes de Computadores*. Rio de Janeiro : Campus, 1994.
- [61] THOMSON, K.; MILLER, G. J.; WILDER, R. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, v. **11**, n. 6, p. 10 – 23, Nov. 1997.
- [62] TROFINO, A.; COUTINHO, D.; BARBOSA, K. A. *Sistemas Multivariáveis: Uma abordagem via LMIs*. Florianópolis: UFSC, 2003.
- [63] WANG, Z.; CROWCROFT, J. A New Congestion Control Scheme: Slow Start and Search (Tri-S). *ACM Computer Communication Review*, v. **21**, n.1, p. 32–43, Jan. 1991.
- [64] XIAO, X.; NI, L. M. Internet QoS: A Big Picture. *IEEE Network*, v. **13**, n. 2, p. 8–18, March/April 1999.