

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
ENGENHARIA ELÉTRICA

GUILHERME MAXIMILIANO TREIN

**SÍNTESE DE COMPENSADOR
ANTI-WINDUP ESTÁTICO PARA
SISTEMAS COM ATRASO: APLICAÇÃO
AO CONTROLE DE
CONGESTIONAMENTO DE REDES TCP**

Porto Alegre
2009

GUILHERME MAXIMILIANO TREIN

SÍNTESE DE COMPENSADOR
ANTI-WINDUP ESTÁTICO PARA
SISTEMAS COM ATRASO: APLICAÇÃO
AO CONTROLE DE
CONGESTIONAMENTO DE REDES TCP

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Engenheiro Eletricista.

ORIENTADOR: Prof. Dr. João Manuel Gomes da Silva Jr.

Porto Alegre
2009

GUILHERME MAXIMILIANO TREIN

**SÍNTESE DE COMPENSADOR
ANTI-WINDUP ESTÁTICO PARA
SISTEMAS COM ATRASO: APLICAÇÃO
AO CONTROLE DE
CONGESTIONAMENTO DE REDES TCP**

Este Projeto foi julgado adequado para a obtenção dos créditos da Disciplina Projeto de Diplomação do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. João Manuel Gomes da Silva Jr., UFRGS
Doutor pela Université Paul Sabatier – Toulouse, França

Banca Examinadora:

Prof. Dr. João Manuel Gomes da Silva Jr., UFRGS
Doutor pela Université Paul Sabatier – Toulouse, França

Prof. Dr. Walter Fetter Lages, UFRGS
Doutor pelo Instituto Tecnológico de Aeronáutica – São José dos Campos, Brasil

Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universität Paderborn – Paderborn, Alemanha

Chefe do DELET: _____
Prof. Dr. Marcelo Soares Lubaszewski

Porto Alegre, dezembro de 2009.

Dedicatória

Dedico este trabalho aos meus pais pela dedicação e incentivo em todos os momentos difíceis, contribuindo assim para a realização dessa conquista.

Igualmente dedico essa conquista à minha namorada que me acompanhou e me apoiou durante todo o desenvolvimento desde trabalho.

Agradecimentos

Aos meus pais, João Carlos e Maria Julia, e ao meu irmão, Gustavo, pelo apoio e auxílio dado durante todo o curso.

À minha namorada, Luiza, pela paciência e suporte no desenvolvimento deste trabalho.

Ao professor João Manoel Gomes da Silva Jr. por sua orientação e ajuda durante o trabalho

Ao Fernando Bender pelo auxílio nas diversas dúvidas e questões referentes ao tema de controle.

Aos colegas do curso de engenharia elétrica, pelo auxílio na solução de dúvidas e nas tarefas desenvolvidas durante o curso.

Abstract

The present work is divided in two parts. In the first part a revision of recent control techniques employed in the congestion of TCP/IP networks control. In particular, the congestion control called AQM will be studied (*Active Queue Management*), when it employs the selective package drop in a congested router buffer. This kind of buffer management is called RED (*Random Early Detection*) and has been subjected to research in recent years for professionals in the domain of networks and telecommunication.

It will be seen, in this application, the control signal determines the drop probability that the routers buffer packages are subjected. The probability can be determined by a PI controller, a function of the system states, i.e. the transmission rate of data and the size of the routers buffer. As the drop probability cannot assume any real value, the input of the system is subjected to a saturation, which, many times, can cause losses of performance and destabilize the system. In particular, if the system presents a time delay in its states and inputs (as it will be presented, it is the case of TCP/IP networks) the saturation can still be more severe on the systems behavior.

Researchers on control processes has been developing methodologies to mitigate the negative effects of the saturation in delayed systems. Amongst them, the synthesis of *anti-windup* compensators deserves prominence for its applicability in the industry of processes. That is why, in the second part of this work, we propose the synthesis of a static *anti-windup* compensator for time continuous linear systems with uncertain delay in order to minimize possible effects for disturbances.

The synthesis technique consider Liapunov-Krasovskii functions and a generalized sector condition in the attainment of enough conditions for the enter-state system stability in closed loop, subject or not to limited disturbances \mathcal{L}_2 . We have as a goal as well the maintenance of a certain level of performance of the system output. The obtained conditions are dependents of the delay that is unknown, but limited.

With the main goal of analyze the proposed solution performance, a simulator software of networks, the NS-2, developed especially for simulation of data networks will be used. In this simulator, some of the theoretical results found in literature will be implemented, as the control law of package drop in a congested router. This implementation will allow to the evaluation of methodologies of synthesis of compensators of *anti-windup* in one same environment.

Keywords: *Anti-windup* , Delayed Systems, Disturbance, AQM/RED, TCP/IP Networks, TCP, UDP, Network Simulator 2

Resumo

O presente trabalho é dividido em duas partes. Na primeira parte faz-se uma revisão das recentes técnicas de controle empregadas no controle de congestionamento de redes TCP/IP. Em particular, será estudo o controle de congestionamento chamado AQM (*Active Queue Management*), quando este emprega o descarte seletivo de pacotes em uma fila de um roteador congestionado. Esta modalidade de gestão de fila é chamada RED (*Random Early Detection*), e tem sido objeto de pesquisa nos últimos anos na literatura que trata de redes e telecomunicação.

Será visto que, nesta aplicação, o sinal de controle determina a probabilidade de descarte a que os pacotes na fila do roteador estão sujeitos. Esta probabilidade pode ser determinada por um controlador PI, em função dos estados do sistema, i.e. a taxa de transmissão do fluxo de dados, e o tamanho da fila do roteador. Como a probabilidade de descarte não pode assumir qualquer valor real, a entrada do sistema está sujeito à saturação, o que muitas vezes é a causa de perdas de desempenho e pode levar o sistema à instabilidade. Em particular, se o sistema apresenta atrasos nos estados e nas entradas (como será apresentado, é o caso de redes TCP/IP) a saturação pode ser ainda mais severa em seus efeitos sobre o sistema.

Pesquisadores de controle de processos vem desenvolvendo metodologias para dirimir esses efeitos negativos da saturação em sistemas com atraso. Dentre elas, a síntese de compensadores *anti-windup* merece destaque pela sua aplicabilidade na indústria de processos. Dessa forma, na segunda parte deste trabalho, propõe-se a síntese de um compensador *anti-windup* estático para sistemas lineares de tempo contínuo com atraso incerto de forma a minimizar eventuais efeitos por perturbações.

As técnicas de síntese consideram funcionais de Liapunov-Krasovskii e uma condição de setor generalizada na obtenção de condições suficientes para a estabilidade entrada-estado do sistema em malha fechada, sujeito ou não a perturbações limitadas em norma \mathcal{L}_2 . Tem-se por objetivo igualmente manter um certo nível de performance da saída do sistema (tamanho da fila do roteador). As condições obtidas são dependentes do atraso que é desconhecido, mas contido dentro de um intervalo.

Com o objetivo de avaliar o desempenho da solução proposta, será utilizado um software simulador de redes, o NS-2, desenvolvido especialmente para simulação de redes de dados. Neste simulador, alguns dos resultados teóricos da literatura serão implementados, como a lei de controle do descarte de pacotes em um roteador de dados com tráfego congestionado. Esta implementação permitirá a avaliação de metodologias de síntese de compensadores de *anti-windup* em um mesmo ambiente.

Palavras-chave: *Anti-windup* , Sistemas com Atraso, Perturbações, AQM/RED, Redes TCP/IP, TCP, UDP, Network Simulator 2

Lista de Ilustrações

Figura 1:	Representação das camadas dos modelos TCP/IP comparada com o modelo OSI	28
Figura 2:	Estrutura do cabeçalho de um pacote TCP (retirado de (TANENBAUM, 1997))	31
Figura 3:	Exemplo de estabelecimento de uma conexão TCP	33
Figura 4:	Exemplo de finalização de uma conexão TCP	33
Figura 5:	Exemplo de transmissão TCP	34
Figura 6:	Algoritmo slow start combinado com o congestion avoidance (retirado de (TANENBAUM, 1997))	36
Figura 7:	Algoritmo fast retransmit combinado com o slow start	37
Figura 8:	Função de probabilidade do RED (retirado de (MISRA; GONG; TOWSLEY, 2000))	41
Figura 9:	RED como filtro passa-baixa em cascata com função de descarte (retirado de (MISRA; GONG; TOWSLEY, 2001))	42
Figura 10:	Comportamento da janela de congestionamento W com a excitação de um degrau limitado	47
Figura 11:	Comportamento da fila de pacotes q com a excitação de um degrau limitado .	47
Figura 12:	Diagrama de blocos de um sistema com saturação	53
Figura 13:	Sinal de saída do processo com e sem compensador <i>anti-windup</i> (retirado de (GOMES DA SILVA JR.; BAZANELLA, 2005))	53
Figura 14:	Sinal de controle do processo com e sem compensador <i>anti-windup</i> (retirado de (GOMES DA SILVA JR.; BAZANELLA, 2005))	54
Figura 15:	Comportamento de um sistema com compensador <i>anti-windup</i>	54
Figura 16:	Função determinante das condições iniciais do sistema	62
Figura 17:	Sinal de controle do modelo de rede TCP/IP	74
Figura 18:	Tamanho da fila de pacotes do modelo de rede TCP/IP	74
Figura 19:	Comparação do sinal de controle para variações do nível de perturbação . . .	75
Figura 20:	Comparação do sinal de saída para variações do nível de perturbação	76
Figura 21:	Comparação do sinal de controle para variações do nível de atraso	76
Figura 22:	Comparação do sinal de saída para variações do nível de atraso	77
Figura 23:	Topologia usada na simulação comparativa	80

Figura 24:	Comportamento da fila de pacotes no roteador - Droptail	81
Figura 25:	Evolução da janela de congestionamento das fontes - Droptail	82
Figura 26:	Comportamento da fila de pacotes no roteador - RED	83
Figura 27:	Probabilidade de descarte de pacotes - RED	84
Figura 28:	Evolução da janela de congestionamento das fontes - RED	84
Figura 29:	Comportamento da fila de pacotes no roteador - PI	86
Figura 30:	Probabilidade de descarte de pacotes - PI	87
Figura 31:	Evolução da janela de congestionamento das fontes - PI	88
Figura 32:	Comparação do nível da fila de pacotes - PI vs PI-AW	91
Figura 33:	Comparação do nível da fila de pacotes (zoom) - PI vs PI-AW	92
Figura 34:	Comparação da probabilidade de descarte - PI vs PI-AW	92
Figura 35:	Comparação da probabilidade de descarte (zoom) - PI vs PI-AW	93

Lista de Tabelas

Tabela 1:	Compensador com rejeição à perturbação utilizando os dados didáticos	72
Tabela 2:	Compensador com rejeição à perturbação utilizando os dados da aplicação . .	73

Lista de Abreviaturas

AQM	Active Queue Management
BMI	Bilinear Matrix Inequality
FTP	File Transfer Potocol
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LMI	Linear Matrix Inequality
NS	Network Simulator
PI	Proportional Integral
POP	Post Office Protocol
RED	Random Early Detection
RFC	Request For Comments
SMTP	Simple Mail Service Protocol
TCP	Transfert Control Protocol
UDP	User Datagram Protocol

Sumário

1	Introdução	23
2	Redes TCP/IP	27
2.1	Redes de computadores	27
2.1.1	Introdução	27
2.2	Transmission Control Protocol (TCP)	29
2.2.1	Controle de fluxo de pacotes	30
2.2.2	Controle de congestionamento de pacotes	34
2.2.3	Atrasos nodais	38
3	Active Queue Management (AQM)	39
3.1	Introdução	39
3.2	AQM Random Early Detection (RED)	40
3.3	Descrição do controle do RED	41
3.4	Modelo dinâmico do AQM RED em uma rede TCP/IP	42
3.4.1	Resultados anteriormente obtidos	43
3.4.2	Linearização do modelo	45
3.4.3	Análises sobre modelo linearizado	47
4	Controle de sistemas com atraso sob saturação	51
4.1	Sistemas sob saturação	51
4.1.1	Compensação <i>anti-windup</i>	52
4.2	Sistemas com atraso	55
4.3	Teoria de Lyapunov	56
4.4	Perturbações	57
5	Aplicação da teoria de controle no gerenciamento de fluxo de pacotes em redes TCP/IP	59

5.1	Definições iniciais	59
5.2	Desenvolvimento do problema	63
5.2.1	Estabilização do sistema	63
5.2.2	Problema de otimização	70
5.2.3	Resultados numéricos	71
5.2.4	Estudos comparativos para o caso TCP/IP	73
6	Simulações de topologias TCP	79
6.1	Introdução	79
6.2	Topologia de simulação	79
6.3	Simulação do comportamento dos AQMs	81
6.3.1	Simulação do comportamento da fila de pacote sem AQM	81
6.3.2	Simulação do comportamento do AQM RED	82
6.3.3	Simulação do comportamento do AQM PI de Hollot	83
6.4	Comparação AQM PI de Hollot com compensador <i>anti-windup</i> sintetizado	87
6.4.1	Implementação do compensador <i>anti-windup</i> sintetizado	89
6.4.2	Resultados	90
7	Conclusão	95
7.1	Trabalhos futuros	96
	Referências	97
	Apêndice A Ferramental matemático	99
	A.1 Linear matrix inequalities (LMIs)	99
	A.2 Complemento de Schur	100
	A.3 S-Procedure	100
	Apêndice B Linearização do modelo para fluxo de pacotes	103
	B.1 Definições	103
	B.2 Desenvolvimento	103
	Apêndice C Simulador NS-2 e o controle de congestionamento em redes TCP	105

C.1	Instalação	105
C.2	Operação	107
C.2.1	Simulações de tráfego em rede TCP	107
C.2.2	Simulações de tráfego com AQM implementado	110

1 Introdução

No primórdios dos anos 70, um projeto de pesquisa da *Defense Advanced Research Projects Agency* (DARPA) do Ministério de Defesa dos Estados Unidos criava o um conjunto de protocolos de comunicação chamado TCP/IP (*Transmission Control Protocol / Internet Protocol*). O objetivo inicial era de simplesmente obter uma maneira de estabelecer conexão entre equipamentos de diferentes fabricantes em uma rede comum. Embora parecesse algo extremamente útil, não se imaginava o potencial que eles iriam apresentar anos mais tarde com a difusão de suas aplicações, como na rede mundial de computadores (internet).

Dentro de uma rede de computadores (leia-se rede TCP/IP), o *Transmission Control Protocol* (TCP) é responsável pelo controle da circulação de pacotes para transferência de dados. Ele se encarrega do controle de todo o fluxo de informação que trafega entre os equipamentos que compõem a topologia formada, desde os envio de confirmações de recebimento dos pacotes até o controle de tráfego desses dados. Hoje em dia, devido à sua grande disseminação e larga utilização através da internet e outras diversas aplicações, ele pode ser considerado como o protocolo de transporte mais importante já criado.

Pode-se observar ainda, nos dias de hoje, uma grande tendência rumo a chamada convergência dos dispositivos inteligentes. Essa convergência estabelece o paradigma de que o mais simples equipamento terá como se comunicar com outros dispositivos ao seu redor de forma eficiente e segura, o que motiva de forma significativa a procura por melhoras nos sistemas de comunicação utilizados atualmente.

Por essa razão, muitos pesquisadores dedicam seus estudos na área de redes. Encontram-se já na literatura muitos resultados interessantes sobre esse tema. Autores de (MISRA; GONG; TOWSLEY, 2000) desenvolveram um modelo matemático representando o comportamento de um roteador congestionado atuando em uma rede regida pelo TCP.

Devido à contínua dispersão de redes de computadores interligadas, o roteador tornou-se um dispositivo elementar para a transmissão de dados. Sua função é aplicar uma política de roteamento nos pedaços de informação *transeunte* (ditos quadros, pacotes, ou segmentos), aumentando a eficiência da transferência de dados na rede. Contudo, ainda com essa otimização, existem diversos aspectos que degradam a performance desse sistema. Devido ao grande número de rotas possíveis entre dois ponto na rede, é muito difícil prever duração de um processo de transferência de dados. Assim sendo, à comunicação associa-se um atraso variante no tempo (BENDER, 2006).

Ainda, o tráfego de dados entrante em um roteador pode freqüentemente superar sua capacidade de redirecionamento. Com o objetivo de evitar perdas de informações, existe um espaço reservado para a armazenagem de dados chamado de *buffer*. Dessa maneira, uma fila de pacotes pode se formar quando a demanda excede a capacidade do roteador, o que acresce ainda mais a incerteza do tempo consumido na transferência.

As chamadas políticas de *Active Queue Management* foram estabelecidas para auxiliar no

controle do tráfego de pacotes em roteadores no TCP. Esses algoritmos agem diretamente no enfileiramento de pacotes formado no roteador, utilizando critérios previamente estabelecidos para o descarte seletivo de dados. Cada pacote nessa fila recebe uma probabilidade de descarte associada.

Dessa maneira, aplica-se um controle de congestionamento intrínseco ao já existente em redes TCP/IP. Seguindo as regras estabelecidas pelo protocolo, um emissor que detecta a perda de seus pacotes enviados, diminui sua taxa de envio. Por esse motivo, os *Active Queue Management* compõem hoje em dia uma área de pesquisa ativa, envolvendo diversas áreas como engenharia de redes e até mesmo engenharia de controle.

Para uma abordagem de engenharia de controle nesse processo, a necessidade de um modelo matemático representando os principais aspectos do fenômeno físico que se quer controlar é essencial. A performance do controle desenvolvido depende do grau de fidelidade desse modelo em relação ao processo físico real, já que as leis de controle são estabelecidas sobre este. Todos os limites físicos relevantes devem ser obrigatoriamente representados no modelo e respeitados pelas leis de controle. Caso contrário, o controle implementado pode gerar comportamentos imprevistos, fazendo com que a ação de controle não mais conduza o comportamento do sistema conforme o projeto original.

Um dos primeiros trabalhos que abordam de fato elementos de redes computacionais como um problema de controle é aquele de (HOLLOT et al., 2002). HOLLOT utiliza o modelo matemático desenvolvido no trabalho de (MISRA; GONG; TOWSLEY, 2000) para estudar a gestão da fila de pacotes em um roteador congestionado pelo excesso de tráfego em suas interfaces. Apesar de precursor na abordagem de controle, o sistema apresentado ainda não leva em consideração alguns parâmetros importantes na modelagem do controlador.

Assim, este trabalho propõe uma abordagem de controle para gerência do fluxo de dados em uma rede TCP/IP a nível de transporte em um roteador. Como uma rede de transferência de dados freqüentemente é formada por diversos nós clientes conectados a um servidor através de um roteador, o estudo do trabalho se focará nessa topologia.

Dessa maneira, serão aplicadas técnicas de controle linear sobre as equações que representam o comportamento do TCP implementado em um roteador congestionado, aproveitando o trabalho (MISRA; GONG; TOWSLEY, 2000). Mesmo com as simplificações já aportadas ao modelo pelos autores, o sistema proposto ainda é sujeito à saturação de controle bem como à atraso incerto nos estados e controle.

A abordagem adotada se baseará nos resultados de dois trabalhos importantes. Para a avaliação da estabilidade do sistema em malha fechada, se fará uso de uma funcional de Lyapunov-Krasovskii conforme enunciado por (NICULESCU, 2001). Uma condição de setor modificada, como proposto em (GOMES DA SILVA JR.; TARBOURIECH, 2005), dará o suporte matemático para levar em consideração a saturação do sistema modelada como uma não linearidade. A partir destas referências, condições suficientes para a existência de um compensador são expressas diretamente em forma de LMI, de modo a assegurar a estabilidade de entrada e saída do sistema, bem como a estabilidade assintótica interna da origem do sistema em malha fechada, na ausência perturbações.

Simulações de redes de computadores são consideravelmente complexas devido aos diversos parâmetros devem ser levados em consideração para que as simulações possam ao máximo aproximar-se de um ambiente real da aplicação simulada.

Afim de analisar os resultados obtidos e compará-los com outros oriundos de diferentes sis-

temas e implementações, utilizar-se-á o programa *Network Simulator* para avaliar as teorias propostas. Em razão da precariedade de documentação sobre esse simulador, uma seção deste trabalho será reservada para a explicação detalhada do funcionamento do programa assim como das instruções para sua instalação. Com a análise dessas simulações, pretende-se propor pontos de melhoria para a metodologia desenvolvida e futuras linhas de pesquisa sobre o assunto.

2 Redes TCP/IP

Neste capítulo, se descreverá o contexto da aplicação propriamente dita de toda a teoria aqui desenvolvida. Uma breve introdução sobre redes de computadores será feita focando o objetivo do trabalho. Conceitos fundamentais para o pleno entendimento da problemática serão revisados aqui.

Em seguida, o *Transmission Control Protocol* será introduzido, descrevendo-se seu funcionamento assim como seus principais algoritmos de controle e eventuais limitações.

2.1 Redes de computadores

2.1.1 Introdução

Uma rede de computadores consiste em dois ou mais computadores e outros dispositivos conectados entre si de modo a poderem compartilhar seus serviços. Contudo, em razão da infinidade de produtores e métodos fabricação desses dispositivos, para que essa comunicação possa ser efetivamente feita, um conjunto de regras único (protocolo) deve ser seguido por esses equipamentos.

No mundo de hoje, não se pode falar de arquitetura de redes sem citar o TCP/IP. Este conjunto de protocolos, originalmente desenvolvido pela Universidade da Califórnia em Berkeley, sob contrato para o Departamento de Defesa dos EUA, se tornou o conjunto de protocolos padrão das redes locais e remotas, suplantando conjuntos de protocolos bancados por grandes empresas como a IBM, Microsoft e Novell.

O grande motivo de todo este sucesso pode ter sido justamente o fato do TCP/IP não ter nenhuma grande empresa associada ao seu desenvolvimento. Talvez isso tenha possibilitado a sua implementação e utilização por diversas aplicações em praticamente todos os tipos de hardware e sistemas operacionais existentes.

2.1.1.1 Modelo de pilha do Transmission Control Protocol/Internet Protocol (TCP/IP)

A arquitetura de redes TCP/IP foi desenhada segundo um modelo de pilha, onde diversas camadas de software interagem somente com as camadas acima e abaixo. Este desenvolvimento agrega diversas semelhanças com o modelo OSI da ISO (para maiores informações, consultar (TANENBAUM, 1997)), contudo, devido a formalização anterior do TCP/IP, algumas especificações não são seguidas. Deve-se notar que o modelo de camadas OSI é simplesmente conceitual, visto que nenhuma implementação fiel de sua estrutura foi feita e utilizada.

O acrônimo TCP/IP vem dos nomes dos protocolos mais utilizados desta pilha, o IP (*Internet Protocol*) e o TCP (*Transmission Control Protocol*). Mas a pilha TCP/IP possui ainda muitos outros protocolos, vários deles necessários para que o TCP e o IP desempenhem corretamente as

suas funções (TANENBAUM, 1997). A figura 1 apresenta as camadas da estrutura do modelo TCP/IP seguido de exemplos de protocolos para cada uma delas.

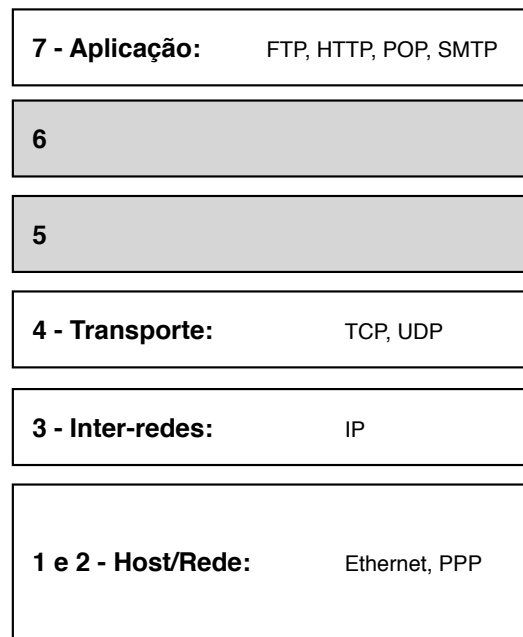


Figura 1: Representação das camadas dos modelos TCP/IP comparada com o modelo OSI

Considera-se o TCP/IP como sendo constituído por quatro camadas apenas. A camada superior, camada de Aplicação/Processo, é onde estão localizados os protocolos de mais alto nível do modelo (TANENBAUM, 1997).

A aplicação cliente em geral está em um equipamento mais simples e com uma boa interface com usuário. Esta aplicação envia requisições à aplicação servidor que normalmente está em uma plataforma mais robusta e que tem capacidade para atender várias requisições diferentes de clientes diferentes. Os protocolos mais utilizados dessa camada são o *Hypertext Transfer Protocol* (HTTP), *Simple Mail Service Protocol* (SMTP), *Post Office Protocol* (POP) e *File Transfer Potocol* (FTP).

O HTTP é um protocolo responsável pelo tratamento de pedidos e respostas entre cliente e servidor na *World Wide Web*. A versão atual desse protocolo é HTTP/1.1 e maiores informações podem ser obtidas no documento (FIELDING; GETTYS; MOGUL, 1999), onde seu funcionamento é descrito em detalhes. Existem igualmente os protocolos SMTP e POP destinados aos serviços de correio eletrônico.

Em redes TCP/IP, o FTP é o protocolo padrão para a transferência de arquivos de grande porte. Ele conta com restrições de acesso e propriedades dos mesmos. Maiores informações podem ser encontradas em (REYNOLDS; POSTEL, 1985).

A camada que segue, camada de Transporte ou *fim a fim*, tem a função principal de permitir que as entidades pares dos *hosts* de origem e de destino mantenham uma conversação, exatamente como acontece com a camada de transporte OSI (TANENBAUM, 1997). Existem dois protocolos principais presentes nesse nível: *Transfert Control Protocol* (TCP) e *User Datagram Protocol* UDP. A principal diferença entre eles reside no fato do UDP não possuir qualquer confiabilidade na transmissão de dados. Isso ocorre pois os pacotes enviados por UDP não possuem qualquer garantia de recebimento pelo destinatário enquanto que, utilizando o TCP, existe sempre um

aviso de recebimento para cada pacote.

O TCP conta ainda, devido ao seu mecanismo de transferência de dados, com diversos mecanismos de controle de tráfego e congestionamento. Em uma conexão seguindo o protocolo TCP ainda é possível efetuar processos de correção e verificação de erros. Maiores detalhes serão dados sobre o protocolo TCP na seção 2.2.

A camada de Inter-redes define um formato de pacote oficial e um protocolo chamado *Internet Protocol* (IP). Sua tarefa principal é entregar pacotes IP onde eles são necessários. O roteamento é uma questão de grande importância nessa camada, assim como a necessidade de se evitar congestionamentos (TANENBAUM, 1997).

A camada de Host/Rede é a responsável pelo roteamento. Comparativamente, ela corresponde no modelo ISO/OSI a camada de Rede (nível 3) e parte da camada Enlace (nível 2). Esta camada é usada para rotear a informação para a rede correta e para fazer as ligações entre as camadas superiores. Em essência podemos afirmar que sem esta camada, as aplicações teriam que ser desenvolvidas para cada tipo de arquitetura de rede como por exemplo *Ethernet* ou *Token Ring* (HUNT, 1997).

Cada camada oferece serviços específicos para a próxima camada seguindo um protocolo comum. Igualmente, essa divisão estabelece um interface de comunicação comum entre camadas para dispositivos de fabricantes diferentes.

2.2 Trasmision Control Protocol (TCP)

O *Trasmision Control Protocol* (TCP) é um protocolo de nível quatro do modelo TCP/IP que controla a circulação de segmentos dentro da rede (ARIBA, 2005-2006). Ele foi projetado especificamente para oferecer um fluxo de bytes *fim a fim* confiável em uma inter-rede não confiável. Uma inter-rede pode ser considerada diferente de uma única rede devido ao fato de se poder ter uma infinidade de topologias, larguras de banda, retardos e outros parâmetros completamente diferentes. O TCP foi desenvolvido para se adaptar dinamicamente às propriedades da inter-rede e ser robusto diante dos muitos tipos de falhas que podem ocorrer durante uma transmissão (TANENBAUM, 1997).

Esse conjunto de protocolos foi totalmente definido na *Request For Comments* (RFC) da publicação de (POSTEL, 1981). RFC são as especificações e proposições de novos protocolos e mecanismo relativos a redes, que podem ser livremente propostos. Mediante uma RFC, as proposições são numeradas e ficam sujeitas a comentários por seis meses. Terminado esse período, não havendo críticas, elas podem ser empregadas na indústria.

De acordo com (POSTEL, 1981), o principal objetivo do TCP pode ser definido como proporcionar uma forma de transferência de dados confiável e relativamente segura. Todas as transmissões de dados feitas possuem confirmação da integridade da informação. Todos os dados que forem perdidos pela rede são recuperados, dados duplicados são descartados e dados entregues de forma desordenada são reordenados corretamente antes de serem entregues ao processo de destino.

A versatilidade e robustez deste protocolo tornou-o adequado a redes globais, já que este verifica se os dados são enviados de forma correta, na seqüência apropriada e sem erros, pela rede. Hoje, ele pode ser considerado como o protocolo de transporte mais importante já criado, principalmente pela grande disseminação e larga utilização por aplicações da internet (SILVA CA-

RISSIMI; JUERGEN; GRANVILLE, 2009).

2.2.1 Controle de fluxo de pacotes

Pelo fato do protocolo TCP estabelecer uma conexão *full-duplex* entre dois sistemas finais, cada lado da conexão tem um *buffer* de envio e 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.

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. Esse controle é implementado através de uma variável, chamada de janela 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 ao *buffer*.

2.2.1.1 Definições

Para uma boa compreensão do funcionamento do processo de transferência de dados dentro de um protocolo TCP, deve-se estabelecer algumas definições.

- Aviso de recepção (Acknowledgement - ACK):
Como será explicado em seguida, no protocolo TCP, todos os dados transmitidos são confirmados e verificados. Esse aviso trata-se então da mensagem emitida do receptor ao emissor quando um pacote é recebido com sucesso.
- Tempo de timeout:
Este é um tempo definido como tempo limite de espera por um aviso de recepção por parte do emissor. Tendo este tempo esgotado para o recebimento de um aviso de recepção (ACK) de um pacote, o emissor deve reenviar o mesmo.
- Buffer do roteador:
Trata-se de uma zona da memória do roteador ou equipamento que intermedia as transferências na qual os pacotes (isto é, dados) são temporariamente armazenados antes de serem tratados. De uma forma menos formal, é o espaço onde os dados são colocados quando a taxa de recepção é superior a capacidade do equipamento de tratar esses mesmos dados.
- Janela de congestionamento:
É a definição do número de pacotes que o emissor pode enviar antes que o mesmo tenha que esperar por um aviso de recepção. Aqui, existe uma estratégia diferente para justamente tentar aumentar a performance do protocolo. Enviam-se diversos pacotes e aguarda-se as respectivas confirmações. Mais detalhes serão explicados a seguir.
- Janela de recepção:
Pode-se definir como o número máximo de pacotes que o receptor é capaz de receber sem que haja perdas de dados. Essa informação corresponde praticamente ao tamanho do *buffer* do equipamento.
- Round trip time (RTT):
É definido como a soma do tempo que o pacote leva para atingir sua destinação acrescido do tempo de retorno do aviso de recepção desse mesmo pacote. Pode-se dizer, logo, que

esse é o tempo que determina a duração da troca de um segmento de dados entre duas entidades.

2.2.1.2 Pacotes TCP

Para se ter um bom entendimento do processo de transporte do protocolo TCP, um conhecimento básico da estrutura do pacote TCP é imprescindível. Como não é o escopo desse trabalho uma caracterização profunda sobre esses conceitos, apenas as características principais da estrutura desse componente serão apresentadas.

O pacote TCP é formado por diversas palavras de 32 bits. Pode-se dividi-lo em duas partes distintas: um cabeçalho de informações e os dados propriamente ditos. A figura 2 ilustra a estrutura do cabeçalho TCP.

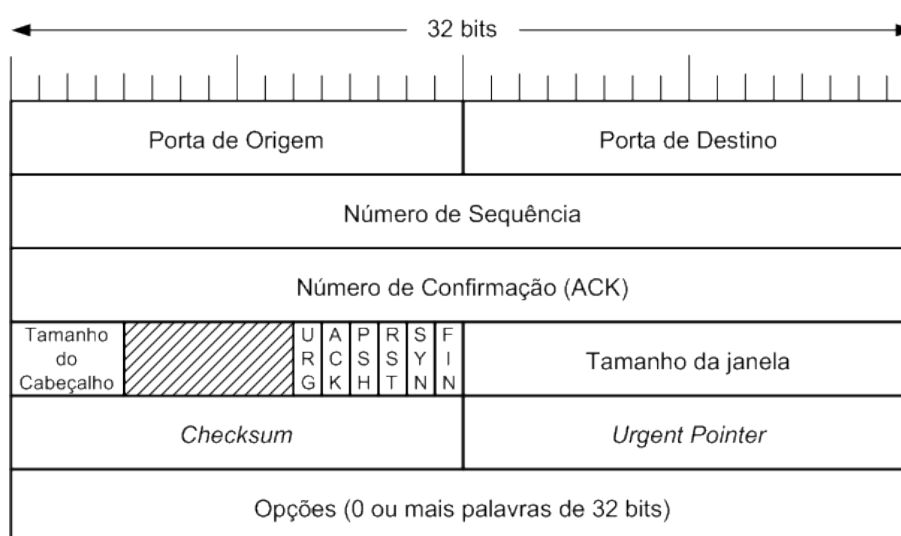


Figura 2: Estrutura do cabeçalho de um pacote TCP (retirado de (TANENBAUM, 1997))

Nas duas primeiras posições do pacote TCP, as portas de destino e origem são armazenadas com 16 bits cada uma. Em seguida, com um tamanho de 32 bits, tem-se o chamado número de seqüência do pacote. Trata-se da identificação única do pacote dentro de um sentido de comunicação do protocolo.

Como será explicado nas seções que seguem, todos os pacotes TCP têm seu recebimento confirmado. Essa verificação se faz justamente pelo campo de número de confirmação também em um tamanho de 32 bits.

O cabeçalho do pacote traz ainda informações sobre o seu tamanho em palavras de 32 bits. Esse item tem um valor mínimo de 5, correspondendo a um pacote TCP sem campos de opções e de dados de usuário (tamanho total de 20 bytes = $5 * 32 / 8$).

Seguindo a estrutura, pode-se identificar 6 bits reservados para uso futuro e mais 6 bits correspondentes às mensagens de verificação interna do protocolo identificadas pelos acrônimos (POSTEL, 1981):

- URG: Urgent Pointer field
- ACK : Acknowledgment field

- PSH: Push Function
- RST: Reset the connection
- SYN : Synchronize sequence
- FIN: No more data from sender

Nas seções que seguem, essas mensagens (ou *flags*) serão explicadas com mais detalhes.

O campo chamado de tamanho da janela é, para a abordagem dada a esse trabalho, um campo bastante importante do pacote TCP. O TCP, como será detalhado na seqüência, utiliza o algoritmo de janelas deslizantes para o controle de fluxo. Esse tamanho da janela deslizante (ou também chamada de janela de recepção) de cada processo participante de uma conexão é passado ao lado remoto através dos 16 bits desse campo (SILVA CARISSIMI; JUERGEN; GRANVILLE, 2009).

A integridade da informação transferida é assegurada pelo campo *checksum*. Trata-se de um campo utilizado para identificação da integridade dos dados contidos no pacote. Os demais campos, por não terem aplicações diretas à nossa aplicação, não serão tratados aqui.

Logo após o cabeçalho, o chamado *payload* é armazenado. Essa região do pacote é composta pelos dados que serão transmitidos. É realmente a informação da mensagem e geralmente compõe a parte mais pesada do pacote.

2.2.1.3 Estabelecimento e finalização de conexões

O TCP realiza o estabelecimento de conexões por troca de três mensagens. Essas mensagens são identificadas pela combinação das *flags* SYN e ACK . O processo de conexão acontece da seguinte maneira:

- Um primeiro pacote é enviado do cliente para o servidor com as *flags* SYN ativada e ACK desabilitada.
- O servidor, ao identificar as *flags*, entende que o cliente quer sincronizar o número de seqüência inicial para iniciar uma transmissão. Da mesma forma, o servidor interpreta que campo de confirmação desse pacote não deve ser tratado.
- Em seguida, caso o servidor aceite a conexão, ele responde com uma mensagem com ambos *flags* ACK e SYN habilitados. Essa mensagem conterà um número de seqüência dado pelo servidor e um número de confirmação incrementado do número de seqüência da mensagem enviada pelo cliente.
- O cliente, ao receber a mensagem com as *flags* ACK e SYN habilitadas, trata os campos número de seqüência e número de confirmação recebidos. Sendo o número de confirmação um incremento do número de seqüência da mensagem enviada anteriormente, o cliente entende que sua requisição foi confirmada.
- Estando a requisição confirmada, o cliente envia outra mensagem ao servidor com a *flag* ACK habilitada contendo o número de seqüência igual ao enviado (logo, o identificador esperado para a próxima mensagem) pelo servidor anteriormente e com o número de confirmação sendo um incremento do número de seqüência da mensagem anterior do servidor.

A figura 3 ilustra o processo de estabelecimento de conexão entre o cliente e o servidor. Logo após a identificação da *flag*, está o valor dos campos número de seqüência e confirmação.

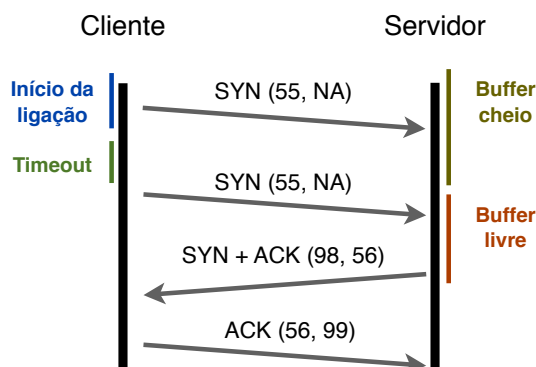


Figura 3: Exemplo de estabelecimento de uma conexão TCP

Após o estabelecimento de uma conexão, a transmissão de dados pode ser iniciada. Seu processo será tratado à seguir na seção 2.2.1.4.

A finalização de conexões no protocolo é realizado de maneira semelhante ao do estabelecimento. Agora, a *flag* FIN é utilizada para solicitar a finalização de uma conexão por ambos os lados da transmissão. O interessante nesse procedimento é que ambos atores devem enviar o pedido de finalização e confirmar os respectivos recebimentos com a *flag* ACK. A figura 4 ilustra o processo (SILVA CARISSIMI; JUERGEN; GRANVILLE, 2009).

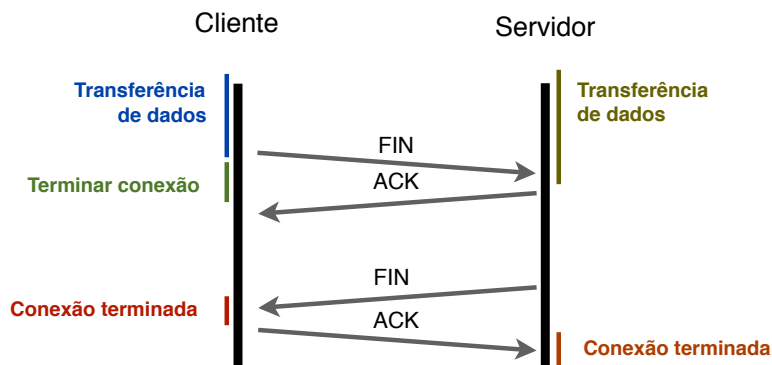


Figura 4: Exemplo de finalização de uma conexão TCP

2.2.1.4 Estabelecimento da transmissão

Sendo a principal característica do TCP sua capacidade de verificar a recepção de dados emitidos a um destinatário, o cliente espera uma confirmação de recebimento para todos os pacotes enviados. Contudo, um mecanismo de aviso de recebimento para cada pacote enviado traria algumas desvantagens quanto à performance do sistema. Ainda, é extremamente inconveniente deixar que um transmissor seja mais rápido e sobrecarregue um receptor mais lento (TANENBAUM, 1997).

Por essa razão, o cliente utiliza o parâmetro chamado tamanho da janela ou janela do receptor, disponível no cabeçalho do pacote TCP de confirmação enviado pelo servidor. Esse dado é então

utilizado pelo emissor para determinar o número máximo de pacotes que podem ser enviados sem sobrecarregar o receptor. Assim, o emissor sabe que só poderá ter em trânsito aquela quantidade de informação até esperar pela confirmação (ACK) de um dos pacotes (SILVA CARISSIMI; JUERGEN; GRANVILLE, 2009).

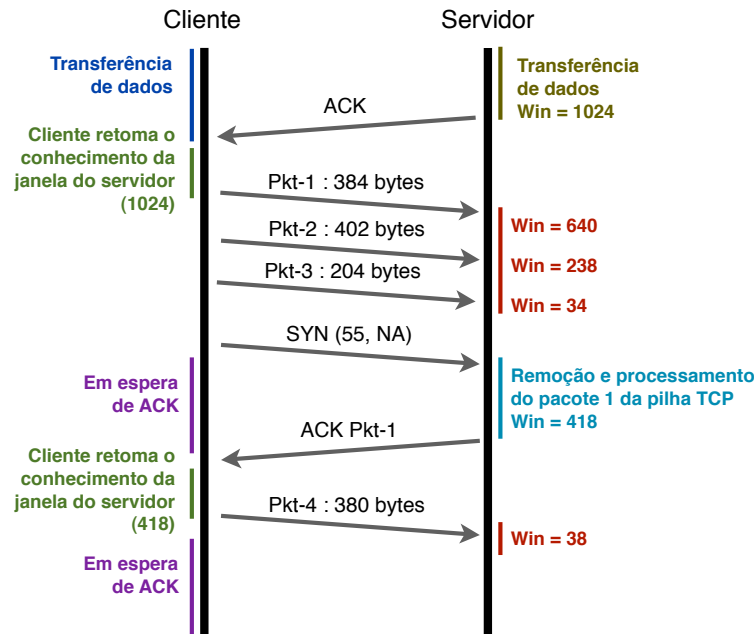


Figura 5: Exemplo de transmissão TCP

A figura 5 mostra um exemplo de transferência usando esse processo. O emissor (cliente) primeiramente toma conhecimento do espaço disponível no *buffer* do receptor (servidor) na confirmação do último pacote enviado. O emissor, por sua vez, sabendo que não é possível enviar uma quantidade de dados que sobrecarregue o *buffer* do receptor, transmite pacotes com tamanho menor ou igual ao tamanho da janela recebido. O receptor recebe os dados e atualiza o tamanho de sua janela para enviar na mensagem de ACK. Assim, o processo se restabelece.

Caso o tamanho da janela informado seja nulo (zero), o emissor não irá enviar pacotes ao receptor. Ele fica aguardando a que os processos do receptor venham requisitar os dados armazenados em seu *buffer* de forma disponibilizar algum espaço. Assim que o *buffer* do receptor atinge um nível de espaço disponível, um aviso é enviado ao emissor informando o novo tamanho da janela de recepção. Todo esse processo é definido em detalhes na publicação (POSTEL, 1981).

Dessa forma, o cliente tem a percepção do estado da janela do servidor a cada ACK recebido e só enviará dados quando existir uma janela disponível.

2.2.2 Controle de congestionamento de pacotes

Quando a carga oferecida a qualquer rede é maior que sua capacidade, acontece um congestionamento. A Internet não é exceção a essa regra (TANENBAUM, 1997). 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.

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. Isso ocorre pois o congestionamento na rede é causado principalmente pelo comportamento exercido dos inúmeros computadores interligados a ela, os quais tentam transmitir dados na maior velocidade possível.

Existem, nesse nível, diversos algoritmos que foram adicionados para tornar o protocolo ainda mais eficaz nessa questão de tráfego de pacotes. Essas melhoras geraram ainda sub-versões de TCP nomeadas como Tahoe, Reno, New Reno, Vegas entre outros (ARIBA, 2005-2006). Em seguida, alguns algoritmos básicos serão explicados em razão de seus mecanismos interessantes para a abordagem do trabalho.

2.2.2.1 Algoritmo de Slow Start

Antigas implementações do TCP começavam uma conexão injetando múltiplos segmentos na rede, até o limite permitido pela janela anunciada pelo receptor. Este mecanismo não acarretaria problemas se os dois nós estivessem na mesma *Local Area Network* (LAN), pois a janela compartilhada entre o dois terá, em teoria, uma variação suave.

Contudo, se existem roteadores ou outras comutações de conexão entre a origem e destino, estes problemas podem vir a ocorrer em decorrência de um possível congestionamento. Algum roteador intermediário pode ficar sem espaço de *buffer* ou aplicar alguma política de descarte de pacotes. Em (JACOBSON, 1988), é mostrado como isto pode reduzir drasticamente o desempenho de uma conexão TCP.

O algoritmo usado para evitar esse problema é o chamado de *slow start*. Seu mecanismo se baseia na relação da taxa de pacotes e confirmações de recebimento que chegam ao emissor. Deve-se estabelecer um equilíbrio entre os pacotes injetados na rede e as confirmações (ACK) que são enviadas pelo destino.

O *slow start* requer que outra janela seja mantida pelo TCP emissor: a janela de congestionamento, chamada igualmente de *cwnd*. Quando uma nova conexão é estabelecida com outro *host* na rede, a janela de congestionamento é inicializada com um pacote, ou seja, o tamanho do pacote anunciado pelo nó oposto (tipicamente 536 ou 512 bytes). Toda vez que um novo ACK é recebido, a janela de congestionamento incrementa seu valor. Na prática, o tamanho dessa janela duplica a cada rajada confirmada (TANENBAUM, 1997).

A *cwnd* é mantida em bytes, mas o *slow start* sempre a incrementa em pacotes. O emissor pode transmitir até o mínimo entre a janela de congestionamento e a janela anunciada pelo receptor (janela de recepção, definida em 2.2.1). Basicamente, a janela de congestionamento é o controle de fluxo imposto pelo emissor, enquanto a janela anunciada é controle de fluxo imposto pelo receptor.

2.2.2.2 Algoritmo de Congestion Avoidance

Diferentemente do *slow start*, o qual faz com que a janela de congestionamento tenha um comportamento exponencial, o algoritmo de *congestion avoidance* permite um crescimento linear da *cwnd*. Esses dois comportamentos são importantes para tentar maximizar a quantidade de pacotes enviados sem que se produza um congestionamento.

Com o início da transmissão, o algoritmo de *slow start* é lançado fazendo a janela de congestionamento ter seu tamanho duplicado a cada ACK recebido. Como descrito anteriormente, a

quantidade de pacotes enviados é sempre o menor valor entre essa janela e a janela de recepção do destino.

Se, por algum motivo, o cronômetro de retransmissão do emissor expirar antes do recebimento do ACK do último pacote enviado, o TCP interpreta a situação corrente como o princípio de um congestionamento. Dessa forma, a janela de congestionamento é reinicializada (valor de um pacote) e o processo recomeça.

Contudo, existe uma diferença marcante quanto ao comportamento do protocolo após esse *timeout*. Como anteriormente, o algoritmo de *slow start* irá aumentar a *cwnd* exponencialmente, mas ao chegar ao valor correspondente à metade do valor que ocasionou a expiração do cronômetro, um outro algoritmo será executado. O valor da janela no qual houve expiração do cronômetro é definido como valor limiar ou *ssthresh*.

O *congestion avoidance* é iniciado então para manter o crescimento da janela de congestionamento com um comportamento linear. Assim, a cada confirmação de recebimento de pacote, a janela de congestionamento terá seu tamanho aumentada de um pacote. Isso é perseguido pois se procura manter a transmissão de dados em um fluxo crescente, sempre tentando utilizar o máximo de banda disponível, mas sem comprometer o comportamento geral da rede.

Caso ocorra novamente um *timeout*, o processo recomeça com a execução do *slow start* e, logo após a *cwnd* atingir o valor correspondente à metade do *ssthresh*, *congestion avoidance* é aplicado.

A figura 6 ilustra o funcionamento do algoritmo *slow start* combinado com o *congestion avoidance* em uma rede hipotética.

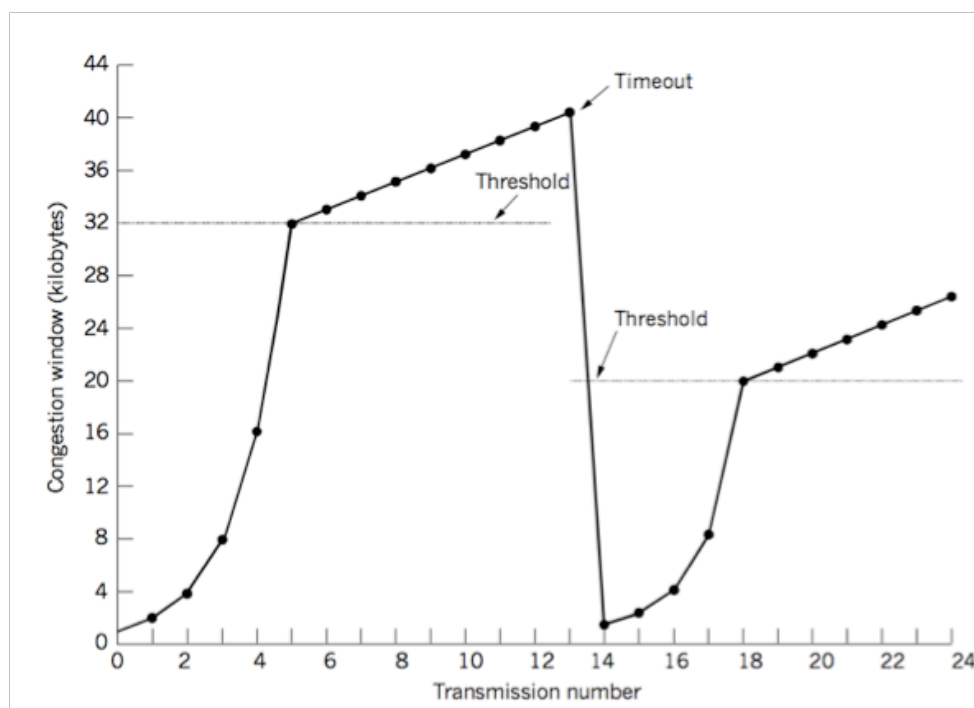


Figura 6: Algoritmo *slow start* combinado com o *congestion avoidance* (retirado de (TANENBAUM, 1997))

De uma forma sucinta, se a *cwnd* for menor ou igual a *ssthresh*, o TCP estará em *slow start*. Caso contrário, ele está realizando o *congestion avoidance*. O *slow start* prossegue até que a janela

de transmissão do TCP esteja com metade do tamanho de quando ocorreu o congestionamento, e, então, passa-se para a fase de *congestion avoidance*.

2.2.2.3 Algoritmo de Fast Retransmit

O algoritmo do *fast retransmit* foi implementado para evitar expiração do cronômetro de retransmissão (*timeout*). O TCP é obrigado a gerar uma confirmação imediata (um ACK duplicado) quando um segmento fora de ordem é recebido. A finalidade deste ACK duplicado é indicar ao emissor que um segmento foi recebido fora de ordem e qual o número de seqüência esperado.

Partindo do fato que não se sabe se um ACK duplicado foi causado por um segmento perdido ou somente uma reordenação de segmentos, espera-se que um pequeno número de ACK duplicados seja recebido antes que qualquer atitude seja tomada. É assumido que, se for somente uma reordenação de segmentos, só serão recebidos um ou dois ACK duplicados antes do segmento fora de ordem alcançar o destino e ser processado, o que implicará em um novo ACK .

Se três ou mais ACK duplicados forem recebidos em seguida, é um forte indício que um segmento foi perdido. O comportamento adquirido pelo TCP é executar o algoritmo de *fast retransmit*. A janela de congestionamento é então reduzida a metade do valor que estava quando houve o recebimento dos ACK e TCP requisita a retransmissão imediata do que aparenta ser o segmento perdido ao emissor, sem esperar que o cronômetro de retransmissão expire (*timeout*).

O comportamento seguinte adotado pela transmissão é a execução do algoritmo *congestion avoidance*. O crescimento linear da janela de congestionamento irá continuar até o aparecimento de um *timeout* ou de três (ou mais) ACK duplicados. Estes eventos que determinarão o comportamento seguinte, reinicializarão da janela e execução do *slow start* no caso de *timeout* ou início do *fast retransmit*, reduzindo o tamanho da janela a sua metade.

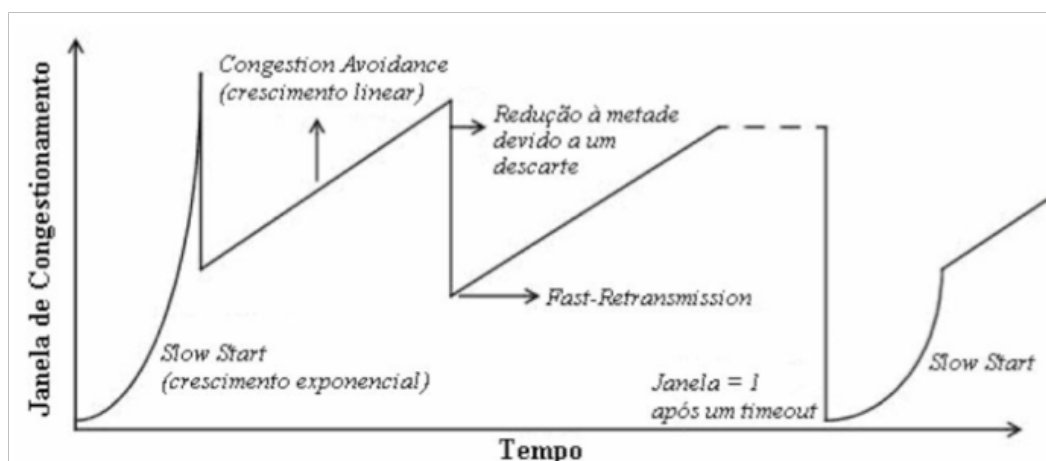


Figura 7: Algoritmo fast retransmit combinado com o slow start

A figura 7 ilustra o funcionamento desse algoritmo combinado com os algoritmos de *slow start* e *congestion avoidance*. Pode-se observar que, em um primeiro momento, só o algoritmo de *slow start* está agindo. Contudo, havendo o recebimento de três ou mais ACK duplicados, a janela de congestionamento é reduzida a metade do valor para a retransmissão do pacote requisitado e o *congestion avoidance* é iniciado. Os comportamentos seguintes da janela de congestionamento são determinados então pela ocorrência dos eventos de *timeout* ou ACK duplicados.

2.2.3 Atrasos nodais

Foi comentado que a ordem dos pacotes que chegam a um destinatário pode não ter a mesma ordem 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 (OLIVEIRA JESUS COSTA, 2006).

Pode-se separar os tipos possíveis de atraso em quatro diferentes: atraso de propagação, de fila, de processamento e 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.

Basicamente, 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 A e o roteador B. 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 sua vez, o atraso de transmissão é o tempo levado pelo roteador para verificar o cabeçalho do pacote e 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.

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. 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.

O atraso de fila pode ser definido como 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.

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.

O atraso de processamento é constituído de várias partes. As que se destacam mais são: o tempo necessário para o roteador A 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.

Assim, o atraso nodal é a soma dos vários tipos de atrasos descritos. Desta forma, denomina-se t_{fila} , t_{proc} , t_{trans} , t_{prop} , respectivamente, como os atrasos de fila, de processamento, de transmissão e de propagação.

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} . 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. O t_{fila} sofre forte influência da intensidade de tráfego na rede e da capacidade de armazenamento do *buffer*. Quanto maior a capacidade do *buffer* e sua posição dentro do mesmo, maior será t_{fila} .

3 Active Queue Management (AQM)

O presente capítulo tem por objetivo a apresentação dos chamados *Active Queue Management*, mecanismos de controle adicionais a nível de transporte implementados para funcionar em conjunto com outros protocolos de redes TCP/IP.

Em seguida, se aprofundará o estudo no AQM chamado *Random Early Detection (RED)*, o algoritmo de gerenciamento mais conhecido e estudado no momento. Aspectos técnicos sobre seu funcionamento serão apresentados assim como comparações de seu desempenho com outros mecanismos.

Por fim, a modelagem matemática da interação do fluxo de pacotes transferidos em uma rede TCP/IP e o algoritmo de *Active Queue Management* serão apresentados. Ainda, aqui será definido o sistema que será estudado e se apresentará o problema a ser analisado.

3.1 Introdução

No capítulo 2, apresentou-se os tipos de controle presentes em transmissões seguindo as regras do TCP. Concluiu-se que a camada de rede não fornece nenhum suporte sobre o comportamento do congestionamento uma vez que sua detecção é realizada através da análise de perda, retransmissão ou atraso de pacotes transmitidos. Entretanto, esse mecanismo de detecção de congestionamento por meio da análise do comportamento da rede pode muitas vezes levar ao descarte prematuro de pacotes além de subutilizar a capacidade dos enlaces (FLOYD; JACOBSON, 1993).

Devido à contínua dispersão de redes de computadores interligadas, o roteador tornou-se um dispositivo elementar para a transmissão de dados. Sua função é aplicar uma política de roteamento nos pedaços de informação (ditos quadros, pacotes, ou segmentos), aumentando a eficiência da transferência de dados na rede. Contudo, ainda com essa otimização, existem diversos aspectos que degradam a performance desse sistema. Devido ao grande número de rotas possíveis entre dois pontos na rede, é muito difícil prever duração de um processo de transferência de dados. Assim sendo, à comunicação associa-se um atraso variante no tempo (BENDER, 2006).

Ainda, o tráfego de dados entrante em um roteador pode freqüentemente superar sua capacidade de redirecionamento (a ver processamento). Com o objetivo de evitar perdas de informações, existe um espaço reservado para a armazenagem de dados chamado de *buffer*. Dessa maneira, uma fila de segmentos pode se formar quando a demanda excede a capacidade do roteador, o que acresce ainda mais a incerteza do tempo consumido na transferência.

Assim, as chamadas políticas de *Active Queue Management* foram estabelecidas para auxiliar no controle do tráfego de pacotes em roteadores. Deve-se ressaltar que elas se aplicam para o TCP, mas igualmente para diversos outros protocolos. Esses algoritmos agem diretamente no enfileiramento de pacotes formado no roteador, utilizando critérios previamente estabelecidos para o descarte seletivo de dados. Cada pacote nessa fila recebe uma probabilidade de descarte

associada.

Estes mecanismos agem de forma conjunta com os algoritmos apresentados anteriormente, ou seja, a lógica do TCP é mantida intacta. O princípio básico de transferência estabelecido pelo uso do parâmetro janela de congestionamento é completamente conservado. A única diferença é que, com um gerenciamento ativo de fila, se executará o controle pelo descarte seletivo de pacotes recebidos pelo equipamento nó da rede.

O descarte de pacotes atua diretamente no fluxo de pacotes de uma rede TCP/IP uma vez que, a cada confirmação de pacote não recebido pelo receptor, a janela de congestionamento estabelecida entre o emissor e receptor será diminuída e, em consequência disso, o tráfego entre eles.

É exatamente nesse gerenciamento que este trabalho aportará uma nova abordagem de controle, tentando desenvolver um novo método com o objetivo de tornar essa administração mais eficiente que as estratégias clássicas. Se usará a abordagem trazida por (FLOYD; JACOBSON, 1993) na definição do AQM *Random Early Detection* (RED), onde a marcação de pacotes é feita utilizando o *Explicit Congestion Notification* (ECN). O funcionamento detalhado desse mecanismo será detalhado a seguir.

Dessa maneira, aplica-se um controle de congestionamento intrínseco ao já existente em redes TCP/IP. Seguindo as regras estabelecidas pelo protocolo, um emissor que detecta a perda de seus pacotes enviados, diminui sua taxa de envio. Por esse motivo, os *Active Queue Management* compõem hoje em dia uma área de pesquisa ativa, envolvendo diversas áreas como engenharia de redes e até mesmo engenharia de controle. Cada vez mais procura-se o otimizar as transmissões de dados mantendo sempre o menor atraso possível e um fluxo de informações elevado no nível dos *buffers* dos roteadores de forma utilizar ao máximo os recursos disponíveis na rede.

3.2 AQM Random Early Detection (RED)

No início de sua implementação, os roteadores não possuíam algoritmo algum dirigido para o controle do nível de seus *buffers*. Quando havia uma saturação do *buffer*, o comportamento adotado era o de simplesmente descartar todos os próximos pacotes recebidos. Não existia uma seleção dos pacotes a serem desprezados, a ordem de chegada que definia o descarte. Assim, esse comportamento foi definido como *droptail*.

Como se pode imaginar, esse comportamento apresenta diversas desvantagens. Em algumas situações, o *droptail* 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*. Da mesma forma, ele 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.

O chamado RED (FLOYD; JACOBSON, 1993) foi um dos primeiros AQM a empregar um controle mais rigoroso na gestão de descarte de pacotes TCP. Com o objetivo de antecipar o congestionamento do roteador, cada pacote recebido e armazenado no *buffer* recebia uma probabilidade de descarte associada.

Quanto maior o tamanho médio fila do equipamento, maior o valor dessa probabilidade de descarte associada será. A equação (1) formaliza matematicamente a função associada. Uma representação gráfica dessa modelagem é apresentada na figura 8.

$$p(x) = \begin{cases} 0 & 0 \leq x \leq t_{min} \\ \frac{x-t_{min}}{t_{max}-t_{min}}p_{max} & t_{min} < x \leq t_{max} \\ 1 & t_{max} < x \end{cases} \quad (1)$$

Onde:

- x : tamanho médio da fila de espera (buffer)
- p_{max} , t_{min} e t_{max} : parâmetros que serão configurados

A figura 8 ilustra a função de probabilidade $p(x)$ definida. Com o valor p_{max} definido, se estabelece o nível máximo de probabilidade que ocasionará o descarte de pacotes. Para cada pacote recebido que fizer o valor médio do tamanho da fila de pacotes entrar no intervalo $[t_{min}, t_{max}]$ receberá uma probabilidade associada.

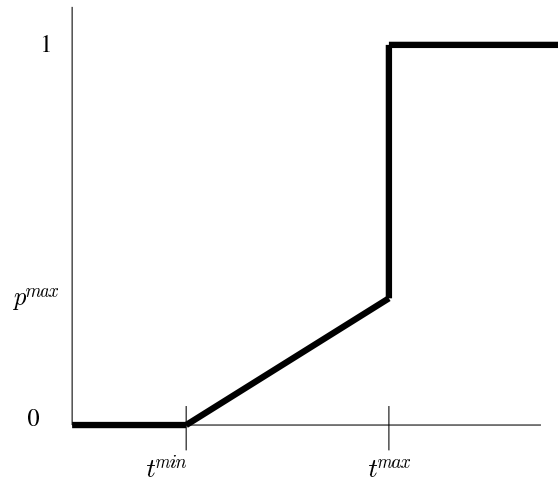


Figura 8: Função de probabilidade do RED (retirado de (MISRA; GONG; TOWSLEY, 2000))

Diversas vantagens surgem desse método. Mantendo o tamanho médio da fila pequeno, o AQM provê uma maior capacidade de absorver naturalmente a ocorrência de rajadas de pacotes sem descartá-los, ou seja, diminui a perda de dados na rede. Além do mais, 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.

Existem, entretanto, diversos problemas práticos em relação a esse algoritmo. A grande sensibilidade à configuração da rede e a difícil regulação de seus parâmetros tornam a eficiência desse método bastante discutida (ARIBA, 2005-2006).

3.3 Descrição do controle do RED

Em contraste com o tradicional comportamento do *droptail*, o qual descarta pacotes somente quando a fila do roteador está cheia, o algoritmo RED descarta os novos pacotes probabilística-

mente.

Esse algoritmo introduzido por (FLOYD; JACOBSON, 1993) foi altamente importante para a evolução dos métodos de gerenciamento de fluxo de dados. 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.

De acordo com MISRA em (MISRA; GONG; TOWSLEY, 2001), o RED pode ser visto como um filtro passa-baixa e uma função de descarte de pacotes. A figura 9 representa essa implementação na forma de um diagrama de blocos. O descarte aleatório de pacotes provocado por esta função não linear tem como objetivo eliminar a sincronização dos fluxos de dados. O filtro, por sua vez, através do cálculo da fila média, é introduzido para atenuar os efeitos das rajadas de pacotes.

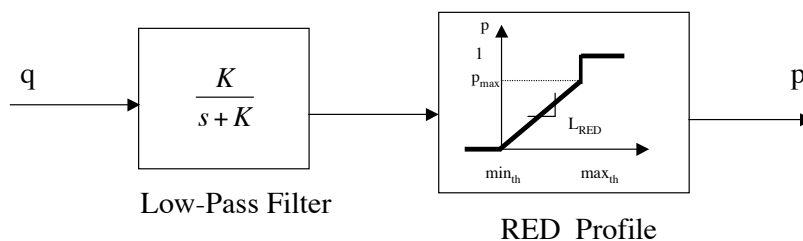


Figura 9: RED como filtro passa-baixa em cascata com função de descarte (retirado de (MISRA; GONG; TOWSLEY, 2001))

Nesse sentido, com o algoritmo RED, o roteador calcula o tamanho médio da fila q usando um filtro passa-baixa EWMA (*Exponential Weighted Moving Average*) sobre o tamanho instantâneo da fila q . O tamanho médio da fila é, então, comparado com dois limites: um limite mínimo t_{min} e um limite máximo t_{max} . 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 ressaltar que, quando se fala de marcação de pacotes no AQM RED, entende-se por descartar esse da fila do roteador (probabilidade associada $p = 1$) ou marcá-lo com uma probabilidade de descarte dentro dos limites estabelecidos.

Como o RED trabalha com a média da fila do roteador, ele não toma decisões levando em conta o comportamento instantâneo da mesma. 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.

3.4 Modelo dinâmico do AQM RED em uma rede TCP/IP

O trabalho proposto neste relatório consiste em desenvolver um controle mais eficiente para o fluxo de pacotes em redes TCP no nível dos roteadores, utilizando as leis de controle aprendidas durante o curso acadêmico e outras extensões mais elaboradas. Trata-se, logo, do desenvolvi-

mento de um novo AQM usando uma abordagem de controle.

Para tal, necessita-se de um modelo matemático descrevendo o comportamento da fila de pacotes no dispositivo comutador da rede sobre o qual toda a teoria de controle será desenvolvida. A performance do controle desenvolvido depende do grau de fidelidade desse modelo em relação ao processo físico real, já que as leis de controle são estabelecidas sobre este. Todos os limites físicos relevantes devem ser obrigatoriamente representados no modelo e respeitado pelas leis de controle. Caso contrário, o controle implementado pode gerar comportamentos imprevisíveis, fazendo com que a ação de controle não mais conduza o comportamento do sistema conforme o projeto original.

3.4.1 Resultados anteriormente obtidos

Este trabalho partirá de resultados bastante conhecidos obtidos pelos autores das publicações (MISRA; GONG; TOWSLEY, 1999) e (MISRA; GONG; TOWSLEY, 2000), os quais utilizaram a análise de fluxo-fluído e a teoria de Equações Diferenciais Estocásticas para modelar a interação entre os fluxos TCP e o *Active Queue Management* em uma rede. Ainda com um resultado de (MISRA; GONG; TOWSLEY, 2000), demonstrou-se que essas Equações Diferenciais Estocásticas podem ser transformadas em um conjunto de Equações Diferenciais Ordinárias.

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 AQM que é caracterizada por uma função de descarte de pacotes $p(\mu)$ que tem como argumento uma estimativa do tamanho médio da fila no roteador, obtida através de um filtro passa-baixa.

Esses desenvolvimentos se baseiam em duas hipóteses principais quanto o tráfego de pacotes e aos eventos de perdas de pacotes. Para o modelo desenvolvido em (MISRA; GONG; TOWSLEY, 1999), o tráfego de pacotes no nível do protocolo TCP é modelado como fluído. Em outras palavras, a janela de congestionamento evolui de forma contínua.

Da mesma forma, considera-se que as indicações de perdas de pacotes podem ser descritas como processos de Poisson. Assim, um contador de Poisson pode ser modelado para representar as indicações de perdas de pacotes que chegam ao remetente da conexão. Assim, parte-se da premissa que a transmissão está sempre sendo efetuada com a execução do algoritmo *congestion avoidance* e/ou *fast retransmit*. O caráter não linear do *slow start* não é considerado (mais detalhes, ver capítulo 2).

$$dN \begin{cases} 1 & \text{caso exista perda} \\ 0 & \text{caso contrario} \end{cases} \quad (2)$$

A extensão do desenvolvimento para a modelagem do AQM RED pôde ser efetuada em (MISRA; GONG; TOWSLEY, 2000). Nessa obra, modela-se o caso do fluxo de pacotes de conexões TCP passando por um roteador congestionado. As mesmas premissas levantadas pelo modelo anterior são mantidas.

A partir dessas proposições, a equação modelando a taxa de crescimento da janela de congestionamento em um emissor i foi estabelecida como

$$dW_i(t) = \frac{dt}{R_i(q(t))} + dN_i(t) \left(-\frac{W_i(t)}{2} \right) \quad (3)$$

$$R_i(t) = \frac{q(t)}{C}, \quad i = 1, \dots, N \quad (4)$$

onde

- $R_i(q(t))$: soma de todos os atrasos de transferência do pacote
- $q(t)$: tamanho instantâneo da fila de pacotes no roteador
- $dN_i(t)(-\frac{W_i(t)}{2})$: redução da janela em função da reprodução de um evento de erro na transmissão (três ACK duplicados ocasionando a execução do algoritmo *fast retransmit*)
- N : número de conexão TCP passando pelo roteador
- C : capacidade do *link* entre o roteador e destino dos pacotes

Deve-se notar que, apesar de pouco visual, a expressão (3) tem um significado bastante intuitivo, pois ela descreve exatamente o funcionamento dos algoritmos de controle de fluxo do TCP apresentados anteriormente em 2.2.1. O primeiro termo da equação representa o aumento-aditivo da janela ocasionada pelo envio do pacote e o segundo como sendo o termo de diminuição-multiplicativa causado por um descarte de pacote (ACK duplicados).

Da mesma forma, utilizando uma forma diferencial da equação de LINDLEY, obteve-se a modelagem do comportamento do tamanho da fila de pacotes no roteador

$$\frac{dq(t)}{dt} = -1_{q(t)}C + \sum_{i=1}^N \frac{W_i}{R_i(q)} \quad (5)$$

onde pode se definir

- $-1_{q(t)}$: parcela que representa o decrescimento da fila por pacotes servidos
- $q(t)$: tamanho instantâneo da fila de pacotes no roteador

Foi então com o desenvolvimento da média (esperança) da equação (3) e (5) no trabalho (MISRA; GONG; TOWSLEY, 2000) que se obteve um modelo dinâmico do comportamento da fila de pacotes de um roteador imerso em uma rede TCP/IP em equações diferenciais ordinárias. O modelo é definido por

$$\begin{cases} \dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)) \\ \dot{q}(t) = \frac{W(t)}{R(t)}N(t) - C \\ R(t) = \frac{q(t)}{C} + T_p \end{cases} \quad (6)$$

com os termos

- $W(t)$: tamanho médio da janela de congestionamento do TCP medido em pacotes
- $q(t)$: tamanho médio da fila de espera medido em pacotes

- $R(t)$: round trip time (RTT) medido em segundos
- C : capacidade do link medido em pacotes por segundo
- T_p : atraso de propagação medido em segundos
- N : fator de carga, ou seja, numero de conexões TCP
- $p(t)$: probabilidade de descarte associada do pacote

Assim, este modelo representa o valor médio das variáveis chaves de uma rede e é descrito por equações diferenciais ordinárias (ODEs). Pode-se notar que, mesmo após a imposição da simplificação em relação a possíveis *timeouts*, o modelo resultante exige ainda a necessidade de uma análise não linear.

Através da definição física dessas variáveis pode-se dizer que q e W são variáveis sempre positivas e limitadas. Sendo assim, pode-se fazer as seguintes considerações (ARIBA, 2005-2006):

- $q \in [0, \bar{q}]$: é a capacidade máxima do *buffer*
- $W \in [0, \bar{W}]$: é a capacidade máxima da janela de congestionamento

A variável p foi introduzida durante a modelagem do algoritmo AQM do RED e constituirá a entrada do sistema. Trata-se da probabilidade de descarte associada ao pacote da fila do *buffer*.

3.4.1.1 Limitações do modelo

Embora o modelo proposto por (MISRA; GONG; TOWSLEY, 2000) representar uma grande contribuição para a modelagem de uma rede seguindo o protocolo TCP de controle de pacotes, ele apresenta ainda alguns problemas que restringem seu domínio de aplicações.

O modelo matemático proposto se baseia sobre o princípio da fluidez do tráfego de pacotes dentro da rede. O crescimento do tamanho da janela de congestionamento é considerado contínuo ao invés de um crescimento do tipo degrau. Existe um incremento de 1 pacote em cada *Round Trip Time* (RTT) e assim o crescimento contínuo é representado por dt/RTT (MISRA; GONG; TOWSLEY, 1999).

Em relação ao seu decréscimo, assume-se um comportamento ideal podendo ser representado por um contador de Poisson. Dessa maneira, para os casos de perdas de pacotes, três ACK duplicados são enviados fazendo o tamanho da janela ser reduzido à metade (MISRA; GONG; TOWSLEY, 1999).

Além disso, tem-se um grande problema quanto a modelagem do evento chamado de *timeout*. Quando há uma probabilidade de descarte muito elevada, existe a possibilidade de que muitos pacotes venham a ser descartados e assim o mecanismo de *slow start* do protocolo é lançado. Como esse mecanismo faz com que a janela de congestionamento adquira um comportamento exponencial, a modelagem dessa variação é invalidada.

3.4.2 Linearização do modelo

Como visto na equação (6), o comportamento dinâmico do TCP é modelado por um sistema de equações diferenciais não lineares. Como este trabalho tem o interesse em uma abordagem do

comportamento do sistema com atraso, mas operando próximo ao seu ponto de equilíbrio, faz-se uma linearização dessas expressões.

Com o objetivo de deixar a leitura mais fluida e menos pesada, todo o desenvolvimento aqui do resultado apresentado pode ser encontrado nas seções anexas (apêndice B). Considerando-se para esse modelo o número de conexões TCP $N(t)$ e a capacidade do *link* conexão $C(t)$ constantes assim como $[W, q]$ os estados, pode-se estabelecer as seguintes expressões para quando se tem o processo em equilíbrio (W_o, q_o, p_o) :

$$\begin{cases} \dot{W}(t) = 0 & \implies W_o^2 p_o = 2 \\ \dot{q}(t) = 0 & \implies W_o = \frac{R_o C}{N}, \quad R_o = \frac{q_o}{C} + T_p \end{cases} \quad (7)$$

Assim, em torno dos pontos de equilíbrio anteriormente definidos, podemos aproximar a equação (6) pelo sistema linear (8) representado já em espaço de estados

$$\begin{aligned} \begin{bmatrix} \delta \dot{W}(t) \\ \delta \dot{q}(t) \end{bmatrix} &= \begin{bmatrix} -\frac{N}{R_o^2 C} & -\frac{1}{R_o^2 C} \\ \frac{N}{R_o} & -\frac{1}{R_o} \end{bmatrix} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} + \begin{bmatrix} -\frac{N}{R_o^2 C} & -\frac{1}{R_o^2 C} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta W(t - R(t)) \\ \delta q(t - R(t)) \end{bmatrix} \\ &+ \begin{bmatrix} -\frac{C^2 R_o}{2N^2} \\ 0 \end{bmatrix} [\delta p(t - R(t))] \end{aligned} \quad (8)$$

onde $\delta W = W - W_o$, $\delta q = q - q_o$ e $\delta p = p - p_o$ representam as variações dos estados e da entrada do sistema em torno de seus respectivos pontos de equilíbrio. Se supõe os sinais δW , δq e δp possuem amplitude pequena.

Os autovalores desse sistema podem ser encontrados sem complicações algébricas pesadas. Eles assumem a forma de

$$\lambda_1 = -\frac{2N}{R^2 C} \text{ ou } \lambda_1 = -\frac{2}{W_o R} \text{ e } \lambda_2 = -\frac{1}{R} \quad (9)$$

Esse modelo linear foi desenvolvido em (ARIBA, 2005-2006), onde se utilizou o método de linearização similar ao apresentado em (HOLLOT et al., 2002) a não ser pelo fato desse último aplicar uma simplificação de atraso constante $h(t) = h$. Embora este trabalho se interessar em uma abordagem com atrasos invariantes no tempo, a linearização trazida pela primeira obra é a demonstrada aqui por se apresentar mais geral e útil para a aplicações futuras que levam em consideração essa variação.

Exemplo 3.4.1 (Exemplo numérico). *A fim de analisar o comportamento do modelo linearizado, realizou-se uma simulação no programa Matlab. Os parâmetros utilizados foram similares aos descritos por (ARIBA, 2005-2006) e retirados de (HOLLOT et al., 2001, 2002), considerando $R_o = 0.246$ segundos, $C = 3750$ pacotes/s, $N = 60$ conexões e um $q_o = 175$. O sistema toma a forma de*

$$\begin{aligned} \begin{bmatrix} \delta \dot{W}(t) \\ \delta \dot{q}(t) \end{bmatrix} &= \begin{bmatrix} -0.2644 & -0.0044 \\ 243.9024 & -4.0650 \end{bmatrix} \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} \\ &+ \begin{bmatrix} -0.2644 & -0.0044 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta W(t - R(t)) \\ \delta q(t - R(t)) \end{bmatrix} + \begin{bmatrix} -480.47 \\ 0 \end{bmatrix} [\delta p(t - R(t))] \end{aligned} \quad (10)$$

Para esses dados, tem-se como pontos de equilíbrio do sistema $W_o = 15$ e $p_o = 0.008$. Utilizando uma função step iniciando no instante $t = 0s$, voltando a zero em $t = 0.246s$ e com amplitude $\delta p = 0.003$ para a entrada do sistema, obteve-se o comportamento das variáveis de estado δW e δq mostrado nas figuras 10 e 11 respectivamente.

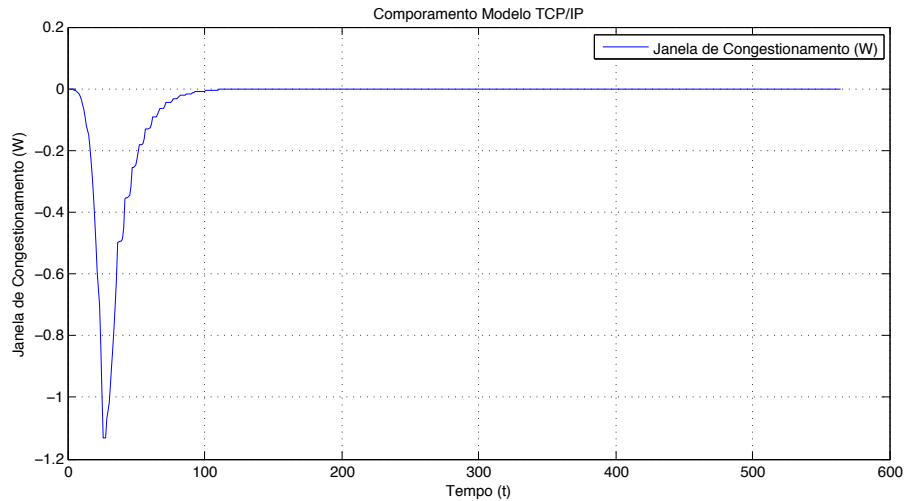


Figura 10: Comportamento da janela de congestionamento W com a excitação de um degrau limitado

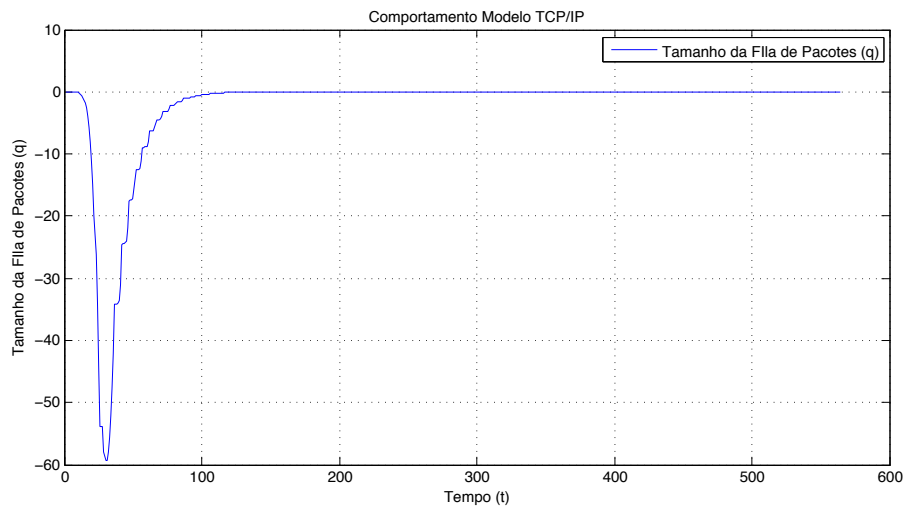


Figura 11: Comportamento da fila de pacotes q com a excitação de um degrau limitado

Deve-se notar o retorno do sistema ao seus pontos de equilíbrio W_o e q_o após a perturbação exercida. A comprovação da estabilidade em malha aberta do sistema pode ser observada.

3.4.3 Análises sobre modelo linearizado

Toda e qualquer linearização de um modelo matemático possui limites de validade. O caso da modelagem do comportamento do protocolo TCP não é diferente. Todo o desenvolvimento proposto pela equação (8) é válido para variações dos sinais pequenas em torno do ponto de equilíbrio identificado.

Contudo, existem diversos fatores intrínsecos no comportamento do protocolo que podem vir a invalidar a modelagem. Descreve-se a seguir alguns desses fatores.

3.4.3.1 Presença de atraso

Como indicado anteriormente em 2.2.3, o protocolo TCP apresenta um atraso originário de sua topologia funcional. Isto é, a resposta de recebimento de um pacote não se dá instantaneamente do receptor ao emissor. Existe um intervalo de tempo determinado que irá se esgotar antes que o ACK seja recebido (ou não, em caso de perda/descarte do pacote).

Assim, pode-se dizer que existe um atraso em seu estado e controle. Este atraso pode vir a variar bruscamente e em uma grande faixa de intervalo de tempo.

Essa característica do atraso de variar em grandes faixas de tempo, pode ser abordada de duas formas diferentes. Uma delas é utilizando uma metodologia de análise independente desse atraso. Existem trabalhos na literatura, como em (ARIBA, 2005-2006), os quais abordam essa metodologia para análise de sistemas desse tipo. Entretanto, os resultados não são satisfatórios devido ao fato de se tratar de uma abordagem muito restritiva e conservadora, em geral.

Da mesma forma, existem métodos mais flexíveis que permitem uma análise mais realista do problema. As abordagens dependentes do atraso trazem ferramentas extremamente úteis para a síntese de leis de controle com um reduzido grau de conservadorismo. É, inclusive, essa abordagem que este trabalho irá utilizar para a análise do problema de controle de pacotes em um roteador imerso em uma rede TCP/IP.

3.4.3.2 Presença de saturação

O fenômeno de saturação está presente na ação de controle do sistema considerado. Devido ao significado físico do sinal de controle (uma probabilidade de descarte), existe uma limitação intrínseca, pois $p(t) \in [0, 1]$.

Uma vez linearizado o sistema em torno do ponto de equilíbrio, a ação de controle passa a ser a variação da probabilidade de descarte em torno do valor de equilíbrio p_o e não o valor propriamente dito $p(t)$. Tem-se, então $p(t) = p_o + \delta p(t)$.

O sistema fica com limites claramente assimétricos para $\delta p(t)$. Nestas condições, assumir limites simétricos, ignorando toda a ação de controle que excede no outro sentido o mais estreito dos limites, é renunciar a grande amplitude de esforço de controle. Não obstante, pelas características do TCP, desvios significativos do ponto de equilíbrio, além de invalidar o modelo, podem induzir no protocolo um comportamento claramente exponencial (o chamado *slow start*), o que invalidaria totalmente a modelagem, inutilizando a ação de controle disponível (BENDER, 2008).

De uma certa forma, a saturação contribui para manter o modelo linearizado proposto válido durante a operação do protocolo. Caso houvesse o descarte sucessivo de pacotes em diferentes transmissões, eventos de *timeout* poderiam ser ativados e, como consequência, o regime de *slow start*, tornando o comportamento do TCP altamente não linear.

3.4.3.3 Presença de não linearidades

Como visto no capítulo 2, o TCP tem um comportamento por construção não linear. Existem períodos nos quais seu funcionamento é praticamente linear, mas eventos de *timeout*, por exemplo, podem fazer com que seu comportamento fuja desse princípio.

Dessa maneira, o sistema linear proposto é válido somente quando se investiga aspectos do comportamento do protocolo em uma estreita vizinhança em torno do equilíbrio. Desvios do ponto de equilíbrio mediante incrementos nos descartes podem conduzir o sistema a um evento de *timeout*, o que dispara o regime de *slow start* no TCP e invalida toda a modelagem linear estabelecida.

Existem trabalhos que tratam do problema de congestionamento do protocolo TCP com uma abordagem de controle, como o desenvolvido em (ARIBA, 2005-2006) e (PARK; YOON; YOON, 2004). Este último desenvolveu um compensador *anti-windup* dinâmico para mitigar os efeitos nefastos provocados pela presença de saturação no sinal de controle desse sistema.

Este trabalho, contudo, pode ser dito com uma extensão dos trabalhos de (ARIBA, 2005-2006) e (PARK; YOON; YOON, 2004), uma vez que se tem como objetivo principal o aperfeiçoamento do controlador PI proposto em (HOLLOT et al., 2002). Será considerada a saturação de controle e eventuais perturbações que não haviam sido levados em conta até então.

Essa abordagem é altamente válida visto que o fenômeno de saturação está constantemente presente nesse processo de gerenciamento. A variável sobre a qual o controle é exercido tem um valor máximo e mínimo. Isto é, a probabilidade de descarte associada ao pacote tem um valor mínimo de 0 e máximo de 1, não importando o estado do *buffer* do roteador.

O exemplo evidente seria o comportamento do sistema quando se tem o *buffer* do roteador quase totalmente preenchido e apresentando uma taxa de recepção de pacotes extremamente grande. Para impedir a saturação da fila, não seria possível elevar a probabilidade de descarte a um valor superior a 1. Logo, o controlador irá permanecer nesse valor limite (irá saturar durante um certo intervalo de tempo) até que o sistema volte ao ponto de equilíbrio que foi estabelecido. O inconveniente desse processo é que, como se apresentará na seqüência deste trabalho, essa situação traz diversos pontos negativos para o desempenho do sistema.

Adicionalmente, a presença de perturbações no tráfego de redes TCP/IP é um evento que degrada de forma significativa a modelagem do comportamento do fluxo de congestionamento. Isso ocorre pois existem diversos outros fluxos de outros protocolos que não podem ser controlados efetivamente pelo descarte seletivo de pacotes ou pela marcação. A título de exemplo, fluxos de vídeo regidos pelo protocolo de transporte UDP não são responsivos à esse controle.

4 Controle de sistemas com atraso sob saturação

Nos últimos anos, muitos autores têm se dedicado ao desenvolvimento de pesquisas na área de controle e estabilidade de sistemas com atraso. Esse fato é absolutamente válido visto que uma grande parte de sistemas físicos pode ser modelada por equações diferenciais apresentando essa característica (GOMES DA SILVA JR. et al., 2009).

Adicionalmente, a dificuldade de controle de sistemas com atraso é ainda aumentada quando o sinal de controle possui limitações em sua amplitude. Devido à impossibilidade física de se transferir amplitude ou energia ilimitada do atuador para a planta, essa restrição prática deve ser considerada de forma se obter uma lei de controle menos conservadora e com melhor desempenho.

Neste capítulo, o foco é voltado então para a apresentação de teorias de controle para sistemas com atraso e sob efeito de algum tipo de saturação. Uma revisão de conceitos será estabelecida assim como uma comparação com os trabalhos realizados e pesquisas em desenvolvimento. Ambos casos de sistemas sob saturação e sistemas com atraso serão tratados e analisados.

4.1 Sistemas sob saturação

A implementação da lei de controle dá-se através de um controlador, e invariavelmente utilizam-se sensores e atuadores. A saturação do atuador ocorre quando o sinal de controle calculado excede os limites físicos do sinal possível ao atuador alimentar a planta. Como descrito anteriormente, essa ocorrência é absolutamente coerente na medida que, na prática, é fisicamente impossível transferir uma quantidade de energia ou amplitude ilimitada para o sistema (BENDER, 2006).

Em um modelo matemático para um fenômeno físico, a ocorrência imprevista dessa saturação pode levar o sistema à instabilidade ou ao aparecimento de respostas indesejáveis inerentes ao comportamento não-linear do mesmo em malha fechada.

Embora não seja dada a devida importância a esse fenômeno por alguns autores, já existem diversos trabalhos que consideram a saturação do atuador. Eles são divididos em duas abordagens principais. Na primeira, chamada de *Síntese Direta*, leva em consideração o efeito da saturação do sistema desde a síntese do controlador. Considerações sobre a estabilização do sistema em malha fechada são feitas em contextos globais, semi-globais e locais, vide (BENDER, 2006; SUSSMANN; SONTAG; YANG, 1994).

Já no segundo método, pressupõe-se o controlador previamente implementado e se tem como principais objetivos a garantia da estabilidade do sistema em malha fechada e o estabelecimento de um nível mínimo de desempenho. São as chamadas técnicas *anti-windup* nas quais um grande esforço de síntese tem sido aplicado atualmente. Pode-se citar diversos trabalhos, dentre eles (BENDER, 2008; GOMES DA SILVA JR.; TARBOURIECH; GARCIA, 2006; GOMES DA SILVA JR.; TARBOURIECH, 2005).

Em (GOMES DA SILVA JR.; TARBOURIECH; GARCIA, 2006), a síntese do *anti-windup* tem um objetivo suplementar à recuperação do desempenho nominal do sistema apesar da presença da saturação. Busca-se projetar um compensador *anti-windup* que maximize a estimativa da região de atração da origem em malha fechada. Nestes trabalhos, o processo de síntese de compensador utiliza uma abordagem baseada em LMIs. No apêndice A.1, alguns conceitos sobre LMIs são explicados para não sobrecarregar a leitura. Contudo, essas publicações não consideram o efeito de perturbações atuantes sobre o sistema.

Um resultado bastante significativo foi obtido em (GOMES DA SILVA JR.; TARBOURIECH, 2005), no qual estabeleceu-se uma condição de setor generalizada. Esta, por sua vez, permite a obtenção de condições suficientes para a estabilidade da origem do sistema em malha fechada já sob forma de Desigualdades Matriciais Lineares (LMI).

O trabalho aqui proposto, devido aos interesses relacionados à aplicação dos conceitos no controle de pacotes de uma rede TCP, seguirá a segunda abordagem de desenvolvimento. Assim, para minimizar esse efeito sobre o laço de controle e garantir a estabilidade do sistema em malha fechada, será sintetizado um compensador *anti-windup* estático para o sistema proposto.

O método seguido baseia-se no uso de um funcional de Liapunov-Krasovskii (NICULESCU, 2001) aliada a condição de setor generalizada, como proposto em (GOMES DA SILVA JR.; TARBOURIECH, 2005). A partir destas referências, condições suficientes para a existência de um compensador são expressas diretamente em forma de LMI, de modo a assegurar a estabilidade de entrada e saída do sistema, bem como a estabilidade assintótica interna da origem do sistema em malha fechada, na ausência perturbações.

Adicionalmente, o trabalho se propõe a melhorar a resposta do sistema quando o mesmo encontra-se submetido a perturbações quando a saturação ocorre. Como as condições suficientes encontradas formam um problema convexo, procura-se otimizar os resultados para duas situações:

- Maximizar a norma da maior perturbação possível, de forma se obter um resultado para o problema de *tolerância à perturbações*.
- Minimizar o ganho superior \mathcal{L}_2 da transferência da perturbação para a saída, o chamado problema de *rejeição à perturbações*.

4.1.1 Compensação *anti-windup*

Em controle de processos, freqüentemente existem casos onde o elemento final de controle pode vir a ficar saturado. Um exemplo clássico desse caso é o controle de fluxo de um encanamento utilizando válvulas. É evidente que, em algum momento, chega-se a situação onde a válvula encontra-se totalmente aberta ou fechada. Não importa a ação requisitada ao controlador, o limite físico foi alcançado e a vazão não terá como ser aumentada. Diz-se então que houve a saturação do controle.

Em processos controlados por controladores proporcional-integral a performance é bastante degradada pela presença de saturação. Em geral, observa-se um comportamento mais lento de acomodação e a ocorrência de oscilações da resposta do sistema em relação à mudança do parâmetro de controle. Isso ocorre devido à componente integral presente no processo:

$$I = \frac{K_c}{T_i} \int_0^t e(t) dt = \frac{K_c}{T_i} \int_0^t [y^{sp}(t) - y(t)] dt \quad (11)$$

Como o controlador não pode ser implementado para agir além dos valores limites da saída do atuador, durante uma situação como essa, as variáveis manipuladas permanecem em um dos valores extremos. A figura 12 ilustra a situação descrita na forma de um diagrama de blocos.

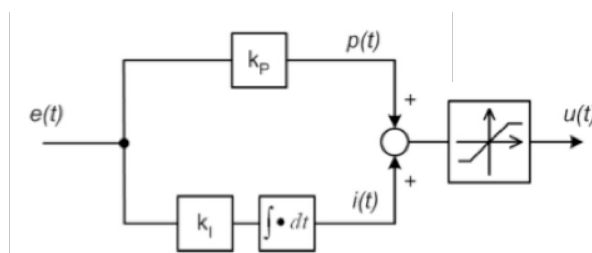


Figura 12: Diagrama de blocos de um sistema com saturação

Devido a estabilidade nesse valor limite, o erro do controlador irá crescer continuamente, o que faz com que o valor do termo integral cresça até o momento em que o erro troque de sinal. Esse comportamento traz desvantagens indesejáveis na resposta do sistema, pois a resposta transitória do sistema tenderá a ficar extremamente lenta e oscilatória.

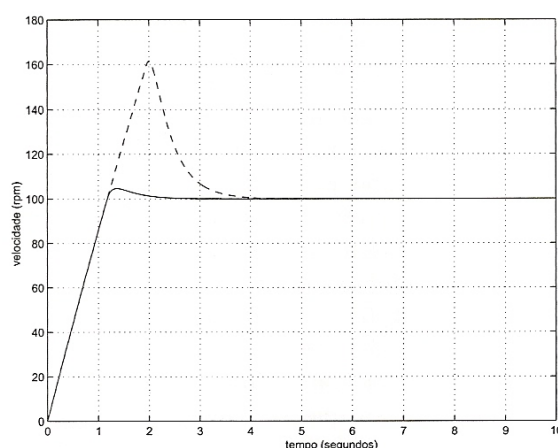


Figura 13: Sinal de saída do processo com e sem compensador *anti-windup* (retirado de (GOMES DA SILVA JR.; BAZANELLA, 2005))

A figura 13 ilustra uma comparação entre o sinal de saída de um sistema genérico sujeito ao efeito de saturação. No gráfico pontilhado, a saída do sistema sem um compensador é representada. A curva em traço contínuo representa a resposta do sistema com um compensador *anti-windup*. Utilizando o mesmo princípio, a figura 14 ilustra para o caso do sinal de controle.

Existem várias maneiras de se evitar esse fenômeno, mas a mais comum é a implementação de um compensador que irá atuar na atualização dinâmica do valor do termo integral quando ocorrer a saturação do sinal de controle. A diferença do valor de entrada e da saída do atuador constituem em um erro e_s que é realimentado à entrada do integrador com um ganho $K_s = 1/T_i$ (GOMES DA SILVA JR.; BAZANELLA, 2005).

Deve-se notar que, quando esse erro de saturação é nulo, o laço não influencia no comportamento do sistema. Contudo, com a presença de saturação, e_s é diferente de zero e o sinal aplicado no integrador não será mais $e(t)K_f$ e sim $e(t)K_f + e_s(t)K_fK_s$. A figura 15 mostra o diagrama de blocos do sistema resultante.

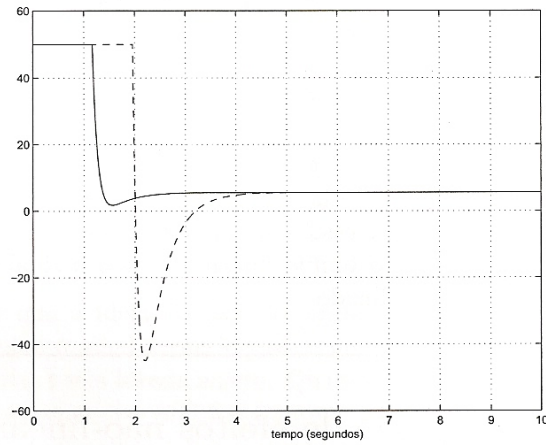


Figura 14: Sinal de controle do processo com e sem compensador *anti-windup* (retirado de (GOMES DA SILVA JR.; BAZANELLA, 2005))

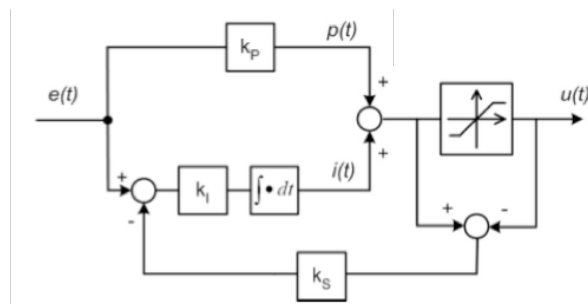


Figura 15: Comportamento de um sistema com compensador *anti-windup*

Assim, de uma forma resumida, a síntese de um compensador *anti-windup* tem dois objetivos principais. O primeiro pode ser definido como a minimização da degradação de performance do sistema devido a ação integral explicada anteriormente. O outro visa justamente aumentar a região de atração do sistema em malha fechada para garantir sua estabilidade (BENDER, 2006).

Nas seções que seguem, se complementar-se esse princípio básico com teorias modernas e mais sofisticadas para o desenvolvimento de uma ferramenta teórica com o objetivo de aplicar as teorias de controle em redes TCP.

4.2 Sistemas com atraso

Como apresentado na introdução deste capítulo, muitos dos problemas físicos de interesse podem ser modelados por equações diferenciais funcionais. Contudo, com o objetivo de tornar o modelo mais fiel possível ao fenômeno físico, freqüentemente essas equações apresentam características, como não linearidades e atrasos temporais, que podem vir a necessitar de um tratamento particular para o estabelecimento das leis de controle.

Os atrasos temporais podem aparecer em um sistema como retardos nas variáveis de estado, entrada ou saída. Esses tipos de sistemas são chamados de *sistemas com atraso*. Da mesma forma, existem casos onde o atraso pode estar presente nas derivadas das variáveis de estado. São os ditos *sistemas neutros*. Na síntese descrita em (GOMES DA SILVA JR.; TARBOURIECH; GARCIA, 2006), esse último tipo de sistema é analisado. Novamente, com o objetivo de focar o estudo na aplicação em controle de redes TCP, a síntese proposta nesse trabalho tratará especificamente do caso de sistemas com atraso.

Adicionalmente, sabe-se que a presença desse tipo de comportamento em sistemas de controle pode acarretar em baixo desempenho no domínio tempo além de poder deixar o sistema em malha fechada instável. Assim, considerando esses parâmetros, duas abordagens principais podem ser utilizadas para avaliar a estabilidade, performance e robustez do sistema em malha fechada: análise dependente do atraso e análise independente do atraso.

Nos métodos independentes do atraso, a estabilidade do sistema é estudada independentemente do valor do atraso. As condições independentes do atraso são capazes de garantir, quando satisfeitas, a estabilidade de sistemas com atraso para qualquer valor positivo de τ . Dentre publicações que utilizaram essa abordagem para o tratamento de sistemas com atraso pode-se citar (GOMES DA SILVA JR.; TARBOURIECH, 2000; CAO; LIN; HU, 2002; ARIBA, 2005-2006).

Com o objetivo de melhorar a análise da natureza do sistema e assim tornar os resultados menos conservadores (em geral), uma abordagem dependente do atraso pode ser adotada. Nessa técnica, informações sobre a ordem de grandeza do atraso são levadas em conta na análise da estabilidade do sistema. Trabalhos como (ARIBA, 2005-2006), além de apresentar uma abordagem independente do atraso, trazem igualmente uma análise considerando restrições sobre o atraso. Em (GOMES DA SILVA JR.; TARBOURIECH; GARCIA, 2006), a síntese de um compensador *anti-windup* é feita levando em conta a saturação de controle assim como a presença de atrasos formulando as condições diretamente em LMIs.

4.3 Teoria de Lyapunov

Resumidamente, pode-se separar o tipo de estabilidade de um sistema em três categorias diferentes: estabilidade global, semi-global e local. Para a descrição de cada um desses tipos, a definição de *região de atração da origem* deve ser estabelecida. Segundo (KHALIL, 1996), região de atração corresponde ao conjunto de todos os pontos do espaço da dimensão do sistema em questão onde o mesmo pode ser iniciado tendo todas suas trajetórias convergindo para origem na ausência de perturbações. Resumidamente, são os pontos no espaço onde o sistema permanece estável.

Na maioria dos casos, busca-se a estabilidade global da planta considerada. Nesse contexto, a região de atração toma a dimensão de todo o espaço do sistema. Para qualquer ponto do espaço considerado na inicialização do sistema, as trajetórias do mesmo convergirão em direção a origem. Todavia, leis de controle que visam a estabilidade global da planta, freqüentemente, apresentam um comportamento dinâmico insatisfatório. Adicionalmente, visto que estabilidade global não pode ser alcançada por sistemas lineares com presença de saturação instáveis em malha aberta, outros resultados devem ser considerados.

A estabilidade semi-global pode ser definida como uma região de atração da origem limitada no espaço do sistema. Pode-se estabelecê-la tão grande quanto se queria, contudo sua dimensão permanecerá sempre limitada a um valor.

Finalmente, o caso de estabilidade local é tratado como a única opção quando o sistema em questão possui autovalores instáveis em malha aberta. Dessa forma, a estabilidade é garantida dentro de uma região restrita. À primeira vista, essa categoria parece extremamente restritiva, pois a garantia da estabilidade é afirmada somente para um limitado conjunto de valores. Contudo, considerando-se o fato de a maioria dos sistemas lineares analisados é derivado de uma estrutura não linear linearizada em torno de um ponto de equilíbrio e que essa abordagem é válida somente para a operação na vizinhança desses valores, a premissa não demonstra um grau de conservadorismo alto.

Lyapunov, em 1899, desenvolveu dois métodos para investigar a estabilidade de um sistema. O primeiro, método indireto de Lyapunov, permite investigar a estabilidade local de um sistema não linear através de seu modelo linearizado. O segundo método, conhecido como método direto de Lyapunov, é baseado em um conceito análogo ao de energia. Estes métodos são de grande relevância prática, pois servem de base para o projeto de controladores estabilizantes para sistemas dinâmicos.

A idéia central da teoria de Lyapunov é descobrir uma função positiva definida que dependa dos estados do sistema 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 dessas expressões fornecem condições necessárias e suficientes para a análise da estabilidade do sistema nominal.

Utilizando uma analogia ao meio físico do sistema, a função de Lyapunov é uma abstração matemática da energia do sistema. Assim, as relações estabelecidas anteriormente são facilmente compreendidas uma vez que somente sistemas que dissipam energia podem ser estáveis.

4.4 Perturbações

Dentro dos critérios de desempenho exigidos a um sistema com estabilidade local, a performance em relação à perturbação merece destaque pela frequência com que é encontrada em processos reais. Nestes termos, a tolerância a uma perturbação com determinada norma \mathcal{L}_2 pode ser exigida dentro de uma dada região. A prática comum é, então, garantir-se que a ação de uma perturbação com norma \mathcal{L}_2 finita e conhecida sobre o sistema não conduza suas trajetórias em malha fechada para fora da região de atração da origem.

Por outro lado, a rejeição à perturbação avalia a capacidade que o sistema tem de atenuar em um ou mais de seus terminais de saída o efeito de uma perturbação limitada em norma \mathcal{L}_2 a que esteja sujeito em um ou mais de seus terminais de entrada. Esta rejeição pode ser quantificada através de um limitante superior para o ganho \mathcal{L}_2 que a perturbação apresente aos terminais de saída do sistema de interesse à esta questão.

Deve-se notar que são duas formulações de problemas de otimização distintas e chegarão, normalmente, em resultados diferentes. A síntese descrita nesse trabalho se propõe a tratar desses dois problemas.

5 Aplicação da teoria de controle no gerenciamento de fluxo de pacotes em redes TCP/IP

Este trabalho se propõe a realizar a síntese de um compensador *anti-windup* para o controlador PI proposto por (HOLLOT et al., 2002) usado para regular a fila de pacotes de um roteador em uma rede TCP/IP. Dessa forma, neste capítulo é apresentado todo o desenvolvimento matemático feito para a elaboração do resultado final: a síntese do compensador *anti-windup* estático para sistemas com atraso nos estados e na entrada.

Em um primeiro momento, são explicados alguns conceitos preliminares necessários para a boa compreensão da síntese.

Em seguida, descreve-se o desenvolvimento propriamente dito, destacando as partes mais importantes e referenciando alguns conceitos utilizados que não são completamente apresentados neste trabalho.

Finalmente, apresenta-se o problema de otimização final que será resolvido por um *solver* matemático e alguns resultados numéricos como exemplo de aplicação.

5.1 Definições iniciais

A partir da equação (8), pode-se estabelecer o sistema generalizado

$$\begin{cases} \dot{x}(t) = Ax(t) + A_d x(t - \tau) + Bu(t - \tau) + B_w w(t) \\ y(t) = Cx(t) \end{cases} \quad (12)$$

onde se define os parâmetros

- $x(t)' = [W(t)' \ q(t)']' \in \mathfrak{R}^n$: Vetor das variáveis de estado representando o tamanho médio da janela de emissão e da fila de pacotes no roteador
- $w(t) \in \mathfrak{R}^q$: Função que estabelece a modelagem de perturbações na rede TCP/IP (tráficos UDP, etc.)
- $u(t) \in \mathfrak{R}^m$: Sinal de entrada da planta
- $y(t) \in \mathfrak{R}^p$: Sinal de saída da planta

O conjunto de equações descrito em (12) representará o sistema de estudo. Assume-se que a entrada da planta está sujeita a perturbações limitadas em norma \mathcal{L}_2 . As matrizes A , A_d , B , B_w e C são reais, constantes e de dimensões apropriadas. O par (A, B) é tido como estabilizável.

A constante real positiva τ representa o atraso ao qual estão sujeitos os estados do sistema. No caso de redes TCP/IP este atraso varia para cada pacote do fluxo de dados, mas para fim de compreensão da natureza do congestionamento, é possível assumir que todos os pacotes estão sujeitos a um único atraso, cujo valor τ representativo para um dado intervalo de tempo da rede, pode ser este determinado por ensaios experimentais.

O vetor de perturbação $w(t)$ é assumido como limitado em energia ($w(t) \in \mathcal{L}_2$). Para um escalar qualquer δ seguindo a relação $0 \leq \frac{1}{\mu} < +\infty$, a perturbação $w(t)$ é limitada conforme

$$\|w\|_2^2 = \int_0^{+\infty} w(t)' w(t) dt < \frac{1}{\mu} \quad (13)$$

Além, é sabido que o sinal de controle pode vir a ficar sujeito a saturações, degradando a performance do sistema ou levando o mesmo a instabilidade. O trabalho desenvolvido por (HOLLOT et al., 2002) foi uma das primeiras aplicações da teoria de controle dentro do domínio de redes TCP/IP. HOLLOT sugere melhorias no desempenho do AQM RED, tratado no capítulo 3, substituindo a dinâmica estática original de (FLOYD; JACOBSON, 1993) através da síntese de um controlador PI para o sistema modelado em (MISRA; GONG; TOWNSLEY, 2000). Contudo, em sua síntese, a saturação no sinal de controle foi totalmente desconsiderada.

De uma forma geral, a representação desse controlador pode ser feita pelas equações

$$\begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) \\ y_c(t) = C_c x_c(t) + D_c u_c(t) \end{cases} \quad (14)$$

com os parâmetros

- $x_c(t) \in \mathfrak{R}^{n_c}$: Estado do controlador
- $u_c(t) \in \mathfrak{R}^p$: Entrada do controlador
- $y_c(t) \in \mathfrak{R}^m$: Saída do controlador

No caso do controlador PI desenvolvido por (HOLLOT et al., 2002), para a determinação das matrizes A_c , B_c , C_c e D_c , tem-se as seguintes equações:

$$\begin{aligned} A_c &= a_c = 0 \\ B_c &= b_c = 1 \\ C_c &= c_c = \frac{K_{PI}}{w_c} = \left| \frac{\frac{jw_g}{p_{queue}} + 1}{\frac{(R_o C)^3}{(2N)^2}} \right| \\ D_c &= d_c = B_c K_{PI} = b_c K_{PI} \end{aligned} \quad (15)$$

onde

- K_{PI} : Ganho do controlador PI

- w_g : Freqüência de corte do controlador PI
- p_{queue} : Um dos pólos do sistema linearizado definido como $\frac{1}{R_o}$

Deve-se notar que, devido a ordem do controlador ser unitária $n_c = 1$, as matrizes B_c , C_c e D_c são na verdade escalares e podem ser definidos como b_c , c_c e d_c .

A prática corrente no controle de congestionamento de redes TCP/IP consiste em medir o tamanho da fila de pacotes do *buffer* e, em função desta medida, determinar a probabilidade de descarte associada aos pacotes ali contidos.

Considerando uma rede TCP/IP de fato, $y_c(t)$ (correspondente à probabilidade de descarte imputável aos pacotes presentes no *buffer* do roteador) não atua imediatamente sobre os estados da planta, ou seja, sobre o tamanho da janela de congestionamento. Isso ocorre devido à estruturação do protocolo TCP, é implícita a necessidade de um atraso no tempo até que o sistema reaja (como visto na seção 2.2.1).

No modelo matemático representando o comportamento do TCP utilizado neste trabalho, o fenômeno de saturação é representado pela limitação do valor da probabilidade de descarte associada ao pacote.

Dessa forma, considera-se que a saída do controlador é sujeita a saturação simétrica de amplitude, como mostra a equação

$$-u_{o(i)} \leq u_{(i)} \leq u_{o(i)} \quad \text{com} \quad u_{o(i)} > 0 \quad \text{e} \quad i = 1, \dots, m \quad (16)$$

Como a interligação entre a planta o controlador é efetuada por $y_c(t) = u(t)$, o vetor de entrada da planta refletirá a saturação à qual o sistema está sujeito, conforme definido na equação (17).

$$u(t) = \text{sat}(y_c(t)) = \text{sat}(C_c x_c(t) + D_c u_c(t)) \quad (17)$$

O termo $\text{sat}(u_c(t))$ modela a limitação de amplitude da saída do controlador e será definido da forma

$$\text{sat}(y_{c(i)}(t)) = \text{sign}(y_{c(i)}(t)) \min(|y_{c(i)}(t)|, u_{o(i)}) \quad (18)$$

Assim, propõe-se o compensador *anti-windup* estático para o controlador PI definido genericamente pelo conjunto de equações (14). O controlador recebe agora um termo adicional de forma mitigar os efeitos da saturação no sinal de controle.

$$\begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) + E_c (\text{sat}(u_c(t)) - u_c(t)) \\ y_c(t) = C_c x_c(t) + D_c u_c(t) \end{cases} \quad (19)$$

Realizando as devidas operações algébricas para avaliar o sistemas (12) e (19) em malha fechada, pode-se estabelecer a equação

$$\begin{cases} \dot{\xi}(t) = \mathbb{A}_o \xi(t) + \mathbb{A}_d \xi(t - \tau) - \mathbb{B} \psi(\mathbb{K} \xi(t - \tau)) - \mathbb{R} E_c \psi(\mathbb{K} \xi(t)) + \mathbb{B}_w w(t) \\ y(t) = \mathbb{C} \xi(t) \end{cases} \quad (20)$$

onde $\xi(t)' = [x(t)' \quad x_c(t)']'$ representa os vetores de estado so sistema em malha fechada e as matrizes \mathbb{A}_o , \mathbb{A}_d , \mathbb{B} , \mathbb{B}_w , \mathbb{R} , \mathbb{K} e \mathbb{C} definidas como

$$\begin{aligned} \mathbb{A}_o &= \begin{bmatrix} A & 0 \\ B_c C & A_c \end{bmatrix} & \mathbb{A}_d &= \begin{bmatrix} A_d + B D_c C & B C_c \\ 0 & 0 \end{bmatrix} \\ \mathbb{B} &= \begin{bmatrix} B \\ 0 \end{bmatrix} & \mathbb{B}_w &= \begin{bmatrix} B_w \\ 0 \end{bmatrix} & \mathbb{R} &= \begin{bmatrix} 0 \\ I_{n_c} \end{bmatrix} \\ \mathbb{K} &= \begin{bmatrix} D_c C & C_c \end{bmatrix} & \mathbb{C} &= \begin{bmatrix} C & 0 \end{bmatrix} \end{aligned} \quad (21)$$

A função $\psi(v) = v - \text{sat}(v)$ corresponde à não linearidade do tipo zona morta decentralizada definida da forma

$$\psi(v_{(i)}) = \begin{cases} v_{(i)} - u_{o(i)} & \text{se } v_{(i)} > u_{o(i)} \\ 0 & \text{se } -u_{o(i)} \leq v_{(i)} \leq u_{o(i)} \\ v_{(i)} + u_{o(i)} & \text{se } v_{(i)} < -u_{o(i)} \end{cases} \quad (22)$$

Devido à presença de atrasos, as condições iniciais do sistema (20) têm a forma de uma função definida como $\phi(\theta)$, onde $\theta \in [-\tau, 0]$. A figura 16 ilustra a forma genérica da função $\phi(\theta)$.

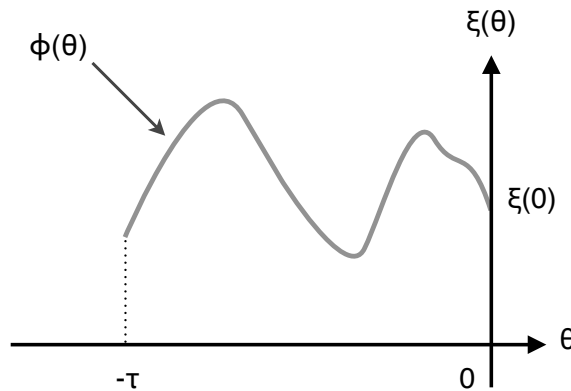


Figura 16: Função determinante das condições iniciais do sistema

Logo, o sistema (20) admite uma condição inicial aumentada $\xi(t_o - \theta) = \phi_\xi(\theta)$ para $\forall \theta \in [-\tau, 0]$.

Dessa forma, pode-se formular os problemas que serão abordados no desenvolvimento deste trabalho:

Problema 5.1.1 (Síntese de compensador *anti-windup* estático: Rejeição a perturbações). *Dado o sistema composto por (12) e (19), interconectados através de $u_c(t) = y(t)$ e $u(t) = y_c(t)$, sujeito à restrição (16), deve-se determinar o compensador anti-windup que garanta a estabilidade do sistema em malha fechada quando o mesmo estiver sobre efeito de saturação e que minimize o ganho superior \mathcal{L}_2 da transferência de $w(t)$ para $y(t)$.*

Problema 5.1.2 (Síntese de compensador *anti-windup* estático: Tolerância a perturbações). Dado o sistema composto por (12) e (19), interconectados através de $u_c(t) = y(t)$ e $u(t) = y_c(t)$, sujeito à restrição (16), deve-se determinar o compensador anti-windup que garanta a estabilidade do sistema em malha fechada quando o mesmo estiver sobre efeito de saturação e que maximize a norma \mathcal{L}_2 da perturbação tolerável pelo sistema.

O desenvolvimento aqui proposto leva em consideração os dois problemas simultaneamente. Contudo, deve-se notar que tratam-se de problemas de otimização independentes.

O primeiro se interessa em encontrar a maior norma \mathcal{L}_2 da perturbação aceitável pelo sistema de forma que ele não venha a ficar instável. O ganho superior \mathcal{L}_2 da transferência de $w(t)$ para $y(t)$ é determinado em consequência, mas seu valor pode não ser ótimo.

O segundo, por sua vez, tem como objetivo, minimizar o limitante superior para o ganho \mathcal{L}_2 de $w(t)$ em $y(t)$ para um dado limitante não nulo da norma \mathcal{L}_2 para a perturbação admissível. Deseja-se assim, tornar a saída do sistema mais resistente possível à perturbação considerada.

Em ambos os casos, assume-se que o sistema parte do equilíbrio. Assim, pode-se estabelecer que as condições iniciais não dependem mais do atraso do sistema, assumindo a forma $\xi(\theta) = 0$ para $\forall \theta \in [0, -\tau]$.

5.2 Desenvolvimento do problema

O desenvolvimento aqui proposto analisa sistemas sujeitos à saturação e com atraso no estado e na entrada. Trata-se de uma extensão do trabalho desenvolvido por (GOMES DA SILVA JR.; TARBOURIECH; GARCIA, 2006).

5.2.1 Estabilização do sistema

Com o objetivo de estudar a estabilidade do sistema (20) em malha fechada, definiu-se o funcional de Lyapunov-Krasovskii

$$V(\xi(t)) = \xi(t)' P \xi(t) + Q(t) + \int_{t-\tau}^t \psi(\mathbb{K}\xi(\sigma))' U \psi(\mathbb{K}\xi(\sigma)) d\sigma \quad (23)$$

onde $P = P' > 0 \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$, $U = U' > 0 \in \mathfrak{R}^{m \times m}$, $\psi(\mathbb{K}\xi(\sigma)) = \psi(y_c(\sigma))$ e $Q(t)$ é uma forma quadrática definida positiva que será definida mais a frente no desenvolvimento.

Conforme enunciado na seção 4.2, a análise de um sistema com atrasos pode ser feita utilizando duas abordagens diferentes: análise dependente do atraso ou independente do atraso. Devido a grande dependência do atraso na aplicação de controle da fila pacotes de roteadores em uma rede TCP/IP, um resultado menos conservador e realista (em geral) geralmente pode ser obtido pela metodologia dependente do atraso. Dessa forma, a desenvolvimento se aterá nessa abordagem.

Para essa análise, um importante recurso algébrico é a fórmula de Leibnitz-Newton definida por

$$\xi(t - \tau) = \xi(t) - \int_{-\tau}^0 \dot{\xi}(t + \theta) d\theta \quad (24)$$

Utilizando a equação (24), pode-se rescrever os termos do sistema (20) dependentes do atraso τ . Assim, de acordo com (NICULESCU, 2001), para garantir a estabilidade de (20) é suficiente assegurar a estabilidade do sistema

$$\begin{aligned}
\dot{\xi}(t) &= (\mathbb{A}_o + \mathbb{A}_d)\xi(t) - \mathbb{B}\psi(\mathbb{K}\xi(t - \tau)) - \mathbb{R}E_c\psi(\mathbb{K}\xi(t)) + \mathbb{B}_w w(t) \\
&\quad - \int_{-\tau}^0 \mathbb{A}_d \mathbb{A}_o \xi(t + \beta) d\beta - \int_{-\tau}^0 \mathbb{A}_d \mathbb{A}_d \xi(t + \beta - \tau) d\beta \\
&\quad + \int_{-\tau}^0 \mathbb{A}_d \mathbb{B}\psi(\mathbb{K}\xi(t + \beta - \tau)) d\beta + \int_{-\tau}^0 \mathbb{A}_d \mathbb{R}E_c\psi(\mathbb{K}\xi(t + \beta)) d\beta \\
&\quad - \int_{-\tau}^0 \mathbb{A}_d \mathbb{B}_w w(t + \beta) d\beta
\end{aligned} \tag{25}$$

com a condição inicial $\xi(t_o + \theta) = \phi_\xi(\theta)$ para $\forall \theta \in [-2\pi, 0]$.

A teoria de Lyapunov estabelece que, para um sistema ser estável, deve existir uma função quadrática positiva definida que dependa dos estados do sistemas de forma que a sua energia decresça, ou seja, que a derivada temporal do fucional de Lyapunov seja estritamente negativa para todas as trajetórias do sistema. Dessa forma, computamos a derivada temporal da equação (23) substituindo os termos definidos em (25).

$$\begin{aligned}
\dot{V}(\xi(t)) &= \dot{\xi}(t)' P \xi(t) + \xi(t)' P \dot{\xi}(t) + \dot{Q}(t) \\
&\quad + \psi(\mathbb{K}\xi(t))' U \psi(\mathbb{K}\xi(t)) - \psi(\mathbb{K}\xi(t - \tau))' U \psi(\mathbb{K}\xi(t - \tau))
\end{aligned} \tag{26}$$

Buscando solucionar os dois problemas descritos em 5.1.1 e 5.1.2, é interessante adicionar dois termos extras na equação (26): $\frac{1}{\gamma} y(t)' y(t)$ e $-w(t)' w(t)$. Define-se, assim, a função auxiliar $\mathcal{L}(t)$

$$\mathcal{L}(t) = \dot{V}(\xi(t)) + \frac{1}{\gamma} y(t)' y(t) - w(t)' w(t) \tag{27}$$

A condição estabelecida pelo problema de tolerância à perturbação pode ser encontrada facilmente integrando (27) no intervalo $t \in [0, T]$. Obtêm-se

$$\int_0^T \mathcal{L}(t) dt = V(T) - V(0) + \frac{1}{\gamma} \int_0^T y(t)' y(t) dt - \int_0^T w(t)' w(t) dt \tag{28}$$

O que implica em

$$\begin{aligned}
V(T) &\leq V(0) + \int_0^T w(t)' w(t) dt \\
&\leq V(0) + \|w(t)\|_2^2 \\
&\leq V(0) + \frac{1}{\mu}
\end{aligned} \tag{29}$$

Considerando ainda condições iniciais nulas $\xi(\theta) = 0$ para $\forall \theta \in [0, -\tau]$, tem-se que $V(0) = 0$. Pode-se assim definir

$$V(T) \leq V(0) + \frac{1}{\mu} \leq \mu^{-1} \quad (30)$$

Essa relação mostra que, se obedecida, as trajetórias do sistema não deixam o conjunto $\mathcal{S}(P, \mu^{-1})$, ou de outra forma, que a origem do sistema em malha fechada é estável mesmo com a presença de uma perturbação satisfazendo (13).

Igualmente, a condição do problema de rejeição à perturbação pode ser obtida da relação (27) integrando sobre o mesmo intervalo e fazendo $T \rightarrow +\infty$. Pode se escrever dessa maneira

$$\begin{aligned} \frac{1}{\gamma} \int_0^{\infty} y(t)' y(t) dt &< V(0) + \|w(t)\|_2^2 \\ \frac{1}{\gamma} \|y(t)\|_2^2 &< V(0) + \|w(t)\|_2^2 \end{aligned} \quad (31)$$

Assim, para o sistema inicializado na origem, tem-se

$$\|y(t)\|_2^2 < \gamma \|w(t)\|_2^2 \quad (32)$$

o que constitui γ o limitante superior do ganho \mathcal{L}_2 de $w(t)$ para $y(t)$.

A função quadrática sobre a qual toda a síntese será desenvolvida fica definida como

$$\begin{aligned} \mathcal{L}(t) &\leq \dot{\xi}(t)' P \xi(t) + \xi(t)' P \dot{\xi}(t) + \dot{Q}(t) \\ &+ \frac{1}{\gamma} \xi(t)' C' C \xi(t) - w(t)' w(t) \\ &+ \psi(\mathbb{K} \xi(t))' U \psi(\mathbb{K} \xi(t)) - \psi(\mathbb{K} \xi(t - \tau))' U \psi(\mathbb{K} \xi(t - \tau)) \end{aligned} \quad (33)$$

onde $\gamma > 0 \in \mathfrak{R}$ é o ganho \mathcal{L}_2 de $w(t)$ para $y(t)$.

Como já explicado anteriormente, o sistema estudado apresenta elementos não lineares em sua modelagem. Esses elementos são a saturação do sinal de controle i em um valor denotado de $u_{o(i)}$.

Usando a abordagem de setores, o trabalho de (GOMES DA SILVA JR.; TARBOURIECH, 2005) contribui de forma significativa no estabelecimento de condições suficientes menos conservadoras para a estabilidade de sistemas sob saturação.

Lemma 5.2.1 (Condição de Setor Generalizada (GOMES DA SILVA JR.; TARBOURIECH, 2005)). *Considere a matriz $G \in \mathfrak{R}^{m \times m}$ e o conjunto*

$$\mathcal{S}(u_o) \triangleq \{\xi \in \mathfrak{R}^n; |(\mathbb{K}_{(i)} - G_{(i)})\xi| \leq u_{o(i)} \quad \forall i = 1, \dots, m\} \quad (34)$$

Assim, se $\xi \in \mathcal{S}(u_o)$, para toda matriz $T \in \mathfrak{R}^{m \times m}$ diagonal e definida positiva

$$\psi(\mathbb{K}\xi(t))^T T[\psi(\mathbb{K}\xi(t)) - G\xi(t)] \leq 0 \quad (35)$$

A condição de setor é um artifício matemático extremamente útil para o estabelecimento das condições suficientes para a estabilidade de sistemas nos quais algum fenômeno de não linearidade está presente.

Assim, $\forall \xi(t) \in \mathcal{S}(u_o)$ (definido em (34)) e usando a condição de setor generalizada em (35) pode-se escrever:

$$\begin{aligned} \mathcal{L}(t) &\leq \dot{\xi}(t)' P\xi(t) + \xi(t)' P\dot{\xi}(t) + \dot{Q}(t) \\ &+ \frac{1}{\gamma} \xi(t)' \mathbb{C}' \mathbb{C}\xi(t) - w(t)' w(t) \\ &+ \psi(\mathbb{K}\xi(t))' U\psi(\mathbb{K}\xi(t)) - \psi(\mathbb{K}\xi(t-\tau))' U\psi(\mathbb{K}\xi(t-\tau)) \\ &- 2\psi(\mathbb{K}\xi(t))' T[\psi(\mathbb{K}\xi(t)) - G\xi(t)] \end{aligned} \quad (36)$$

Aplicando as condições estabelecidas em (25) em (36), chega-se à equação

$$\begin{aligned} \mathcal{L}(t) &\leq \xi(t)' [(\mathbb{A}'_o + \mathbb{A}'_d)P + P(\mathbb{A}_o + \mathbb{A}_d)]\xi(t) - 2\xi(t)' P\mathbb{B}\psi(\mathbb{K}\xi(t-\tau)) \\ &- 2\xi(t)' P\mathbb{R}E_c\psi(\mathbb{K}\xi(t)) + 2\xi(t)' P\mathbb{B}_w w(t) \\ &+ \mu(\xi(t)) + \zeta(\xi(t)) + \delta(\xi(t)) + \alpha(\xi(t)) + \gamma(\xi(t)) \\ &+ \frac{1}{\gamma} \xi(t)' \mathbb{C}' \mathbb{C}\xi(t) - w(t)' w(t) \\ &+ \psi(\mathbb{K}\xi(t))' U\psi(\mathbb{K}\xi(t)) - \psi(\mathbb{K}\xi(t-\tau))' U\psi(\mathbb{K}\xi(t-\tau)) \\ &- 2\psi(\mathbb{K}\xi(t))' T[\psi(\mathbb{K}\xi(t)) - G\xi(t)] + \dot{Q}(t) \end{aligned} \quad (37)$$

onde

$$\begin{aligned} \mu(\xi(t)) &= -2 \int_{-\tau}^0 \xi(t)' P\mathbb{A}_d\mathbb{A}_o\xi(t+\beta)d\beta \\ \zeta(\xi(t)) &= -2 \int_{-\tau}^0 \xi(t)' P\mathbb{A}_d\mathbb{A}_d\xi(t+\beta-\tau)d\beta \\ \delta(\xi(t)) &= 2 \int_{-\tau}^0 \xi(t)' P\mathbb{A}_d\mathbb{B}\psi(\mathbb{K}\xi(t+\beta-\tau))d\beta \\ \alpha(\xi(t)) &= 2 \int_{-\tau}^0 \xi(t)' P\mathbb{A}_d\mathbb{R}E_c\psi(\mathbb{K}\xi(t+\beta))d\beta \\ \gamma(\xi(t)) &= -2 \int_{-\tau}^0 \xi(t)' P\mathbb{A}_d\mathbb{B}_w w(t+\beta)d\beta \end{aligned} \quad (38)$$

Analisando a expressão (37), pode-se observar que os termos integrais não possuem uma definição clara de sinal. Dessa forma, aplica-se a desigualdade triangular $2u'v \leq u'Mu + v'M^{-1}v$ onde u e v são vetores com dimensões apropriadas e M uma matriz simétrica positiva e definida

($M = M' > 0$). Por uma questão notação, para simplificar as expressões algébricas, denota-se $\psi(\mathbb{K}\xi(t)) = \psi(t)$. Os termos integrais tornam-se então

$$\begin{aligned}
\mu(\xi(t)) &\leq \tau \xi(t)' P \mathbb{A}_d X \mathbb{A}'_d P \xi(t) + \int_{-\tau}^0 \xi(t+\beta)' \mathbb{A}'_o X^{-1} \mathbb{A}_o \xi(t+\beta) d\beta \\
\zeta(\xi(t)) &\leq \tau \xi(t)' P \mathbb{A}_d R \mathbb{A}'_d P \xi(t) + \int_{-\tau}^0 \xi(t+\beta-\tau)' \mathbb{A}'_d R^{-1} \mathbb{A}_d \xi(t+\beta-\tau) d\beta \\
\delta(\xi(t)) &\leq \tau \xi(t)' P \mathbb{A}_d H \mathbb{A}'_d P \xi(t) + \int_{-\tau}^0 \psi(t+\beta-\tau)' \mathbb{B}' H^{-1} \mathbb{B} \psi(t+\beta-\tau) d\beta \\
\alpha(\xi(t)) &\leq \tau \xi(t)' P \mathbb{A}_d L \mathbb{A}'_d P \xi(t) + \int_{-\tau}^0 \psi(t+\beta)' \mathbb{R}' E_c L^{-1} \mathbb{R} E_c \psi(t+\beta) d\beta \\
\gamma(\xi(t)) &\leq \tau \xi(t)' P \mathbb{A}_d J \mathbb{A}'_d P \xi(t) + \int_{-\tau}^0 w(t+\beta)' \mathbb{B}'_w J^{-1} \mathbb{B}_w w(t+\beta) d\beta
\end{aligned} \tag{39}$$

Define-se a forma quadrática $Q(t)$ como

$$\begin{aligned}
Q(t) &= \int_{-\tau}^0 \int_{t+\beta}^t \xi(\theta)' \mathbb{A}'_o X^{-1} \mathbb{A}_o \xi(\theta) d\theta d\beta \\
&+ \int_{-\tau}^0 \int_{t+\beta-\tau}^t \xi(\theta)' \mathbb{A}'_d R^{-1} \mathbb{A}_d \xi(\theta) d\theta d\beta \\
&+ \int_{-\tau}^0 \int_{t+\beta-\tau}^t \psi(\theta)' \mathbb{B}' H^{-1} \mathbb{B} \psi(\theta) d\theta d\beta \\
&+ \int_{-\tau}^0 \int_{t+\beta}^t \psi(\theta)' \mathbb{R}' E_c L^{-1} \mathbb{R} E_c \psi(\theta) d\theta d\beta \\
&+ \int_{-\tau}^0 \int_{t+\beta}^t w(\theta)' \mathbb{B}'_w J^{-1} \mathbb{B}_w w(\theta) d\theta d\beta
\end{aligned} \tag{40}$$

Considerando a derivada temporal de (40), tem-se

$$\begin{aligned}
\dot{Q}(t) &= \\
&- \int_{-\tau}^0 \xi(t+\beta)' \mathbb{A}'_o X^{-1} \mathbb{A}_o \xi(t+\beta) d\beta + \tau \xi(t)' \mathbb{A}'_o X^{-1} \mathbb{A}_o \xi(t) \\
&- \int_{-\tau}^0 \xi(t+\beta-\tau)' \mathbb{A}'_d R^{-1} \mathbb{A}_d \xi(t+\beta-\tau) d\beta + \tau \xi(t)' \mathbb{A}'_d R^{-1} \mathbb{A}_d \xi(t) \\
&- \int_{-\tau}^0 \psi(t+\beta-\tau)' \mathbb{B}' H^{-1} \mathbb{B} \psi(t+\beta-\tau) d\beta + \tau \psi(t)' \mathbb{B}' H^{-1} \mathbb{B} \psi(t) \\
&- \int_{-\tau}^0 \psi(t+\beta)' \mathbb{R}' E_c L^{-1} \mathbb{R} E_c \psi(t+\beta) d\beta + \tau \psi(t)' \mathbb{R}' E_c L^{-1} \mathbb{R} E_c \psi(t) \\
&- \int_{-\tau}^0 w(t+\beta)' \mathbb{B}'_w J^{-1} \mathbb{B}_w w(t+\beta) d\beta + \tau w(t)' \mathbb{B}'_w J^{-1} \mathbb{B}_w w(t)
\end{aligned} \tag{41}$$

Logo, para $\forall \xi \in \mathcal{S}(u_o)$, a expressão de $\mathcal{L}(t)$ assume a forma

$$\begin{aligned}
\mathcal{L}(t) \leq & \xi(t)' [P(\mathbb{A}_o + \mathbb{A}_d) + (\mathbb{A}'_o + \mathbb{A}'_d)P] \xi(t) \\
& - 2\xi(t)' P\mathbb{B}\psi(t - \tau) - 2\xi(t)' P\mathbb{R}E_c\psi(t) \\
& + \psi(t)' U\psi(t) - \psi(t - \tau)' U\psi(t - \tau) \\
& + 2\xi(t)' P\mathbb{B}_w w(t) - 2\psi(t)' T\psi(t) + 2\xi(t)' (TG)' \psi(t) \\
& + \tau\xi(t)' P\mathbb{A}_d X \mathbb{A}'_d P \xi(t) + \tau\xi(t)' \mathbb{A}'_o X^{-1} \mathbb{A}_o \xi(t) \\
& + \tau\xi(t)' P\mathbb{A}_d R \mathbb{A}'_d P \xi(t) + \tau\xi(t)' \mathbb{A}'_d R^{-1} \mathbb{A}_d \xi(t) \\
& + \tau\xi(t)' P\mathbb{A}_d H \mathbb{A}'_d P \xi(t) + \tau\psi(t)' \mathbb{B}' H^{-1} \mathbb{B}\psi(t) \\
& + \tau\xi(t)' P\mathbb{A}_d L \mathbb{A}'_d P \xi(t) + \tau\psi(t)' \mathbb{R}' E_c L^{-1} \mathbb{R}E_c \psi(t) \\
& + \tau\xi(t)' P\mathbb{A}_d J \mathbb{A}'_d P \xi(t) + \tau w(t)' \mathbb{B}'_w J^{-1} \mathbb{B}_w w(t) \\
& + \frac{1}{\gamma} \xi(t)' \mathbb{C}' \mathbb{C} \xi(t) - w(t)' w(t)
\end{aligned} \tag{42}$$

Rescreve-se a equação (42) em forma matricial para facilitar o tratamento.

$$\mathcal{L}(t) \leq \begin{bmatrix} \xi(t)' & \psi(t - \tau)' & \psi(t)' & w(t)' \end{bmatrix} \begin{bmatrix} Q'_1 & -P\mathbb{B} & -P\mathbb{R}E_c - (TG)' & P\mathbb{B}_w \\ \star & -U & 0 & 0 \\ \star & \star & Q'_2 & 0 \\ \star & \star & \star & Q'_3 \end{bmatrix} \begin{bmatrix} \xi(t) \\ \psi(t - \tau) \\ \psi(t) \\ w(t) \end{bmatrix} \tag{43}$$

com Q'_1 , Q'_2 e Q'_3 na forma

$$\begin{aligned}
Q'_1 &= \Delta + \tau\mathbb{A}'_d R^{-1} \mathbb{A}_d + \tau\mathbb{A}'_o X^{-1} \mathbb{A}_o + \frac{1}{\gamma} \mathbb{C}' \mathbb{C} \\
Q'_2 &= \tau\mathbb{B}' H^{-1} \mathbb{B} + \tau(\mathbb{R}E_c)' L^{-1} \mathbb{R}E_c - 2T + U \\
Q'_3 &= \tau\mathbb{B}'_w J^{-1} \mathbb{B}_w - I
\end{aligned} \tag{44}$$

onde $\Delta = (\mathbb{A}_o + \mathbb{A}_d)' P + P(\mathbb{A}_o + \mathbb{A}_d) + \tau P\mathbb{A}_d(X + R + H + L + J)\mathbb{A}'_d P$.

Assim, uma condição suficiente para termos $\mathcal{L}(t) < 0$ é dada pela seguinte desigualdade matricial

$$\begin{bmatrix} Q'_1 & -P\mathbb{B} & -P\mathbb{R}E_c - (TG)' & P\mathbb{B}_w \\ \star & -U & 0 & 0 \\ \star & \star & Q'_2 & 0 \\ \star & \star & \star & Q'_3 \end{bmatrix} < 0 \tag{45}$$

Deve-se notar que a desigualdade encontrada em (45) é uma BMI. Isso ocorre justamente pela multiplicação dos termos que representam as variáveis a serem determinadas (ex: $P\mathbb{R}E_c$). Para eliminar esses termos e transformar a BMI em uma LMI, multiplica-se ambos os lados por

$$\begin{bmatrix} P^{-1} & 0 & 0 & 0 \\ 0 & T^{-1} & 0 & 0 \\ 0 & 0 & T^{-1} & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \text{ para se chegar na relação}$$

$$\begin{bmatrix} Q_1 & -\mathbb{B}M & -\mathbb{R}Z - Y' & \mathbb{B}_w \\ \star & -F & 0 & 0 \\ \star & \star & Q_2 & 0 \\ \star & \star & \star & Q_3 \end{bmatrix} < 0 \quad (46)$$

tendo como definidos Q_1 , Q_2 e Q_3 da forma

$$\begin{aligned} Q_1 &= \Gamma + \tau W \mathbb{A}'_d R^{-1} \mathbb{A}_d W + \tau W \mathbb{A}'_o X^{-1} \mathbb{A}_o W + \frac{1}{\gamma} W C' C W \\ Q_2 &= \tau M \mathbb{B}' H^{-1} \mathbb{B} M + \tau (\mathbb{R}Z)' L^{-1} \mathbb{R}Z - 2M + F \\ Q_3 &= \tau \mathbb{B}'_w J^{-1} \mathbb{B}_w - I \end{aligned} \quad (47)$$

com as relações $\Gamma = W(\mathbb{A}_o + \mathbb{A}_d)' + (\mathbb{A}_o + \mathbb{A}_d)W + \tau \mathbb{A}_d(X + R + H + L + J)\mathbb{A}'_d$, $M'UM = F$, $P^{-1} = W$, $T^{-1} = M$, $E_c = ZM^{-1}$ e $Y = GW$.

Aplicando o complemento de Schur (mais detalhes na seção A.2 do apêndice A) para os componentes internos de (46), quando aplicável, pode-se estabelecer a seguinte LMI

$$\begin{bmatrix} \Gamma & \tau W \mathbb{A}'_o & \tau W \mathbb{A}'_d & W C' & -\mathbb{B}M & -\mathbb{R}Z - Y' & 0 & 0 & \mathbb{B}_w & 0 \\ \star & -\tau X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \star & \star & -\tau R & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \star & \star & \star & -\gamma I & 0 & 0 & 0 & 0 & 0 & 0 \\ \star & \star & \star & \star & -F & 0 & 0 & 0 & 0 & 0 \\ \star & \star & \star & \star & \star & -2M + F & \tau M \mathbb{B}' & \tau Z' \mathbb{R}' & 0 & 0 \\ \star & \star & \star & \star & \star & \star & -\tau H & 0 & 0 & 0 \\ \star & \star & \star & \star & \star & \star & \star & -\tau L & 0 & 0 \\ \star & \star & \star & \star & \star & \star & \star & \star & -I & \tau \mathbb{B}'_w \\ \star & \star & \star & \star & \star & \star & \star & \star & \star & -\tau J \end{bmatrix} < 0 \quad (48)$$

5.2.1.1 Inclusão elipsoidal

Para o estabelecimento da segunda restrição, parte-se da premissa da satisfação da LMI (49). Deve-se agora provar que o $\mathcal{S}(P, \mu^{-1}) \subset \mathcal{S}(u_o)$ de forma que as restrições estabelecidas para o uso da *condição setor generalizada* sejam efetivamente respeitadas.

$$\begin{bmatrix} W & W(\mathbb{K}_{(i)} - Y_{(i)})' \\ \star & \mu u_{o(i)}^2 \end{bmatrix} \geq 0 \quad (49)$$

Deve-se notar que as matrizes W e Y são as mesmas utilizadas em (48).

Multiplicando-se ambos os lados por $\begin{bmatrix} W_0^{-1} & 0 \\ 0 & I \end{bmatrix}$, chega-se a a LMI (50) que faz intervir agora as matrizes P , G e \mathbb{K} .

$$\begin{bmatrix} P & (\mathbb{K}_{(i)} - G_{(i)})' \\ \star & \mu u_{o(i)}^2 \end{bmatrix} \geq 0 \quad (50)$$

Aplicando o complemento de Schur na desigualdade (50), tem-se que é igualmente verdadeira a relação

$$(\mathbb{K}_{(i)} - G_{(i)})' \frac{1}{\mu u_{o(i)}^2} (\mathbb{K}_{(i)} - G_{(i)}) - P \leq 0 \quad (51)$$

O que leva a relação

$$\xi(t)' (\mathbb{K}_{(i)} - G_{(i)})' \frac{1}{u_{o(i)}^2} (\mathbb{K}_{(i)} - G_{(i)}) \xi(t) \leq \xi(t)' \mu P \xi(t) \leq 1 \quad (52)$$

Pela equação (48) pode-se assim concluir que, para todo $t > 0$, as trajetórias $\xi(t)$ do sistema estudado não deixam o elipsóide definido por $\mathcal{S}(P, \mu^{-1})$. Adicionalmente, a relação (52) implica o elipsóide $\mathcal{S}(P, \mu^{-1})$ deve estar contido no conjunto $\mathcal{S}(u_o)$, o que garante a validade da condição de setor (35) utilizada para se obter (48).

Dessa forma, pode-se concluir que $\xi(t)$ pertence à $\mathcal{S}(u_o)$ e, assim, qualquer $\xi(t)$ satisfaz a *condição setor generalizada* enunciada na definição 5.2.1.

5.2.2 Problema de otimização

De posse das equações (48) e (49), pode-se enunciar o seguinte teorema:

Teorema 5.2.1 (Síntese do compensador *anti-windup* estático para sistemas com atraso na entrada e nos estados). *Se existem as matrizes simétricas positivas $W \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$, $X \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$, $R \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$, $H \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$, $L \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$ e $J \in \mathfrak{R}^{(n+n_c) \times (n+n_c)}$, uma matriz diagonal positiva definida $M \in \mathfrak{R}^{m \times m}$, uma matriz $F \in \mathfrak{R}^{m \times m}$, uma matriz $Z \in \mathfrak{R}^{n_c \times m}$ e escalares positivos μ e γ satisfazendo as relações (48) e (49) para $E_c = ZM^{-1}$, logo existe o compensador *anti-windup* definido em (19) que assegura:*

- *Todas as trajetórias do sistema são limitadas para qualquer condição inicial nula, isto é, $\xi(t) \in \mathcal{S}(P, \mu^{-1})$, $\forall t \geq 0$.*
- *Existe um limitante superior do ganho \mathcal{L}_2 de $w(t)$ para $y(t)$ tal que $\|y(t)\|_2^2 < \gamma \|w(t)\|_2^2$.*
- *Em ausência de perturbação $w(t)$, a origem do sistema (20) em malha fechada é localmente assintoticamente estável.*

A satisfação das condições dadas no teorema 5.2.1 asseguram que o sistema em malha fechada (20) apresenta trajetórias limitadas para qualquer perturbação com norma \mathcal{L}_2 finita e menor do que $\frac{1}{\mu}$. Neste caso, é também assegurado que o ganho \mathcal{L}_2 entre $w(t)$ e $y(t)$ não supera o escalar $\sqrt{\gamma}$.

A idéia aqui é utilizar as condições suficientes do teorema 5.2.1 de modo a obter-se um compensador *anti-windup* estático que considere os seguintes problemas de otimização:

5.2.2.1 Maximização da tolerância à perturbação

A idéia é maximizar a norma \mathcal{L}_2 da perturbação para a qual pode-se assegurar que as trajetórias do sistema permaneçam limitadas. Considerando as condições iniciais como nulas ($\phi_\xi(\theta) = 0$,

$\forall \theta \in [-\tau, 0]$), isto pode ser feito de acordo com o seguinte problema de otimização convexa (BENDER; GOMES DA SILVA JR.; TARBOURIECH, 2010):

$$\begin{cases} \min \mu \\ \text{sujeito à (48) e (49)} \end{cases} \quad (53)$$

Na maximização da tolerância à perturbação não se está interessado no valor que corresponderá a γ . De fato, γ assumirá algum valor finito que assegure que ambos (48) e (49) sejam válidos.

5.2.2.2 Maximização da rejeição à perturbação

Para um dado limitante não nulo da norma \mathcal{L}_2 para a perturbação admissível (dado por $\frac{1}{\mu}$), a idéia é minimizar o limitante superior para o ganho \mathcal{L}_2 de $w(t)$ em $y(t)$. Considerando novamente as condições iniciais como nulas, isto pode ser obtido mediante a solução do seguinte problema de otimização convexa (BENDER; GOMES DA SILVA JR.; TARBOURIECH, 2010):

$$\begin{cases} \min \gamma \\ \text{sujeito à (48) e (49)} \end{cases} \quad (54)$$

5.2.3 Resultados numéricos

O método apresentado aqui é agora aplicado na resolução do problema de tolerância à perturbação e de rejeição à perturbação em uma planta, cujo modelo pode ser expresso conforme (12). Em um primeiro momento, aplica-se a desenvolvimento feito para um conjunto de dados genéricos para uma exemplificação didática retirados de (BENDER; GOMES DA SILVA JR.; TARBOURIECH, 2010).

Em seguida, os resultados são aplicados no modelo que descreve o comportamento do fluxo de pacotes de uma rede TCP/IP em um roteador. Os dados utilizados nesse último seguem o exemplo numérico feito no capítulo 3 seção 3.4.2.

5.2.3.1 Exemplo numérico didático

Utilizando primeiramente dados hipotéticos para fins didáticos, estabeleceu-se os parâmetros para a planta definida em (12)

$$\begin{aligned} A &= \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix} & A_d &= \begin{bmatrix} -0.1 & 0.1 \\ 0 & -0.2 \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} & B_w &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} & C &= [1 \quad 0] \end{aligned} \quad (55)$$

O atraso nos estados assumido é como $\tau = 1$. Pressupõe-se que o sinal de controle da planta é limitado em amplitude, representado por $u_o = 1$. Igualmente, sua dinâmica é sujeita à perturbações do tipo (13). Parte-se da premissa que para esta planta foi projetado a priori um controlador linear PI, conforme (19). Assume-se que este controlador tem os seguintes valores para suas matrizes

$$A_c = 0 \quad B_c = -0.2 \quad C_c = 1 \quad D_c = -2 \quad (56)$$

Maximização da tolerância à perturbação:

A partir da solução do problema de otimização (53) obteve-se um compensador *anti-windup* de acordo com (19), definido segundo os parâmetros

$$E_c = 1.0386 \quad \mu_{min} = 17.0241 \quad \frac{1}{\mu_{min}} = 58.7402 \cdot 10^3 \quad (57)$$

Maximização da rejeição à perturbação:

Considerando o problema de rejeição à perturbação, realizou-se uma sequência de simulações para variantes limitantes de norma da perturbação $w(t)$ aplicada. O resultado é mostrado na tabela 2.

Tabela 1: Compensador com rejeição à perturbação utilizando os dados didáticos

	E_c	γ_{min}
$\mu = 18$	0.7637	82.5785
$\mu = 22$	0.3132	25.1199
$\mu = 26$	0.1709	19.8006

5.2.3.2 Exemplo numérico aplicado a redes TCP/IP

Os dados utilizados para a simulação deste exemplo são os mesmos do exemplo numérico descrito na seção 3.4.2. Assume-se que a entrada controlada (probabilidade de descarte) não supera o valor $u_o = 0.991$ (pois o ponto de equilíbrio $p_o = 0.008$), que o atraso recebe o valor $\tau = 0.246s$ e as demais considerações feitas no enunciado do exemplo. O sistema pode ser representado pelas matrizes

$$\begin{aligned}
 A &= \begin{bmatrix} -0.2644 & -0.0044 \\ 243.9024 & -4.0650 \end{bmatrix} & A_d &= \begin{bmatrix} -0.2644 & -0.0044 \\ 0 & 0 \end{bmatrix} \\
 B &= \begin{bmatrix} -480.47 \\ 0 \end{bmatrix} & B_w &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} & C &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}
 \end{aligned} \quad (58)$$

Como controlador, utiliza-se o PI implementado em (HOLLOT et al., 2002) considerando os valores de C , N , q_o e $R_o = \tau$ como no exemplo numérico de 3.4.2 de forma haver compatibilidade nas topologias.

Tais dados, resultaram na seguinte configuração de PI

$$A_c = 0 \quad B_c = 1 \quad C_c = 9.6426 \cdot 10^{-6} \quad D_c = 18.194 \cdot 10^{-6} \quad (59)$$

Maximização da tolerância à perturbação:

A partir da solução do problema de otimização (53) obteve-se um compensador *anti-windup* de acordo com (19), definido segundo os parâmetros

$$E_c = 1.1986 \cdot 10^4 \quad \mu_{min} = 5.1988 \cdot 10^{-13} \quad \frac{1}{\mu_{min}} = 1.9235 \cdot 10^{12} \quad (60)$$

Maximização da rejeição à perturbação:

Considerando o problema de rejeição à perturbação, realizou-se uma seqüência de cálculos para variantes de limitantes de norma da perturbação $w(t)$ aplicada. O resultado é mostrado na tabela 2.

Tabela 2: Compensador com rejeição à perturbação utilizando os dados da aplicação

	E_c	γ_{min}
$\mu = 5 \cdot 10^{-13}$	$8.4613 \cdot 10^4$	0.1100
$\mu = 5 \cdot 10^{-12}$	$9.9412 \cdot 10^4$	0.1051
$\mu = 5 \cdot 10^{-11}$	$1.9633 \cdot 10^5$	0.0928

5.2.4 Estudos comparativos para o caso TCP/IP

Como o modelo de comportamento de redes TCP/IP é globalmente estável, mesmo com a pior perturbação limitada, o mesmo terá suas trajetórias convergindo para a origem em um tempo infinito. Pode-se comparar, contudo, o desempenho trazido pela implementação do compensador *anti-windup* no controle do sistema para um mesmo nível de perturbação dado. Nas simulações feitas no programa Matlab, plota-se os gráficos do sinal de controle aplicado na planta e o sinal de saída da mesma com os parâmetros iniciais:

$$E_c = 1.4271 \cdot 10^5 \quad \gamma = 0.0974 \quad \frac{1}{\mu} = 3.0 \cdot 10^{10} \quad u_o = 0.991 \quad \tau = 0.246s \quad (61)$$

5.2.4.1 Comparação com PI de Hollot

Pode-se observar com o gráfico 17 o menor tempo que o sinal de controle permanece saturado para conseguir fazer com que o sistema venha a seguir o comportamento desejado. É aqui justamente que se pode verificar a eficácia do compensador *anti-windup* agindo no controle. O compensador faz com que o controlador, no momento em que a saturação está ocorrendo, fique pronto para reagir. A resposta do controle se torna mais rápida devido à diminuição da carga do termo integral do PI ocasionada pelo sinal de erro calculado, que nessa situação estaria continuamente diferente de zero.

Da mesma forma, com a figura 18, nota-se uma grande melhora na velocidade de resposta do sistema para reverter os efeitos causados pela perturbação inserida. O sistema com compensador *anti-windup* agregado ao controlador PI tem uma reposta mais rápida. O tempo de acomodação do sistema é em torno de duas vezes menor para este caso.

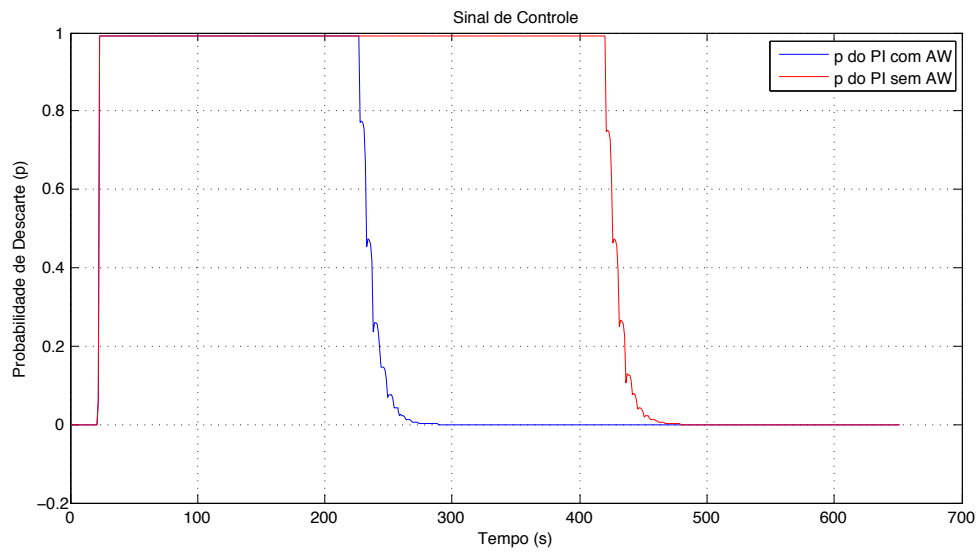


Figura 17: Sinal de controle do modelo de rede TCP/IP

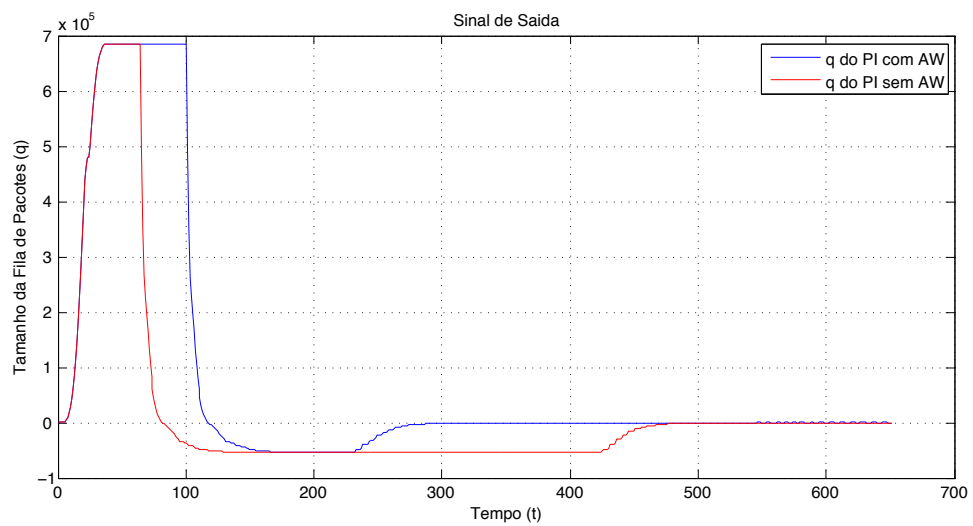


Figura 18: Tamanho da fila de pacotes do modelo de rede TCP/IP

5.2.4.2 Variação do nível de perturbação

Algumas outras simulações foram efetuadas com o objetivo de avaliar o desempenho e robustez do compensador desenvolvido. Compara-se a seguir o comportamento do sistema para o caso de variação nos valores da norma da perturbação μ^{-1} .

Aumentou-se a perturbação presente no sistema em um passo de 20% até que se atingisse um valor de μ^{-1} 80% superior ao valor para o qual o ganho do compensador foi calculado e determinou-se a resposta do sistema e o sinal de controle resultante.

A figura 19 ilustra o resultado do sinal de controle obtido. Pode-se notar uma pequena diferença entre os sinais devido ao aumento significativo da perturbação gerada. O sistema com o maior nível de perturbação tende a ficar saturado por um tempo maior para garantir o seguimento da referência do sinal de saída.

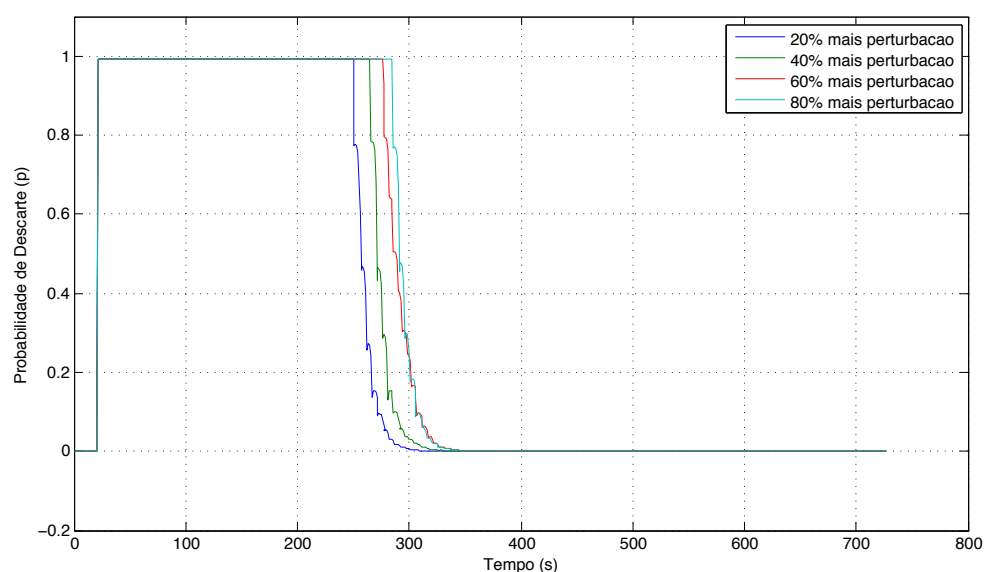


Figura 19: Comparação do sinal de controle para variações do nível de perturbação

Intuitivamente, pode-se estabelecer uma relação direta entre a variação do tamanho da fila de pacotes do roteador e o aumento do nível de perturbação presente no sistema. A figura 20 mostra então o resultado obtido para o sinal de saída do sistema para os níveis de perturbação simulados. Nota-se que, apesar da grande diferença nos níveis de perturbação gerados, os sistemas convergem ao seu ponto de equilíbrio quase que simultaneamente. A diferença do tempo de convergência entre os respectivos sinais de saída é pequena apesar de se ter uma grande variação nos valores de perturbação.

5.2.4.3 Variação do nível de atraso

Para a comparação do efeito da variação do atraso sobre a resposta do sistema, utilizou-se o mesmo raciocínio adotado para a análise das variações dos níveis de perturbação. Fazendo o valor do atraso variar 20% a mais e a menos do valor para o qual o ganho *anti-windup* foi calculado, plotou-se os sinais de controle e de saída do sistema.

A figura 21 mostra o sinal de controle p resultante para a variação do parâmetro τ . Pode-se notar que, como na análise de perturbação, o aumento do nível de atraso faz com que o sinal

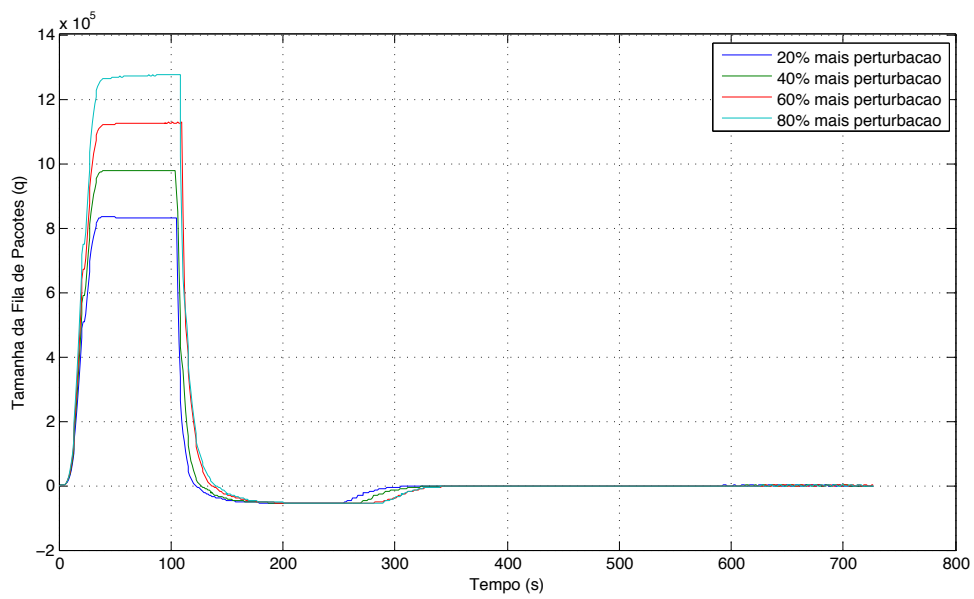


Figura 20: Comparação do sinal de saída para variações do nível de perturbação

de controle permaneça saturado por um intervalo de tempo maior. Essa figura pode ajudar a entender como atraso muito elevado pode levar o sistema a ter um nível de desempenho medíocre ou ainda levá-lo a uma região de instabilidade.

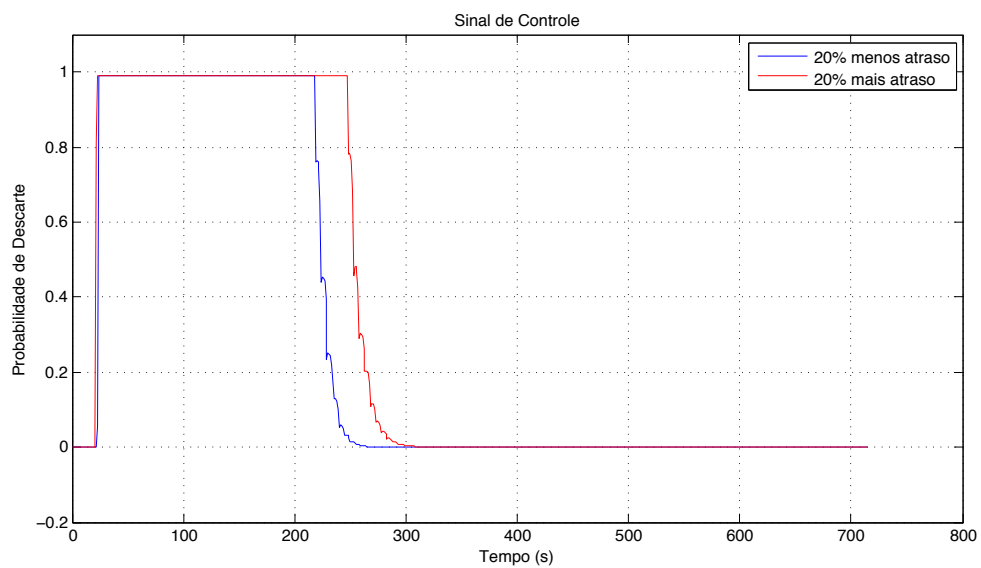


Figura 21: Comparação do sinal de controle para variações do nível de atraso

O sinal de saída do sistema para a comparação da variação do nível de atraso é mostrado na figura 22. O sistema com um menor atraso consegue reagir mais rapidamente à perturbação inserida e, assim, chegar ao ponto de equilíbrio em um menor tempo.

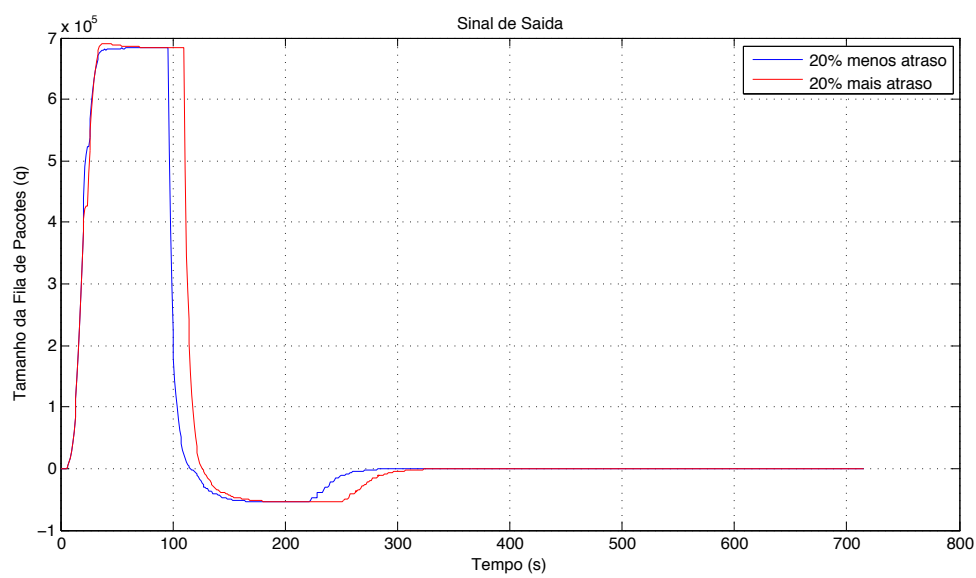


Figura 22: Comparação do sinal de saída para variações do nível de atraso

6 Simulações de topologias TCP

6.1 Introdução

A execução de simulações computacionais é uma das mais importantes formas de avaliação de novos protocolos e mecanismos de rede. Entretanto, estudos completos envolvendo simulação tipicamente exigem um grande poder computacional, tanto pelo fato de que uma única simulação pode demandar várias horas de processamento, quanto pelo fato de que, durante o processo de avaliação, inúmeras execuções independentes se fazem necessárias, variando os parâmetros de entrada (por exemplo, número de *hosts*, largura de banda, etc.) e as sementes aleatórias (para que uma boa precisão estatística seja obtida).

Simulações de redes de computadores são consideravelmente complexas. Muitos parâmetros devem ser levados em consideração para que as simulações possam ao máximo aproximar-se de um ambiente real da aplicação simulada. Sendo mais próximo possível da situação real, os resultados obtidos na simulação são mais confiáveis à medida que são analisados e utilizados com o propósito de auxiliar em tomadas de decisão.

Um simulador tem basicamente a função de colher as informações relativas à topologia da rede de comunicação, do tráfego da rede (conexões e transmissão de dados) e, através dessas informações, simular o ambiente desejado. O *Network Simulator* versão 2 (NS-2) mostrou-se mais completo e flexível que os outros programas analisados (como por exemplo AdventNet 6) devido à sua fácil disponibilidade e por possuir maiores possibilidades de o usuário interferir nas características da rede simulada.

O funcionamento do simulador NS-2 limita-se à linha de comando. Ou seja, não existe nenhum tipo de visualização gráfica durante a execução da simulação. Para essa visualização é necessária a utilização de outras ferramentas abordadas no decorrer do trabalho.

Devido à documentação relativamente pobre disponível na literatura sobre esse programa, no apêndice C.1, uma descrição detalhada do processo de instalação e de operação é feita sobre o simulador.

6.2 Topologia de simulação

Com o auxílio do *Network Simulator 2* (NS-2), realizou-se uma simulação do tráfego de dados em uma topologia com cinco nós conectados alterando o tipo de política de gerenciamento de pacotes utilizada no nó roteador. A figura 23 ilustra a topologia utilizada para essa simulação.

O cenário utilizado consiste de diversas fontes ($Cliente_x$, $Cliente_y$ até $Cliente_z$) de envio de dados, um roteador e somente um destino. As fontes estabelecerão conexões TCP com o destino, aqui chamado de *Servidor*, e transferirão dados por diversos protocolos da camada de aplicação da rede TCP/IP (*FTP*, *HTTP*) internamente a essa conexão.

Fontes: FTP, HTTP e UDP
 T_p : variando linearmente de 80ms a 120ms
 Capacidade do link: 100Mbps
 AQM do link: sem AQM (droptail)

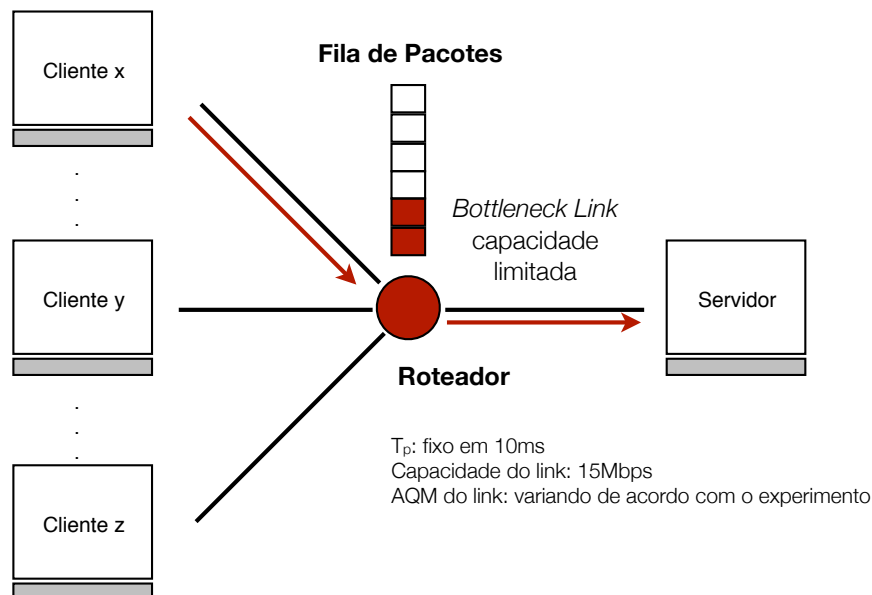


Figura 23: Topologia usada na simulação comparativa

A ligação entre o roteador e nó de destino foi definida como um *link* do tipo *bottleneck*, uma vez que sua capacidade de dados é relativamente menor que os *links* entre o roteador e emissores. Definiu-se a capacidade dessa ligação como 15 Mbps para que a capacidade dos *links* entre *Cliente* e roteador sejam aproximadamente seis vezes maior. Dessa forma, o roteador será obrigado, em alguns momentos, a armazenar pacotes em sua fila antes de enviá-los aos seus destinatários. Sendo esse *buffer* do roteador limitado a 800 pacotes via simulador, poderão haver descartes.

A transferência de dados se dará através de conexões do tipo FTP internamente à conexão TCP, enviando pacotes de 500 bytes de dados por um tempo de 200 segundos, iniciadas pelos diversos emissores. Para deixar a simulação mais coerente com as situações normalmente vistas na prática, o início da transmissão de cada fonte é aleatório, ou seja, elas podem começar em tempos diferentes.

Visto que a maioria dos fluxos que passam pela rede mundial de computadores é HTTP, adicionou-se diversas conexões HTTP internas à conexão TCP dos emissores. Para adequar a simulação com a situação corrente na realidade, determinou-se que esse número de conexões deveriam ser três vezes maiores que o número de conexões FTP (HOLLOT et al., 2002). Deve-se notar que, mesmo tendo um número de conexões maior, o fluxo HTTP ainda é pequeno se comparado com o fluxo FTP. O número de conexões FTP foi definido em 60 e, assim, o de HTTP em 180.

Em um segundo momento, uma forma de perturbação freqüentemente presente em redes TCP/IP será adicionada. Trata-se de fluxos UDP que passam obrigatoriamente pelo roteador, ocupando uma largura da banda de transmissão, mas não podem ser controlados pelos mecanismos implementados. Isso acontece pois esses pacotes seguem as regras do protocolo UDP que, devido ao seu mecanismo de funcionamento, não permite controles de congestionamento. Para

a simulação aqui apresentada, adotou-se como o número de fluxos UDP o valor respectivo a 10% do total de conexões FTP. Sendo assim, serão criados 6 conexões UDP passando através do roteador.

6.3 Simulação do comportamento dos AQMs

6.3.1 Simulação do comportamento da fila de pacote sem AQM

Como apresentado anteriormente, o *droptail* sugere o comportamento padrão implementado em equipamentos comutadores do protocolo (roteadores). Seu mecanismo consiste em simplesmente descartar todos os pacotes recebidos quando o *buffer* do equipamento está cheio. Pode-se considerar seu funcionamento extremamente rudimentar por não existir uma seleção dos pacotes que serão descartados.

A simulação feita no NS-2 pôde evidenciar exatamente o comportamento desse mecanismo. Como pode ser visto na figura 24, devido a limitação de tráfego imposta pela capacidade do *link* entre o roteador e o destinatário, o nível de pacotes armazenados no *buffer* do equipamento aumenta continuamente com o tempo, até que se atinja seu nível máximo. Não havendo política alguma no controle da fila, cada pacote adicional será descartado pelo roteador, evento que é detectado pelo TCP como congestionamento.

Na mesma figura, encontra-se o comportamento da fila de pacotes na presença de fluxos UDP. Pode-se notar o aumento na variação de $q(t)$ para um mesmo intervalo de tempo onde os pacotes presentes na fila do roteador são puramente pacotes TCP.

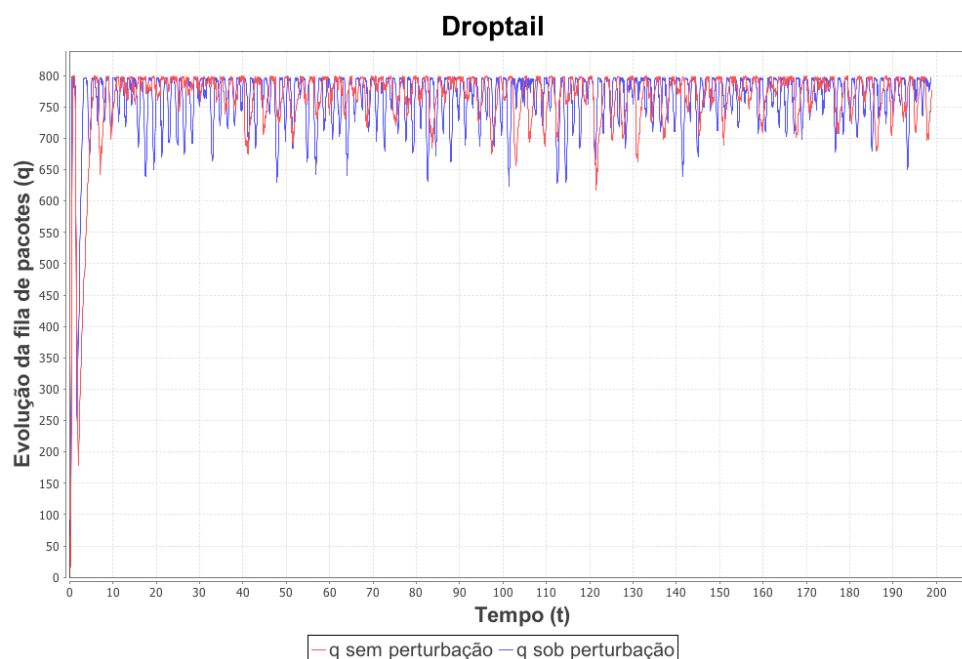


Figura 24: Comportamento da fila de pacotes no roteador - Droptail

Os emissores por sua vez, seguindo as regras do TCP, interpretam o descarte de pacotes como um sinal de congestionamento na rede e assim diminuem o envio de dados. Esse comportamento pode ser evidenciado na evolução da janela de congestionamento dos emissores, como mostra a

figura 25. Para aumentar a clareza do gráfico elaborado, mostra-se o comportamento de somente três emissores. Pode-se notar que a diminuição da janela acontece, dependendo do atraso do sistema, no momento em que os descartes começam ocorrer.

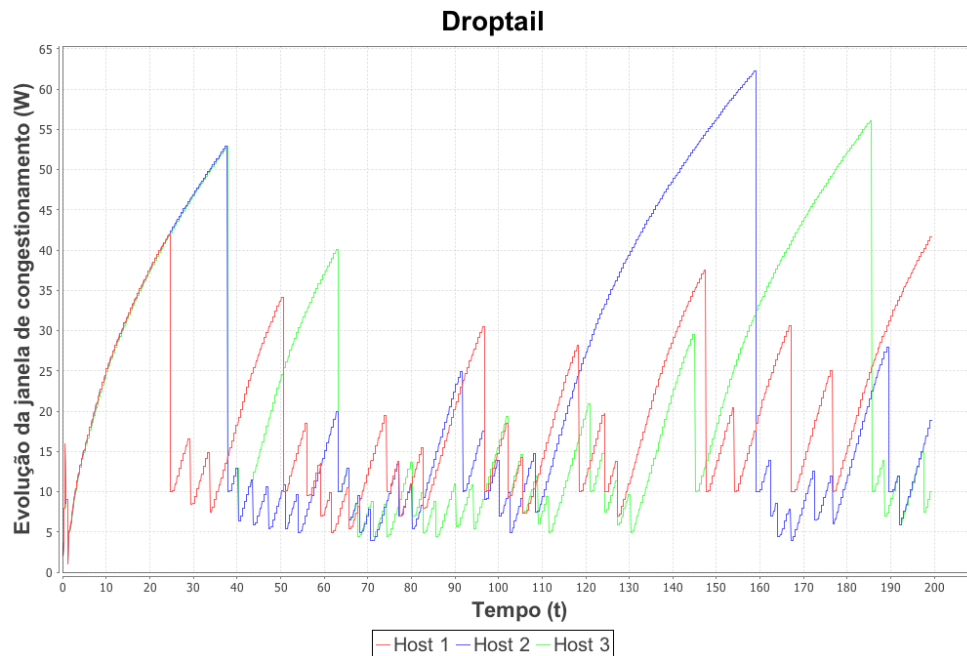


Figura 25: Evolução da janela de congestionamento das fontes - Droptail

Devido ao maior interesse deste trabalho pelo comportamento da fila de pacotes, e por uma questão de clareza, mostra-se a evolução da janela de congestionamento somente para o caso de conexões sob o protocolo TCP, desprezando a influência da presença de fluxos de dados UDP.

Pode-se destacar na figura 25 ainda o alto nível de sincronização entre as janelas dos diferentes emissores, situação a qual evidencia que todos os emissores perdem pacotes ao mesmo tempo.

6.3.2 Simulação do comportamento do AQM RED

Para a mesma topologia estabelecida na simulação do *droptail*, executou-se uma simulação utilizando o AQM RED. Diferentemente do *droptail*, no qual não existe controle anterior a saturação da fila do roteador, para cada pacote que entra no *buffer* desse equipamento, o RED vai associar uma probabilidade de descarte. Assim, quanto maior for a fila de pacotes, maior a probabilidade de descarte marcada no pacote.

Para a simulação do AQM RED, atribuiu-se o valor de 75 pacotes como o parâmetro min_{th} e 500 pacotes como max_{th} . Para maiores informações sobre a utilidade desses parâmetros, consultar a seção 3.2 deste relatório.

O comportamento da fila de pacotes no roteador pode ser ilustrado através da figura 26. É clara a lentidão de resposta desse AQM no início da simulação. Nesse ponto, seu comportamento se assemelha ao apresentado pelo mecanismo *droptail*. Contudo, ao passar do tempo, existe uma convergência em direção a um valor de equilíbrio. A implementação do AQM RED no roteador é, assim, traduzida pela variação da nível de preenchimento do *buffer* do equipamento em torno de um valor de equilíbrio.

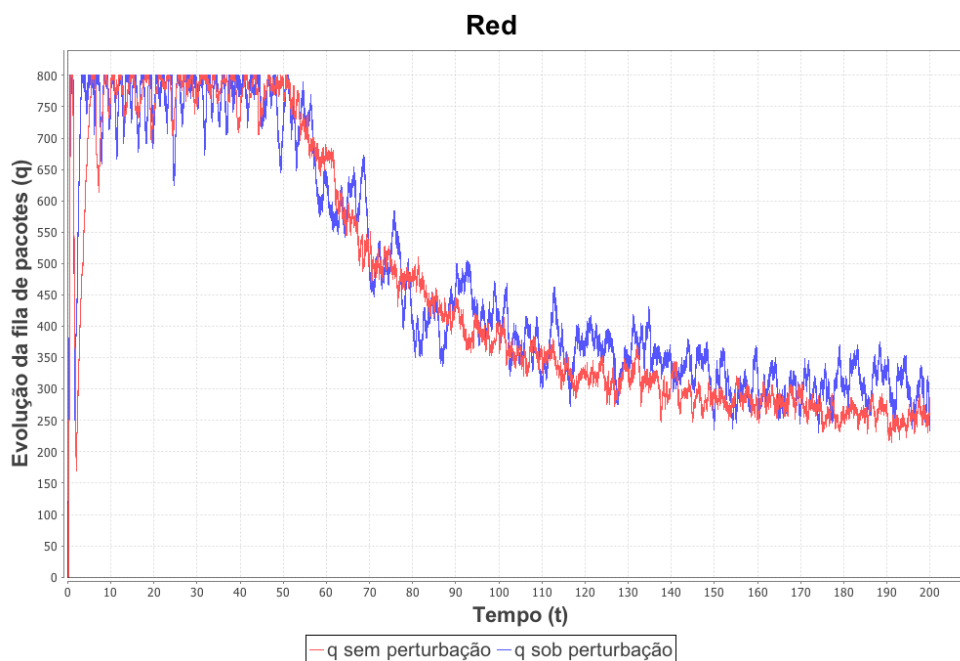


Figura 26: Comportamento da fila de pacotes no roteador - RED

No mesmo gráfico 26 é mostrada a reação do RED quando existem tráficos UDP passando pelo roteador. Um aumento na variação da fila de pacotes pode ser identificado, mas ainda assim existe um controle eficaz fazendo com que o sistema seja levado ao ponto de equilíbrio.

O sinal de controle gerado pelo AQM é mostrado na figura 27. Grandes variações são vistas no início da simulação para ambas as situações de perturbação. Uma probabilidade de equilíbrio maior pode ser evidenciada para o caso do sistema sob efeito de perturbações - como era de se esperar.

Com essa simulação pôde-se igualmente analisar a evolução da janela de congestionamento de três fontes emissoras de dados. A figura 28 mostra o resultado obtido. Deve-se notar a grande variação da janela e a dessincronização das perdas de pacotes, caráter do AQM RED. Devido ao descarte seletivo e de sua marcação de pacotes com probabilidades de descartes, o congestionamento é previsto e a taxa de transmissão é regulada diferentemente para cada fonte de forma reverter a situação e tornar o fluxo mais equilibrado.

6.3.3 Simulação do comportamento do AQM PI de Hollot

A simulação do controlador PI desenvolvido por Hollot é um pouco mais delicada por dois motivos. A primeira razão está relacionada ao fato de ser um algoritmo de controle que deve ser discretizado e introduzido no simulador por meio de uma implementação de código em C++. O segundo é quantidade de parâmetros os quais devem ser determinados para o tipo de sistema a ser controlado.

HOLLOT sintetiza um controlador proporcional integral para o comportamento de uma rede TCP/IP modelado por (MISRA; GONG; TOWSLEY, 2000). Em (HOLLOT et al., 2001), a frequência de corte e o ganho desse PI são equacionados em função dos parâmetros do sistema TCP/IP.

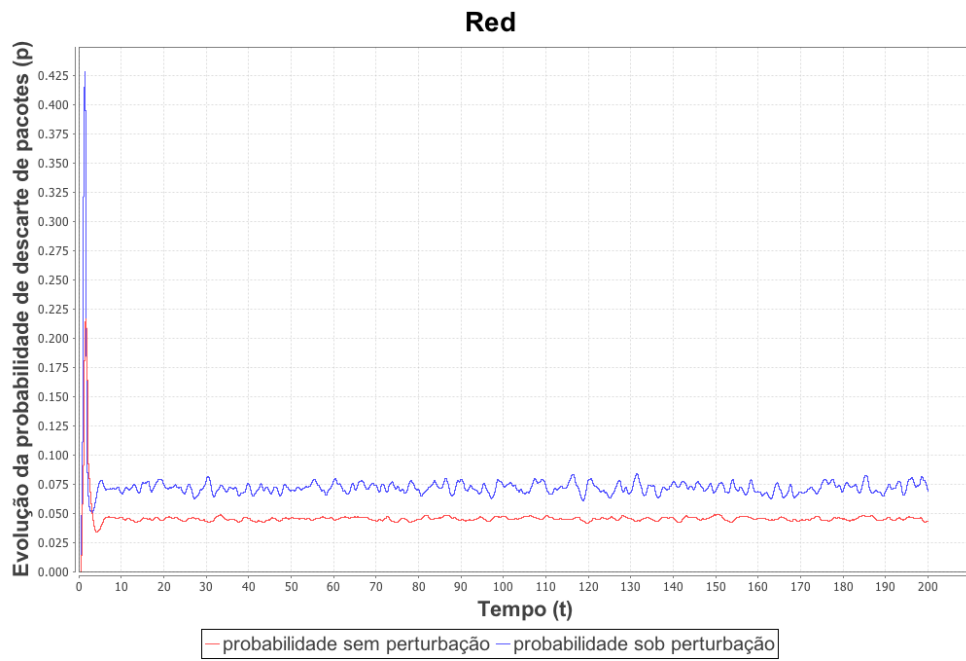


Figura 27: Probabilidade de descarte de pacotes - RED

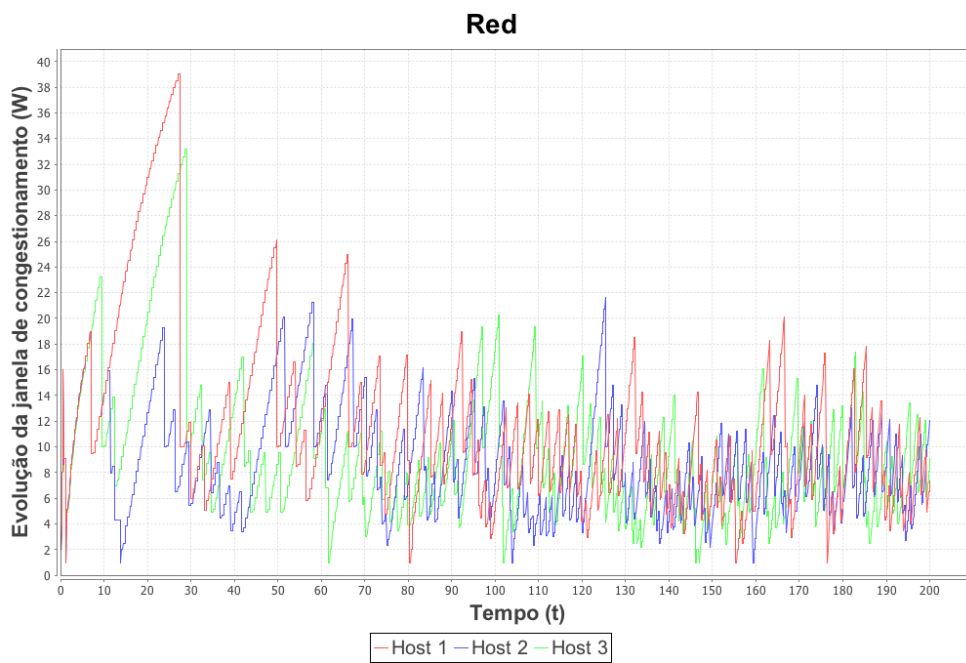


Figura 28: Evolução da janela de congestionamento das fontes - RED

De forma tornar a simulação do PI comparável com as demais realizadas anteriormente, se estabelece:

$$C = \frac{\text{capacidade do link}}{\text{tamanho médio do pacote}} = \frac{15 \text{ Mbits/s}}{500 \times 8 \text{ bits/pacote}} = 3750 \text{ pacotes/segundo}$$

$$N = \text{número de sessões TCP (fator de carga)} = \text{número de conexões FTP} = 60 \quad (62)$$

$$q_{ref} = \text{nível de referência da fila de pacotes} = 200 \text{ pacotes}$$

$$R = \text{tempo de ida e volta} = 0.5 \text{ segundos (máximo permitido)}$$

Apesar de ser um dos protocolos mais utilizados em redes TCP/IP, o fator de carga não leva em conta as sessões HTTP devido ao baixo grau de banda utilizado por elas. Mesmo elas apresentando uma quantidade três vezes maior, é muito difícil que ultrapassem uma parcela muito alta do valor de banda utilizado por uma conexão FTP.

O tempo de ida e volta (*Round Trip Time*) é estabelecido como o máximo dos RTTs do sistema. Como os atrasos de propagação vão linearmente variando, a determinação do valor exato não pode ser estabelecida. Essa é uma das razões que obriga o controlador a ser robusto. Em um sistema real, esses atrasos podem ter uma grande variação por diversos motivos e o controle deve tentar garantir a estabilidade do sistema para essas situações.

Partindo da expressão estabelecida por HOLLLOT para o controlador pode-se encontrar a frequência de corte e o ganho do PI para o sistema da simulação.

$$C(s) = \frac{p(s)}{q(s)} = K_{PI} \frac{\left(\frac{s}{w_g} + 1\right)}{s}$$

$$w_g = \frac{2N}{R^2 C} = 0.53 \text{ rad/s} \quad K_{PI} = \left| \frac{jw_g R + 1}{\frac{(R_o C)^3}{(2N)^2}} \right| = 9.6426 \cdot 10^{-6} \quad (63)$$

Como especificado no início desta subseção, para que o AQM sintetizado possa ser implementado no NS-2, existe a necessidade de se discretizar o controlador. Para tal, HOLLLOT utiliza os recursos Transformada Z aliado às relações desse método com a Transformada de Laplace (Transformação de Tustin). O resultado encontrado em (HOLLLOT et al., 2001) foi

$$C(z) = \frac{\delta p(z)}{\delta q(z)} = \frac{a z - b}{z - 1} = \frac{a z}{z - 1} - \frac{b}{z - 1} \quad (64)$$

o qual pode ser convertido em uma equação de diferenças

$$p(nT) = a \delta q(nT) - b \delta q((n - 1)T) + p((n - 1)T) \quad (65)$$

onde

$$a = K_{PI}\left(w_g + \frac{1}{2f_n}\right) \quad b = K_{PI}\left(w_g - \frac{1}{2f_n}\right) \quad (66)$$

Segundo o autor, para que o controlador digital tenha seu funcionamento adequado assegurado, a frequência de amostragem f_n deve ser escolhida de forma que seu valor seja de 10 à 20 vezes w_g . Sendo um pouco conservador e seguindo o valor sugerido por HOLLLOT, se adotará $f_n = 160$ Hz, o que resulta em $a = 1.822 \cdot 10^{-5}$ e $b = 1.816 \cdot 10^{-5}$.

Com todos os parâmetros necessários estabelecidos, pode-se executar a simulação do AQM PI através da modificação do AQM RED preestabelecido no simulador. Utilizou-se uma biblioteca adicional para agregar esse AQM ao NS já que a versão utilizada não possuía esse suporte. Contudo, na versão atual desse simulador, o AQM PI já pode ser utilizado sem qualquer alteração no núcleo do NS. A figura 29 mostra o resultado obtido para o comportamento da fila de pacotes no roteador.

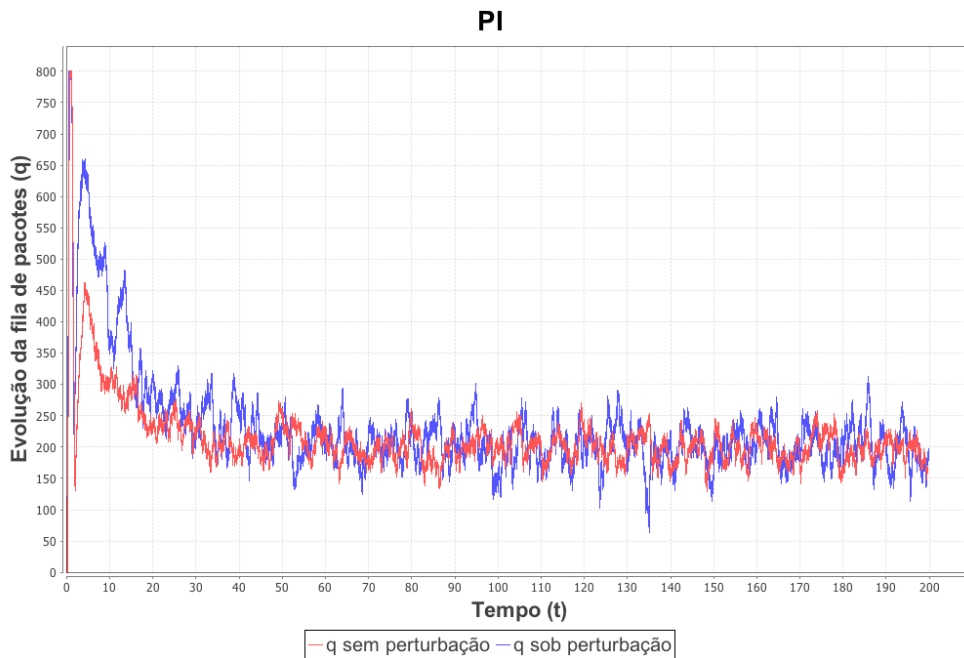


Figura 29: Comportamento da fila de pacotes no roteador - PI

A rápida resposta no tempo e igualmente a regulação da saída para q_{ref} é claramente observada. Mesmo sob efeito de perturbações, o que gera algumas oscilações adicionais, a referência é seguida.

Pode-se dizer que sua vantagem sobre o AQM RED vai além da velocidade. Isso pois, para um bom funcionamento do controlador, a frequência mínima de amostragem fica em torno de 10 à 20 vezes o valor da frequência de corte do controlador não discretizado. O exemplo aqui simulado adotou uma bordagem mais conservadora com $f_n \sim 100 \times w_g$, contudo, mesmo nesse nível, a necessidade de processamento é menor que a exigida pelo RED, o qual necessita da avaliação de todos os pacotes entrantes (3750 pacotes por segundo, o que implica em $f_n = 3750$ Hz).

O sinal de controle gerado pelo controlador de forma estabilizar o sistema é mostrado na

figura 30. Pode-se notar o efeito da perturbação sobre a variação do sinal de controle. Uma probabilidade de descarte maior é necessária para manter a fila de pacotes do roteador no nível de referência.

A inclusão de um fluxo perturbativo na razão de 10% fez a probabilidade de descarte se equilibrar em um valor duas vezes superior ao mesmo nível sem perturbação. É visível aqui a facilidade com que o sinal pode vir a saturar quando sob efeito de perturbações mais intensas.

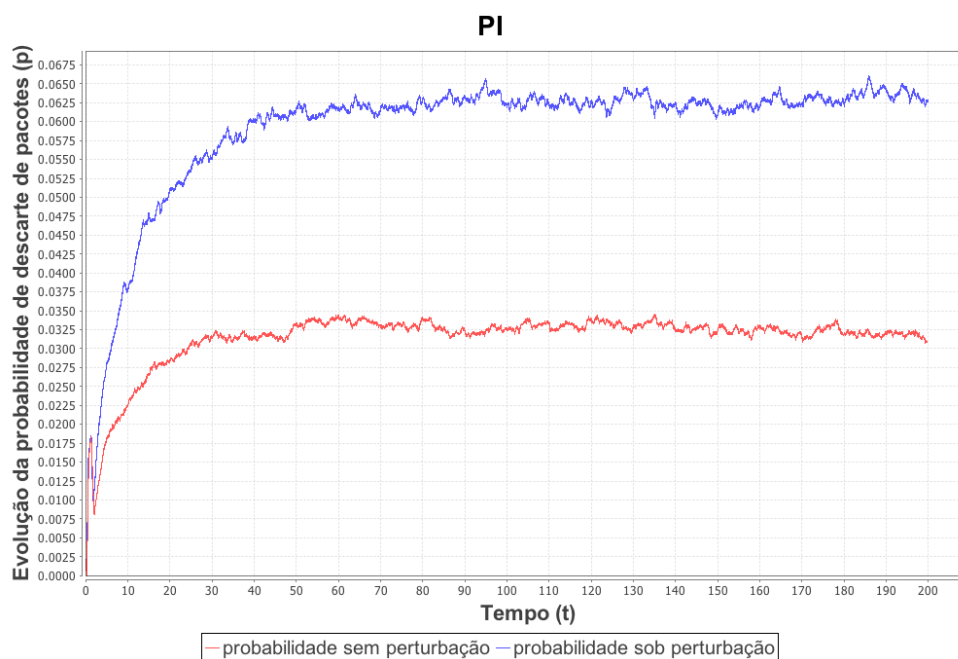


Figura 30: Probabilidade de descarte de pacotes - PI

Igualmente aos outros experimentos, a evolução da janela de congestionamento de três fontes é descrito na figura 31. Como no comportamento de $q(t)$, a velocidade de resposta do PI é largamente mais rápida que a obtida para o *droptail* ou para o AQM RED. Da mesma forma que para o RED, a sincronização da diminuição da janela é amenizada com o descarte seletivo de pacotes implementado.

6.4 Comparação AQM PI de Hollot com compensador *anti-windup* sintetizado

A superioridade da resposta do AQM PI é evidente nos gráficos mostrados até aqui. Como este trabalho tem por objetivo propor uma melhoria para esse mecanismo de gerência de tráfego, nada mais justo que comparar as respostas de um mesmo sistema utilizando as duas soluções propostas.

Essa comparação, entretanto, gera alguns problemas práticos. Os desenvolvedores que implementaram o mecanismo do PI no simulador codificaram igualmente um compensador *anti-windup* heurístico, sem qualquer formalidade. No momento do cálculo da probabilidade $p(n)$ por exemplo, utiliza-se o valor da probabilidade $p(n-1)$ como um dos limites do sinal de controle (0 ou 1) quando esse vinha a ficar saturado. Tal operação impede o incremento do termo integral, tendo quase que o mesmo efeito de um compensador *anti-windup*. Para probabilidades de descarte

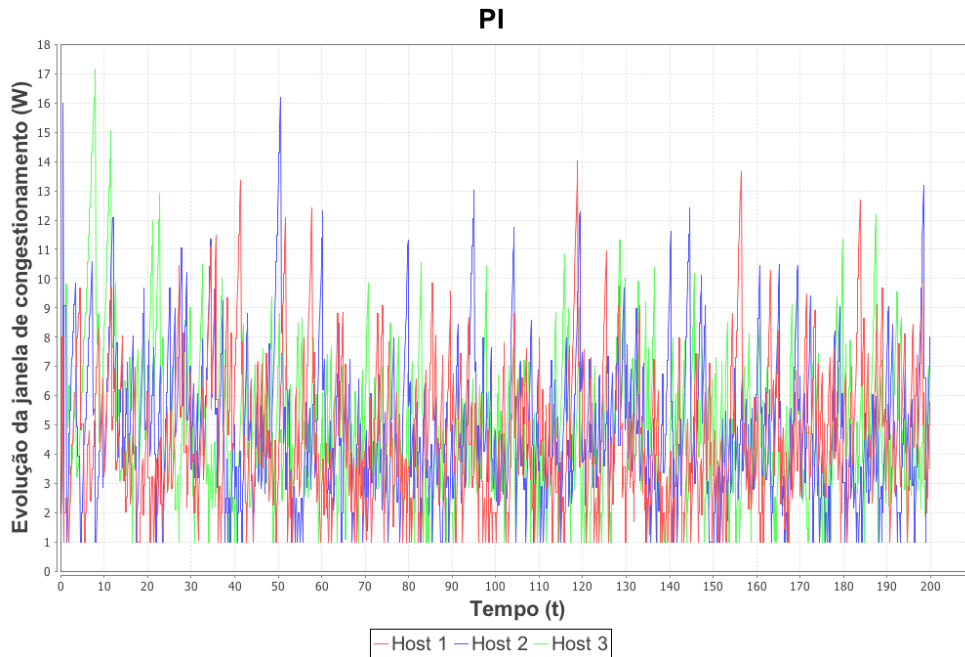


Figura 31: Evolução da janela de congestionamento das fontes - PI

dentro dos limites de saturação, contudo, o sistema implementado apresenta os mesmos resultados de uma implementação fiel do PI de HOLLLOT.

Assim, para que uma comparação justa seja efetuada, três casos foram analisados. Em um deles, retirou-se esse compensador sem qualquer formalismo matemático ou garantia, codificando fielmente o comportamento do PI estabelecido por HOLLLOT. Essa implementação foi chamada de PI-Fiel.

As simulações que seguem diferem das anteriores pelo fato de se aumentar consideravelmente a perturbação presente no sistema, simulando um tráfego UDP intenso. Essa configuração pode ser comparada com uma transmissão de vídeo-conferência sob uma rede TCP/IP. Tal topologia é justificada pela grande difusão e popularidade desse tipo de transferência de dados nos dias de hoje.

Parte-se então de um fluxo TCP semelhante ao anterior. O sistema será sujeito à 60 conexões TCP enviando dados continuamente. Assim que o sistema atingir o equilíbrio, isto é, a fila de pacotes no roteador assumir o valor de referência, 600 conexões UDP serão iniciadas no tempo de 100 segundos e seguirão até o final da simulação. Cada conexão UDP irá injetar pacotes UDP (não controláveis) com tamanho médio de 536 bytes. Tal perturbação fará com que o sinal de controle (probabilidade de descarte) venha a saturar, descartando continuamente os pacotes da fila.

Para essa situação pode-se estabelecer os seguintes parâmetros

$$\frac{1}{\mu} = \frac{\text{Capacidade do Link} \times \text{Tempo de Atividade} \times \text{Número de Conexões}}{\text{Tamanho do Pacote}}$$

$$\frac{1}{\mu} = \frac{100 \cdot 10^6 \text{ bits/s} \times 100 \text{ s} \times 600}{536 \times 8 \text{ bits}} = 1.4 \cdot 10^9 \text{ pacotes} \quad (67)$$

$$\mu = 714.29 \cdot 10^{-12}$$

o que resulta em um ganho *anti-windup* ótimo de rejeição à perturbação $E_c = 1.475 \cdot 10^5$.

Nessa situação, espera-se que o compensador *anti-windup* comprove sua eficiência fazendo com que o sistema atinja o equilíbrio mais rapidamente após a passagem desse tráfego perturbativo minimizando ainda o efeito desse distúrbio na resposta do sistema.

6.4.1 Implementação do compensador *anti-windup* sintetizado

A simulação do desempenho do AQM PI-AW necessita de algumas modificações para ser executada. Isto se mostra necessário devido ao fato do modelo linearizado ser contínuo e a implementação no simulador exigir algo discretizado.

Em razão das equações de controle do modelo ser um sistema linear, pode-se aproveitar grande parte do resultado obtido em (HOLLOT et al., 2002) na discretização do novo AQM. Parte-se então do modelo do controlador apresentado em (19), mas aqui alterado para representar um PI. Tal representação é definida por

$$\begin{cases} \dot{x}_c(t) = b_c q(t) + E_c \psi(t) \\ p(t) = c_c x_c(t) + d_c q(t) \end{cases} \quad (68)$$

Pode-se utilizar o fato do sistema (68) ser linear para facilitar as operações algébricas. Define-se com esse objetivo então

$$\begin{aligned} \dot{x}_c(t) &= x_{c(1)}(t) + x_{c(2)}(t) \\ x_{c(1)} &= b_c q(t) \\ x_{c(2)} &= E_c \psi(t) \end{aligned} \quad (69)$$

e assim

$$\begin{aligned} p(s) &= c_c \frac{x_{c(1)}(s)}{s} + c_c \frac{x_{c(2)}(s)}{s} + d_c q(s) \\ p(s) &= q(s) \left(K_{PI} \frac{\left(\frac{s}{wg} + 1\right)}{s} \right) + c_c \frac{E_c \psi(s)}{s} \end{aligned} \quad (70)$$

Comparando a expressão (70) com a equação (63) pode-se observar somente a adição do termo $c_c \frac{E_c \psi(s)}{s}$. Assim pode-se definir

$$p_{aw}(s) = p_{pi}(s) + c_c \frac{E_c \psi(s)}{s} \quad (71)$$

Logo, a função probabilidade de descarte discreta do controlador PI com o compensador *anti-windup* terá somente a necessidade de discretização e inclusão de um termo aditivo representado

por $c_c \frac{E_c \psi(s)}{s}$. Usando a Transformação de Turstin e utilizando o resultado obtido por (HOLLOT et al., 2002) para $p_{PI}(z)$, tem-se

$$p_{aw}(z) = p_{pi}(z) + c_c E_c \frac{T}{2} \frac{z}{z-1} + c_c E_c \frac{T}{2} \frac{1}{z-1} \quad (72)$$

Aplicando a Transformada Z inversa, encontra-se a função $p(nT)$ pronta para a implementação no simulador NS-2

$$p_{aw}(nT) = p_{pi}(nT) + c_c E_c \frac{T}{2} \delta\psi(nT) + c_c E_c \frac{T}{2} \delta\psi((n-1)T) \quad (73)$$

onde a função $\delta\psi(nT) = \psi(nT) - \psi((n-1)T)$ se considerado que $\psi_{ref} = 0$.

O pseudo-código da implementação pode ser representado por

$$\begin{aligned} p_{pi} &= a(q_{cur} - q_{ref}) - b(q_{old} - q_{ref}) + p_{old} \\ \text{if } (p_{pi} > 1.0) \{ p_{sat} &= 1.0 \} \\ \text{else if } (p_{pi} < 0.0) \{ p_{sat} &= 0.0 \} \\ \text{else } \{ p_{sat} &= p_{pi} \} \\ \\ phi_{cur} &= p_{sat} - p_{pi} \\ \\ delta_{cur} &= c_c E_c \frac{1}{2f_n} (phi_{cur} + phi_{old}) \\ \\ p_{aw} &= p_{pi} + delta_{cur} \\ \\ p_{old} &= p_{aw} \\ \\ \text{if } (p_{aw} > 1.0) \{ p_{sat} &= 1.0 \} \\ \text{else if } (p_{aw} < 0.0) \{ p_{sat} &= 0.0 \} \\ \text{else } \{ p_{sat} &= p_{aw} \} \\ \\ \text{return } p_{sat} \end{aligned} \quad (74)$$

6.4.2 Resultados

Dois gráficos principais foram retirados da simulação: nível da fila de pacotes no roteador e probabilidade de descarte. Na figura 32, o gráfico dessa primeira variável é mostrado. Nota-se a grande variação da fila de pacotes no momento em que a perturbação atua sobre o sistema ($t = 100s$). Com o objetivo de tornar a simulação mais próximo dos resultados teóricos esperados, a saturação do estado q não será considerada (tamanho da fila de pacotes que foi considerado como limitado em 800 pacotes nas primeiras simulações deste capítulo).

Efetuada uma aproximação na zona onde ocorre a saturação, pode-se observar uma convergência mais rápida do estado do PI-AW ao valor de referência se comparado com as respostas do PI e do PI-Fiel. Esse caso é ilustrado pela figura 33. Outra observação importante é sobre

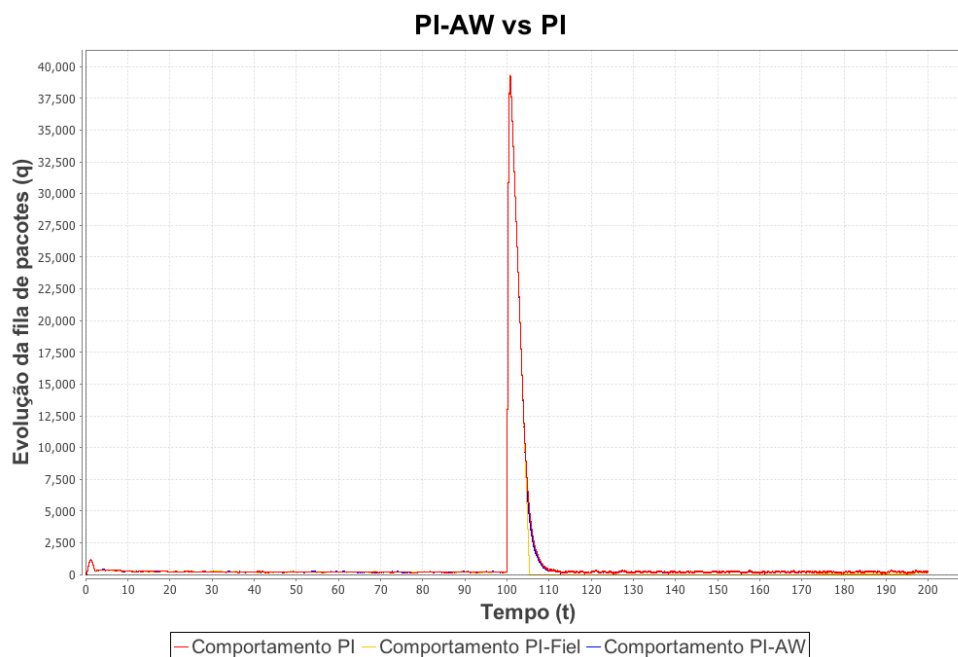


Figura 32: Comparação do nível da fila de pacotes - PI vs PI-AW

o comportamento do PI-Fiel. Percebe-se que o tamanho da fila de pacotes oscila consideravelmente. A natureza da variável q limita o valor mínimo a zero, caso contrário pode-se dizer que teria-se um valor negativo para esse estado. Somente muito perto do final de o PI-Fiel consegue fazer com que o sistema reaja e que a referência volte a ser seguida.

O sinal de controle resultante para essa simulação é mostrado na figura 34. Por alguns instantes, se vê a saturação do sinal de controle para os três mecanismos devido à presença da perturbação no sistema. O controlador tenta reagir ao distúrbio súbito agindo no sistema, mas é impossibilitado de usar toda a excursão que gostaria em razão do limite do sinal de controle.

A figura 35 ilustra uma amplificação do gráfico do sinal de controle na região de saturação. O sinal de controle do PI-Fiel permanece saturado por um tempo muito superior ao o sinal de controle para o PI-AW (PI com o compensador sintetizado aqui). Contudo, se comparados os sinais de controle do PI-AW com o PI implementado originalmente no simulador, percebe-se um melhor desempenho desse último. Deve-se notar que, apesar de um desempenho superior, o mecanismo implementado não formaliza garantia alguma - diferentemente do compensador *anti-windup* sintetizado neste trabalho. Trata-se somente de um mecanismo heurístico que apresentou resultados razoáveis para alguns casos.

Assim, as simulações puderam evidenciar alguns dos resultados teóricos esperados. Apesar de um desempenho inferior do PI-AW em relação ao PI original do NS-2, o objetivo do trabalho foi alcançado com a melhora da resposta do PI-Fiel (PI de HOLLLOT fielmente implementado). Existe assim uma relativa melhora na resposta do sistema com a implementação do compensador *anti-windup* estático.

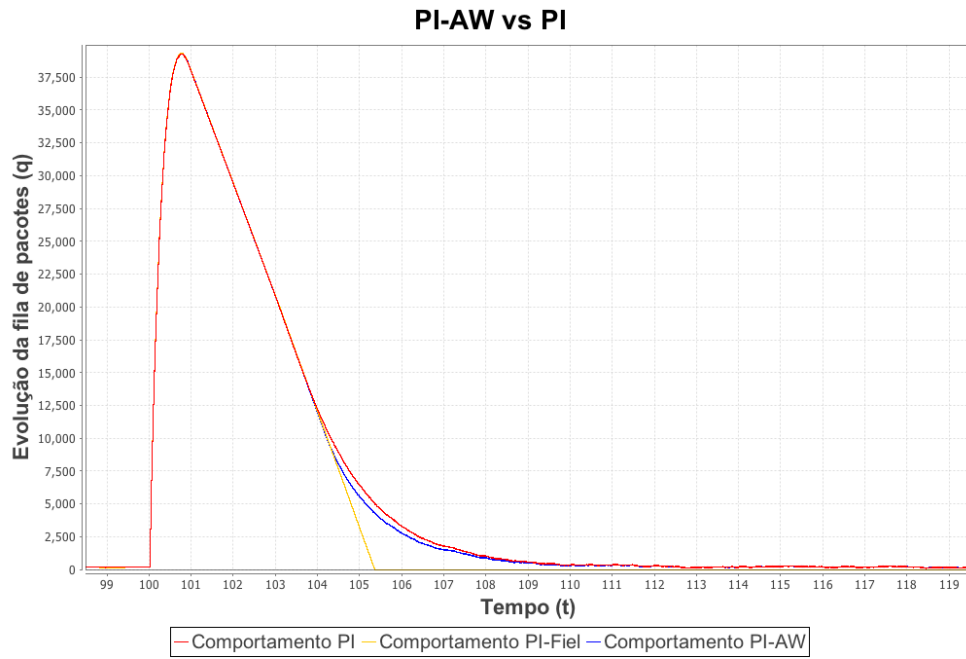


Figura 33: Comparação do nível da fila de pacotes (zoom) - PI vs PI-AW

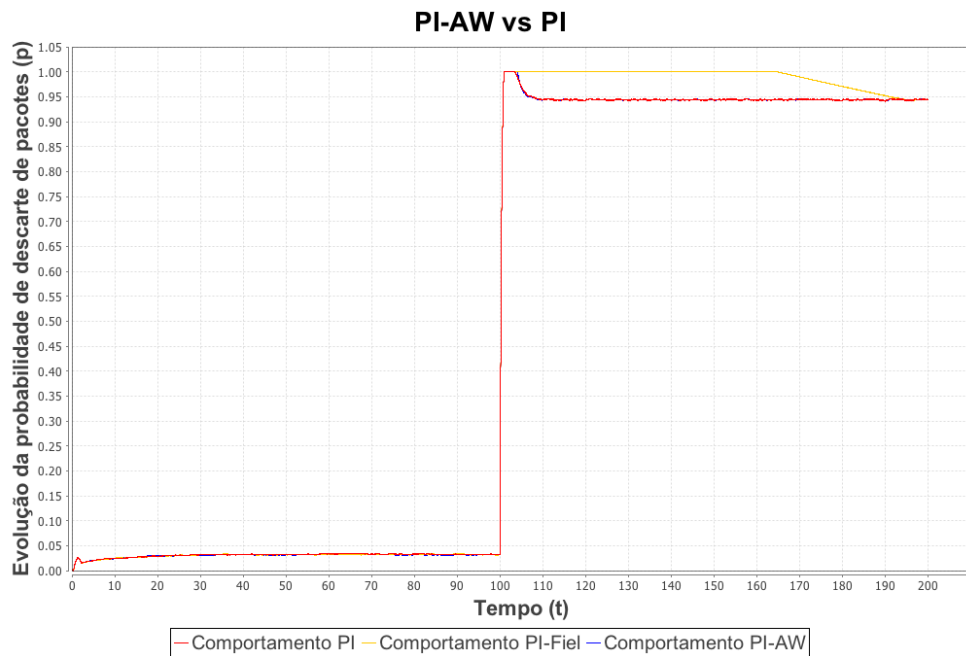


Figura 34: Comparação da probabilidade de descarte - PI vs PI-AW

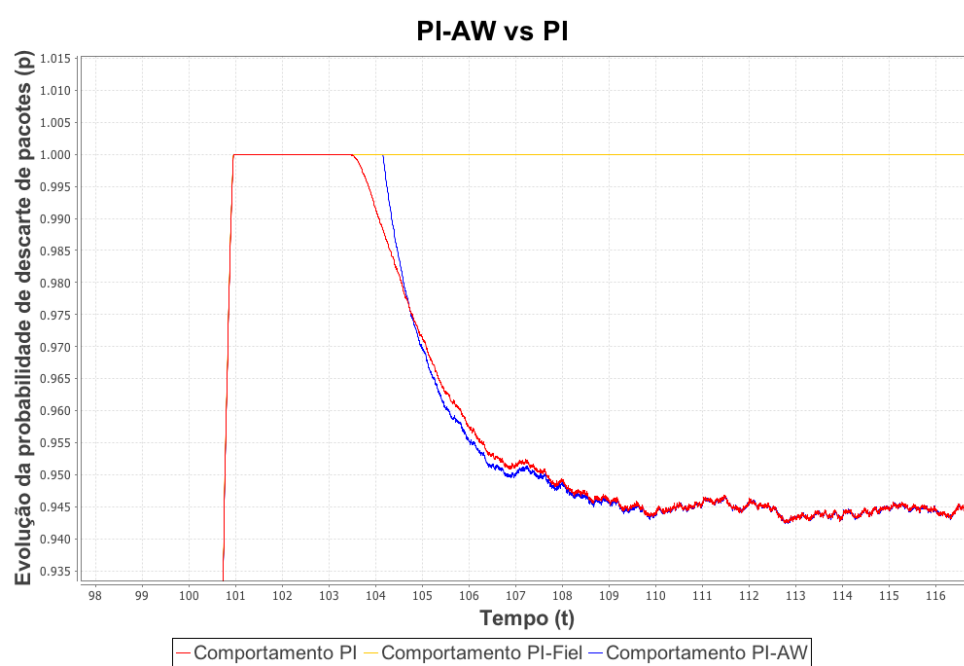


Figura 35: Comparação da probabilidade de descarte (zoom) - PI vs PI-AW

7 Conclusão

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. A abordagem de controle para esse problema é relativamente nova e vem apresentando bons resultados.

Estendendo os resultados de (GOMES DA SILVA JR.; TARBOURIECH; GARCIA, 2006) e (GOMES DA SILVA JR.; TARBOURIECH, 2000) assim como utilizando uma abordagem dependente do atraso com a esperança de se obter resultados menos conservadores, sintetizou-se um compensador *anti-windup* para o controlador PI apresentado em (HOLLOT et al., 2002), previamente desenvolvido para a gerência do tráfego TCP de um roteador congestionado em uma rede TCP/IP.

As condições suficientes de existência do compensador *anti-windup* para os dois requisitos de Rejeição à Perturbação e Tolerância à Perturbação foram encontradas através da utilização de funcionais de Lyapunov-Krasovskii. Um artifício matemático chamado de *condição de setor generalizada* (GOMES DA SILVA JR.; TARBOURIECH, 2005) foi igualmente usado de forma considerar a saturação do sinal de controle modelada como uma não linearidade do sistema. Essas condições foram expressas diretamente em forma de LMIs que, quando respeitadas, garantem a estabilidade assintótica do sistema em malha fechada.

Através das simulações com o programa Matlab e NS-2 pode-se constatar uma melhora significativa da resposta do sistema quando o sinal de controle está sob o efeito de saturação devido à eventuais perturbações. O controlador PI de HOLLOT combinado com o compensador *anti-windup* permitiu uma reação mais rápida do controle sobre o sistema assim como uma redução nas oscilações do sinal de saída.

Paralelamente, devido a importância de simulações computacionais para a avaliação de novos protocolos e mecanismos de rede, realizou-se um grande estudo sobre o simulador de rede NS-2. Embora este simulador seja uma das principais ferramentas de simulação de redes utilizadas no mundo acadêmico nos dias de hoje, não se tem acesso a uma documentação adequada tanto para sua instalação e configuração quanto para a modificação dos mecanismos ali disponíveis a fim de se executar novos experimentos.

Por esse motivo, uma seção deste trabalho foi dedicada à explicação do funcionamento do programa *Network Simulator 2*. Diversas simulações com diferentes mecanismos de gerenciamento de fila do roteador foram implementadas com o objetivo de avaliar e comparar seus desempenhos.

A contribuição deste trabalho, portanto, está na proposição de mecanismos alternativos os quais atenuam os efeitos negativos que as constantes variações dos parâmetros de uma rede podem causar tanto no seu próprio comportamento quanto no comportamento dos roteadores e dos sistemas finais lá contidos.

7.1 Trabalhos futuros

O modelo do protocolo TCP da camada de transporte de redes TCP/IP mostrou-se apropriado, pois possibilitou capturar a dinâmica desse protocolo de forma satisfatória, contribuindo para um bom projeto de sistemas de controle. Contudo, a adição dos fenômenos de *timeout*, e assim conseqüentemente do mecanismo de *slow start*, nas equações do capítulo 3 podem tornar o modelo ainda mais realista.

O trabalho aqui descrito considerou o comportamento do sistema em torno de seus pontos de equilíbrio. A linearização do modelo estabelecido por (MISRA; GONG; TOWSLEY, 2000) se mostrou necessária, adicionando algumas limitações para a validade do compensador sintetizado. Dessa forma, uma abordagem de controle não linear também pode ser avaliada com o objetivo de diminuir as restrições impostas.

Em relação a abordagem utilizada para síntese do compensador *anti-windup*, levou-se em consideração somente a saturação em amplitude do sinal de controle no atuador. Resultados mais completos poderiam ser obtidos com a consideração de sistemas lineares sujeitos à saturação do atuador em amplitude e taxa de variação. Outra forma de saturação não considerada é a saturação de estados do sistema. Devido a limites físicos do roteador, o tamanho de seu *buffer* é limitado, adicionando assim uma restrição adicional no problema de síntese.

Finalmente, como trabalho futuro, pode se implementar os controles propostos para o gerenciamento ativo de fila para uma rede real. Tal fato seria possível com a discretização das leis de controle propostas para a programação em um roteador imerso em uma rede TCP/IP real - uma rede residencial, por exemplo, podendo ser estendido à redes maiores com o objetivo de se adquirir resultados mais completos.

Referências

- ARIBA, Y. **Analyse et Commande de Reseaux TCP**. 2005-2006. 46p. Dissertação (Mestrado em Engenharia Elétrica) — SMIS-EEAS, Toulouse/França.
- BENDER, F. A. **Técnicas de Compensação Dinâmica para Sistemas Lineares com Saturação de Controle**. 2006. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul UFRGS, Porto Alegre/Brasil.
- BENDER, F. A. **Técnicas de Síntese de Compensadores Anti-windup para Sistemas com Atraso no Tempo**: aplicação ao caso de redes TCP/IP. Porto Alegre/Brasil, 2008.
- BENDER, F. A.; GOMES DA SILVA JR., J. M.; TARBOURIECH, S. **A convex framework for the design of dynamic anti-windup for state-delayed systems**. 2010.
- BOYD, S.; GHAOUI, E.; BALAKRISHMAN, V. **Linear Matrix Inequalities in System and Control Theory**. Estados Unidos: PA: SIAM, 1994.
- CAO, Y.; LIN, Z.; HU, T. Stability Analysis of Linear Time-Delay Systems Subject to Input Saturation. **IEEE Control Systems**, New York/Estados Unidos, 2002.
- FIELDING, R.; GETTYS, J.; MOGUL, J. RFC 2616: hypertext transfer protocol - HTTP/1.1. **The Internet Society**, N/A, 1999.
- FLOYD, S.; JACOBSON, V. Random Early Detection Gateways for Congestion Avoidance. **IEEE/ACM Transactions on Networking**, New York/Estados Unidos, 1993.
- GOMES DA SILVA JR., J. M.; BAZANELLA, A. S. **Sistemas de Controle**: princípios e métodos de projeto. Porto Alegre/Brasil: Editora UFRGS, 2005. 246-250p.
- GOMES DA SILVA JR., J. M.; SEURET, A.; FRIDMAN, E.; RICHARD, J.-P. Stabilization of Neutral Systems with Saturating Control Inputs. **International Journal of Systems Science**, N/A, 2009.
- GOMES DA SILVA JR., J. M.; TARBOURIECH, S. Synthesis of Controllers for Continuous-time Delay Systems with Saturating Controls Via LMIs. **IEEE Transactions on Automatic Control**, New York/Estados Unidos, 2000.
- GOMES DA SILVA JR., J. M.; TARBOURIECH, S. Synthesis of Controllers for Continuous-time Delay Systems with Saturating Controls Via LMIs. **IEEE Transactions On Automatic Control**, New York/Estados Unidos, 2005.
- GOMES DA SILVA JR., J. M.; TARBOURIECH, S.; GARCIA, G. Anti-windup Design for Time-Delay Systems Subject to Input Saturation: an LMI-based approach. **European Journal of Control**, N/A, 2006.
- HOLLOT, C.; MISRA, V.; GONG, W.-B.; TOWSLEY, D. A Control Theoretic Analysis of RED. **IEEE INFOCOM**, New York/Estados Unidos, 2001.

- HOLLOT, C.; MISRA, V.; GONG, W.-B.; TOWSLEY, D. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows. **IEEE Transactions on Automatic Control**, New York/Estados Unidos, 2002.
- HUNT, C. **TCP/IP Network Administration**. Estados Unidos: O'Reilly Associates, 1997.
- JACOBSON, V. Congestion Avoidance and Control. **ACM SIGCOMM**, N/A, 1988.
- KHALIL, H. **Non-Linear Systems**. Estados Unidos: Prentice-Hall, 1996.
- MISRA, V.; GONG, W.-B.; TOWSLEY, D. Stochastic Differential Equation Modeling and Analysis of TCP-Window Size Behavior. **ECE-TR-CCS**, N/A, 1999.
- MISRA, V.; GONG, W.-B.; TOWSLEY, D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. **ACM SIGCOMM**, N/A, 2000.
- MISRA, V.; GONG, W.-B.; TOWSLEY, D. A Control Theoretic Analysis of RED. **IEEE/INFOCOM**, N/A, 2001.
- NICULESCU, S. Delay Effects on Stability: a robust control approach. **Springer-Verlag**, Berlin/Alemanha, 2001.
- OLIVEIRA JESUS COSTA, S. S. de. **QoS: quality of service**. 2006.
- PARK, J.-K.; YOON, T.-W.; YOON, S.-S. Dynamic Anti-windup Scheme for Feedback Linearizable Nonlinear Control Systems with Saturating Inputs. **IEEE Transactions on Automatic Control**, New York/Estados Unidos, 2004.
- POSTEL, J. RFC 793: transmission control protocol - TCP. **DARPA Internet Program Protocol Specification**, N/A, 1981.
- REYNOLDS, J.; POSTEL, J. RFC 959: file transfert protocol - FTP. **Network Information Center**, N/A, 1985.
- SILVA CARISSIMI, A. da; JUERGEN, R.; GRANVILLE, L. Z. **Redes de Computadores**. Porto Alegre/Brasil: Bookman, 2009.
- SUSSMANN, H.; SONTAG, E.; YANG, Y. A General Result for the Stabilization of Linear Systems Using Bounded Controls. **IEEE Transactions On Automatic Control**, New York/Estados Unidos, 1994.
- TANENBAUM, A. S. **Computers Network**. Holanda: Pearson, 1997.
- TROFINO, A.; COUTINHO, D.; BARBOSA, K. Sistemas Multivariáveis: uma abordagem via LMIs. **UFSC**, Florianópolis/Brasil, 2002.

Apêndice A Ferramental matemático

A.1 Linear matrix inequalities (LMIs)

O aparecimento das primeiras formas de LMIs datam de 1890, quando Lyapunov, matemático Russo, publicou um trabalho introduzindo o que chama-se hoje de teoria de Lyapunov. Nesse trabalho, Lyapunov provou que a equação diferencial

$$\dot{x}(t) = Ax(t) \quad (75)$$

é estável se e somente se existir uma matriz P ($P > 0$) positiva definida tal que

$$A^T P + PA < 0 \quad (76)$$

Essa desigualdade, mais tarde chamada de *desigualdade de Lyapunov*, foi a primeira LMI usada para analisar a estabilidade de sistemas dinâmicos.

Contudo, 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. O trabalho de (BOYD; GHAOUI; BALAKRISHMAN, 1994) foi decisivo para a incorporação das LMIs na síntese de controladores e análise de sistemas sujeitos a restrições no controle.

Linear Matrix Inequalities emergiram, assim, como poderosas ferramentas para se solucionar problemas que vão desde engenharia controle até *design* de estruturas. Esse fato é motivado principalmente pela facilidade de se especificar uma grande gama de restrições em forma de LMIs. Além, uma vez o problema formulado em forma de LMIs, sua solução pode ser encontrada eficientemente por métodos de otimização convexa disponíveis em software comerciais como MatLab entre outros.

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 A.1.1 (Linear Matrix Inequality). *De acordo com (TROFINO; COUTINHO; BARBOSA, 2002), uma LMI é uma desigualdade matricial do tipo $F(g) > 0$, no qual $F(g) : \mathbb{R}^n \rightarrow \mathbb{R}^{q \times q}$ é simétrica e afim nas variáveis de busca que são representadas pelo vetor g . Assim uma LMI pode ser genericamente representada na forma*

$$F(g) = F_o + \sum_{i=1}^m g_i F_i > 0 \quad \mathbf{g} = \begin{bmatrix} g_1 \\ \dots \\ g_m \end{bmatrix} \quad (77)$$

onde $F_i = F_i^T \in \mathfrak{R}^{q \times q}$ 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.

A.2 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.

Lemma A.2.1 (Complemento de Schur). *Seja $g \in \mathfrak{R}^m$ 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*

$$M_1(g) - M_3(g)M_2(g)^{-1}M_3(g)^T > 0 \quad (78)$$

$$\begin{bmatrix} M_1(g) & M_3(g) \\ M_3(g)^T & 0 \end{bmatrix} > 0 \quad (79)$$

Se as matrizes $M_1(x)$, $M_2(x)$ e $M_3(x)$ dependem finamente da variável x , então a desigualdade matricial 79 é uma LMI.

A.3 S-Procedure

Freqüentemente, na teoria de controle, encontram-se problemas no qual alguma 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 *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.

Lemma A.3.1 (S-Procedure). *Sejam F_0, \dots, F_p funções quadráticas em uma variável $\mu \in \mathfrak{R}^n$ seguindo a expressão*

$$F_i(\mu) = \mu' T_i \mu + 2u_i' + c_i \quad \text{para } i = 0, \dots, p \quad (80)$$

onde $T_i = T_i' \in \mathfrak{R}^{n \times n}$, $u_i \in \mathfrak{R}^n$ e $c_i \in \mathfrak{R}$.

Logo, a condição $F_0 \geq 0$ para todo μ tal que $F_i \geq 0$ para $i = 1, \dots, p$ é satisfeita se existem escalares $\alpha_1, \dots, \alpha_p \geq 0$ tais que

$$F_0(\mu) - \sum_{i=1}^p \alpha F_i(\mu) \geq 0 \quad \text{para } \forall \mu \quad (81)$$

Apêndice B Linearização do modelo para fluxo de pacotes

O modelo desenvolvido por (MISRA; GONG; TOWSLEY, 2001) para modelar o comportamento da janela de congestionamento de transmissões de pacotes em uma rede regida pelo TCP não é linear. Para um estudo em torno do ponto de equilíbrio, pode-se fazer a linearização desse conjunto de equações. Grande parte do desenvolvimento aqui descrito é apresentado igualmente em (ARIBA, 2005-2006).

B.1 Definições

De forma tornar a demonstração mais clara e objetiva, defini-se as funções f e g da seguinte forma

$$\begin{aligned} f(W, W_R, q, q_R, p_R) &\doteq \frac{1}{\frac{q}{C} + T_p} - \frac{W W_R}{2(\frac{q_R}{C} + T_p)} p_R \\ g(W, q) &\doteq \frac{N}{\frac{q}{C} + T_p} W - C \end{aligned} \quad (82)$$

onde $W_R(t) \doteq W(t - h(t))$ e $q_R(t) \doteq q(t - h(t))$ são as variáveis de estado, $p_R(t) \doteq p(t - h(t))$ a entrada do sistema e $h(t) = \frac{q}{C} + T_p$.

B.2 Desenvolvimento

Seguindo as definições previamente estabelecidas, tenta-se exprimir as derivadas parciais de f e g em torno de seus pontos de equilíbrio. Tais ponto são determinados na condição de ausência de dinâmica, ou seja, $\dot{f} = 0$ e $\dot{g} = 0$.

$$\begin{aligned} \dot{f}(W, W_R, q, q_R, p_R) = 0 &\implies W_o^2 p_o = 2 \\ \dot{g}(W, q) = 0 &\implies W_o = \frac{R_o C}{N}, \quad R_o = \frac{q_o}{C} + T_p \end{aligned} \quad (83)$$

Logo, tem-se

$$\begin{aligned}\frac{\partial f}{\partial W} &= \frac{\partial f}{\partial W} \partial W + \frac{\partial f}{\partial W_R} \partial W_R + \frac{\partial f}{\partial p_R} \partial p_R + \frac{\partial f}{\partial q} \partial q + \frac{\partial f}{\partial p_R} \partial q_R \\ \frac{\partial g}{\partial W} &= \frac{\partial g}{\partial W} \partial W + \frac{\partial g}{\partial q} \partial q\end{aligned}\tag{84}$$

onde pode-se definir as derivadas parciais como

$$\begin{aligned}\frac{\partial f}{\partial W} &= -\frac{W_R}{2\left(\frac{q_R}{C} + T_p\right)} p_R = -\frac{W_o}{2\left(\frac{q_o}{C} + T_p\right)} p_o = -\frac{W_o p_o}{2R_o} = -\frac{N}{R_o^2 C} \\ \frac{\partial f}{\partial W_R} &= \frac{\partial f}{\partial W} \\ \frac{\partial f}{\partial p_R} &= -\frac{W W_R}{2\left(\frac{q_R}{C} + T_p\right)} = -\frac{W_o^2}{2R_o} = -\frac{C^2 R_o}{2N^2} \\ \frac{\partial f}{\partial q} &= -\frac{\frac{1}{C}}{\left(\frac{q}{C} + T_p\right)^2} = -\frac{1}{C\left(\frac{q_o}{C} + T_p\right)^2} = -\frac{1}{C R_o^2} \\ \frac{\partial f}{\partial q_R} &= -\frac{W W_R \frac{1}{C}}{2\left(\frac{q}{C} + T_p\right)^2} = -\frac{W_o^2 p_o}{2C\left(\frac{q_o}{C} + T_p\right)^2} = -\frac{1}{C R_o^2} \\ \frac{\partial g}{\partial W} &= \frac{N}{\frac{q}{C} + T_p} = \frac{N}{\frac{q_o}{C} + T_p} = \frac{N}{R_o} \\ \frac{\partial g}{\partial q} &= -\frac{N W \frac{1}{C}}{\left(\frac{q}{C} + T_p\right)^2} = -\frac{N W_o}{C\left(\frac{q_o}{C} + T_p\right)^2} = -\frac{1}{R_o}\end{aligned}\tag{85}$$

Apêndice C Simulador NS-2 e o controle de congestionamento em redes TCP

C.1 Instalação

O NS é uma ferramenta em constante desenvolvimento no projeto VINT (Virtual InterNet Testbed). O VINT é um projeto conjunto entre a Universidade da Califórnia em Berkeley, o LBNL (Lawrence Berkeley National Laboratory), o ISI (Information Sciences Institute) da USC (University of South California) e o laboratório Xerox PARC (Palo Alto Research Center).

O simulador é gratuito e pode ser obtido através do carregamento direto de seus arquivos fontes na página web do projeto <http://www.isi.edu/nsnam/ns/>. Diversas plataformas são suportadas, desde Windows até sistemas baseados em Unix (FreeBSD, Linux, SunOS, Solaris). Existe um suporte para Macintosh, mas ele se restringe às máquinas com processadores mais antigos (antigos IBM).

Existem, basicamente, duas formas diferentes de adquirir o simulador. Como NS-2 é formado por diversos módulos opcionais e dependências, existe a opção de se carregar somente os componentes que serão efetivamente utilizados para a tarefa que se deseja executar. Para usuários menos experientes, a opção *all at once* se revela a melhor escolha, uma vez que o ganho de espaço não é significativo e a dificuldade de instalação decresce consideravelmente.

O processo de instalação do NS-2 se resume na compilação do código fonte do programa. Como já explicado anteriormente, a simulação de cenários de redes pode ser bastante custosa em termos de tempo de cálculo. Devido à natureza do código do programa, sistemas operacionais baseados em Unix apresentam uma performance superior aos demais compatíveis. Por essa razão, descreve-se o processo de instalação utilizando o pacote de instalação *all at once* no sistema operacional do tipo Linux Ubuntu 9.04.

Processo de Instalação

1. Realizar o carregamento do arquivo *all at once* em <http://www.isi.edu/nsnam/ns/ns-build.html#allinone>
2. Descompactar o arquivo carregado no diretório do usuário. Em sistemas do tipo Linux, normalmente o caminho desse diretório é dado por `\home\user`, onde *user* representa o nome de usuário atual logado no computador. O diretório resultante após a descompactação do arquivo será *ns-allinone-x.yz*, onde *x.yz* representa a versão do simulador. Até o momento da confecção deste trabalho, a versão corrente era 2.34.
3. Existe um detalhe importante quanto ao nível de permissões dadas aos arquivos do simulador. Por configuração padrão, o Ubuntu 9.04 bloqueia a execução de qualquer arquivo como programa. Por essa razão, antes de executar o arquivo de instalação, deve-se habilitar essa funcionalidade.

- (a) Clica-se sobre o diretório descompactado com botão direito e em seguida em *Propriedades*.
 - (b) Seleciona-se a aba *Permissões*.
 - (c) Marca-se a opção *Permitir execução do arquivo como programa*.
 - (d) Clica-se sobre o botão *Aplicar as permissões em arquivos internos*.
4. Após a configuração de permissões, deve-se abrir o terminal e digitar alguns comandos para o início da instalação. O leitor deve notar que o código descrito a seguir foi o utilizado para a execução da instalação da versão 2.34 do programa NS-2. Para outras versões, a alteração da linha de acesso ao diretório dos arquivos é necessária.

```
$ cd ns-allinone-2.34
$ sudo apt-get install build-essential autoconf automake libxmu-dev
$ ./install
$ ./configure
```

5. Sendo bem sucedida a instalação, resta agora a atualização das variáveis de ambiente do sistema. Esse processo é necessário para que as chamadas ao simulador pelo terminal de comandos possa ser efetuada de uma forma mais intuitiva. Novamente no terminal, digita-se o comando:

```
$ gedit ~/.bashrc
```

6. O editor padrão de texto será iniciado com o conteúdo do arquivo que contém as declarações das variáveis de ambiente. Devido ao grande número de versões das dependências do simulador, é necessário ter um cuidado adicional para que todos os caminhos dos módulos declarados estejam corretos. Deve-se adicionar o seguinte trecho ao final do arquivo, modificando-o de acordo com o caminho dos módulos do NS-2. É muito importante ressaltar que a modificação correta dessas linhas é imprescindível para a execução das simulações.

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/root/ns-allinone-2.34/otcl-1.13
NS2_LIB=/home/root/ns-allinone-2.34/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/home/root/ns-allinone-2.34/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/home/root/ns-allinone-2.34/bin:
/home/root/ns-allinone-2.34/tcl8.4.18/unix:
/home/root/ns-allinone-2.34/tk8.4.18/unix
NS=/home/root/ns-allinone-2.34/ns-2.34/
NAM=/home/root/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

7. Alterações efetuadas e salvas, deve-se aplicar as modificações. Para que não seja necessária a reinicialização do sistema, executa-se no terminal o comando:

```
$ source ~/.bashrc
```

8. O item final para o processo de instalação é um teste para verificar se a compilação foi realizada com sucesso. Novamente no terminal, deve-se digitar:

```
$ ns
```

Estando as variáveis de ambiente corretamente declaradas, o simulador será executado. Essa ação é representada pelo caractere % impresso no terminal. Para sair do programa, é necessário digitar o comando *exit* e precionar a tecla *return*.

Todas as instruções acima descritas formam um tutorial passo-à-passo para instalação e execução do simulador de redes NS-2. A seção que segue introduzirá alguns comandos básicos da linguagem utilizada para o desenvolvimento de *scripts* para simulação de topologias.

C.2 Operação

O simulador NS compreende diversas funcionalidades dentre aplicações, protocolos, topologia de redes, elementos de redes e modelos de tráfego. O NS foi desenvolvido e atua sobre duas linguagens de programação, a saber: C++ e OTCL. A linguagem de programação C++ é utilizada no núcleo do simulador e, com isso, busca-se um melhor desempenho com relação à execução das simulações devido ao fato de ser uma linguagem compilada. Já a OTCL - extensão orientada a objetos da linguagem TCL - é utilizada para a escrita dos *scripts* de simulação, que contém as diversas informações a respeito da rede de comunicação e que serão interpretados pelo simulador.

C.2.1 Simulações de tráfego em rede TCP

Um conhecimento básico dos métodos implementados no núcleo do NS é fundamental para a síntese de simulações com topologias de rede. Apresenta-se, a seguir, um exemplo que sintetiza a simulação de uma topologia simples, formada por dois nós se comunicando entre si através de um *link* passando uma conexão TCP. Cada trecho será explicado separadamente.

No NS, toda a caracterização de topologias deve ser realizada editando-se um arquivo texto externo. Essa metodologia colabora para que o código se desenvolva de forma limpa e clara, evitando erros e facilitando a depuração.

Como em qualquer linguagem de programação, antes de qualquer definição de valores a objetos ou variáveis, esses devem ser declarados. O NS-2 faz esse procedimento através das linhas abaixo.

```
#Cria um objeto do tipo Simulator.
set ns [new Simulator]
```

Em simulações, existe uma preocupação constante com o registro de atividades executadas. Eventuais erros de processamento ou exceções podem ser analisadas e resolvidas com a utilização desse documento. O procedimento de *logging* é feito da seguinte forma.

```

#Registro de dados da simulacao.
set nf [open out.nam w]
$ns namtrace-all $nf

#Processos padrao para finalizar o
#tratamento do arquivo ao fim da simulacao.
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}

```

Sendo o conjunto de código apresentado anteriormente uma seqüência de operações padrão para a configuração do ambiente de simulação, os métodos que serão descritos daqui para frente que representarão a topologia propriamente dita.

A primeira etapa de declaração é a definição dos nós da topologia. A linguagem OTCL suporta a utilização de estruturas como vetores e matrizes. Devido a facilidade de compreensão desses tipos de variáveis e de sua grande utilidade dentro do escopo de simulações do presente trabalho, se aproveitará essa estrutura para esse *script* simples.

```

#Criacao de nos dentro do ambiente de simulacao.
for {set i 0} {$i < 2} {incr i} {
    set n($i) [$ns node]
}

```

Em seguida, adiciona-se ligações (*links*) entre os dois nós, definindo a capacidade de transferência de dados assim como a política de gerenciamento de filas utilizada. O presente caso utilizará um AQM *droptail* em um *link* com capacidade de 1024 bits *full-duplex* com atraso máximo de 10 milissegundos.

```

#Insere um link entre os dois nós.
for {set i 0} {$i < 2} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i + 1) \% 7]) 1Mb 10ms DropTail
}

```

O próximo passo pode ser definido como a configuração dos parâmetros do protocolo que deverá ser seguido pelo processo de transferência de dados. A escolha dada ao TCP nesse exemplo é natural, visto que todo o trabalho se baseia nas regras desse protocolo para o controle de congestionamento de pacotes.

Para que haja passagem de dados entre dois nós em uma rede TCP, é necessário a criação de pelo menos um Agente e um Dissipador. Pode-se dizer que eles representam a origem dos dados e sua destinação respectivamente. O NS permite a configuração do Agente com diversos parâmetros diferentes, como tamanho da janela de emissão, tamanho de cada pacote entre outros. Após a configuração dessas duas estruturas, deve-se anexá-las aos respectivos nós desejados e ligá-los entre si.

```
#Seta o tamanho da fila de pacotes para o link.
$ns queue-limit $n(0) $n(1) 20
```

```
#Criação de um Agent TCP.
set tcp [new Agent/TCP/Newreno]
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
$ns attach-agent $n(0) $tcp
```

```
#Criação de um Sink TCP.
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n(1) $sink
```

```
#Agora, precisa-se conectar os dois agentes.
$ns connect $tcp $sink
```

O final do *script* é marcado pela configuração da passagem de dados e a definição de parâmetros para a execução da simulação. Defini-se aqui uma conexão do tipo *File Transfert Protocol* internamente da conexão TCP e vincula-se ao Agente TCP do nós criados anteriormente.

```
#Criaremos um FTP dentro da rede TCP.
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$tcp set type_ FTP
```

A simulação em si é definida como uma execução limitada no tempo por um valor previamente definido. Cabe ao operador ativar ou desativar os eventos desejados no tempo correto. Dessa forma, no presente exemplo, inicia-se a transmissão de dados através da conexão FTP no instante $t = 0,5s$ e interrompe-se em $t = 9,5s$, totalizando um tempo total de 9.0 segundos de transferência de dados.

```
#Ordem de disparo e interrupcao da transmissão no tempo.
$ns at 0.5 "$ftp start"
$ns at 9.5 "$ftp stop"
```

```
#Sinaliza ao simulador que o metodo finish deve ser
#executado 10 segundos apos o tempo de simulacao.
$ns at 10.0 "finish"
```

```
#Roda a simulacao.
$ns run
```

Tendo caracterizado toda a topologia desejada, registra-se o arquivo com uma extensão *.tcl* para o reconhecimento do simulador. A execução da simulação é feita através dos seguintes comandos executados no terminal:

```
$ns <file_name>.tcl
```

C.2.2 Simulações de tráfego com AQM implementado

O NS oferece a possibilidade de se executar simulações com uma infinidade de topologias diferentes utilizando inúmeros protocolos e políticas de gerenciamento de filas. Contudo, a flexibilidade da implementação desse programa ainda permite a execução de simulações utilizando implementações especiais de AQM ou protocolo. Nessa seção, um pequeno passo-à-passo é descrito para adicionar novas implementações de gerenciamento de filas no simulador NS-2.

As bibliotecas do programa contêm a maioria dos objetos necessários para a realização das simulações. Somente as implementações dos algoritmos de gerenciamento de fila devem ser modificados para garantir o bom funcionamento do simulador.

C.2.2.1 Arquivos da implementação

As implementações suportadas pelo NS-2 podem ser encontradas no diretório:

```
/home/ns-allinone-2.34/ns-2.34/queue
```

Todos os algoritmos de gerenciamento de filas do programa NS são definidos em dois arquivos: **.cc* e **.h*, onde *** representa o nome da implementação. Esses arquivos são a descrição de algoritmos escritos em C++. Seguindo a arquitetura de desenvolvimento dessa linguagem de programação, o arquivo *.cc* conterá a implementação de todas as funções enquanto que as declarações de variáveis e das funções usadas serão feitas no *.h*.

C.2.2.2 Adicionando os parâmetros *default*

Os algoritmos de gerenciamento de fila usam alguns parâmetros que podem ser modificados sem alterações na estrutura do código em C++ dos arquivos de implementação. Essa modificação é possível através da alteração do arquivo *ns-default.tcl* encontrado em

```
/home/ns-allinone-2.34/ns-2.34/tcl/lib
```

Para cada nova implementação adicionada ao simulador, deve-se incluir o conjunto de parâmetros padrão que serão passados no momento da criação de um novo objeto do tipo. A título de exemplo, abaixo segue uma parte da lista de parâmetros contidos no arquivo *ns-default.tcl* para o AQM RED.

```
...
Queue/RED set bytes_ false
Queue/RED set queue_in_bytes_ false
Queue/RED set thresh_ 0
Queue/RED set maxthresh_ 0
Queue/RED set thresh_queue_ [Queue set limit_]
Queue/RED set maxthresh_queue_ [Queue set limit_]
Queue/RED set mean_pktsize_ 500
Queue/RED set idle_pktsize_ 100
...
```


C.2.2.3 Incluindo as novas referências

Com o objetivo de fazer com que o simulador reconheça o novo gerenciamento de fila implementado, algumas referências devem ser adicionadas a biblioteca principal do simulador. Isso é feito pela inclusão da *string* identificadora no novo método na função *simplex-link* do arquivo *ns-lib.tcl*. É justamente por essa *string* que o nova política de gerenciamento vai ser definida no código descritivo da simulação (arquivo *.tcl*).

O arquivo *ns-lib.tcl* descreve diversas funções utilizadas no núcleo do NS-2 para a descrição das simulações pelo usuário. Ele pode ser encontrado no diretório */home/ns-allinone-2.34/ns-2.34/tcl/lib*. Supondo que o novo algoritmo a ser introduzido se chama *PI-AW*, as seguintes modificações devem ser aportadas ao arquivo

```

Simulator instproc simplex-link { n1 n2 bw delay qtype args } {
...
if {[string first "RED" $qtype] != -1 ||
    [string first "PI" $qtype] != -1 ||
    [string first "Vq" $qtype] != -1 ||
    [string first "REM" $qtype] != -1 ||
    [string first "GK" $qtype] != -1 ||
    [string first "RIO" $qtype] != -1 ||
    [string first "PI-AW" $qtype] != -1 || #referência à nova implementação
    [string first "XCP" $qtype] != -1} {
    $q link [$link_($sid:$did) set link_]
}
...

```

C.2.2.4 Recompilação do NS-2

Para que as modificações sejam efetivadas e possam ser utilizadas nas simulações de topologias é necessário a recompilação de alguns trechos do simulador.

No arquivo *makefile* encontrado em */home/ns-allinone-2.34/ns-2.34/* pode-se sinalizar ao compilador que alterações foram feitas no núcleo do código e que o mesmo deve ser recompilado. Supondo a implementação adicionada têm arquivos nomeados de *pi-aw*, isso pode ser feito através da inclusão do parâmetro *queue/pi-aw.o* na seguinte linha de código

```

OBJ_CC = \
...
queue/red.o queue/pi.o queue/vq.o queue/rem.o queue/pi-aw.o \
...

```

Tendo todas as referências adicionadas aos lugares corretos, deve-se executar o comando de recompilação do simulador. Um terminal deve ser aberto e os seguintes comandos devem ser digitados

```

$cd /home/ns-allinone-2.34/ns-2.34/
$make

```

Após a operação, basta descrever a topologia a ser simulada usando a linguagem OTCL habitual e executar o código pelo processo normal.