Controladores Ótimos para Gerenciamento Ativo de Filas na Arquitetura de Serviços Diferenciados da Internet

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Leonardo Rangel Augusto e aprovada pela Banca Examinadora.

Campinas, 21 de Abril de 2009.

Nelson Luis Saldanha da Fonseca
IC - UNICAMP (Orientador)

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

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Crisllene Queiroz Custódio – CRB8 / 7966

Augusto, Leonardo Rangel
Au45c Controladores ótimos para gerenciamento ativo de filas na arquitetura de serviços diferenciados da Internet / Leonardo Rangel Augusto -- Campinas, [S.P. : s.n.], 2006.

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

1. TCP/IP (Protocolo de rede de computação). 2. Teoria do controle. 3. Redes de computação. I. Fonseca, Nelson Luís Saldanha da. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Optimal Active Queue Management for the Differentiated Services architecture on the Internet

Palavras-chave em inglês (Keywords): 1. TPC/IP (Computer network protocol). 2. Control theory. 3. Computer networks.

Área de concentração: Redes de computadores

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Nelson Luís Saldanha da Fonseca (IC-UNICAMP)
Prof. Dr. Edmundo Roberto Mauro Madeira (IC-Unicamp)
Profa. Dra. Michele Mara de Araújo Espíndula Lima (UFPE)
Prof. Dr. José Cláudio Geromel (FEEC-UNICAMP)

Data da defesa: 26/09/2006

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 26 de setembro de 2006, pela Banca examinadora composta pelos Professores Doutores:

Profa. Dra. Michele Mara Araújo Espindula Lima
UFPE

Prof. Dr. José Cláudio Geromel
FEEC - UNICAMP

Prof. Dr. Edmundo Roberto Mauro Madeira
IC - UNICAMP

Prof. Dr. Nelson Luis Sandanha da Fonseca
IC - UNICAMP

Instituto de Computação
Universidade Estadual de Campinas

Controladores Ótimos para Gerenciamento Ativo de Filas na Arquitetura de Serviços Diferenciados da Internet

Leonardo Rangel Augusto

Setembro de 2006

Banca Examinadora:

- Nelson Luis Saldanha da Fonseca
  IC - UNICAMP (Orientador)
- Edmundo Roberto Mauro Madeira
  IC - UNICAMP
- Michele Mara de Araújo Espíndula Lima
  UFPE
- José Claudio Geromel
  FEEC - UNICAMP

Resumo

A classe de Serviço Assegurado da arquitetura de Serviços Diferenciados (DiffServ) da Internet inclui a provisão de diferenciação de banda passante, o que depende do adequado uso de mecanismos de condicionamento de tráfego e gerenciamento ativo de filas (AQM). Nesta dissertação, propõe-se um controlador ótimo para gerenciamento ativo de filas para a arquitetura Diffserv. Seu projeto considera intrinsecamente a influência de fluxos não adaptativos na dinâmica do controle de congestionamento. Apesar de o controlador obtido ser racional, seu projeto utiliza uma abordagem não-racional, o que aumenta a precisão do modelo. Simulações conduzidas demonstram que o controlador proposto reduz o descarte desnecessário de pacotes, aumentando o goodput e diminuindo a quantidade de RTOs dos emissores TCP.

Abstract

The Assured Service of Differentiated Services Architecture (DiffServ) is currently used for providing throughput differentiation in the Internet. For this, traffic policing and active queue management (AQM) mechanisms must be used. In this dissertation, we use a non-rational approach to develop an optimal AQM controller. Its design considers both adaptative and non-adaptative like UDP. Simulations were conducted for comparison with other proposals. Results show that the proposed controller reduces unnecessary packet drops, increases the goodput and reduces the TCP sender's Retransmission Timeouts.

Agradecimentos

Gostaria de agradecer a meus pais pelo amor, apoio e confiança incondicionais, à minha irmã Alice pelo ombro emprestado nos momentos difíceis e à minha noiva Sânia por me fazer notar que há coisas na vida muito mais importantes do que títulos.

Agradeço também aos meus amigos Augusto Devegili, Daniel Batista, Neumar Malheiros, André Atanásio, André Drummond, Juliana de Santi, Fernanda Michel, Marcelo Dias e Junior Valadares, grandes companheiros de Academia e de vida, que em muito contribuiram para formar o humano que sou hoje. Aos amigos/mestres Carlos Frederico, Elton, Armando e Lucilia, da Universidade Federal de Ouro Preto, de quem sempre tive imenso apoio e confiança.

Ao Nelson, professor, orientador e companheiro nas horas difíceis, minha sincera admiração. Poucos têm percepção tão apurada sobre tudo que envolve o trabalho de orientação. Muito além do direcionamento acadêmico, bons orientadores sabem lidar com pessoas, com suas expectativas, com suas deficiências, sabem o momento de dar um bom "puxão de orelha" mas também o de aplaudir o bom trabalho. Bons orientadores, como gerentes de pessoas, sabem lidar com os momentos difíceis pelos quais cada um passa, criam uma atmosfera positiva entre os seus "filhos" e torcem pelo sucesso deles. Nelson, parabéns pelo sucesso e pela postura exemplar.

Um agradecimento muito especial à Michele, não só pelo excepcional trabalho, cheio de valiosos frutos, mas pela perseverança, paciência, prestatividade e sobretudo pelo fantástico exemplo de vida. Talvez eu não tenha tido outra oportunidade de lhe agradecer, mas saiba que você é uma grande referência que vou carregar para a vida toda. Muito obrigado.

A Vera Ragazzi, Daniel Capeleto, Flávio Luzia e todos os funcionários do IC, que trabalham duro para que os alunos possam estudar tranquilamente e dar o melhor de si. Agradeço também aos professores Jose Geromel e Edmundo Madeira pelos precisos direcionamentos, e à Capes, CNPq e Fapesp pelo fomento à minha pesquisa.

Deus, mais uma vez obrigado por sempre cuidar do meu caminho, pela saúde, pelas felicidades, pelos alertas, e por me apresentar tantas pessoas maravilhosas neste mundo.

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

A Internet consiste de redes com diferentes tecnologias de enlaces interconectadas pelo protocolo IP. O serviço tradicional oferecido pelo IP, o Melhor Esforço, foi, por muito tempo, adequado para as aplicações tradicionais até então existentes na Internet, tais como transferência de arquivos e acesso remoto. Entretanto, com o surgimento de novas aplicações multimídia, esse modelo de serviço tornou-se insuficiente para prover garantias mínimas de banda passante e de níveis máximos de atraso, necessários a essas aplicações.

Dessa necessidade, surgiram propostas para a provisão de Qualidade de Serviço na Internet, como a arquitetura de Serviços Diferenciados (*DiffServ*). *DiffServ* é uma proposta que possibilita o oferecimento de diferentes níveis de serviço de forma escalável. Um de seus *Per Hop Behaviors* (PHBs) padronizados, o Encaminhamento Assegurado *AF - Assured Forwarding*, possibilita o desenvolvimento de estratégias para diferenciação de banda passante aos usuários, de acordo com taxas preestabelecidas.

Este trabalho trata da provisão de garantias de banda passante a usuários da classe AF do modelo *DiffServ* e tem como foco o projeto de um controlador ótimo para gerenciamento ativo de filas em tal classe.

Em [12], é proposta uma arquitetura para oferecimento de tais garantias, baseando-se em estratégias de marcação e diferenciação de pacotes. Para se realizar essas tarefas, são empregados dois controladores, um para regular a marcação (coloração) dos pacotes e outro para prover uma diferenciação de tratamento entre tais pacotes no núcleo da rede. Os controladores usados nesse esquema, ARM e dsPI-AQM, foram originalmente propostos pelos mesmos autores em [11]. Seu desenvolvimento baseou-se na Teoria de Controle Moderno, resultando em um par de controladores *Proportional Integral* capaz de estabilizar o sistema enquanto almejava o alcance de certas métricas de desempenho. O desenvolvimento desses controladores, porém, baseou-se em um modelo simplificado da dinâmica do comportamento do tráfego, o que limita a validade dos mecanismos.

No presente trabalho, propõe-se a utilização da Teoria de Controle Ótimo e um modelo

mais apurado da dinâmica do sistema a fim de desenvolver um novo mecanismo de AQM que supere as dificuldades e limitações de desempenho de dsPI-AQM, atuando na classe de Serviço Assegurado de *DiffServ* em prol do oferecimento de garantias mínimas de banda passante a seus usuários.

O restante da dissertação está dividido da seguinte maneira: no Capítulo 2, serão sucintamente explanados alguns conceitos sobre Qualidade de Serviço em redes IP, estratégias como *DiffServ*, modelos de serviço por ela oferecidos, problemas existentes e atuais soluções para resolvê-los. Em seguida, no Capítulo 3, é abordado o problema da prevenção e controle de congestionamento. Adiante, no Capítulo 4, é apresentada a arquitetura para provisão de banda passante utilizando o serviço AF *DiffServ*. Após uma breve revisão sobre Teoria de Controle, vista no Capítulo 5, é introduzida a modelagem matemática do sistema, no Capítulo 6, e o processo de síntese do controlador, no Capítulo 7. Por fim, serão apresentados os resultados da avaliação do controlador proposto em confronto com as propostas existentes, no Capítulo 8, e as conclusões deste trabalho, no Capítulo 9.

Capítulo 2

Qualidade de Serviço em redes IP

Neste capítulo, é introduzido o problema da provisão de Qualidade de Serviço em redes IP, particularmente na Internet. Será apresentada a conjectura atual, que esboça os rumos de uma nova Internet, com novas e mais diversas aplicações, nas quais há o freqüente uso de recursos multimídia. As particularidades desse conjunto de aplicações motivaram o desenvolvimento de novas estratégias que, usadas em conjunto com a atual estrutura da Internet, possam garantir Qualidade de Serviço às aplicações. Adiante, são abordadas duas das principais propostas para o estabelecimento de uma Internet com Qualidade de Serviço.

2.1 Aplicações multimídia

Nos últimos anos, tem sido observada uma forte tendência para que a transmissão de dados multimídia, para diversos fins, concentre-se sobre um mesmo meio físico, em substituição a seus meios tradicionais. Nesse processo, a Internet mostra-se como uma boa alternativa, o que é evidenciado pelo surgimento de aplicações como a telefonia sobre IP e a transmissão de vídeo sob demanda através da web.

A Internet, que inicialmente fora desenvolvida para fins militares e acadêmicos, agora transforma-se em uma infra-estrutura comercial de alcance mundial, permitindo prover múltiplas funcionalidades e potencialmente agregar inúmeras aplicações, desde o usual correio eletrônico até as recentes aplicações multimídia, como videoconferência e jogos interativos.

Aplicações multimídia, especialmente, possuem requisitos peculiares quanto à forma como seus dados são transportados de um usuário a outro através da rede. Só a satisfação desses requisitos pode garantir que elas sejam executadas com a qualidade desejada. Sua

principal característica é a alta sensibilidade ao atraso fim-a-fim[1] e variação do atraso[2], porém possuem uma certa tolerância a ocasionais perdas (descartes) de pacotes. Esses requisitos em muito diferem dos requisitos de aplicações para as quais a Internet originalmente fora projetada, como correio eletrônico e transferência de arquivos, que são, em geral, sensíveis à perda de pacotes mas tolerantes ao atraso [39].

Apesar de a Internet agregar tipos tão díspares de aplicações, seu serviço original de transmissão de dados dispõe apenas de um único nível de qualidade, conhecido como Melhor Esforço (*BE - Best Effort*). Esse tipo de serviço é igualmente oferecido a todas as aplicações.

O serviço de Melhor Esforço não provê garantia alguma sobre o encaminhamento do tráfego, seja de atraso fim-a-fim ou variação de atraso. Todo ele é processado da melhor forma localmente possível, o que, em geral, se resume a, em cada roteador, encaminhá-lo segundo um determinado critério predefinido. Dessa forma, não é dada às aplicações garantia alguma de que seus requisitos são atendidos fim-a-fim com a qualidade que lhes é necessária.

Como consequência, ao oferecer um único serviço, a Internet atual não possibilita diferenciação alguma no tratamento dado aos diferentes tipos de aplicações. Sendo assim, mesmo possuindo requisitos diferentes, essas aplicações recebem um mesmo nível de qualidade de serviço.

Na Seção 2.2, são apresentadas propostas de extensão da atual estrutura da Internet, desenvolvidas a fim de oferecer diferentes níveis de qualidade de serviço às aplicações.

2.2 Qualidade de Serviço

Diante do cenário exposto na Seção 2.1, mostrou-se necessária a criação de arquiteturas para extensão da Internet, dando-lhe a capacidade de suportar novos e múltiplos níveis de serviço. Dessa forma, através da especificação dos requisitos das aplicações, um nível apropriado de qualidade de serviço pode ser escolhido para satisfazê-los.

Essa escolha é testificada em um contrato firmado entre o cliente e seu provedor de serviços de Internet (*ISP - Internet Service Provider*). Nesse documento, denominando contrato de nível de serviço (*SLA - Service Level Agreement*), o cliente compromete-se a respeitar as restrições impostas pelo provedor de serviços e este, por sua vez, compromete-se a encaminhar o tráfego do cliente garantindo-lhe um determinado nível de qualidade de serviço. Especificamente, um *serviço* define as características da transmissão de dados unidirecional entre dois integrantes de uma rede (transmissão fim-a-fim). Características estas que podem ser especificadas de forma quantitativa ou estatística, em relação à vazão,

[1] tempo decorrido entre a emissão de um pacote e sua chegada ao destinatário

[2] diferença de tempo entre os atrasos sofridos por pacotes de um mesmo conjunto

atraso, variação do atraso ou níveis de descarte de pacotes. Podem também estar ligadas a alguma relação de prioridade para acesso aos recursos de uma rede [7]. Por exemplo, pode-se definir um serviço que ofereça banda passante garantida ou baixos níveis de descarte ao tráfego.

As faixas de valores aceitáveis para as métricas de desempenho de determinado serviço são especificadas em um outro documento, derivado do SLA. Esse documento é denominado especificação de nível de serviço (*SLS - Service Level Specification*).

No contexto da Internet, Qualidade de Serviço (*QoS - Quality of Service*) é um termo genérico que reflete a capacidade de uma rede prover resultados previsíveis sobre o transporte do tráfego. Ou seja, é esperado que redes com suporte a QoS assegurem a seus clientes que, uma vez especificados seus requisitos de qualidade, um transporte justo será dado ao seu tráfego. Dessa forma, tendo seus requisitos satisfeitos, as aplicações podem obter o desempenho esperado.

Em suma, aplicações de rede são construídas utilizando serviços fim-a-fim. Existem serviços com diferentes níveis de qualidade, logo, cabe ao cliente escolher um serviço compatível com as necessidades (requisitos) de sua aplicação. O nível de qualidade de serviço é assegurado ao cliente por meio de um contrato (*SLA*) firmado entre ele e o *ISP*.

A especificação de requisitos, contudo, só define as metas de desempenho que devem ser alcançadas para o êxito da aplicação mas não como isso deve ser feito, em termos práticos. Para prover à aplicação do cliente a qualidade de serviço de que ela necessita, podem ser utilizadas diferentes estratégias. Na Seção 2.3, são abordadas duas das principais propostas para se prover Qualidade de Serviço na Internet, *DiffServ* e *IntServ*.

2.3 Arquiteturas para provisão de QoS

Nesta seção, são apresentadas duas das soluções sugeridas pelo IETF (*Internet Engineering Task Force*) [34] para provisão de Qualidade de Serviço na Internet, as arquiteturas *IntServ* e *DiffServ*. Nesta última, foca-se o presente trabalho.

2.3.1 Arquitetura de Serviços Integrados (*IntServ*)

A arquitetura de Serviços Integrados (*IntServ - Integrated Services*) foi a primeira arquitetura proposta para garantir Qualidade de Serviço às aplicações na Internet [9]. A idéia de *IntServ* é prover classes de serviço adicionais ao serviço de Melhor Esforço das redes IP. Duas classes foram propostas:

Serviço Garantido (*Guaranteed Service*) - Criada para atender aplicações com requisitos explícitos de transmissão, tais como garantia de banda passante, limites para

atraso fim-a-fim e ausência de perdas nas filas. Esta classe de serviço é adequada para aplicações de tempo real ou aplicações que sejam sensíveis a perdas e atrasos [56, 57].

Serviço de Carga Controlada (*Controlled Load Service*) - Apesar de não prover garantias estritas de serviço como as da classe de Serviço Garantido [57, 59], esta classe oferece às aplicações um serviço mais confiável e eficiente que o Melhor Esforço. Seu desempenho é adequado para aplicações que suportam pequenos atrasos e perdas, tais como aplicações adaptativas de tempo real.

O nível de qualidade que cada uma dessas classes provê é programado de acordo com os requisitos das aplicações. Tais requisitos devem ser explicitamente repassados aos roteadores que compõem o caminho pelo qual o tráfego passará. Essa reserva prévia de recursos é essencial para a arquitetura *IntServ*. Por meio dela, consegue-se garantir que os requisitos de qualidade das aplicações sejam atendidos fim-a-fim.

Na arquitetura *IntServ*, a solicitação dos recursos necessários para determinado fluxo é feita sob demanda, pouco antes de sua transmissão se iniciar. Isso é feito através de protocolos específicos, como o RSVP (*Resource Reservation Protocol* [57]). O RSVP tenta fazer a reserva de recursos para as aplicações ao longo do caminho entre a origem e o destino. Somente no caso de haver recursos suficientes ao longo de todo o caminho, o fluxo que os solicitou poderá utilizá-los [57].

Nessa arquitetura, os roteadores devem fazer o controle do tráfego que recebem, com o objetivo de garantir às aplicações que os requisitos de cada fluxo são devidamente proporcionados. Esse controle de tráfego é composto por três mecanismos, obrigatoriamente implementados em todos os roteadores:

Mecanismo de controle de admissão - Avalia a possibilidade de atender a requisição de reserva de recursos feita por novos fluxos. Caso isso não seja possível, é oferecido ao novo fluxo o serviço Melhor Esforço.

Classificador - Mapeia os pacotes recebidos pelo roteador para uma das classes de serviço e os coloca em uma fila específica para ela. Conseqüentemente, pacotes pertencentes a uma mesma classe recebem o mesmo tratamento por parte da rede.

Escalonador - Implementa as prioridades entre as filas e coordena a transmissão dos pacotes.

A reserva prévia mostra-se, de fato, uma estratégia eficaz para se prover os recursos necessários para uma aplicação. Todavia, a abordagem adotada por *IntServ* tem algumas desvantagens:

- Caso haja falha em algum dos roteadores por onde o tráfego passa ou modificações na rota do fluxo, a garantia de recursos pode ficar comprometida.
- Os mecanismos de controle de admissão, classificador, escalonador e o suporte a RSVP, fazem com que uma carga adicional seja atribuída aos roteadores;
- É imprescindível que todos os roteadores no caminho entre a origem e o destino, possuam todos os mecanismos de controle de tráfego supracitados.
- A alocação de recursos e a quantidade de informações de estado a ser processada nos roteadores cresce com o número de fluxos ativos;

Estas três últimas características fazem com que a arquitetura *IntServ* seja considerada não *escalável*. Escalabilidade é um conceito que indica a capacidade que um sistema computacional tem de crescer sem que a arquitetura desse sistema seja um limitante ao crescimento.

Na Seção 2.3.2, é apresentada uma proposta escalável para provisão de Qualidade de Serviço em redes IP, a arquitetura *DiffServ*.

2.3.2 Arquitetura de Serviços Diferenciados (*DiffServ*)

A arquitetura de Serviços Diferenciados (*DiffServ*) foi proposta como uma alternativa para provisão de QoS baseada no mesmo paradigma original da Internet, segundo o qual "ações complexas devem ser tomadas nas extremidades da rede, de modo a preservar a simplicidade no núcleo" [15].

DiffServ foi concebida com o objetivo de ser escalável e de garantir QoS através da diferenciação de serviços. Essa escalabilidade é conseguida através da agregação de diversos fluxos em um número fixo de classes de serviço, às quais são providos diferentes níveis de qualidade de transporte. Sendo assim, o desempenho do sistema não se degrada com o aumento de fluxos ativos na rede [6, 7].

Domínios *DiffServ*

Um domínio *DiffServ* é formado por um conjunto de nós (roteadores) contíguos, com suporte a serviços diferenciados e que tenham todos um mesmo conjunto de Comportamentos por Nó (PHBs - *Per-Hop-Behaviors*) implementados. Os PHBs descreve o comportamento que cada roteador do domínio fornece na expedição do tráfego agregado das diferentes classes.

Cada domínio deve garantir os recursos necessários para dar suporte aos SLAs por ele firmados. Caso a provisão de um serviço fim-a-fim dependa da passagem dos fluxos por

outros domínios, é necessário que os ISPs que os controlam firmem SLAs entre si. Com o firmamento desses contratos, busca-se garantir que, em todos os domínios, os PHBs aplicados àquela porção de tráfego sejam suficientes para satisfazer seus requisitos de qualidade.

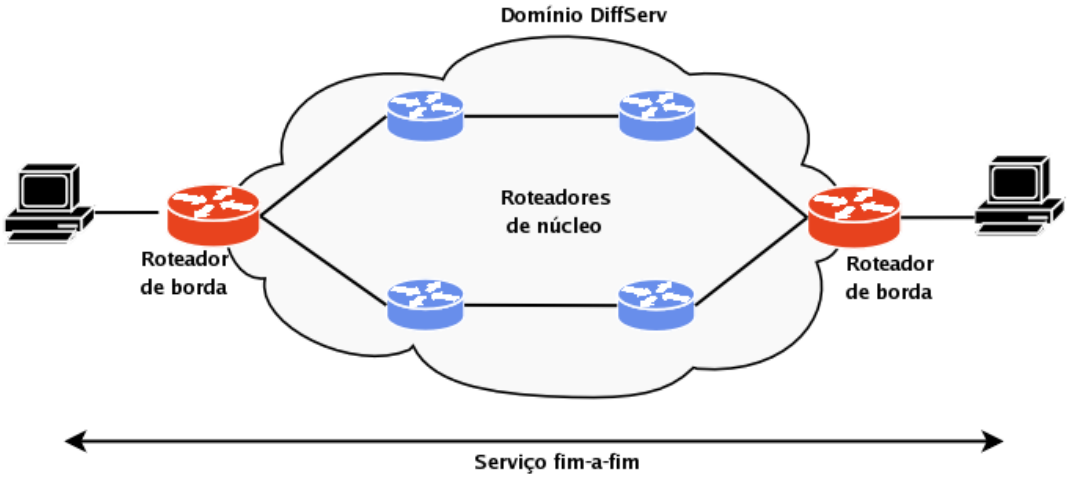

Figura 2.1: Domínio *DiffServ*

Como se observa na Figura 2.1, existem dois tipos básicos de roteadores em um domínio *DiffServ*, definidos de acordo com sua localização:

Roteadores de borda ou fronteira - Ficam na fronteira de um domínio e têm a função de fazer a comunicação com outros roteadores que não pertencem ao domínio. Dependendo do sentido do tráfego, eles podem ser denominados de roteadores de entrada ou de saída.

Roteadores internos ou de núcleo - Todos os roteadores presentes em um domínio que não são roteadores de borda.

Tratamento do tráfego nos roteadores de borda

A arquitetura *DiffServ* é composta por uma série de elementos funcionais implementados nos roteadores do domínio, cuja atividade de maior complexidade computacional é o condicionamento do tráfego ingressante. Esse procedimento é dividido em algumas etapas, detalhadas a seguir.

A primeira ação aplicada ao tráfego que almeja ingressar no domínio *DiffServ*, e uma das mais complexas, é a classificação. Nessa etapa, um *classificador* seleciona, de acordo com regras preestabelecidas, os pacotes que ingressarão no domínio através da análise

das informações contidas em seus cabeçalhos. O classificador é, em termos práticos, constituído de uma seqüencia de filtros que selecionam os pacotes que se encaixam em determinados padrões.

Após a classificação, *medidores* estimam a taxa de entrada do tráfego de cada classe. Posteriormente, o valor obtido nessa medição será usado para verificar se a taxa de tráfego enviado por cada classe está dentro dos limites especificados em contrato.

Na etapa seguinte, *marcadores* são usados para marcar (rotular) os pacotes com o DSCP (*DiffServ Code Point*), um identificador da classe de serviço à qual ele pertence. É essa marcação que possibilitará a diferenciação do tratamento dado ao tráfego no núcleo do domínio. Contudo, é importante frisar quer a marca carregada por cada pacote não determina, em si, qual tratamento ou qualidade de serviço ele receberá no núcleo do domínio. A marcação é apenas um dos mecanismos para distingui-los em classes. A marcação dos pacotes é feita em um campo específico de seus cabeçalhos IP. No IPv4 [50], é utilizado o campo *ToS* (*Type of Service*). Já no IPv6 [18], o campo utilizado é o *DS* (*Differentiated Service Field*).

Após a etapa de marcação, o tráfego passa pela etapa de policiamento. O *policiador* tem, literalmente, a função de fiscalizar o tráfego ingressante e a ele aplicar a ação cabível. Normalmente, essa ação se restringe a três opções: i) Caso o volume de tráfego esteja totalmente dentro do limite estabelecido, todos os pacotes são marcados como "dentro da especificação", ou simplesmente *in*; ii) Caso contrário, todo o **excedente** é marcado como "fora da especificação", ou simplesmente *out*; iii) Todo o tráfego não-conformante com alguma especificação do contrato pode ser descartado ou, alternativamente, associado ao Serviço Melhor Esforço.

O policiamento é feito através de mecanismos denominados *Token Buckets*. Além da fiscalização, os *Token Buckets* podem ainda efetuar a "suavização" do tráfego. Esse processo tem por objetivo regular as taxas de emissão de pacotes ao interior do domínio, absorvendo as rajadas de tráfego e repassando ao núcleo um tráfego em taxa constante. A etapa de condicionamento tem a função adicional de impedir que fluxos não-conformantes (fora das especificações de contrato) prejudiquem a qualidade de serviço dos demais fluxos. Esse tipo de proteção é um importante princípio na provisão de QoS e é comumente chamado de *isolamento* entre classes.

Na Figura 2.2, é ilustrado graficamente o relacionamento entre as etapas de classificação, marcação, medição e policiamento do tráfego em um roteador de borda *DiffServ*. O fluxo representado pela linha contínua indica as etapas nas quais são aplicadas ações ao tráfego. O fluxo da linha pontilhada é de controle, usado apenas para sugerir que a etapa de medição, além de ocorrer em paralelo com a marcação, ocorre de forma passiva, não aplicando ações ao tráfego.

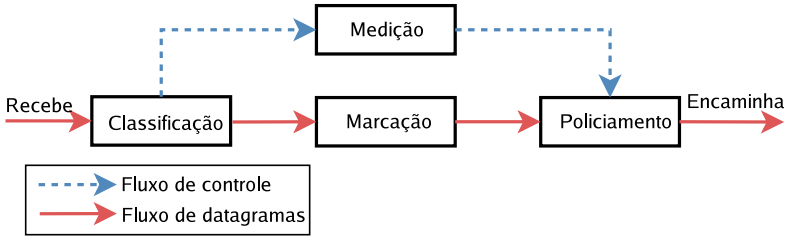

Figura 2.2: Etapas do condicionamento do tráfego nos roteadores de borda

Tratamento do tráfego nos roteadores do núcleo

O tratamento que o tráfego de uma determinada classe recebe no núcleo do domínio depende de um conjunto de regras predefinidas. Elas especificam forma de classificação, policiamento e encaminhamento dos pacotes de cada classe. Esse conjunto de regras é denominado Comportamento por Nó (*PHB Per-Hop Behavior*).

A definição das classes de serviços na arquitetura *DiffServ* consiste na definição dos PHBs implementados pelos roteadores para oferecer tais serviços. Existem apenas dois PHBs padronizados atualmente: Encaminhamento Expresso (*EF-PHP - Expedited Forwarding-PHB*) [17] e Encaminhamento Assegurado (*AF-PHB - Assured Forwarding-PHB*) [31].

Como visto anteriormente nesta seção, os pacotes ingressantes no domínio são marcados com um determinado DSCP. Cada um desses DSCPs corresponde diretamente a um PHB na arquitetura *DiffServ*. Na Tabela 2.1 é explicitada essa correspondência.

DSCP	Classe de serviço
111	(reservado)
110	(reservado)
101	EF
100	AF-4
011	AF-3
010	AF-2
001	AF-1
000	Melhor Esforço

Tabela 2.1: Relação entre DSCPs e PHBs na Arquitetura *DiffServ*

Como pode-se ver na Tabela 2.1, o serviço Melhor Esforço pode ainda ser oferecido dentro da arquitetura *DiffServ*, mas agora como *uma* das classes de serviço, a de mais

baixa prioridade. Em geral, apenas são associados a esse serviço os pacotes de fluxos não-conformantes com as especificações de contrato. Nesse caso, o pacote recebe como marca o DSCP 000 na etapa de condicionamento do tráfego.

Usando os PHBs AF e EF da arquitetura *DiffServ*, três novas classes de serviço, além do Melhor Esforço, podem ser oferecidas:

Serviço Premium - O EF-PHB define uma classe de serviço conhecida como Serviço Premium. Essa classe é adequada para aplicações como vídeo-conferência e voz sobre IP, que possuem requisitos explícitos de atraso, variação de atraso e que geram tráfego a uma taxa constante predefinida [17]. O usuário desse serviço deve especificar no SLS a taxa de envio desejada e se responsabilizar a não exceder esse limite, sob pena de ter o excedente descartado. Por sua vez, o ISP deve garantir essa banda passante quando o tráfego for enviado dentro dos limites especificados. Alguns mecanismos são primordiais para garantir a reserva de banda passante: filas de prioridade, mecanismos de policiamento e algoritmos de escalonamento, tais como *WFQ - Weight Fair Queuing* ou *CBQ - Class Based Queuing*.

Serviço Assegurado - Esse serviço não oferece garantias de banda passante ao tráfego, mas assegura que ele é transmitido com confiabilidade e alta prioridade, mesmo na presença de congestionamento. Esse serviço é oferecido através de *uma* das classes de serviço definidas pelo AF-PHB, a AF4. Uma forma de implementá-lo é utilizar a marcação feita pelos *Token Buckets* nas bordas do domínio. Dessa forma, os pacotes *out* podem ser descartados preferencialmente na presença de congestionamento, preservando, assim, os pacotes *in*. Isso é feito por um mecanismo de gerenciamento ativo de filas, tal como RIO.

Serviço Olímpico - Esse é um serviço que oferece três níveis de qualidade: ouro, prata e bronze [31]. Uma das formas de implementá-lo é atribuir diferentes fatias de banda para cada um desses níveis. O excedente de banda é redistribuído proporcionalmente entre as classes. Por exemplo, caso não haja algum pacote dos níveis ouro e prata, os pacotes do nível bronze podem utilizar toda a banda passante disponível. Numa outra forma de se implementar esse serviço, a diferenciação pode ser feita oferecendo níveis de prioridade de descarte baixo, médio e alto ao invés de banda passante. O Serviço Olímpico pode ser implementado através do mapeamento das classes bronze, prata e ouro para as classes AF1, AF2 e AF3, definidas no AF-PHB.

Desde a proposição do Serviço Assegurado em [16], diversos trabalhos vêm sendo desenvolvidos com o objetivo de avaliar e melhorar seu desempenho, como [30, 54]. O mecanismo de AQM RIO, em particular, apresenta dificuldades para se chegar a uma adequada configuração de seus parâmetros, o que, caso não seja conseguido, acarreta em

baixo desempenho. Mais adiante, no Capítulo 3, são abordadas algumas novas soluções para mecanismos de AQM que visam suprir as deficiências de RIO.

A provisão de Qualidade de Serviço, de forma geral, depende sobretudo de um correto dimensionamento da demanda suportada em relação à capacidade da rede. Mesmo estando bem dimensionada, uma rede (ou domínio, como é o caso específico a ser tratado) pode enfrentar períodos de pico de utilização, onde há congestionamentos temporários. Nesses períodos, a manutenção da Qualidade de Serviço depende da adoção de mecanismos de contenção ou, preferencialmente, prevenção de congestionamento. No próximo Capítulo, o problema do congestionamento nas redes IP e, em particular, na Arquitetura *DiffServ*, é abordado. Serão apresentadas ainda algumas estratégias e soluções para superar esse problema.

Como descrito, a classe de Serviço Assegurado não provê, originalmente, garantia alguma em termos de banda passante ao seu tráfego. No Capitulo 4, é apresentada uma recente proposta de extensão dos mecanismos atualmente utilizados na arquitetura *DiffServ* a fim de permitir a diferenciação em termos de banda passante entre o tráfego da classe de Serviço Assegurado.

Capítulo 3

Controle de congestionamento

Neste capítulo, serão abordadas a motivação e alternativas para o controle de congestionamento na Internet. Primeiramente, serão observadas a origem do problema de congestionamento e a importância do seu controle. Em seguida, é apresentado o mecanismo de controle de congestionamento do protocolo TCP e os mecanismos de gerenciamento ativo de filas (AQM). Será visto, ainda, que esses mecanismos, quando usados em conjunto, complementam-se e formam um sistema eficiente para o controle de congestionamento.

3.1 Visão geral acerca do controle do congestionamento

Aplicações multimídia são, hoje em dia, muito populares na Internet. Contudo, seu desempenho só é satisfatório quando o tráfego por elas gerado está sujeito a baixos níveis de atraso e variação do atraso e, sobretudo, não está sujeito a congestionamento. Para oferecer a essas aplicações a qualidade de transmissão necessária, uma solução intuitiva é a utilização de redes com largura de banda abundante. No entanto, redes dessa natureza também estão sujeitas a congestionamentos, mesmo que puntuais. Assim, a qualidade de transmissão das aplicações multimídia pode se deteriorar até níveis inaceitáveis assim que o fluxo de pacotes da aplicação atinja um enlace moderadamente congestionado [39]. Dessa forma, mostra-se necessário não só o controle do congestionamento já formado, mas também a adoção de estratégias que evitem que ele se forme.

O congestionamento é formado quando a demanda por determinado recurso excede a sua disponibilidade, seja qual for o ambiente considerado, incluindo redes de computadores. Isso pode ocorrer devido ao mal dimensionamento dos recursos oferecidos ou à sua má utilização, que pode se dar, por exemplo, quando se sobrecarrega um recurso em detrimento dos demais. A resposta de uma rede ao congestionamento pode variar depen-

dendo da sua intensidade e duração. Por exemplo, congestionamentos de curta duração podem ser provocados pela irregularidade nas taxas de envio dos dados (transmissão em rajadas), característica de protocolos como o TCP. Nesse caso, a simples adoção de filas para absorção das rajadas de pacotes e posterior encaminhamento já ameniza o problema.

Porém, quando o congestionamento perdura por longos intervalos de tempo ou quando se apresenta de forma muito intensa, o volume de pacotes pode aumentar demasiadamente, de forma que o armazenamento de todos eles em filas torna-se impraticável, provocando o descarte de alguns.

A ocorrência de descarte, que no serviço Melhor Esforço deteriora ainda mais seus baixos níveis de qualidade, pode acabar prejudicando a provisão da qualidade desejada em arquiteturas com suporte a QoS, como *IntServ* e *DiffServ*. Por isso, nesses ambientes, é preciso mais do que a escolha de uma política de escalonamento/descarte apropriada para alcançar os objetivos de qualidade desejados. É preciso *controlar* o congestionamento de forma satisfatória quando ele ocorrer e, sobretudo, tentar evitar que ele se forme. O descarte de pacotes, quando inevitável, deve ainda ser feito de forma a não comprometer o objetivo de obtenção de QoS das aplicações.

Uma das formas de controlar o congestionamento na rede é regulando a taxa de envio dos emissores de dados. Assim, em períodos de congestionamento, tais emissores, ao reduzir suas taxas de transmissão, fariam com que o estado de congestionamento regredisse. Através da observação do andamento da transmissão, os emissores de dados podem estimar o estado de utilização do núcleo da rede e, havendo indícios de congestionamento eles próprios podem regular suas taxas de transmissão. Essa é a abordagem do mecanismo de controle de congestionamento do protocolo de transporte TCP.

O protocolo de transporte TCP (*Transmission Control Protocol*) [51], em sua versão mais difundida, a TCP Reno [3], possui incluso um mecanismo de controle de congestionamento. Uma abordagem mais detalhada sobre o funcionamento do protocolo TCP Reno, especialmente do seu mecanismo de controle de congestionamento, é vista a seguir, na Seção 3.2,

Como é visto adiante, os mecanismos de controle de congestionamento situados nas bordas da rede não são suficientes para impedir que o congestionamento ocorra em todas as situações. Isso se deve ao fato de que, atuando apenas nos sistemas finais da rede (clientes), eles não tem uma visão precisa sobre o real estado de utilização do núcleo da rede, onde possivelmente o congestionamento ocorre. O núcleo da rede é tratado como uma "caixa preta" e a ocorrência de congestionamento é presumida a partir de indícios como a perda de pacotes, mesmo que esta seja devida a problemas físicos no enlace. Sendo assim, pode ocorrer a detecção de falsos congestionamentos, o que, considerando-se o método de redução da janela de transmissão do emissor, prejudicaria sua transmissão de dados.

Os mecanismos de controle de congestionamento no núcleo da rede têm, por outro lado, condições de monitorar o estado de utilização dos enlaces e filas, avaliando de forma mais precisa a dimensão do congestionamento já formado ou incipiente. Eles podem ser usados de forma complementar aos mecanismos de borda, fornecendo-lhes informações mais precisas sobre o estado de utilização do núcleo da rede. O principal desses mecanismos, o gerenciamento de filas, é abordado logo em seguida, na Seção 3.3.

3.2 O Protocolo TCP Reno

O TCP é o protocolo de transporte dominante atualmente na Internet, sendo responsável por cerca de 75% dos pacotes, 83% dos bytes e 56% dos fluxos transmitidos [28]. Sendo assim, a compreensão e análise do seu funcionamento, principalmente do seu mecanismo de controle de congestionamento, mostra-se fundamental para o estudo do controle de congestionamento na Internet, como um todo.

No funcionamento padrão do TCP, para cada pacote que um emissor envia, é esperado que o destinatário lhe retorne uma confirmação de recebimento. Essa confirmação de reconhecimento (*ACK - Acknowledgement*) pode ainda ser cumulativa, ou seja, um mesmo ACK pode confirmar o recebimento de vários pacotes. Isso é possível porque todos os pacotes enviados por um emissor TCP carregam consigo um número de seqüencia. Sendo assim, uma mensagem ACK pode ser usada não só para acusar o recebimento de um único pacote, mas também o de todos os demais que antes foram enviados daquela seqüencia.

O TCP usa também um mecanismo de janela deslizante, denominado "janela de congestionamento" ou "janela de transmissão", que limita a quantidade de pacotes sem reconhecimento (ACK) que o emissor pode manter na rede. *Grosso modo*, essa janela de congestionamento impõe um limitante superior para a diferença entre número de pacotes enviados e o de pacotes confirmadamente recebidos. Esse mecanismo permite ao TCP adequar a taxa de transmissão de um emissor de acordo com sua estimativa de estado da carga da rede, inclusive diminuindo sua taxa de transmissão em períodos de congestionamento da rede. Protocolos com essa característica são comumente chamados de adaptativos ou bem comportados.

A implementação atual mais popular do TCP, denominada TCP Reno [28], possui um mecanismo de controle de congestionamento composto de quatro algoritmos: Partida Lenta (*Slow Start*), Prevenção de Congestionamento (*Congestion Avoidance*), Recuperação Rápida (*Fast Recovery*) e Retransmissão Rápida (*Fast Retransmit*) [3].

Os dois primeiros algoritmos são utilizados pelo emissor TCP para controlar a quantidade de dados enviada à rede. No início de uma transmissão, o emissor TCP tem que lidar com condições desconhecidas da rede. Nesse estágio, o algoritmo Partida Lenta é utilizado para estimar gradativamente a capacidade de transmissão pela rede suportada. Isso é feito

com o objetivo de evitar que, logo de início, a rede seja sobrecarregada com um volume de tráfego superior a sua capacidade, o que certamente causaria o congestionamento.

Durante a fase de Partida Lenta, o emissor TCP incrementa sua janela de transmissão em uma unidade para cada ACK recebido. Sendo assim, a janela de transmissão dobrará de tempos em tempos. O crescimento exponencial do tamanho da janela de transmissão dos emissores TCP certamente faria com que a rede logo enfrentasse períodos de congestionamento. Para evitar que isso aconteça, quando a janela de congestionamento atinge o limiar *ssthresh* (*slow start threshold*), o algoritmo Partida Lenta é finalizado e o algoritmo de Prevenção de Congestionamento é iniciado. Nessa fase, o crescimento do tamanho da janela de transmissão passa a ser linear e não mais exponencial. É importante notar que, mesmo durante a fase de Partida Lenta, o congestionamento pode ocorrer. Uma situação possível seria, por exemplo, a de vários emissores iniciando suas transmissões (ainda em fase de Partida Lenta) em um curto intervalo de tempo, sobrecarregando assim um determinado roteador por onde seus fluxos passam.

Para um emissor TCP, toda perda de pacotes é considerada indício de congestionamento já formado no caminho pelo qual os pacotes passam. Essa perda pode ser notada por um emissor TCP de duas formas: quando o ACK de um pacote não lhe é retornada dentro de um período de tempo estipulado (expiração do temporizador); ou quando esse ACK é recebido duplicadamente para determinado pacote, ou seja, quando o receptor envia repetidas mensagens solicitando o envio de determinado pacote. São justamente esses indícios de congestionamento que o TCP usa para melhor dimensionar sua janela de transmissão.

Os outros dois algoritmos do TCP Reno, o Retransmissão Rápida e o Recuperação Rápida, são usados pelo emissor TCP para se recuperar de perdas de pacotes ocorridas sem expiração do temporizador (RTO - *Retransmission TimeOut*).

Quando um receptor TCP recebe um segmento fora de ordem, ele deve enviar imediatamente um ACK informando ao emissor TCP a ocorrência deste fato e indicando qual o segmento que ele espera receber (o segmento que está faltando). Para o emissor TCP este é um ACK duplicado e esta informação poderá ser usada futuramente pelo seu algoritmo de Retransmissão Rápida. No TCP Reno, quando são recebidos três ACKs duplicados, o segmento faltante apontado por esses ACKs é reenviado imediatamente. Pelo funcionamento do TCP padrão, esse segmento faltante seria reenviado somente após a expiração do seu temporizador. A função do algoritmo Retransmissão Rápida é justamente a de evitar que isso aconteça e, por precaução, já reenviar o pacote faltante antes mesmo que seu temporizador expire.

A ocorrência de RTOs em um fluxo pode acarretar uma grande variação no atraso sofrido pelos seus pacotes e comprometer, assim, a obtenção da qualidade de serviço desejada. Sendo assim, a utilização do algoritmo de Retransmissão Rápida justifica-se pelo

fato de evitar que se aguarde o transcorrer de todo esse intervalo para que então se reenvie, então, o pacote supostamente faltante. A utilização do algoritmo de Retransmissão Rápida não elimina, porém, a ocorrência de *timeouts*, mas a minora em situações de congestionamento moderado [37].

Depois de o TCP emissor retransmitir o pacote aparentemente perdido, o algoritmo Recuperação Rápida passa a governar a transmissão de novos pacotes até que um ACK não duplicado seja recebido. O algoritmo de Recuperação Rápida faz com que o algoritmo Prevenção de Congestionamento seja executado ao invés do Partida Lenta, que seria o curso natural do TCP padrão logo após a perda de um pacote. Essa alteração justifica-se porque, como os ACKs continuam a chegar seguidamente, assim como eles próprios, outros pacotes continuam a trafegar normalmente pela rede. Sendo assim, o evento que causou a perda do pacote em questão pode ter sido um congestionamento brando, não tão severo a ponto de se exigir a redução abrupta na janela de transmissão do emissor. Dessa forma, o emissor continua a transmitir os pacotes sem ter que baixar drasticamente sua taxa de transmissão.

3.3 Gerenciamento de filas

Como já introduzido nas seções anteriores, os mecanismos de controle de congestionamento situados nas bordas da rede, incluindo o do protocolo TCP, não são capazes de evitar o congestionamento de forma eficiente pois falta-lhes o conhecimento sobre o estado do núcleo da rede, onde o congestionamento de fato ocorre.

Os mecanismos de prevenção e controle de congestionamento situados nos roteadores do núcleo da rede, tendo conhecimento sobre o estado de utilização do núcleo, têm condições de detectar o congestionamento incipiente e de tomar decisões sobre a sua notificação. Dessa forma, podem agir de forma complementar ao mecanismo de borda, provendo-lhes as informações necessárias sobre o estado da rede. De posse dessas informações, os emissores podem adequar suas taxas de transmissão à realidade em que se encontra a rede.

Os mecanismos de gerenciamento de filas realizam o controle de congestionamento no núcleo da rede através da monitoração do comportamento das filas nos roteadores ao longo do tempo.

Mecanismos mais simplistas, como o *DropTail*, padrão na Internet, apenas limitam o tamanho das filas e descartam todo o tráfego que exceder esse limite. A simplicidade do *DropTail* não inibe, porém, o aparecimento de alguns problemas no gerenciamento das filas, como o monopólio por parte de certas conexões e a manutenção de altos níveis de ocupação, que acabam por provocar descartes desnecessários e injustos [41].

Problemas como esses ocorrem porque o mecanismo *DropTail*, por natureza, não

previne a formação do congestionamento, tomando atitudes para combatê-lo (descarte de pacotes) apenas quando ele já está de fato estabelecido, quando as filas encontram-se cheias. A manutenção de altos níveis de ocupação das filas traz ainda outras consequências, como a incapacidade de absorver novas rajadas de tráfego.

Na literatura recente, têm sido propostos mecanismos mais eficientes para suprir essas deficiências. Os mecanismos de gerenciamento ativo de filas (*AQM - Active Queue Management*) permitem que os roteadores tenham maior controle sobre a gerência das filas.

O gerenciamento ativo de filas é abordado com mais detalhes na próxima seção, onde serão apresentados os algoritmos de AQM recomendados para uso na Internet e algumas novas propostas de melhorias.

3.4 Gerenciamento Ativo de Filas

Na seção anterior, foram descritos os mecanismos de gerenciamento de filas, como *DropTail* e levantadas algumas de suas deficiências, as principais são relacionados às filas cheias.

Uma solução intuitiva para esse problema seria o aumento da capacidade das filas de forma que todas as rajadas pudessem ser absorvidas. Contudo, isso elevaria por demais o atraso de transmissão do pacote, o que não é desejável para aplicações de tempo real. Por conta disso, o tamanho médio das filas deve ser fundamentalmente bem escolhido, balanceando a capacidade de absorver rajadas de tráfego e o nível de atraso proporcionado aos pacotes devido ao enfileiramento.

O controle sobre o tamanho da fila é feito através do descarte (ou marcação [52]) dos pacotes nela enfileirados. A fim de evitar o problema do monopólio nas filas, uma abordagem muito conveniente é o descarte aleatório de pacotes.

Os mecanismos de gerenciamento ativo de filas têm por objetivo suprir as deficiências de *DropTail*. Através de uma apropriada política de descarte de pacotes, eles são capazes de manter o nível de utilização das filas próximo a algum limiar convenientemente estabelecido, permitindo a absorção de rajadas de tráfego, limitando o atraso inserido pelo enfileiramento e, sobretudo, prevenindo a formação do congestionamento.

Mecanismos de AQM atuam de forma mais sistemática juntamente aos emissores em prol do controle de congestionamento. Eles monitoram a variação do tamanho da fila ao longo do tempo, o que os torna capazes de detectar um congestionamento incipiente. Sendo assim, podem notificar antecipadamente os emissores de dados, fazendo com que estes reduzam suas taxas de transmissão antes que o congestionamento efetivamente se forme.

Além de manter a ocupação da fila em níveis convenientes, pode-se ainda tentar minimizar a oscilação em seu tamanho, contribuindo para a redução da variação do atraso

sofrido pelos pacotes (*jitter*), fator de suma importância para a provisão de QoS.

Nas próximas subseções, serão abordadas os principais mecanismos de gerenciamento ativo de filas (AQM) para a Internet, tanto para o cenário de Melhor Esforço quanto para a arquitetura *DiffServ*. A abordagem dos mecanismos de AQM para o cenário de Melhor Esforço é importante devido ao fato de que grande parte dos mecanismos de AQM para a arquitetura *DiffServ* é inspirada ou baseada em mecanismos equivalentes para o Melhor Esforço.

Inicialmente, serão explanados os mecanismos de AQM padrão para o Melhor Esforço, RED, e sua variação desenvolvida para a arquitetura *DiffServ*, RIO. Desde suas proposições, vários novos mecanismos de AQM têm sido propostos a fim de superar as deficiências de RED e RIO. Serão explanados aqui, em especial, os mecanismos desenvolvidos a partir da Teoria de Controle, os quais modelam intrinsecamente a interação entre emissores de dados e AQM como um sistema integrado de controle de congestionamento.

3.4.1 Mecanismos de AQM

RED - *Random Early Detection* [26] é o algoritmo de AQM recomendado pelo IETF para a Internet. Além de prevenir a ocorrência do congestionamento, RED tem por objetivo a manutenção de baixos nível de utilização nas filas. Adicionalmente, RED previne de forma eficiente a penalização do tráfego em rajadas e a sincronização global [26]. Seu funcionamento é dividido em duas etapas: a estimativa do tamanho médio da fila e o descarte (ou marcação) dos pacotes.

O cálculo do tamanho médio das filas é feito utilizando-se uma fórmula de média ponderada móvel (*weighted moving average*). Essa fórmula utiliza um filtro passa baixa para excluir pequenas variações ao cálculo da média. O valor obtido é comparado com dois limiares (*thresholds*), min_{th} e max_{th}, previamente definidos. Nesse momento, inicia-se a segunda fase do algoritmo, ilustrada na Figura 3.1. Se o valor obtido estiver abaixo de min_{th}, o algoritmo está na zona normal de operação e nenhum pacote é descartado/marcado; Caso o valor esteja entre os dois limiares, cada pacote que chega é descartado/marcado com uma probabilidade p_a, que cresce linearmente com o tamanho médio da fila estimado; Se o valor estiver acima de max_{th}, todos os pacotes que chegam são descartados.

RIO (*RED with IN/OUT*) [16] é uma variação de RED que foi desenvolvida para ser utilizada na classe de Serviço Garantido, da arquitetura *DiffServ*. Seu objetivo é diferenciar e penalizar fluxos que estão fora das especificações de contrato. O processo de marcação usado para discriminar essas duas classes de fluxos é descrita em detalhes na Subseção 2.3.2.

RIO utiliza dois algoritmos RED, um para os pacotes que estão em conformidade com as especificações de contrato (*in*) e outro para os pacotes que não estão em conformidade

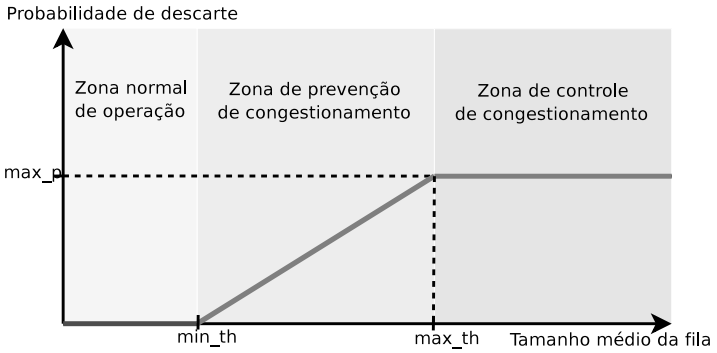

Figura 3.1: Fases do algoritmo RED

(*out*), característica que lhe rendeu o nome. Cada algoritmo, porém, possui o seu próprio conjunto independente de limiares.

RIO diferencia o tratamento dado aos pacotes *in* e *out* através da definição de alguns parâmetros nos dois algoritmos. Primeiramente, considera-se que a fila dos pacotes *in* é apenas um subconjunto da fila geral de pacotes e não uma fila à parte. Assim, quando um pacote *in* chega ao roteador, é calculado o valor do tamanho médio da fila *in*, *InQavg*, e o tamanho médio da fila geral, *TotQavg*. Caso um pacote *out* chegue ao roteador, apenas *TotQavg* é recalculado. O cálculo da probabilidade de descarte dos pacotes *in* depende apenas de *InQavg*. Já o cálculo da probabilidade de descarte dos pacotes *out* depende de *TotQavg*.

O objetivo de RIO é descartar prioritariamente pacotes *out* na presença de congestionamento. A Figura 3.2 ilustra como isso pode ser feito através da definição dos parâmetros de ambos os algoritmos. Ao escolher *Out_min_th* menor que *in_min_th*, faz-se com que a fase de prevenção de congestionamento e os descartes de pacotes *out* iniciem-se antes que a dos fluxos *in*, à medida que o tamanho médio da fila cresce; Ao escolher *out_p_max* maior que *in_p_max*, na fase de prevenção de congestionamento, garante-se que os pacotes *out* serão descartados com alta probabilidade; Ao escolher *Out_max_th* menor que *In_max_th*, faz-se com que a fase de controle de congestionamento dos pacotes *out* inicie-se antes que a dos pacotes *in*.

Em suma, RIO descarta os pacotes *out* com maior probabilidade quando detecta um congestionamento incipiente, descarta todos os pacotes *out* quando o congestionamento persiste, e em último caso, quando chega ao roteador um número excessivo de pacotes *in*, ele os descarta. Neste último caso, há um indicativo de que a rede foi mal projetada para atender os requisitos dos fluxos *in* definidos em contrato, já que a meta é evitar que os estes sejam punidos com o descarte.

RIO herdou todas as vantagens de RED bem como suas desvantagens [41], mas é bas-

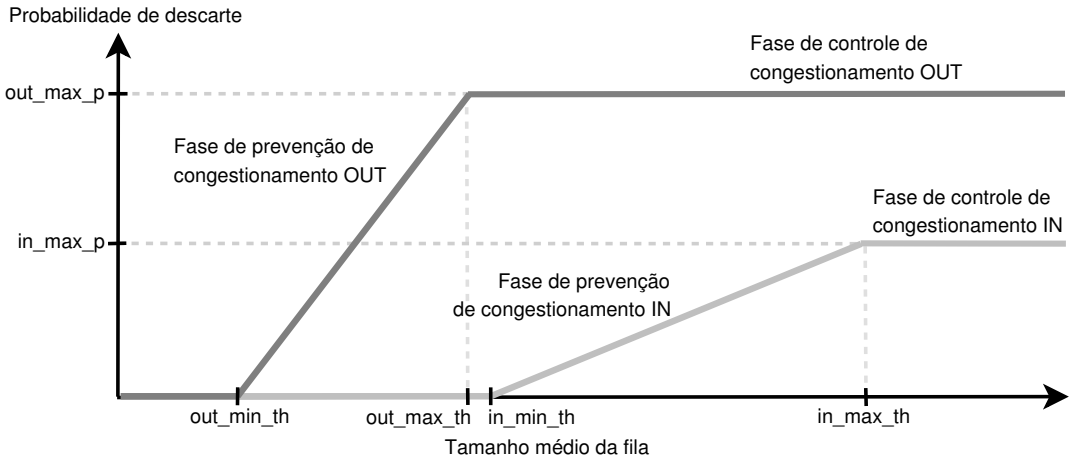

Figura 3.2: Fases do algoritmo RIO

tante simples e é mais justo que RED no sentido de penalizar os fluxos mal comportados.

A principal dificuldade de se utilizar RED e, conseqüentemente, RIO é determinar corretamente os valores de seus limiares para que se possa atingir um ponto de operação ideal. Caso tais parâmetros não sejam corretamente ajustados, RED oferece um baixo desempenho.

Diversos estudos têm sido propostos para determinar os valores dos *thresholds* de RED baseados em heurísticas e simulações [19,20,22–24]. Todas essas configurações, entretanto, não garantem que um ponto de equilíbrio seja atingido nem a estabilidade do tamanho da fila. Há também algumas propostas de uma forma mais sistemática para determinação desses valores [21, 40].

Na Seção 3.4.2, serão abordados alguns mecanismos de AQM baseadas em Teoria de Controle.

3.4.2 Mecanismos de AQM baseados em Teoria de Controle

Nos mecanismos de AQM desenvolvidos utilizando a Teoria de Controle, o sistema integrado de controle de congestionamento (dos emissores e AQM) é modelado como um sistema retroalimentado. Nesse sistema, observável na Figura 3.3, a taxa de transmissão dos emissores (r) é ajustada dinamicamente, de acordo com o nível de congestionamento da rede. Este é determinado pelo nível de ocupação das filas do núcleo (q) que, por sua vez, determina a probabilidade de descarte dos pacotes (p).

Nesse cenário, o papel dos controladores AQM é o de determinar uma probabilidade de descarte adequada que estabilize o tamanho da fila e o sistema como um todo, independentemente das suas variações de estado.

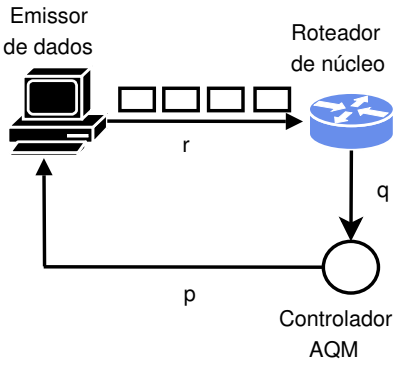

Figura 3.3: Sistema de controle de congestionamento retroalimentado

O controlador PI-AQM (*Proportional Integral*) [33], utiliza um controlador homônimo e baseia-se no modelo simplificado da dinâmica do comportamento do TCP introduzido em [46]. Como a planta apresentada nessa proposta não representa toda a dinâmica do sistema, faz-se necessário identificar qual a sensibilidade desse sistema, bem como estabelecer as condições da rede para as quais o controlador consegue estabilizá-lo.

O controlador H2-AQM [41], utiliza o mesmo modelo da dinâmica comportamental do TCP que foi usado para derivar o controlador PI-AQM. No entanto, a planta utilizada no desenvolvimento do controlador H2-AQM representa a dinâmica do sistema com maior detalhamento, garantindo a sua estabilidade independente das condições de rede. Diferentemente de PI-AQM, H2-AQM usa técnicas de Controle Ótimo ao invés de Controle Clássico.

A síntese do controlador H2-AQM usou uma abordagem não racional, incorporando os termos com atraso. Ou seja, sua modelagem considera, além dos parâmetros instantâneos da rede, um histórico de seus últimos valores. No ambiente de redes de computadores, isso se traduz em uma melhor utilização dos recursos da rede, em se comparando com abordagens racionais [36]. Em [41], há demonstrações de que H2-AQM supera tanto RED quanto PI-AQM na obtenção dos objetivos de desempenho.

Tanto PI-AQM quanto H2-AQM, foram modelados apenas ao modelo de Melhor Esforço da Internet, o que não permite sua utilização direta em arquiteturas como *DiffServ*, já que lhes falta a capacidade de diferenciação de tratamento entre classes de tráfego.

Em [11], a modelagem proposta em [33] foi estendida com o objetivo de se derivar um controlador PI que comportasse a diferenciação de pacotes em duas classes, verde e vermelha, aqui referenciado como dsPI.

dsPI-AQM é constituído de um par de controladores PI que atua de forma complementar, gerando as probabilidades de descarte p_g e p_r a partir do tamanho da fila q medido. Cada um desses controladores possui um conjunto independente de parâmetros, dentre os

quais destaca-se o valor-referência para o tamanho da fila desejado, q_r para o controlador dos fluxos vermelhos e q_g para o dos verdes, com $q_r < q_g$. Dessa forma, ambos os controladores computam em paralelo, as probabilidades de descarte para os fluxos baseando-se cada um em seu próprio conjunto de parâmetros.

Na Seção 6.2, a modelagem utilizada na síntese de dsPI é vista em maior detalhamento.

Capítulo 4

Diferenciação de Banda Passante no Serviço Assegurado de *DiffServ*

Neste capítulo, é abordado o tema da diferenciação de banda passante para o tráfego da classe AF-4 (Serviço Assegurado) da arquitetura *DiffServ*. Será introduzida a arquitetura para tal fim, proposta em [11] e, mais recentemente, estendida em [12]. Essa arquitetura, na qual baseia-se o presente trabalho, é descrita nas seções seguintes.

4.1 Visão geral

Como visto na Seção 2.3.2, a classe de Serviço Assegurado não provê, originalmente, garantia alguma em termos de banda passante ao tráfego a ela associado.

Uma solução intuitiva para este problema seria limitar, nas bordas da rede, o volume de tráfego que adentra o núcleo a partir de cada borda, descartando todo o tráfego que excedesse um determinado limite. No entanto, trabalhos recentes concluíram que a vazão obtida pelo tráfego de determinado agregado não é afetada somente pela política de marcação feita nas bordas da rede, mas também pela presença de outras fontes de tráfego, retardos na propagação etc [30, 54, 61]. Isso acontece porque o tráfego na Internet é, predominantemente, gerado por emissores TCP e os mecanismos de controle e prevenção de congestionamento do TCP possuem um comportamento um tanto complexo [11].

Em [11], é apresentada uma proposta de extensão dos mecanismos empregados em *DiffServ* que possibilita a diferenciação de largura de banda oferecida ao tráfego da classe AF-4. Tal trabalho teve caráter inovador por utilizar uma abordagem baseada em Teoria de Controle na modelagem desse problema.

Apesar dos avanços obtidos em tal trabalho, o uso da arquitetura proposta não possibilita o oferecimento de *garantias* reais de banda passante para o tráfego durante toda sua travessia através do núcleo do domínio *DiffServ*. O trabalho busca aproximar tal objetivo

ao regular a taxa média de envio dos agregados de tráfego próxima a níveis preestabelecidos. Uma vez reguladas as taxas de envio e não havendo situações de congestionamento intenso no núcleo, os agregados de tráfego poderiam experimentar uma banda passante proporcional a suas taxas de envio.

Essas taxas de envio refletem, mais precisamente, a taxa com que cada roteador de borda envia tráfego (por ele recebido dos emissores e em seguida agregado em classes) ao roteador congestionado do núcleo. A regulagem dessas taxas de envio é feita dinamicamente por um conjunto de controladores, implementados tanto nos próprios roteadores de borda quanto nos de núcleo. Ela utiliza valores-referência previamente configurados, que especificam as taxas mínimas de transmissão que cada agregado deve experimentar.

Para atingir tal objetivo, são utilizados mecanismos de gerenciamento ativo de filas, nos roteadores de núcleo, e mecanismos para o processo de condicionamento do tráfego (CT), que gerenciam a coloração (marcação) feita pelos *Token Buckets*, nas bordas do domínio.

Os mecanismos de CT têm o papel de regular a coloração dos pacotes de cada agregado de acordo com duas métricas específicas de cada um: a taxa de chegada do tráfego (gerada pelos emissores de dados ligados àquele roteador de borda); e um determinado valor-alvo (de referência) para essa mesma métrica. Já os mecanismos de AQM têm como objetivo principal prevenir e controlar de forma justa o nível de congestionamento da rede.

Tais mecanismos podem atuar de forma complementar: enquanto os mecanismos de AQM buscam otimizar a utilização de recursos da rede, de forma global, os mecanismos de CT objetivam, de modo mais específico, "balancear" as taxas de envio de cada agregado, a fim de garantir que elas, individualmente, atinjam ou superem os valores das suas taxas-alvo.

A seguir, na próxima seção, são abordados os avanços introduzidos nessa arquitetura desde a sua proposição.

4.2 Evolução da arquitetura de diferenciação de banda

Na proposta original da arquitetura de provisão de banda [11], são sugeridos dois mecanismos para implementá-la, denominados ARM e dsPI-AQM.

Nessa proposta, a eficiência da arquitetura foi avaliada apenas para tais mecanismos, em específico. Ademais, apenas experimentos práticos foram conduzidos e nenhuma prova formal da convergência ou eficiência da solução proposta foi apresentada.

Ainda assim, várias propostas de melhorias para os mecanismos de CT [13, 14, 38, 60] e AQM [1] foram derivadas a partir de tal trabalho.

Já em [12], o mesmo grupo de autores estendeu a proposta do trabalho anterior. Nessa extensão, além de provada matematicamente a eficácia da arquitetura proposta, a

validade dos resultados foi estendida a todo e qualquer mecanismo de CT ou AQM que respeite algumas condições preestabelecidas. Tais restrições são detalhadas mais adiante, na Seção 4.4.

Nas próximas seções, a arquitetura para provisão de diferenciação de banda é apresentada de forma mais detalhada.

4.3 A arquitetura para diferenciação de banda

Nesta seção, é apresentada uma visão geral da arquitetura para diferenciação de banda proposta em [12] e os mecanismos usados para regular as taxas de transmissão do tráfego.

Na Seção 3.4.2, a interação entre o mecanismo de AQM e os emissores de dados foi apresentada como um sistema de controle de congestionamento. Em seguida, foi sugerida a modelagem desse sistema como um problema de Controle com retroalimentação.

A fim de melhor ilustrar a interação entre tais componentes, o sistema de retroalimentação (Figura 3.3) pode ser extendido para incorporar o papel do mecanismo de CT, como ilustrado na Figura 4.1.

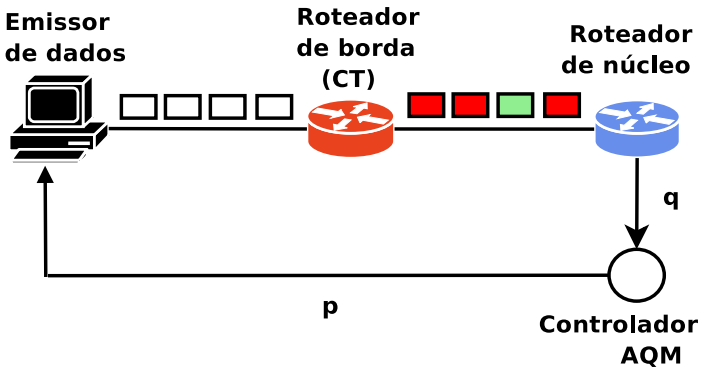

Figura 4.1: Interação entre CT, AQM e emissores de dados

Os mecanismos de CT são desenvolvidos para operar sobre os *Token Buckets*, gerenciando a coloração dos pacotes nos roteadores de borda da rede. Esse processo mantém a característica original dos *Token Buckets*, que é marcar os pacotes ingressantes *in* (nesse trabalho, referenciados como "verdes") quando eles estão em conformidade com uma taxa de envio preestabelecida ou *out* (chamados de "vermelhos"), caso contrário. No núcleo da rede, os roteadores dão preferência ao encaminhamento de pacotes *in*. Isso implica que, em períodos de congestionamento, os pacotes *out* são descartados com maior freqüência.

Cada CT é configurado com uma determinada taxa-alvo, ou valor-referência, $\tilde{r}$. A função do CT nesse esquema é o de garantir, em cooperação com o mecanismo de AQM,

que a taxa de envio r do tráfego agregado naquele roteador de borda alcance, no mínimo, $\tilde{r}$ preestabelecido, independente das variações nas condições da rede, como atraso, probabilidade de descarte ou carga.

Havendo a discriminação entre os pacotes *in* e *out* (verde ou vermelho), as taxas de entrada dessas classes de pacotes, em relação ao todo, podem ser analisadas em separado e representadas pelas frações f_i e $(1 - f_o)$.

Cada porção de tráfego, *in* ou *out*, é submetida a uma probabilidade de descarte diferente no núcleo da rede. Normalmente, objetiva-se manter uma probabilidade de descarte dos pacotes *in* sempre menor que a dos pacotes *out*. Dessa forma, consegue-se privilegiar os pacotes conformantes com as especificações de contrato em situações de congestionamento, preservando-os mesmo quando o descarte de pacotes mostra-se inevitável.

Tais probabilidades de descarte são representadas por p_i (*in*) e p_o (*out*), respectivamente. Sendo assim, a probabilidade de descarte geral, p, experimentada pelo tráfego como um todo, é definida através de uma média ponderada que envolve ambas as probabilidades de descarte e as frações de tráfego a elas associadas:

$$p = (f_i)p_i + (1 - f_i)p_o \tag{4.1}$$

As taxas de envio dos agregados e as garantias a elas oferecidas, podem ainda ser usadas para classificar as redes como: i) superprovida, caso $\sum_{j=1}^{m} \tilde{r}_j < C$; ii) subprovida, caso $\sum_{j=1}^{m} \tilde{r}_j > C$; e exatamente provida, caso $\sum_{j=1}^{m} \tilde{r}_j = C$. Essas definições consideram uma rede com m roteadores de borda (cada um equipado com um CT) e apenas um enlace-gargalo, de capacidade C.

Alguma confusão pode surgir a partir dessas definições. A característica de superprovisão ou subprovisão de uma rede não está diretamente relacionada à taxa média de dados enviada ao enlace gargalo, mas sim às *garantias* oferecidas pelo ISP aos agregados de tráfego. Uma rede subprovida é aquela onde são "garantidos" pelo ISP mais recursos do que ele pode efetivamente disponibilizar. Uma rede é subprovida, independente de estar congestionada ou não.

4.4 Requisitos para os CTs e AQMs

O conjunto de ações descrito na seção anterior, baseado em [12], permite o oferecimento de diferenciação de vazão entre o tráfego da classe de Serviço Assegurado de *DiffServ*. Os resultados demonstrados em tal trabalho dependem, porém, da utilização de mecanismos de CT e AQM conformantes com um determinado conjunto de restrições.

Nesta seção, são sucintamente apresentadas as restrições que envolvem os mecanismos de CT e AQM. Uma análise mais extensa e detalhada, incluindo as demonstrações matemáticas, pode ser encontrada no trabalho original [12].

A restrição do mecanismo de CT é a de que ele seja *fully color*, ou seja, que nas condições de equilíbrio valha[1]:

$$\begin{cases} \hat{r} < \tilde{r} \Rightarrow f_i = 1 \\ \hat{r} > \tilde{r} \Rightarrow f_i = 0 \end{cases} \tag{4.2}$$

Em termos práticos, um CT *fully-color* colore como *in* os fluxos que persistentemente não alcançam suas taxas-alvo e, de forma oposta, marca como *out* os fluxos que persistentemente as ultrapassam.

Essa política de coloração do CT está intimamente relacionada ao comportamento dos emissores de dados, em particular os de natureza adaptativa, como o TCP. Como visto na Seção 3.2, o mecanismo de controle de congestionamento do TCP reduz a taxa de transmissão do emissor ao receber uma notificação de congestionamento (por exemplo, o descarte de um pacote).

Dessa forma, considerando que a probabilidade de descarte dos fluxos *in* p_i é sempre menor do que a dos *out* p_o, e que esse fato se refletirá, estatisticamente, em taxas de descarte equivalentes, infere-se que: i) marcar determinado fluxo como *out* significa atribuir a ele uma probabilidade de descarte maior, o que provocará a redução das suas taxas de envio. ii) marcar um fluxo como *in* significa atribuir a ele uma probabilidade de descarte nula (igual a zero), o que contribuirá para que os mecanismos de controle de congestionamento do seu emissor naturalmente aumentem suas taxas de envio gradativamente até que se atinja o nível esperado.

Já para os mecanismos de AQM, a restrição é a de que eles sejam *non-overlapping*, ou seja, que nas condições de equilíbrio, valha o seguinte:

$$\begin{cases} \hat{p}_o < 1 \Rightarrow \hat{p}_i = 0 \\ \hat{p}_i > 0 \Rightarrow \hat{p}_o = 1 \end{cases} \tag{4.3}$$

De forma prática, essas condições refletem a necessidade de se preservar ao máximo os pacotes *in* em detrimento dos *out*. Dessa forma, pacotes *in* apenas são descartados somente no caso extremo onde o congestionamento persiste mesmo em se descartando todos os pacotes *out* ($p_o = 1$). Graficamente, pode-se dizer, então, que as curvas de probabilidade de descarte dos fluxos *in* e *out* nunca se sobrepõe, o que rende ao mecanismo de AQM a denominação de *non-overlapping*.

[1] A notação $\hat{x}$ é usada para indicar o valor da variável x nas condições de equilíbrio

O caso no qual pacotes *in* são descartados é considerado extremo porque ele indicaria que, mesmo em se descartando todos os pacotes *out* na fila do roteador, o número de pacotes *in* injetado no núcleo do domínio ainda estaria ultrapassando a capacidade de transmissão do enlace. Esse seria um forte indício de mal dimensionamento da rede, onde se oferece garantias de transmissão maiores que os recursos disponíveis.

A seguir, nas próximas duas seções, são sucintamente abordados alguns dos mecanismos de CT e AQM conformantes com essa especificação, ou seja, mecanismos que, uma vez utilizados no esquema proposto, garantem que os objetivos em termos de banda passante dos agregados são atingidos.

4.5 Mecanismo de CT *fully-color*

Como abordado na Seção 4.2, a partir do trabalho desenvolvido em [11] derivaram-se vários outros, com o objetivo de melhorar o desempenho da arquitetura proposta.

Contudo, diante das recentes restrições impostas no trabalho de extensão de tal arquitetura([12]), ainda não se tem conhecimento de outros trabalhos relacionados que apresentem mecanismos de CT, equivalentes ao ARM, conformantes com tais especificações.

Uma análise sobre a capacidade de adaptação dos mecanismos anteriormente propostos às novas especificações da arquitetura é requerida e é um possível objeto de estudo posterior, mas foge, todavia, ao escopo do presente trabalho. Sendo assim, o único mecanismo de CT abordado neste trabalho é o ARM *fully-color* apresentado em [12].

4.6 Mecanismos de AQM *non-overlapping*

Nesta seção, são sucintamente apresentados alguns mecanismos de AQM *non-overlapping* que podem ser usados na arquitetura proposta, segundo a investigação conduzida em [12]. Inicialmente, é explanado como o mecanismo RIO pode ser regulado de forma a obedecer a essa propriedade. Em seguida, é visto como PI-AQM provê tal funcionalidade.

4.6.1 RIO

Em [32], o mesmo grupo de autores do PI-AQM promoveu uma investigação analítica do mecanismo de AQM RED, no qual se baseia RIO. De tal trabalho, concluiu-se que RED pode ser modelado como um controlador do tipo P (*Proportional*). RIO, sendo constituído de dois algoritmos RED, pode ser considerado como um par de controladores P, que estabelecem uma relação proporcional entre o comprimento da fila medido e a probabilidade de descarte por eles gerada.

Esse par de controladores de RIO pode ser facilmente configurado de forma a respeitar a condição de *non-overlapping*. Para isso, basta ajustar seus parâmetros assim como fora anteriormente apresentado na Figura 3.2, com $out\_max\_p = 1$ e $out\_max\_th < in\_min\_th$. Nessa figura, nota-se que não há sobreposição das curvas de probabilidade na fase de prevenção de congestionamento de ambos os algoritmos ($0 < p_i, p_o < 1$). Pode-se também notar claramente a observância de tal configuração de RIO com as duas restrições da especificação de *non-overlapping*.

A condição de *non-overlap* pode ser considerada como uma flexibilização das premissas de RIO, já que ela considera apenas valores da condição de equilíbrio do sistema. Por outro lado, RIO garante a observância a essa regra para todo estado do sistema, independente das condições em que a rede se encontra.

No Capítulo 5, algumas definições de Teoria de Controle, inclusive a de condição de equilíbrio, são apresentadas de forma mais clara. Para mais detalhes sobre o funcionamento de RIO e a relação entre seus parâmetros, consulte a Seção 3.4.1.

4.6.2 dsPI-AQM

dsPI-AQM é um mecanismo de AQM proposto em [11], juntamente com a arquitetura de diferenciação de banda, projetado para regular o nível de ocupação das filas próximo a limites preestabelecidos.

O controle I (*Integral*) de PI faz com que seus integradores naturalmente impliquem a característica *non overlapping* aos controladores [12]. Isso deve-se à forma como eles foram projetados, estabelecendo limiares para o valor do tamanho da fila q, e determinado a atuação exclusiva de apenas um dos controladores para cada faixa de valores. Esses valores são q_o, para o controlador *out*, e q_i, para o *in*, sendo que $q_o < q_i$.

Baseado nesses limiares, o integrador do controlador *out* garante que a probabilidade $0 < p_o < 1$ quando $q < q_o$ (e um saturador, aplicado à saída do controlador), garante que $p_o = 1$ quando $q \geq q_o$. Já o integrador do controlador *in* (e um saturador aplicado a sua saída) garante que a probabilidade $p_i = 0$ enquanto $q < q_o$, $0 < p_i < 1$ quando $q_o \geq q \geq q_i$ e $p_i = 1$ quando $q \geq q_i$.

Sendo assim, dsPI-AQM é, por natureza, um mecanismo AQM *non overlapping*.

Capítulo 5

Introdução aos Sistemas de Controle

Neste capítulo, são cobertos de forma sucinta alguns conceitos básicos sobre a Teoria de Controle e a modelagem matemática envolvida no desenvolvimentos de controladores.

5.1 Introdução

O Controle Automático tem desempenhado papel fundamental no avanço da tecnologia e da ciência, produzindo meios para otimizar o desempenho de sistemas dinâmicos. Sua utilização abrange desde os tradicionais sistemas robóticos a modernos processos industriais e de produção.

Recentes trabalhos introduzem o uso do Controle Automático para o desenvolvimento de mecanismos de controle de congestionamento em redes de computadores [5, 11–13, 32, 41].

Neste capítulo, são introduzidos brevemente alguns conceitos de controle de sistemas e explanados alguns detalhes sobre o processo de modelagem de tais sistemas.

Na seção seguinte, são enumerados alguns conceitos sobre controle que são constantemente referenciados ao longo deste trabalho.

5.2 Conceitos a respeito de controle

A seguir, são apresentadas algumas definições necessárias para a descrição dos Sistemas de Controle. Cada uma delas é exemplificada de forma a melhor contextualizá-las no ambiente de controle de congestionamento de redes de computadores, objeto de estudo deste trabalho. Algumas definições [29, 47]:

Sinal - Expressão quantitativa de determinado fenômeno (Ex: métricas ou valores passados como entrada ou saída de um Sistema).

Sistema a controlar ou Planta - Conjunto de objetos "físicos" que funciona de maneira integrada e que é controlado (Ex: uma rede com emissores de dados, roteadores etc).

Sistema de Controle - Sistema constituído de componentes abstratos dinâmicos cujas variáveis interagem de forma predefinida em prol de determinado objetivo (Ex: sistema integrado de controle de congestionamento TCP/AQM).

Modelo - Conjunto de relações matemáticas entre as variáveis do Sistema de Controle (Ex: equações que descrevem o comportamento dos emissores TCP e das filas nos roteadores, apresentadas no Capítulo 6).

Variável controlada - Grandeza medida e controlada pelo controlador (Ex: tamanho da fila do roteador congestionado).

Variável manipulada - Grandeza modificada pelo controlador de modo a afetar o estado da variável controlada (Ex: probabilidade de descarte).

Valor de referência - Valor desejado da variável controlada (Ex: o tamanho da fila esperado).

Erro ou desvio - Diferença entre o valor instantâneo da variável controlada e seu valor referência.

Controlar - Ação que consiste em medir o valor da variável controlada do sistema e utilizar a variável manipulada para corrigir eventuais desvios daquela com relação ao seu valor de referência (Ex: manipular a probabilidade de descarte com o objetivo de manter o tamanho da fila próximo a um valor preestabelecido).

Distúrbio - Sinal interno ou externo ao sistema que tende a afetar de modo adverso o seu funcionamento.

Controle em malha fechada ou realimentado - Na presença de distúrbios, utiliza o desvio da variável controlada em relação a seu valor referência como informação de forma a corrigi-lo ou atenuá-lo.

Equações diferenciais - Equações onde os valores das variáveis relacionam-se uns com os outros através das suas derivadas, ou seja, uma equação que contém derivadas de uma função.

Equações diferença - Equações que definem uma seqüência de termos recursivamente, onde cada termo é definido em função dos termos anteriores.

Função de transferência - Representação matemática de uma relação entre uma entrada e uma saída de um sistema linear invariante no tempo.

Processo - Operação que evolui de forma natural e progressiva, caracterizada por uma série de alterações graduais sucessivas direcionadas a um determinado resultado.

5.3 Modelagem de Sistemas de Controle

O projeto de Sistemas de Controle baseia-se na modelagem matemática do sistema em questão. Os modelos matemáticos que os representam podem ser determinados através da experimentação prática, avaliando a relação entre entradas e saídas do sistema, ou a partir do conhecimento da sua estrutura interna. Neste último caso, especificamente, são determinadas as equações diferenciais que descrevem o sistema. A partir dessas equações, pode-se obter as funções de transferência ou as equações de estado que o representam.

A definição de um modelo matemático que represente o sistema é uma tarefa um tanto quanto ardilosa. Isso deve-se ao fato de que, quanto maior a precisão do modelo construído, em termos de variáveis e características do sistema consideradas, maior é a complexidade do modelo obtido. Sendo assim, deve-se estabelecer, na modelagem, uma conciliação entre a simplicidade do modelo criado e a precisão dos resultados por ele fornecidos.

Sendo necessária ou desejável a simplificação de um modelo, algumas de suas propriedades, convenientemente escolhidas, devem ser ignoradas, principalmente caso o efeito por elas provocado na resposta do sistema seja pequeno. Dessa forma, ignorando certos aspectos do sistema em sua modelagem, busca-se obter modelos mais simplificados que forneçam boas aproximações quanto à precisão da resposta. Um modelo mais simplificado pode, além de oferecer um bom desempenho em termos de tempo de resposta, facilitar, ainda, o processo de modelagem e implementação do controlador obtido.

Um sistema é dito linear caso todas as operações realizadas sobre as variáveis de entrada sejam lineares, como adições/subtrações, multiplicações/divisões, derivadas/integrais em relação ao tempo etc. Já os sistemas invariantes no tempo são aqueles cujas operações mantém-se com o decorrer do tempo.

Uma importante propriedade à qual todos os sistemas lineares invariantes no tempo (SLIT) obedecem é denominada princípio da superposição. Esse princípio dita que um sinal de saída, formado pela combinação linear de dois diferentes sinais de entrada, é igual à mesma combinação aplicada aos sinais de saída gerados por cada sinal de entrada individualmente.

Dado o alto grau de complexidade envolvido na análise e projeto de sistemas não lineares, os métodos de análise e projeto de sistemas de Controle mais usados restringem-

se a SLITs. Sendo assim, mesmo que o sistema modelado seja não linear, pode-se a ele aplicar um processo de linearização, que resultará em um sistema linear equivalentemente aproximado.

No processo de linearização, mais alguns aspectos do sistema podem ser eliminados, reduzindo ainda mais a precisão do modelo obtido em relação ao real. Por esse motivo, a perda de precisão implicada pela linearização deve ser muito bem avaliada para que o sistema obtido, mesmo sendo linear, represente uma boa aproximação do sistema a ser controlado.

Em linhas gerais, o processo de linearização é uma simplificação do sistema linear que considera um sistema estável em torno de um ponto de equilíbrio. O *ponto de equilíbrio* é uma condição na qual o sistema se estabiliza e nela permanece indefinidamente. Já o conceito de *estabilidade* pode ser um pouco estendido, incorporando a capacidade que um sistema tem de, diante de perturbações, voltar ao ponto de equilíbrio ou dele não se afastar.

Os sistemas lineares obtidos através do processo de linearização são utilizados apenas para o projeto do controlador, cuja função é manter o sistema no ponto de equilíbrio [41]. Em geral, essa estratégia é utilizada principalmente quando se faz estudos sobre como limitadas perturbações em torno de um ponto de equilíbrio variam de acordo com o tempo.

Em [42], introduz-se uma técnica de linearização aplicável à maioria dos sistemas não-lineares. Sobre a estabilidade do sistema não linear, a mesma referência discorre sobre o critério de estabilidade de Lyapunov, que estabelece uma relação direta entre a estabilidade de um sistema não-linear e seu equivalente linear. Basicamente, ao se provar a estabilidade de um sistema linearizado em torno de um ponto de equilíbrio, consegue-se inferir que o sistema não-linear que o originou é estável na região que inclui tal ponto de equilíbrio.

Detalhando mais o processo de modelagem dos sistemas, a seguir, são apresentadas duas abordagens de modelagem, uma baseada em Funções de Transferência e outra em Espaço de Estados.

5.4 Função de Transferência

Na Teoria de Controle, a relação entre a entrada e a saída de determinados componentes (ou sistemas completos) é, geralmente, caracterizada através de funções de transferência (FTs). Esse conceito, porém, só é aplicável a sistemas que podem ser descritos por equações diferenciais lineares invariantes no tempo.

A utilização de FTs permite a representação da dinâmica de um sistema (ou de uma parte dele) por meio de uma equação algébrica. Essas equações, em geral, apresentam-se

na seguinte forma:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \ldots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n}$$

onde o numerador representa a função de saída do sistema, $Y(s)$, e o denominador, $X(s)$, representa a função de entrada. Em tal representação, a *ordem* do sistema é determinada pelo expoente do termo de maior grau do denominador da FT, no caso, n.

Apesar de representarem toda a dinâmica do sistema, os coeficientes que figuram numa FT não fornecem nenhuma informação relativa à estrutura ou composição do sistema que ela descreve. Por se tratar de uma equação, é possível, inclusive, que sistemas completamente diferentes possuam FTs semelhantes ou até mesmo idênticas. Ou seja, a FT é capaz de fornecer uma descrição completa das características dinâmicas de um sistema, independente da sua descrição "física". Por conta dessa capacidade de abstração, seu uso é muito comum na análise e no projeto de SLITs [47].

5.5 Modelagem de Sistemas de Controle no Espaço de Estados

A Teoria de Controle convencional é fundamentada na relação entrada-saída, como introduzido na seção anterior. Já a Teoria de Controle Moderno fundamenta-se na descrição de um sistema de n equações diferenciais de primeira ordem, nas quais o expoente do termo de maior grau é igual a um. Vale lembrar, como visto na seção anterior, a descrição do sistema por meio de FTs considera equações de grau n.

Essa abordagem mais moderna, apesar de aumentar o número de variáveis de estado consideradas, simplifica em muito o processo de análise e modelagem de SLITs, principalmente aqueles que relacionam muitas entradas e saídas [29].

Apesar dessas diferenças, ambos os tipos de representações podem ser convertidos mutuamente. Para esse fim, há muitas técnicas matemáticas [47] e softwares especializados, como [44], que os implementam, o que, de fato, agiliza muito o processo de análise e projeto de Sistemas de Controle.

Diferentemente a abordagem de FTs, a de Espaço de Estados permite caracterizar o comportamento *interno* do sistema. Em termos gerais, o *estado* de um sistema pode ser definido como um conjunto de n valores $x_1(t), x_2(t), \ldots, x_n(t)$, cujo conhecimento num determinado instante $t = t_0$, aliado ao conhecimento do sistema para um dado $t > t_0$, é suficiente para determinar o comportamento do sistema para todo $t \geq t_0$, inclusive para o futuro.

Nesse cenário, $x_1(t), x_2(t), \ldots, x_n(t)$ são as *variáveis de estado*. As equações de primeira ordem que resultam da representação de um determinado sistema dinâmico por variáveis de estado são denominadas *equações de estado*. *Variáveis de saída* são aquelas que podem ser observadas e medidas através de sensores. O controle dessas variáveis de saída é, essencialmente, o objetivo da análise e projeto no Espaço de Estados.

Um Sistema Linear Variante no Tempo é normalmente representado da seguinte forma:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{5.1}$$
$$y(t) = C(t)x(t) + D(t)u(t) \tag{5.2}$$

onde as Equações (5.1) e (5.2) determinam, respectivamente, a equação de estado e da equação de saída do sistema. Nesse conjunto de equações, estão representados: $x(t)$, o vetor de estados; $u(t)$, o vetor de entrada de controle no sistema; $y(t)$, o vetor de saída do sistema; $A(t)$, a matriz de estado; $B(t)$, a matriz de entrada; $C(t)$, a matriz de saída; e $D(t)$, a matriz de transmissão direta.

A abordagem de Espaço de Estados é usada no presente trabalho a fim de se analisar o Sistema de Controle integrado de congestionamento (emissores e AQM). Assim, nos próximos capítulos, são vistos mais a fundo os processos de determinação das equações de estado do sistema, a simplificação do modelo, a escolha do ponto de equilíbrio, a linearização e a derivação dos controladores. Na etapa de derivação dos controladores, o sistema é expresso na forma de Espaço de Estados, como definido acima, e são vistos em detalhes os processos de avaliação dos objetivos do sistema e preenchimento adequado das matrizes do Espaço de Estados.

Capítulo 6

Modelagem do sistema de controle de congestionamento

Neste capítulo, o sistema integrado de controle de congestionamento, que inclui emissores de dados e mecanismos de AQM, é modelado como um sistema de Controle. As características do tráfego da classe de Serviço Assegurado (AF) de *DiffServ* são modeladas, bem como o comportamento das filas nos roteadores. Esse modelo, já simplificado, é então linearizado para que, futuramente, a ele possam ser aplicados métodos de análise e projeto de controladores ótimos para regular o sistema de controle de congestionamento.

Na primeira seção, é visto de forma geral o modelo de rede considerado, que representa um domínio *DiffServ* simplificado, com apenas um enlace gargalo (congestionado) e vários emissores adaptativos (TCP) e não adaptativos (UDP) gerando o tráfego que a ele é enviado.

6.1 Comportamento do mecanismo de AQM no núcleo da rede

Nesta seção, é introduzido o modelo de rede elaborado em [11], que representa um domínio *DiffServ* cujo tráfego é composto, basicamente, por fluxos agregados da classe AF-4 (Serviço Assegurado).

O cenário de rede considerado, é um domínio *DiffServ* como o ilustrado na Figura 2.1. Para $0 < j < m$, onde m é o número de roteadores de entrada (borda) do domínio, considere $r_j(t)$ como sendo a taxa de tráfego enviada por cada roteador de borda ao núcleo do domínio. Em conjunto, esses roteadores de borda alimentam um mesmo enlace gargalo, de capacidade C, de forma que $\sum_{j=1}^{m} r_j = C$.

Considere agora $\tilde{r}_j$ como sendo a banda mínima garantida ao agregado de fluxos de

um determinado roteador de entrada. Essa banda mínima refere-se especificamente à taxa de envio de tráfego que tal roteador envia ao núcleo do domínio.

Seja $A_j(t)$ uma fração $f_j^i(t)$ de $r_j(t)$ que caracteriza a taxa de envio de pacotes do tipo *in* com relação ao total. Essa taxa é o reflexo direto da taxa média de marcação dos *Token Buckets*. A fração do tráfego que é marcada como *in* é definida por

$$f_j^i(t) = min\left(1, \frac{A_j(t)}{r_j(t)}\right) \tag{6.1}$$

Já a fração do tráfego marcada como *out* é definida, de forma complementar, como $f_j^o = 1 - f_j^i$.

No núcleo, p_i e p_o denotam, respectivamente, as probabilidades de descarte dos pacotes *in* e *out*. Consistentemente com os conceitos de *DiffServ*, abordados na Seção 2.3.2, os valores dessas probabilidades são limitados por

$$0 < p_i(t) f_j^i(t) < p_o(t)(1 - f_j^i(t)) < 1 \tag{6.2}$$

Dessa forma, a taxa de descarte dos fluxos *in* é sempre menor que a dos *out*, ou seja, os pacotes *out* são preferencialmente descartados na ocorrência de congestionamento.

A seguir, na próxima seção, é apresentada a modelagem do sistema de controle de congestionamento. Essa modelagem baseia-se na análise comportamental dos emissores de tráfego e da variação das filas nos roteadores.

6.2 Modelo de comportamento dinâmico do tráfego da classe *Assured Service*

Nesta seção, é especificado o modelo de rede e as equações que descrevem o sistema de controle de congestionamento em tal ambiente.

O tráfego considerado é basicamente gerado por emissores TCP Reno e UDP. A seguir, é apresentado o modelo dinâmico simplificado baseado em análise de fluidos e equações diferenciais estocásticas que caracterizam o comportamento da variação da janela de transmissão do TCP ao longo do tempo. Em seguida, é apresentada a equação que descreve a variação do tamanho da fila em função da capacidade do enlace e da demanda.

A principal simplificação do modelo que é apresentado é a não incorporação do mecanismo de RTO (*timeout*) do TCP Reno. Ignora-se também na modelagem as retransmissões provocadas por *timeouts*, considerando-se apenas que todas as perdas são notadas pelo emissor através do recebimento de ACKs duplicados. Tem-se como foco neste modelo, o comportamento da variação da janela TCP durante sua fase de estabilização.

O domínio *DiffServ* considerado neste trabalho possui m roteadores de borda, cada um deles: (i) servindo como agregador para um tráfego composto por N_j fluxos TCP iguais e (ii) possuindo *Token Buckets* com taxa de marcação $A_j(t)$ e tamanho de balde $b_j \gg 1, j = 1, \ldots, m$. Conjuntamente, esses roteadores de borda alimentam um único roteador do núcleo, que possui um enlace de saída com capacidade C e uma fila de tamanho q.

Em um dado instante $t > 0$, cada fluxo TCP é caracterizado pelo tamanho de sua janela $W_j(t)$ e tempo de viagem (*RTT - round-trip time*):

$$R_j(t) = T_j + \frac{q(t)}{C} \tag{6.3}$$

Nessa equação, T_j é uma constante que representa o tempo de propagação do meio físico (enlaces por onde passam os pacotes) e $\frac{q(t)}{C}$ representa o tempo introduzido no RTT pelo enfileiramento de pacotes no roteador do enlace gargalo.

A taxa de envio agregada r_j de cada um dos m roteadores de borda é dada por

$$r_j(t) = \frac{N_j W_j(t)}{R_j(t)} \tag{6.4}$$

Nessa equação, considera-se que a taxa de cada agregado é definida pelo número de emissores TCP (N_j) agregados naquele roteador de borda, o tamanho da janela de transmissão de cada um em determinado instante de tempo ($W_j(t)$) e o RTT do momento ($R_j(t)$).

A variação da janela dos emissores TCP segue a seguinte equação diferencial:

$$\dot{W}_j(t) = \frac{1}{R_j(t)} - \frac{W_j(t) W_j(t - R_j(t))}{2 R_j(t - R_j(t))} p(t - R_j(t)) \tag{6.5}$$

Nessa equação, o primeiro termo caracteriza o crescimento aditivo da janela e o segundo o decréscimo multiplicativo (AIMD - *Additive Increase Multiplicative Decrease*) do TCP Reno. Nela, a probabilidade de descarte $p(t - R_j(t))$ é dada por

$$p = \frac{A_j}{r_j(t)} p_i(t - R_j(t)) + \left(1 - \frac{A_j}{r_j(t)}\right) p_o(t - R_j(t)) \tag{6.6}$$

A variação do tamanho da fila no roteador do núcleo é dada como a diferença entre a capacidade do enlace (primeiro termo) e a taxa de chegada (segundo termo), determinada pelo somatório dos m agregados de tráfego:

$$\dot{q}(t) = -C + \sum_{j=1}^{m} \frac{N_j W_j(t)}{R_j(t)} + \omega_q(t) \tag{6.7}$$

Nessas equações, tem-se:

- $W_j(t)$: tamanho médio da janela TCP em pacotes;
- $\dot{W}_j(t)$: variação do tamanho da janela ao longo do tempo;
- $q(t)$: tamanho da fila em pacotes;
- $\dot{q}(t)$: variação do tamanho da fila;
- $\omega_q(t)$: ruído gerado pelo tráfego UDP;
- $R_j(t)$: tempo total de viagem (RTT) em segundos;
- C: capacidade do enlace em pacotes/segundo;
- T_p: tempo de propagação em segundos;
- N_j: número de conexões TCP;
- A_j: taxa de marcação do *Token Bucket*
- p_i: probabilidade de marcação de pacotes *in*;
- p_o: probabilidade de marcação de pacotes *out*;

A Equação (6.7) difere do modelo original, apresentado em [11] pela introdução do termo $\omega_q(t)$, que modela a influência do tráfego não adaptativo, como UDP, no sistema. A modelagem original considera que todo o tráfego é gerado por emissores TCP e a adição de tal termo aproxima a modelagem das condições mais realistas de rede, que inclui outros tipos de tráfego.

6.2.1 Modelagem dos controladores *non-overlapping*

Assim como dsPI-AQM ou RIO, a modelagem de dsH2-AQM terá como objetivo a regulagem do tamanho da fila. Dependendo da condição em que a rede se encontre, é importante que essa regulagem seja feita com base em relação a limiares (comprimentos de fila) convenientemente escolhidos para aquela condição.

No caso de uma rede superprovida ou exatamente provida, onde há recursos disponíveis para sustentar as garantias oferecidas aos usuários. Essa pode ser definida como uma condição normal de operação da rede. Nesse caso, objetiva-se manter o comprimento da fila próximo a um determinado limiar q_out preestabelecido, que ofereça ao usuário um serviço com baixo nível de atraso mas capaz de comportar as variações de fluxo da rede. Nessa condição de rede, o descarte de pacotes *in* é evitado a todo custo.

Já nos casos de redes subprovidas, onde não há recursos suficientes para sustentar tais garantias, até mesmo os pacotes *in* precisam ser descartados. Nesse caso, busca-se levar o sistema a um ponto de funcionamento que possui filas maiores, capazes de comportar excedente de tráfego *in* na rede, com o intuito de evitar que ele seja sumariamente descartado.

A disparidade entre essas condições de rede requer atuações diferentes por parte dos mecanismos de controle de congestionamento em cada uma delas. Baseado nisso, são estabelecidos dois pontos de equilíbrio para os quais deseja-se conduzir o sistema, um para as condições normais de operação e outro para a condição de subprovisão.

Na próxima seção, são definidas as equações que modelam matematicamente os pontos de equilíbrio do sistema.

6.2.2 Pontos de Equilíbrio

Nesta seção é apresentada a obtenção dos pontos de equilíbrio do sistema, tanto com respeito às equações que o descrevem quanto aos valores das variáveis que definem os estados do sistema nos quais se alcança o equilíbrio.

Expressão do Ponto de Equilíbrio

O primeiro passo na análise de um sistema não linear é a identificação do seu estado de equilíbrio. O estado de equilíbrio em um sistema é um ponto de operação que, uma vez alcançado, tende a nele se manter indefinidamente. Em um sistema de equações diferenciais, como é o caso, o ponto de equilíbrio é calculado igualando-se a zero as derivadas de todas as equações.

Para o modelo comportamental do TCP, obtido na Seção 6.2, considera-se algumas de suas incógnitas como constantes, como é o caso do número de conexões TCP ($N(t)$), a capacidade do enlace ($C(t)$), o ruído ($\omega(t)$) e a taxa de marcação dos *Token Buckets* ($A(t)$). As demais incógnitas são consideradas como variáveis nos pontos de equilíbrio, que são dados por (W_{0j}, q_0, p_{o0}) e (W_{0j}, q_0, p_{i0}).

No primeiro dos pontos de equilíbrio, (W_{0j}, q_0, p_{o0}), considera-se as situações normais de operação da rede, com cenários superprovidos ou exatamente providos. Nesse ponto de equilíbrio, a fim de satisfazer as restrições do critério *non-overlapping*, a probabilidade de descarte dos pacotes *in* deve ser nula, enquanto a probabilidade de descarte dos pacotes *out* varia entre zero e um. Por esse motivo, apenas a flutuação das variáveis W_{0j}, q_0 e p_{o0} influi no estado de equilíbrio, razão pela qual elas compõem tal ponto de equilíbrio.

No segundo ponto de equilíbrio, (W_{0j}, q_0, p_{i0}), ainda com a finalidade de satisfazer as restrições do critério *non-overlapping*, enquanto a probabilidade de descarte dos pacotes *out* mantém-se igual a um, a probabilidade de descarte dos pacotes *in* varia entre zero e um. Por esse motivo, apenas a flutuação das variáveis W_{0j}, q_0 e p_{i0} influi no estado de equilíbrio, razão pela qual elas compõem tal ponto de equilíbrio.

A consideração de um número de conexões TCP constante é razoável já que o sistema modelado considera, basicamente, que o tráfego é formado por fluxos de longa duração. Por sua vez, considerar que o tráfego é formado por fluxos de longa duração é também

uma abordagem interessante. Isso justifica-se pelo objetivo de estudar o congestionamento provocado por fluxos que enviam uma quantidade de pacotes suficiente à rede, de forma que, quando esta reagir ao congestionamento, tais fluxos ainda estejam ativos para recebê-la.

Partindo das Equações (6.5) e (6.7) e igualando suas derivadas a zero ($\dot{W}_j(t) = 0$ e $\dot{q}(t) = 0$), as expressões para as componentes dos pontos de equilíbrio (W_{0j}, q_0, p_{o0}) e (W_{0j}, q_0, p_{i0}) são equivalentes, e dadas por:

$$0 = -C_0 + \sum_{j=1}^{m} \frac{N_{0j} W_{0j}}{R_j} = -C_0 + m\frac{N_{0j} W_{0j}}{R_{0j}} \tag{6.8}$$

$$0 = \frac{1}{R_j} - \frac{W_{0j} W_{R0j}}{2R_{Rj}} \left[\frac{A_{0j}}{r_j} p_{i0} + (1 - \frac{A_{0j}}{r_j}) p_{o0} \right]$$

Partindo da Equação (6.3), chega-se à expressão que define q_0:

$$q_0 = C_0(R_j - T_j)$$

Na próxima seção, é conduzida a linearização do sistema em torno dos pontos de equilíbrio, aqui definidos.

6.2.3 Linearização em torno dos pontos de equilíbrio

Nesta seção, é feita a linearização do sistema de equações, definido na Seção 6.2, em torno dos pontos de equilíbrio, definidos na Seção 6.2.2.

Na linearização, foram consideradas algumas simplificações para melhorar a visualização das expressões:

$W(t) = W$:	Tamanho da janela atual;
$W(t - R(t)) = W_R$:	Tamanho da janela na medição anterior;
$R(t) = R$:	RTT atual;
$R(t - R(t)) = R_R$:	RTT na última medição;
$q(t) = q$:	Tamanho da fila atual;
$q(t - R(t)) = q_R$:	Tamanho da fila na última medição.
$p_i(t - R(t)) = p_{iR}$:	Valor de p_i na última medição.
$p_o(t - R(t)) = p_{oR}$:	Valor de p_o na última medição.

Foram consideradas também algumas aproximações:

$R(t) = R_0$: RTT constante;
$r(t) = r$: Taxa de emissão dos agregados constante;
$C(t) = C0$: Capacidade do enlace fixa;
$N(t) = N_0$: Número médio de conexões constante;
$\omega_q(t) = \omega_{q0}$: Ruído constante.

Sejam $f(W_i, q) =$ eq(6.7) e $g_j(W_j, W_{jR}, q, q_R, p_{iR}, p_{oR}) =$ eq(6.5).

Seguindo o método de linearização de [42], previamente introduzido na Seção 5.3, são desenvolvidas, a seguir, as derivadas parciais das equações eq(6.7) e eq(6.5) em relação a cada uma das variáveis dos pontos de equilíbrio.

Em estado de equilíbrio, considera-se que $W = W_R = W_0, q = q_R = q_0, p_i = p_{iR} = p_{i0}$ e $p_o = p_{oR} = p_{o0}$.

As derivadas parciais para $f(W_j, q)$ são:

$$\frac{\partial f}{\partial W_j} = \sum_{j=1}^{m} \frac{N_j}{R_j}$$

$$\frac{\partial f}{\partial q} = -\sum_{j=1}^{m} \frac{N_{0j} W_{0j}}{C_0 (T_j + \frac{q}{C_0})^2} = -\sum_{j=1}^{m} \frac{N_{0j} W_{0j}}{C_0 R_j^2}$$

As derivadas parciais para $g_j(W_j, W_{jR}, q, q_R, p_{iR}, p_{oR})$ são:

$$\frac{\partial g_j}{\partial W_j} = -\frac{W_{0j}}{2R_{Rj}} \left[\frac{A_{0j}}{r_j} (p_{i0} - p_{o0}) + p_{o0} \right] = -\frac{1}{W_{0j} R_{Rj}}$$

$$\frac{\partial g_j}{\partial W_{Rj}} = \frac{\partial g_j}{\partial W_j}$$

$$\frac{\partial g_j}{\partial p_{iR}} = -\frac{W_{0j}^2 A_{0j}}{2R_{Rj} r_j}$$

$$\frac{\partial g_j}{\partial p_{oR}} = -\frac{W_{0j}^2}{2R_{Rj}} \left(1 - \frac{A_{0j}}{r_j} \right)$$

$$\frac{\partial g_j}{\partial q} = -\frac{1}{R_j^2 C_0}$$

$$\frac{\partial g_j}{\partial q_R} = \frac{W_{0j}^2}{2R_{Rj} C_0} \left[\frac{A_{0j}}{r_j} (p_{i0} - p_{o0}) + p_{o0} \right] = \frac{1}{R_{Rj} C_0}$$

Definidas as derivadas parciais, a linearização das Equações (6.7) e (6.5) em torno dos pontos de equilíbrio segue de forma separada.

Primeiro ponto de equilíbrio: condições superprovidas ou exatamente providas

A partir das derivadas parciais, a linearização em torno do ponto de equilíbrio (W_{0j}, q, p_{o0}) resulta em:

$$\begin{aligned}\dot{W}_j(t) &= -\frac{1}{W_{0j}R_j}\partial W_j(t) - \frac{1}{W_{0j}R_j}\partial W_j[t-R_j(t)] - \frac{1}{R_j^2C}\partial q + +\frac{1}{R_jC}\partial q[t-R_j(t)] \\ &\quad + \left[\frac{W_{0j}^2}{2R_{Rj}}\left(\frac{A_{0j}}{r_j}-1\right)\right]\partial p_o[t-R_j(t)] \end{aligned} \tag{6.9}$$

$$\dot{q}(t) = \sum_{j=1}^{m}\frac{N_j}{R_j}\partial W_j(t) - \sum_{j=1}^{m}\frac{N_jW_{0j}}{CR_j^2}\partial q(t) \tag{6.10}$$

Agora, considera-se:

$$x_{1j}(t) = W_j(t) - W_{0j} \tag{6.11}$$

$$x_2(t) = q(t) - q_0 \tag{6.12}$$

$$u(t) = p_o(t - R_j(t)) - p_{o0} \tag{6.13}$$

Essas novas variáveis representam a perturbação (erro) das variáveis em com relação aos seus valores definidos no ponto de equilíbrio. Utilizando essas novas variáveis (6.11), (6.12) e (6.13) em (6.9) e (6.10) tem-se:

$$\begin{aligned}\dot{x}_{1j}(t) &= -\frac{1}{W_{0j}R_j}[x_{1j}(t) - x_{1j}(t-R_0)] - \frac{1}{R_j^2C}x_2(t) + \frac{1}{R_jC}x_2(t-R_0) + \\ &\quad + \left[\frac{W_{0j}^2}{2R_{Rj}}\left(\frac{A_{0j}}{r_j}-1\right)\right]u(t-R_0) \end{aligned} \tag{6.14}$$

$$\dot{x}_2(t) = \sum_{j=1}^{m} \frac{N_j}{R_j} x_{1j}(t) - \sum_{j=1}^{m} \frac{N_j W_{0j}}{C R_j^2} x_2(t) \tag{6.15}$$

Agora, utilizando a Equação (6.4) para fazer $r_j = \frac{N_j W_j(t)}{R_j(t)}$ e $R \approx R_R$ tem-se em (6.14):

$$\begin{aligned} \dot{x}_{1j}(t) &= -\frac{1}{W_{0j} R_j}[x_{1j}(t) - x_{1j}(t - R_0)] - \frac{1}{R_j^2 C} x_2(t) + \frac{1}{R_j C} x_2(t - R_0) + \\ &+ \left(\frac{W_{0j} A_{0j}}{2Nj} - \frac{W_{0j}^2}{2R_j} \right) u(t - R_0) \end{aligned} \tag{6.16}$$

Considerando-se que fluxos TCP da Equação (6.15) têm comportamento igual, tem-se:

$$\dot{x}_2(t) = m \frac{N_j}{R_j} x_{1j}(t) - m \frac{N_j W_{0j}}{C R_j^2} x_2(t) \tag{6.17}$$

Isolando-se W_{0j} na equação(6.8), tem-se que $W_{0j} = \frac{C R_j}{N_j m}$. Sendo assim, as Equações (6.16) e (6.15) podem ser reescritas da seguinte forma:

$$\begin{aligned} \dot{x}_{1j}(t) &= -\frac{N_j m}{C R_j^2}[x_{1j}(t) - x_{1j}(t - R_0)] - \frac{1}{R_j^2 C} x_2(t) + \frac{1}{R_j C} x_2(t - R_0) + \\ &+ \left(\frac{C A_{0j} R_j}{2 N_j^2 m} - \frac{C^2 R_j}{2 N_j^2 m^2} \right) u(t - R_0) \end{aligned} \tag{6.18}$$

$$\dot{x}_2(t) = m \frac{N_j}{R_j} x_{1j}(t) - \frac{1}{R_j} x_2(t) \tag{6.19}$$

Segundo ponto de equilíbrio: redes subprovidas

A partir das derivadas parciais, a linearização em torno do ponto de equilíbrio (W_{0j}, q, p_i) resulta em:

$$\begin{aligned} \dot{W}_j(t) &= -\frac{1}{W_{0j} R_j} \partial W_j(t) - \frac{1}{W_{0j} R_j} \partial W_j[t - R_j(t)] - \frac{1}{R_j^2 C} \partial q + + \frac{1}{R_j C} \partial q[t - R_j(t)] \\ &- \frac{W_{0j}^2 A_{0j}}{2 R_{Rj} r_j} \partial p_i[t - R_j(t)] \end{aligned} \tag{6.20}$$

$$\dot{q}(t) = \sum_{j=1}^{m} \frac{N_j}{R_j} \partial W_j(t) - \sum_{j=1}^{m} \frac{N_j W_{0j}}{C R_j^2} \partial q(t) \tag{6.21}$$

Como no primeiro ponto de equilíbrio, considera-se:

$$x_{1j}(t) = W_j(t) - W_{0j} \tag{6.22}$$

$$x_2(t) = q(t) - q_0 \tag{6.23}$$

$$u(t) = p_i(t - R_j(t)) - p_{i0} \tag{6.24}$$

Utilizando as novas variáveis (6.22), (6.23) e (6.24) em (6.20) e (6.21) tem-se:

$$\begin{aligned} \dot{x}_{1j}(t) &= -\frac{1}{W_{0j} R_j}[x_{1j}(t) - x_{1j}(t - R_0)] - \frac{1}{R_j^2 C} x_2(t) + \frac{1}{R_j C} x_2(t - R_0) + \\ &\quad -\frac{W_{0j}^2 A_{0j}}{2 R_{Rj} r_j} u(t - R_0) \end{aligned} \tag{6.25}$$

$$\dot{x}_2(t) = \sum_{j=1}^{m} \frac{N_j}{R_j} x_{1j}(t) - \sum_{j=1}^{m} \frac{N_j W_{0j}}{C R_j^2} x_2(t) \tag{6.26}$$

Agora, utilizando a Equação (6.4) para fazer $r_j = \frac{N_j W_j(t)}{R_j(t)}$ e $R \approx R_R$ tem-se em (6.25):

$$\begin{aligned} \dot{x}_{1j}(t) &= -\frac{1}{W_{0j} R_j}[x_{1j}(t) - x_{1j}(t - R_0)] - \frac{1}{R_j^2 C} x_2(t) + \frac{1}{R_j C} x_2(t - R_0) + \\ &\quad -\frac{W_{0j} A_{0j}}{2 N_j} u(t - R_0) \end{aligned} \tag{6.27}$$

Considerando-se que fluxos TCP da Equação (6.26) têm comportamento igual, tem-se:

$$\dot{x}_2(t) = m \frac{N_j}{R_j} x_{1j}(t) - m \frac{N_j W_{0j}}{C R_j^2} x_2(t) \tag{6.28}$$

Considerando-se que $W_{0j} = \frac{C R_j}{N_j m}$, as Equações (6.27) e (6.26) podem ser reescritas da seguinte forma:

$$\dot{x}_{1j}(t) = -\frac{N_j m}{C R_j^2}[x_{1j}(t) - x_{1j}(t - R_0)] - \frac{1}{R_j^2 C} x_2(t) + \frac{1}{R_j C} x_2(t - R_0) + \\ -\frac{C A_{0j} R_j}{2 N_j^2 m} u(t - R_0) \quad (6.29)$$

$$\dot{x}_2(t) = m\frac{N_j}{R_j} x_{1j}(t) - \frac{1}{R_j} x_2(t) \quad (6.30)$$

Determinação das condições de Equilíbrio

Na Seção 6.2.2 foram obtidas as equações que representam as variáveis do ponto de equilíbrio do sistema. Nesta seção, são avaliados pontos de equilíbrio para as duas condições básicas da rede, discutidas na Seção 6.2.1. A escolha dos valores das variáveis que determinam os estados nos quais o sistema se encontra em equilíbrio é, neste trabalho, fundamentada em estudos realizados recentemente que objetivam caracterizar o tráfego da Internet. Dessa forma, busca-se obter pontos de equilíbrio que reflitam condições próximas às reais em redes como a Internet.

Primeiramente, são apresentados os valores que formam o estado onde um sistema em condições normais alcança o equilíbrio. Em seguida, é apresentado o estado onde sistemas subprovidos alcançam o equilíbrio.

Equilíbrio em redes sob condições superprovida e exatamente provida

- **Tamanho do pacote**: Estudos conduzidos por [45] avaliaram o tamanho médio dos pacotes na Internet através da coleta de amostras de tráfego durante 10 meses, entre os anos de 1999 e 2000. Chegou-se à conclusão de que a grande maioria dos dados coletados encaixavam-se em quatro faixas de valores para tamanho do pacote: pacotes de 1500 bytes (unidade máxima de transferência – MTU do padrão Ethernet), 40 bytes (tamanho dos acknowledgements), 552 e 576 para implementações de protocolo que não usam o algoritmo *path MTU discovery* para estimar o tamanho dos máximo de pacote suportado pela rede. Por medida de compatibilidade com as diferentes implementações de protocolo, no presente trabalho é utilizada uma aproximação desses valores, com pacotes de tamanho **500 bytes**. A mesma escolha também foi feita em outros trabalhos [11].

- **Tempo de viagem (RTT) -** R: Na recomendação G.114 do ITU (*International Telecommunication Union* [35], é abordado o tema dos limites de atraso de pacotes para transmissão em aplicações multimídia, particularmente para as transmissões

de voz, que são mais sensíveis ao atraso. Do estudo feito, conclui-se que atrasos de até 150ms numa transmissão de áudio em tempo real são imperceptíveis ao ouvido humano; atrasos variando entre 150-400ms são toleráveis e acima disso podem trazer incômodos ou tornar a comunicação ininteligível. Já análises feitas por [55] apontam para um atraso médio de cerca de 200ms para o tempo de viagem em transmissões internacionais e intercontinentais. Como atrasos nessa faixa são considerados toleráveis, é adotado o valor de **256ms**, o que permitirá inclusive uma melhor comparação com os trabalhos já propostos que utilizam esse valor, como [11].

- **Capacidade do enlace -** C: O valor adotado é $C = 155Mbps$, o que corresponde a 38750 pacotes (de 500 bytes) por segundo, mesmo valor adotado em [11].

- **Número de roteadores de entrada -** m: O número de roteadores de entrada definirá o número de agregados de fluxos que alimentarão o núcleo congestionado da rede. Esses agregados de fluxos e os emissores TCP que os geram podem ter necessidades diferentes em termos de taxa de transmissão. Com o objetivo de evidenciar a possibilidade de ocorrência dessas disparidades, mostra-se importante a utilização de uma quantidade apenas não-singular de roteadores de entrada, de forma que se comporte casos em que se deseja definir diferentes taxas-alvo de transmissão para cada um dos agregados. Será arbitrado então um valor de **3** roteadores de entrada.

- **Taxa de envio agregada -** r: Será utilizada na modelagem uma rede exatamente provida ($\sum_{j=1}^{m} \tilde{r}_j = C$), onde a taxa mínima garantida para transmissão do tráfego agregado é exatamente igual à capacidade do enlace. Para isso, buscar-se-á fazer com que a taxa de transmissão efetiva dos emissores TCP r_j seja maior ou igual ao mínimo garantido $\tilde{r}_j$ e ao mesmo tempo comportada pela capacidade de transmissão do enlace. Dessa forma, a taxa de transmissão dos emissores deve ser regulada próxima ao valor mínimo garantido. Por motivo de simplificação do modelo matemático para simulação, são considerados Nm fluxos TCP com comportamento idêntico e agregados de fluxo com taxas r_j e $\tilde{r}_j$ iguais, dessa forma fazendo com que $r_j = \frac{C}{m} = \frac{N_j W_j}{R_j}$ (Equação (6.4)). Sendo assim, r_j assumirá então valor igual a **12916** pacotes/segundo.

- **Tamanho da janela TCP -** W: A prevenção de múltiplos *timeouts* no emissor TCP é condição necessária para a provisão de QoS. Com base nisso, análise feita em [41] infere um limiar inferior de 8 pacotes para W para evitar que, na ocorrência de congestionamento, o tamanho da janela chegue a um valor tão baixo que obstrua a execução do algoritmo Retransmissão Rápida do TCP, para se evitar o *timeout*. Seguindo essa orientação, neste trabalho, a modelagem do sistema adota $W = 8$.

- **Probabilidade de descarte dos fluxos *in* -** p_i: Para alcançar a qualidade de serviço necessitada pelos fluxos *in*, é dada a eles a mais baixa probabilidade de descarte, igual a zero. Dessa maneira, é esperado um baixo número de descartes, o que contribui para a evitar a variação de atraso (*jitter*). Será utilizado o valor $p_i = 0.0$.

- **Probabilidade de descarte dos fluxos *out* -** p_o: Em [22], Sally Floyd considera que em situações normais de funcionamento de redes, as taxas de descarte não devam superar 5% do total de pacotes. O caso contrário reflete uma situação de congestionamento extremo. Já em [49] é proposto o modelo *inverse-square-root of packet-loss* para prevenção de congestionamento, que relaciona o tamanho da janela de um emissor TCP e a taxa de descarte de seus pacotes. Segundo esse modelo, $W = \sqrt{\frac{2}{p}}$. Assim, para uma janela $W = 8$, p não deveria ultrapassar o limiar dos 3.125% (probabilidade 0.03125) para evitar a formação de congestionamento intenso. Considerando $p = 0.03125$ e sendo ele constituído por p_i e p_o, pode-se utilizar a Equação (6.5) para calcular o valor de p_o. Substituindo-se também os valores $p_i = 0.0$ e $r = 12916$ nessa equação, chega-se a uma igualdade que contém duas variáveis: $19.5 = (625 - A)p_o$. Nesse ponto, deve-se escolher os valores de ambas as variáveis arbitrando um deles, cuidando principalmente para conservar o valor de $p_o > p_i$. Não é desejável ainda que o valor de p_o seja alto a ponto de causar grande número de perda aos pacotes *out*, o que lhes poderia acarretar um aumento na sua variação de atraso. Dessa forma, foi arbitrado um valor para $A = 2583$ e conseqüentemente $p_o = 0.0390625$.

- **Número de fluxos TCP -** N: O número de conexões TCP do ponto de equilíbrio provém diretamente do estabelecimento dos valores para os parâmetros anteriores, já que, como derivado na Equação (6.8), $r = m\frac{NW}{R}$, ou igualmente $N = \frac{rR}{Wm}$. Dessa forma, N e W devem ser ajustados para fazer com que a taxa r_j atinja o mínimo estabelecido $\tilde{r}_j$. Considerando os valores já adotados para $r = 38750$ (agregado total), $W = 8$, $m = 3$ e $R = 0.256$, são consideradas por volta de **413 fluxos** TCP por agregado r_j de tráfego.

- **Tamanho da fila -** q: O tamanho da fila reflete, na verdade, não sua capacidade, mas o tamanho das rajadas de tráfego que se deseja absorver em curtos espaços de tempo. Além disso, o tamanho da fila determina o tempo de enfileiramento dos pacotes, fator que influi na contabilização do seu atraso fim-a-fim, o RTT (variável R), descrito pela Equação (6.3). Dessa forma, um bom valor para essa variável deve possibilitar não só a absorção dos fluxos em rajada mas também prover baixos (e limitados) níveis de atraso para os pacotes que nela aguardam. Com um tamanho

de fila igual a um quarto de sua capacidade total, é possível absorver os fluxos em rajada e comportar as variações de fluxo geradas pelos emissores TCP, que no pior caso, durante a fase de Partida Lenta do TCP, pode fazer com que o volume total de dados enviados dobre a cada RTT. O tamanho total da fila é dado pelo produto banda x atraso [39], $q = CR = 9920$. Portanto o tamanho da fila que se quer obter no ponto de equilíbrio proposto deve ser igual a **2480 pacotes**.

- **Tempo de propagação -** T: Apesar de o tempo de propagação não fazer parte do ponto de equilíbrio, seu valor é componente do RTT (variável R) e, seu valor precisa ser estipulado corretamente para que se alcance tal valor. Tendo como base a Equação (6.3) e utilizando os valores de R, q e C anteriormente definidos, chega-se a um atraso de transmissão de 0.192 segundos.

Equilíbrio em redes subprovidas

Alguns parâmetros, como tamanho do pacote, capacidade do enlace (C), número de roteadores de entrada (m), taxa de envio agregada (r), tamanho da janela TCP (W) e tempo de propagação (T) são os mesmos considerados no caso de redes em condição normal de operação. Os demais parâmetros são estabelecidos a seguir.

- **Tempo de viagem (RTT) -** R: Seguindo a mesma orientação G.114 do ITU [35], usada no caso anterior, na condição de subprovisão ainda se buscará oferecer um nível de atraso menor que 400ms, a fim de não prejudicar por demais a inteligibilidade da comunicação. Neste cenário, é arbitrado um atraso máximo 25% maior que o do caso anterior, agora igual a **320ms**.

- **Probabilidade de descarte dos fluxos *out* -** p_o: Em redes subprovidas, tem-se que o tráfego de pacotes *in*, por si só, já proporciona à rede um certo nível de congestionamento. Sendo assim, a fim de preservá-los, são excluídos todos os pacotes *out*, fazendo $p_o = 1$.

- **Probabilidade de descarte dos fluxos *in* -** p_i: Mesmo havendo a necessidade de descarte de pacotes *in* em um cenário de rede subprovido, essa taxa de descarte é minimizada o quanto possível. Dessa forma, ainda se terá como objetivo preservar totalmente os pacotes *in* em caso de congestionamento, fazendo $p_i = 0.0$.

- **Taxa de marcação dos *Token Buckets* -** A: Tendo p_i, p_o e r definidos, a definição de A resulta diretamente em **12513** pacotes.

- **Número de fluxos TCP -** N: O número de conexões TCP do ponto de equilíbrio é definido da mesma forma como feito no caso anterior. Considerando os valores já adotados para $r = 38750$ (agregado total), $W = 8$, $m = 3$ e $R = 0.256$, são consideradas por volta de **516 fluxos** TCP por agregado r_j de tráfego.

- **Tamanho médio da fila -** q: O tamanho médio da fila adotado é igual a metade de sua capacidade total. Dessa forma, é possível absorver todos os pacotes *in* e tentar preservá-los o máximo possível da ocorrência de descartes. Isso resulta em uma fila de tamanho médio **480 pacotes**.

Capítulo 7

Projeto dos controladores ótimos

Nas seções seguintes, o processo de síntese do controlador dsH2-AQM é explanado em detalhes. Inicialmente, é introduzida a metodologia a ser utilizada. Em seguida, é detalhado o processo de síntese dos controladores, que inclui a definição das matrizes do espaço de estados, a propriamente dita síntese do controlador, sua discretização para implementação digital *et cetera*.

7.1 Considerações sobre o projeto dos controladores

A modelagem apresentada no Capítulo 6 estabelece dois pontos de equilíbrio para o sistema, de acordo com os possíveis cenários em que a rede pode se encontrar. Para cada ponto de equilíbrio desses, é sintetizado um controlador capaz de estabilizar o sistema sob tais condições. Ou seja, o controlador dsH2-AQM é, na verdade, composto por dois controladores, um para o cenário de redes subprovido e outro para situações normais de operação (cenários superprovidos ou exatamente providos).

A razão de se haver dois controladores é, inclusive, uma necessidade intrínseca ao sistema. Dos conceitos de Teoria de Controle, sabe-se que cada controlador é representado por uma função de transferência, a qual descreve a relação entre uma entrada e uma saída do sistema. O sistema aqui modelado é do tipo MIMO (*Multiple Input Multiple Output*) [47], pois possui uma entrada (o tamanho da fila q medido) e duas saídas (as probabilidades de descarte p_i e p_o). Logo, são necessários dois controladores (duas funções de transferência) para descrever a relação entre as entradas e saídas do sistema.

Sendo assim, neste capítulo é explanado o desenvolvimento de dois controladores, um para geração da probabilidade de descarte dos fluxos *in*, aqui denominado $H2_{in}$, e outro para os fluxos *out*, denominado $H2_{out}$. O controlador $H2_{in}$ é desenvolvido sobre o ponto de equilíbrio dos cenários subprovidos e, através da manipulação da probabilidade de descarte dos pacotes *in*, objetivará conduzir o sistema ao ponto de equilíbrio definido.

Já o controlador $H2_{out}$ é desenvolvido sobre o ponto de equilíbrio do cenário normal de operação da rede.

Como definido na Seção 4.3, as condições de sub ou superprovisão de uma rede são determinadas pela quantidade de garantias de banda oferecidas por um ISP a seus clientes, em relação à capacidade da rede. O oferecimento dessas garantias aos clientes depende de uma prévia configuração das taxas de marcação e taxas-alvo dos *Token Buckets* nos roteadores de borda. Sendo assim, a condição sub ou superprovisão de uma rede pode ser considerada como uma característica não-transitória. Baseado nisso, o projeto dos controladores $H2_{in}$ e $H2_{out}$ considera seu uso mutuamente exclusivo e adequadamente predeterminado de acordo com a condição estabelecida para a rede. Sendo assim, a escolha do controlador mais adequado para uso em determinada rede depende explicitamente da condição à qual ela foi estabelecida.

Ambos os controladores são desenvolvidos segundo a especificação de mecanismos AQM *non-overlapping*, apresentada na Seção 4.6. Ao final deste capítulo, na Seção 7.6, são apresentados os pontos que garantem a conformidade dos controladores dsH2-AQM com tais especificações.

7.2 Metodologia

No capítulo anterior, o sistema de controle de congestionamento foi modelado matematicamente. Para ele, foram obtidos dois pontos de equilíbrio. Em seguida, tal modelo foi ainda linearizado, obtendo-se como resultado um sistema linear com atraso constante no tempo. Neste capítulo, utiliza-se a teoria de Controle Ótimo para desenvolver um controlador AQM que é conectado a esse sistema.

Na metodologia de Controle Ótimo, o comportamento do sistema desejado é formulada em termos de um critério que deve ser otimizado. O uso de uma função de custo ou objetivo força o projetista do controlador a especificar exatamente os objetivos do sistema, que, uma vez determinados, são alcançados de forma ótima [41].

A modelagem dos controladores é feita na forma de espaço de estados, onde a evolução do sistema de acordo com o tempo é representada como um conjunto de matrizes. Inicialmente, é necessário definir os objetivos de desempenho dos controladores e determinar as métricas a serem observadas a fim de verificar sua eficiência. Em seguida, as matrizes desses controladores capazes de estabilizar o sistema e ao mesmo tempo minimizar métricas específicas em suas saídas são determinadas. Dessa forma, são sintetizados dois controladores que apresentem, cada um, soluções ótimas para as métricas de desempenho definidas em seus respectivos cenários.

A síntese dos controladores baseia-se nos resultados apresentados em [48]. Tal trabalho apresenta uma abordagem na qual a forma do controlador projetado reproduz em detal-

hes a estrutura da planta a ser controlada. Dessa forma, o problema dos controladores para sistemas lineares com atraso, incluindo fatores como sua estabilidade e objetivos de desempenho, pode ser expresso e solucionado utilizando-se desigualdades matriciais lineares (LMIs - *Linear Matrix Inequalities*) [8]. Sendo assim, os parâmetros do controlador podem ser determinados a partir da solução de um problema convexo [1]. Ademais, a abordagem utilizada garante que a estabilização da planta pelo controlador implica na estabilidade do sistema de controle de congestionamento modelado.

A síntese do controlador dsH2-AQM usará uma abordagem não-racional, incorporando os termos com atraso, ou seja, sua modelagem considera, além dos parâmetros instantâneos da rede, um histórico dos seus últimos valores. No ambiente de redes de computadores, isso se traduz em uma melhor utilização dos recursos da rede, em se comparando com abordagens racionais [36].

Apesar de a modelagem introduzida no capítulo anterior utilizar termos dependentes do tempo, é utilizada a técnica apresentada em [48] que permite cancelá-los, fazendo com que o controlador resultante seja racional, ou seja, não incorporando termos dependentes do tempo. Em outras palavras, o controlador se vale de todas as vantagens de uma modelagem não-racional e, por fim, apresenta uma solução racional, o que reduz a complexidade de sua implementação e conseqüente melhora o desempenho.

7.3 Projeto do controlador dsH2-AQM

O sistema linear com atraso contínuo no tempo, modelado no Capítulo 6, pode ser expresso no espaço de estados pelas seguintes equações:

$$\begin{aligned} \dot{x}(t) &= A_0 x(t) + A_1 x(t - R_0) + B_w w(t) + B_u u(t); \\ z(t) &= C_z x(t) + D_{zu} u(t); \\ y(t) &= C_y x(t - R_0) + D_{yw} w(t); \end{aligned} \tag{7.1}$$

Nesse sistema de equações, todas as matrizes ($A_\bullet$, $B_\bullet$, $C_\bullet$, $D_\bullet$) são definidas apropriadamente com respeito ao vetor de estado $x(t) \in \mathbb{R}^n$, à entrada de controle $u(t) \in \mathbb{R}^m$, à entrada referente ao ruído $w(t) \in \mathbb{R}^r$, à saída de referência (desejada) $z(t) \in \mathbb{R}^q$ e à saída medida $y(t) \in \mathbb{R}^p$.

Considere que tal sistema esteja conectado a controladores do tipo:

[1] Para a solução de tal problema, utilizou-se a ferramenta LMI Solver [43], proposta pelos mesmos autores de [48] e disponível livremente para download na Internet, mediante registro prévio.

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}_0\hat{x}(t) + \hat{A}_1\hat{x}(t - R_0) + \hat{B}y(t); \\ u(t) &= \hat{C}_0\hat{x}(t) + \hat{C}_1\hat{x}(t - R_0) + \hat{D}y(t); \end{aligned} \tag{7.2}$$

Apesar de os controladores desenvolvidos serem descritos por conjuntos distintos de matrizes, ambos possuem estruturas semelhantes, como a descrita no sistema de Equações (7.2).

Considerando-se as Equações linearizadas (6.18) e (6.19) para o controlador $H2_{out}$ e, equivalentemente, as Equações (6.29) e (6.30)) para o controlador $H2_{in}$, pode-se definir o vetor de estados do Sistema (7.1) da seguinte forma:

$$x(t) = \left| \begin{array}{c} \sum_{i=1}^{m} x_{1i}(t) \\ x_2(t) \end{array} \right| \tag{7.3}$$

O termo somatório do vetor de estados (7.3) reflete a existência de m roteadores de borda (entrada) e, conseqüentemente, um conjunto de m equações que descrevem a variação das janelas dos emissores TCP de cada agregado. Considerando-se que os agregados são compostos por fluxos TCP de igual comportamento, pode-se reescrever a representação do vetor da seguinte forma:

$$x(t) = \left| \begin{array}{c} mx_{1i}(t) \\ x_2(t) \end{array} \right| \tag{7.4}$$

Esse conjunto de $m + 1$ equações descreve completamente o estado do sistema.

O próximo passo é determinar as matrizes do Controlador (7.2) que estabilizam o Sistema (7.1), minimizando certas medida na saída de referência $z(t)$. Para isso, faz-se necessário definir os objetivos de desempenho desejados para a saída $z(t)$, assim como a métrica observada na saída, $y(t)$. Dessa forma, fica clara a necessidade de expressar os objetivos de desempenho do mecanismo de AQM internamente no controlador $C_{H_2}(s)$.

A seguir, as matrizes do espaço de estados, utilizadas no Sistema (7.1), são definidas. Dadas as semelhanças entre os sistemas linearizados dos controladores $H2_{out}$ (Equações (6.18) e (6.19)) e $H2_{in}$ (6.29) e (6.30)), praticamente todas as matrizes que descrevem o comportamento de ambos os sistemas possuem o mesmo conteúdo, exceto a matriz B_u, que possui valores específicos para cada controlador.

Utilizando os sistemas de Equações linearizadas ((6.18) e (6.19), (6.29) e (6.30)), deve-se separar os termos dependentes do atraso dos termos não dependentes. Em cada linha da matriz A_0 são inseridos os coeficientes dos termos não dependentes do atraso das equações, referentes a W e q. Sua primeira linha é definida por $\frac{\partial g}{\partial W}$ e $\frac{\partial g}{\partial q}$. Sua segunda é definida por $\frac{\partial f}{\partial W}$ e a $\frac{\partial f}{\partial q}$.

De forma semelhante, na matriz A_1 são inseridos os termos do sistema que são dependentes do atraso. Sua primeira linha é definida por $\frac{\partial g}{\partial W_R}$ e $\frac{\partial g}{\partial q_R}$. A segunda linha desta matriz é nula, já que $\frac{\partial f}{\partial W_R} = \frac{\partial g}{\partial q_R} = 0$.

Sendo assim, as matrizes A_0 e A_1 podem ser definidas como segue:

$$A_0 = \begin{bmatrix} \frac{-Nm^2}{CR_0^2} & -\frac{1}{R_0^2 C} \\ \frac{mN}{R_j} & -\frac{1}{R_0} \end{bmatrix}, A_1 = \begin{bmatrix} \frac{-N_j m^2}{CR_j^2} & \frac{1}{R_j^2 C} \\ 0 & 0 \end{bmatrix}$$

A matriz B_u é preenchida com os coeficientes dos termos referentes à entrada de controle, que no caso é a probabilidade de descarte gerada por cada controlador. No caso do controlador $H2_{out}$, apenas a probabilidade p_o é considerada. Sendo assim, a matriz B_u desse controlador fica da seguinte forma:

$$B_u = \begin{bmatrix} -\frac{CA_{0i}R_0}{2N^2 m} - \frac{C^2 R_0}{2N^2 m^2} \\ 0 \end{bmatrix}$$

De forma equivalente, a matriz B_u do controlador $H2_{in}$ fica da seguinte forma:

$$B_u = \begin{bmatrix} -\frac{CA_{0i}R_0}{2N^2 m} \\ 0 \end{bmatrix}$$

A matriz B_w expressa o nível máximo de ruído comportado na modelagem dos controladores. Enquanto o nível de ruído do sistema estiver abaixo do valor especificado, garante-se que o controlador consegue estabilizar a planta próximo ao ponto de equilíbrio. Acima desses níveis de ruído, tal garantia não pode ser dada.

O ruído, gerado no sistema pela presença do tráfego UDP, que havia sido desconsiderado no passo de linearização, volta agora à modelagem do controlador na definição dessa matriz. A matriz B_w possui o mesmo conteúdo para ambos os controladores, o que indica que o nível máximo de ruído suportado é o mesmo para as duas condições de rede modeladas. O valor escolhido, de $0.2C$, permite que até 20% da capacidade do enlace seja ocupada por UDP. Tal valor é razoável, em se considerando que, segundo recentes medições, cerca de 83% de todos os bytes transmitidos pela Internet são gerados por fontes TCP [28].

$$B_W = \begin{bmatrix} 0 & 0 \\ 0.2C & 0 \end{bmatrix}$$

A matriz C_y indica que a variável controlada é o tamanho da fila q, medida no RTT anterior. D_{yw}, pondera o ruído na medição, que é, geralmente, 10% do valor presente na matriz B_w.

$$C_y = \left[\begin{array}{cc} 0 & 1 \end{array}\right], D_{yw} = \left[\begin{array}{cc} 0 & 0.02C \end{array}\right]$$

Resta ainda definir as matrizes C_{z0}, C_{z1} e D_{zu}. Tais matrizes refletem os objetivos de desempenho desejáveis para a saída de referência do sistema, $z(t)$, e são definidas na próxima seção.

O fato de algumas das matrizes serem definidas da mesma forma para ambos os controladores não implica, porém, que elas são consideradas idênticas durante a síntese do controlador. Como visto na Seção 6.2.3, tais controladores são construídos sobre pontos de equilíbrio diferentes, o que possivelmente associará valores diferentes para as variáveis definidas na síntese de cada um deles. Por exemplo, nas matrizes que definem o sistema do controlador $H2_{out}$, a variável N receberá o valor de 413 fluxos, enquanto que, nas matrizes do controlador $H2_{in}$, essa variável receberá o valor de 516 fluxos.

Tanto N quanto todas as demais variáveis utilizadas no preenchimento das matrizes são convenientemente substituídas pelos valores de equilíbrio definidos na Seção 6.2.3 para o cenário de cada controlador.

7.3.1 Objetivo do controlador

Durante o projeto de um controlador, tem-se como objetivo fazer com que ele estabilize a planta do sistema e que essa estabilidade seja alcançada em torno de um certo ponto de equilíbrio, previamente definido.

Independentemente da abordagem escolhida para análise e projeto de um controlador, em caso de sucesso, espera-se que, a partir de qualquer estado inicial, tal controlador sempre conduza o sistema a um mesmo ponto de equilíbrio. As diferenças entre os controladores desenvolvidos sob essas diversas abordagens resume-se à forma *como* esses controladores conduzem o sistema ao ponto de equilíbrio.

A forma como um controlador conduz o sistema a partir de um estado inicial qualquer até a estabilidade em um determinado ponto de equilíbrio pode transcorrer de diferentes formas. Esse caminho seguido pelo controlador é definido pelos seus objetivos de projeto.

Os objetivos dos controladores dsH2-AQM são expressos através das matrizes C_{z0}, C_{z1} e D_{zu}. Como estabelecido no Capítulo 4, ambos os controladores têm como objetivos principais facilitar a garantia de banda passante e minimizar a ocorrência de *jitter*. Isso pode ser conseguido através da minimização da variação do tamanho da janela dos emissores TCP (W) e do tamanho da fila (q).

Para se alcançar tal objetivo, busca-se fazer com que os valores de W e q mantenham-se próximos aos valores definidos no ponto de equilíbrio. Sabe-se que a caracterização do comportamento dessas variáveis no sistema é dada pelas matrizes A_0, A_1 e B_u e sabe-se que o sistema de equações dos controladores é construído de forma a reproduzir internamente a planta do sistema controlado. Logo, para fazer com que os valores de W

e q aproximem-se dos valores definidos no equilíbrio, basta definir as matrizes C_{z0}, C_{z1} e D_{zu} de forma com que elas sejam respectivamente iguais às matrizes A_0, A_1 e B_u do Sistema linearizado (7.1).

7.4 Síntese dos controladores

A síntese dos controladores segue a abordagem introduzida em [48]. Em tal trabalho, controladores para sistemas lineares com atraso são expressos e solucionados por meio de LMIs.

Uma vez determinadas as matrizes que caracterizam o sistema para os dois controladores dsH2-AQM, o processo de síntese transcorre da mesma forma para ambos, por meio da solução de um problema convexo baseado em LMIs, feito com o auxílio do software LMI Solver [43], desenvolvido pelos autores de [48].

Definidos os objetivos do controlador, em seguida, eles devem ser conectados ao Sistema (7.1). Seja $\bar{x}(t)$ o vetor de estado aumentado, que agrupa o vetor de estado $x(t)$ e o vetor de estado do controlador $\hat{x}(t)$:

$$\bar{x}(t) = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} \tag{7.5}$$

A conexão do Sistema (7.1) com o controlador resulta no sistema linear com atraso:

$$\begin{aligned} \dot{\bar{x}}(t) &= \mathcal{A}_0\bar{x}(t) + \mathcal{A}_1\bar{x}(t - R_0) + \mathcal{B}w(t); \\ z(t) &= \mathcal{C}_0\bar{x}(t) + \mathcal{C}_1\bar{x}(t - R_0) + \mathcal{D}w(t); \end{aligned} \tag{7.6}$$

onde:

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

Através do Teorema 4-b de [48], garante-se a estabilidade do Sistema (7.6). Esse teorema especifica que um sistema é assintoticamente estável e $\|H_{wz}(s)\|_2^2 < \gamma$, se $\mathbf{D} = 0$ e se existem matrizes simétricas definidas positivas W, Y_0 e X_j, e matrizes F, R, L_j e Q_j, com $j = 0, 1$, tais que as seguintes LMIs tenham ao menos uma solução factível:

$$
\begin{bmatrix} \mathbf{A}_0 + \mathbf{A}_0^T + X_1 & (\bullet)^T & (\bullet)^T \\ \mathbf{A}_1^T & -X_1 & (\bullet)^T \\ \mathbf{C}_0 & \mathbf{C}_1 & -I \end{bmatrix} < 0 \tag{7.7}
$$

$$
\begin{bmatrix} W & (\bullet)^T \\ \mathbf{B} & \mathbf{P}_0 \end{bmatrix} > 0, \quad trace(W) < \gamma \tag{7.8}
$$

onde $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{B}$, $\mathbf{C}_0$, $\mathbf{C}_1$ e $\mathbf{P}_0$ são dadas por:

$$\mathbf{A}_0 = \begin{bmatrix} A_0X_0 + B_uL_0 & A_0 \\ Q_0 & Y_0A_0 \end{bmatrix},$$

$$\mathbf{A}_1 = \begin{bmatrix} A_1X_0 + B_uL_1 & A_1 \\ Q_1 & Y_0A_1 + FC_y \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} B_w \\ Y_0B_w + FD_{yw} \end{bmatrix},$$

$$\mathbf{P}_0 = \begin{bmatrix} X_0 & I \\ I & Y_0 \end{bmatrix},$$

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

$$\mathbf{C}_1 = \begin{bmatrix} D_{zu}L_1 & 0 \end{bmatrix};$$

Obtida uma solução factível, busca-se, então, determinar os parâmetros do controlador. Primeiramente, matrizes arbitrárias U_0 e V_0 devem ser escolhidas de forma que $V_0U_0 = I - Y_0X_0$. As matrizes utilizadas foram $U_0 = X_0$ e $V_0 = X_0^{-1} - Y_0$. Assim, os parâmetros do controlador podem ser determinados por:

$$\begin{bmatrix} \hat{A}_0 & \hat{A}_1 & \hat{B} \\ \hat{C}_0 & \hat{C}_1 & \hat{D} \end{bmatrix} = \mathcal{K}.\mathcal{M}.\mathcal{N}. \tag{7.9}$$

onde:

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

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

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

Nas soluções obtidas, as matrizes $\hat{A}_1$ e $\hat{C}_1$ são aproximadamente zero, sendo portanto ignoradas. Tem-se, conseqüentemente, o cancelamento do termo de atraso do sistema, e os controladores tornam-se racionais. O cancelamento do termo de atraso, quando possível,

é a estratégia global ótima para solucionar o problema de minimização da norma H_2 [48].

Obtido o controlador, sua apresentação foi convertida da forma de espaço de estados para função de transferência, a fim de facilitar a sua implementação tanto em ambientes contínuos, como o software Matlab [44], quanto os discretos, como o simulador de redes NS-2 [25].

A função de transferência obtida para o controlador $H2_{out}$ foi:

$$C_{H2_{out}}(s) = \frac{0.045793(s + 0.0005042)}{(s + 1.07e04)(s + 1.056e - 07)}; \quad (7.10)$$

Já a função de transferência obtida para o controlador $H2_{in}$ foi:

$$C_{H2_{in}}(s) = \frac{0.030116(s + 0.0003085)}{(s + 1.031e04)(s + 1.283e - 07)}; \quad (7.11)$$

7.5 Discretização dos controladores

A implementação digital dos controladores obtidos depende da escolha de uma freqüência de amostragem f_s para que se obtenha uma representação no domínio de tempo discreto. A freqüência escolhida leva em consideração a capacidade de transmissão do enlace em termos de número de pacotes. O mecanismo de AQM RED e seus derivados, como o RIO, por exemplo, utilizam $f_s = 1$ para redes com carga em 100%. Isso significa que a cada chegada de um pacote, a probabilidade de descarte é completamente recalculada. Com a probabilidade de descarte variando de forma tão intensa em curtos intervalos de tempo, RED e RIO não conseguem marcar os pacotes de forma suficientemente aleatória a ponto de evitar o fenômeno da sincronização global, além de provocar perdas desnecessárias [41].

Dessa forma, mostra-se fundamental a escolha de uma freqüência de amostragem adequada para dar ao mecanismo de AQM a capacidade de reagir de forma apropriada às mudanças abruptas no tamanho da fila sem, no entanto, ser responsivo demais a ponto de ser influenciado por pequenas variações transientes [27].

Em [11], a freqüência de amostragem utilizada foi de 1%. Para a escolha da freqüência de amostragem adequada, foram realizados testes com o controlador dsH2-AQM discretizado com freqüência variando entre 1% e 50% da capacidade do enlace. Nesses testes, a mudança de frequência em pouco afetou os coeficientes dos controladores discretizados. Por fim, optou-se pela freqüência de amostragem de 10% da capacidade do enlace, medida de maior responsividade mas também muito comum dentre mecanismos de AQM propostos na literatura [33, 41].

Algoritmo 7.1 Algoritmo utilizado para o cálculo da probabilidade de descarte por ambos os controladores, $H2_{out}$ e $H2_{in}$

dsH2-AQM-Função-de-Probabilidade()
$p_0 \Leftarrow$ valor definido no ponto de equilíbrio;
$p \Leftarrow a(q - 2q_0 + q_{old}) - b.p_{old} + p_0(1 + b)$;
$p_{old} \Leftarrow p$;
$q_{old} \Leftarrow q$;

O projeto dos controladores $H2_{out}$ e $H2_{in}$ foi baseado em pontos de equilíbrio cujas capacidades do enlace era de 38750 pacotes por segundo. Logo, sua discretização tomou $f_s = 3875$. As funções de transferência para os controladores no domínio-z resultou em:

$$C_{H2_{out}}(z) = \frac{a(z+1)}{z+b} = \frac{2.4822^{-6}(z+1)}{z+0.1598}; \quad (7.12)$$

$$C_{H2_{in}}(z) = \frac{a(z+1)}{z+b} = \frac{1.6677^{-6}(z+1)}{z+0.1417}; \quad (7.13)$$

Como introduzido no Capítulo 5, funções de transferência representam a relação entre as entradas e saídas do sistema. No caso do sistema de controle de congestionamento modelado no Capítulo 6, a entrada é o comprimento da fila q e as saídas são as probabilidades de descarte p_o e p_i. Assim, as duas funções-transferência 7.12 e 7.13 representam a relação entre um δq de entrada e um δp de saída do controlador.

Essas funções de transferência entre $\delta p = p - p_0$ e $\delta q = q - q_0$, podem ser convertidas em equações-diferença, em tempo discreto kT, onde $T = \frac{1}{f_s}$. Essas equações possuem o seguinte formato:

$$\delta p_o(kT) = a[\delta q(kT) + \delta q((k-1)T)] - b\delta p_o((k-1)T); \quad (7.14)$$

$$\delta p_i(kT) = a[\delta q(kT) + \delta q((k-1)T)] - b\delta p_i((k-1)T); \quad (7.15)$$

Essa representação em equações-diferença serve como base para a implementação dos algoritmos de cálculo de probabilidade dos controladores $H2_{out}$ e $H2_{in}$. Como essas funções de cálculo de probabilidade de ambos possuem o mesmo formato, os algoritmos para implementação dessas funções também têm o mesmo formato:

7.6 O controlador dsH2-AQM e a condição de *non-overlapping*

Como introduzido no Capítulo 4, arquitetura para provisão de banda proposta em [12] estabelece algumas restrições que os mecanismos de Condicionamento de Tráfego e AQM devem seguir a fim de que se possa oferecer garantias de banda passante.

Para os mecanismos de AQM, a restrição é a de que eles sejam *non-overlapping*, ou seja, que nas condições de equilíbrio, valha o seguinte:

$$\begin{cases} \hat{p}_o < 1 \Rightarrow \hat{p}_i = 0 \\ \hat{p}_i > 0 \Rightarrow \hat{p}_o = 1 \end{cases}$$

No mesmo trabalho, são discutidas formas de configuração de alguns mecanismos de AQM para que esses possam se tornar compatíveis com a arquitetura proposta, dentre eles, dsPI-AQM e RIO.

Assim como dsPI-AQM e RIO, dsH2-AQM também obedece à restrição de *non-overlap*. Isso se deve à própria abordagem adotada na análise e projeto do controlador. Em tal abordagem, proposta em [48], o controlador projetado reproduz com precisão a planta do sistema e esse projeto considera explicitamente os pontos de equilíbrio estabelecidos. Dessa forma, uma vez estabelecido um ponto de equilíbrio e, havendo solução factível para o problema de controle proposto, o teorema 4-b de tal trabalho garante que o sistema é estabilizável, que o controlador resultante o estabiliza e que o ponto de equilíbrio estabelecido é atingido. Assim, ao se estabelecer um ponto de equilíbrio *non-overlapping* e encontrando-se ao menos uma solução factível, garante-se que o controlador resultante também será *non-overlapping*.

Como descrito na Seção 6.2.2, os pontos de equilíbrio sobre os quais foram desenvolvidos os dois controladores de dsH2-AQM foram cuidadosamente escolhidos a fim de respeitar a restrição *non-overlapping*. Sendo assim, como foram encontradas soluções factíveis para o problema proposto, os controladores resultantes, $H2_{out}$ e $H2_{in}$, também obedecem à restrição *non-overlapping*.

Além de obedecer à restrição *non-overlapping* nas condições de equilíbrio, os controladores de dsH2-AQM o fazem a todo instante, independentemente das condições da rede, assim como RIO o faz [12]. Isso se deve ao fato de que, na modelagem desenvolvida, em ambos as condições de rede, nunca é permitido que os valores das duas probabilidades de descarte flutuem simultâneamente. Em ambos os casos, enquanto uma das probabilidades é mantida constante, a outra é conduzida em direção ao ponto de equilíbrio. Sendo assim, as duas probabilidades nunca se sobreporão, mesmo em condições reais de utilização.

Capítulo 8

Implementação e simulação do dsH2-AQM

Neste capítulo, são apresentados os resultados das simulações desenvolvidas a fim de avaliar o desempenho de dsH2-AQM em relação aos mecanismos dsPI-AQM [11] e RIO [26], que é o padrão na arquitetura *DiffServ*.

Inicialmente, os experimentos são descritos de forma geral. Em seguida, são discutidas as métricas de desempenho consideradas, os *scripts* de simulação, a análise comparativa dos resultados e, por fim, o que se pode explorar em trabalhos futuros.

8.1 Descrição dos experimentos

Nesta seção, o processo de simulação dos controladores AQM, os cenários e os parâmetros usados nos experimentos são detalhados. As simulações consideram apenas um enlace congestionado, havendo simulações envolvendo tráfego exclusivamente adaptativo e outras envolvendo adicionalmente tráfego não adaptativo.

Em [11], é considerado um enlace-gargalo com capacidade de 15Mbps. Já em [12], considera-se um enlace-gargalo de 155Mbps. Na derivação do controlador dsH2-AQM a definição dos pontos de equilíbrio foi feita com o intuito de aproximar ao máximo os valores utilizados no primeiro artigo, a fim de facilitar as comparações. Para que as comparações fossem feitas com os cenários de [12], o ponto de equilíbrio dos controladores ótimos foi alterado para se adequar aos novos valores. Na prática, as mudanças ocorreram, basicamente, aumentando-se de forma proporcional o número de emissores TCP adotados (N), na taxa dos *Token Buckets* (A), nos tamanho médio das filas (q_i e q_o) e na capacidade do enlace congestionado (C).

Outros valores adotados nas simulações são:

Controlador $H2_{out}$: Tamanho médio da fila: 2480 pacotes, RTT médio: 0.256 seg, $p_i = 0.0$, $p_o = 0.0390625$ (Valores do ponto de equilíbrio).

Controlador $H2_{in}$: Tamanho médio da fila: 4960 pacotes, RTT médio: 0.320 seg, $p_i = 0.0$, $p_o = 1$ (Valores do ponto de equilíbrio).

Parâmetros gerais: tamanho dos pacotes: 500 bytes, tempo total de propagação nos enlaces (T): 0.192 seg, número de *Token Buckets*: 3 ou 4, dependendo do cenário de simulação, tamanho do *buffer* do roteador congestionado: 960 pacotes.

O tamanho do *buffer* no roteador congestionado reflete a capacidade máxima de armazenamento da fila. Considerando um enlace-gargalo de 155Mbps, o tamanho máximo do *buffer* deve ser:

$$buffer = \frac{155.000.000bits * RTT}{(500bytes * 8bits}) \tag{8.1}$$

Considerando os dois RTTs definidos nos pontos de equilíbrio dos controladores (0.256 seg e 0.320 seg), seria necessário um *buffer* com capacidade de 9930 pacotes ou 12400 pacotes. Foi adotado o maior dos valores, que é um valor realista do tamanho dos atuais buffers em roteadores da Internet.

A janela de recepção dos *sinks* (nós receptores) TCP é ajustada para um valor alto, por exemplo 1000, de modo que a capacidade de recepção do nós não influa na variação da janela de transmissão dos emissores TCP. Sendo assim, o controle de fluxo do TCP foi desativado, fazendo com que a transmissão seja afetada somente pelo estado da rede e ação dos mecanismos de AQM, como é desejado e previsto na modelagem matemática dos controladores.

O misto de tráfego TCP foi gerado pelo software TrafficGen [53]. Gera-se o tráfego web (de curta duração) segundo uma distribuição híbrida Lognormal/Pareto, na qual o corpo (88%)é modelado por uma distribuição Lognormal e a cauda (12%) é modelada por uma distribuição Pareto [4]. Segundo [10], a média da distribuição Pareto é 10558 e o *shape* 1.383, enquanto a distribuição Lognormal tem média 7247 e desvio padrão 28765. O tráfego FTP (longa duração) é modelado por uma distribuição exponencial com média 512KB, um valor comum para divisão de grandes arquivos que trafegam na web.

Os experimentos descritos a seguir têm como finalidade comparar o desempenho do mecanismo dsH2-AQM com as propostas atuais para a arquitetura de provisão de banda [12]. As comparações consideram o *goodput* por conexão, quantidade de *timeouts* por conexão etc.

A seguir, são descritos os cenários de simulação.

8.1.1 Experimentos com tráfego exclusivamente adaptativo

O primeiro bloco de experimentos compara o desempenho do dsH2-AQM com as propostas já existentes, em especial com o dsPI-AQM.

Nos testes, é usado o TrafficGen para gerar tráfego web similar ao utilizado por [12], no qual, apenas fontes TCP Reno são utilizadas para se gerar os dados. Assim, nestes primeiros testes, apenas esse tipo de tráfego, adaptativo, é considerado. Uma diferença dos cenários simulados nesta dissertação e os simulados em [12] é que, ao invés de considerar um número fixo de emissores, o uso do TrafficGen permite variar esse número, mantendo uma carga constante.

Foram realizados, inicialmente, dois experimentos. No Experimento 1, foi avaliado o funcionamento dos controladores sob condições normais de operação, em ambientes superprovidos e com níveis de congestionamento baixo ou moderado. No Experimento 2, foi analisado o comportamento de ambos os controladores em um ambiente de rede subprovido.

Experimento 1

Neste experimento, foi utilizada a topologia Dumbbell, simulando um domínio *DiffServ* composto por seis roteadores de borda, três de entrada e três de saída. Os roteadores de entrada são equipados com *Token Buckets* e controladores ARM para regular a taxa de entrada dos fluxos conformantes no domínio. Eles agregam o tráfego gerado por emissores TCP Reno a eles conectados e encaminham, todos, o tráfego agregado a um nó comum, que possui um enlace-gargalo.

Na topologia considerada, tanto os enlaces que ligam emissores TCP e roteadores de borda, quanto os que que ligam roteadores de borda ao roteador de núcleo, possuem todos capacidade de 100Mbps. Os roteadores de núcleo são interligados por um enlace de 155Mbps. Considerando que o tempo de propagação (T) definido no ponto de equilíbrio dos controladores é fixo e definido em 0.192 seg, o atraso em cada um desses enlaces foi devidamente configurado de modo a alcançar esse valor total.

Assim como em [12], foi utilizada neste experimento, uma rede 20% superprovida, com MGRs (*Minimum Guaranteed Rates*) iguais a $x_1 = 70.5Mbps$, $x_2 = 17Mbps$ e $x_3 = 42.5Mbps$. Como o tráfego de curta duração (web) já faz parte do misto de tráfego gerado pelo TrafficGen, não é estudada a inserção de um tráfego web transiente adicional.

Experimento 2

Neste experimento, foi avaliado o desempenho dos controladores em um ambiente de rede subprovido, onde a soma das MGRs ultrapassa a capacidade do enlace-gargalo.

O cenário utilizado é o mesmo do Experimento 1, com três roteadores de entrada e tráfego misto TCP gerado pelo TrafficGen. Assim como realizado em [12], as MGRs de todos os agregados são aumentadas em 20%, com relação ao valor utilizado no Experimento 1, fazendo com que a rede passe, então, a ser subprovida.

8.1.2 Experimentos com tráfego misto

No segundo grupo de experimentos, além de tráfego adaptativo, tanto de longa duração quanto de curta duração, foi utilizado também o tráfego não adaptativo. O cenário-base do primeiro bloco de experimentos, com 3 roteadores de entrada e um enlace-gargalo, foi mantido, e a ele foi adicionado um novo roteador de entrada. Esse roteador injeta no domínio *DiffServ* apenas tráfego não adaptativo UDP.

A opção por separar agregados de tráfego TCP e UDP é conveniente sobretudo na análise dos resultados, quando são coletadas informações específicas do tráfego de cada protocolo. Essa opção, todavia, não torna menos realista o cenário ao qual os mecanismos de AQM são expostos, uma vez que o roteador localizado na entrada do enlace-gargalo continua a agregar todo o tráfego gerado. Neste caso, composto por fluxos TCP e UDP.

O tráfego gerado pelo quarto agregado (UDP) pertence à classe de Serviço Assegurado de *DiffServ*. Sendo assim, os pacotes UDP ainda são marcados segundo a política do mecanismo ARM nos *Token Buckets* e estão sujeitas a marcação (ou descarte) por parte dos mecanismos de AQM do núcleo do domínio, da mesma forma que os pacotes TCP.

A inserção do tráfego UDP nas simulações é um ponto importantíssimo que, além de diferenciar o presente trabalho dos anteriores, avalia o controlador desenvolvido em ambiente realista, onde há presença de fluxos não adaptativos interferindo no sistema de controle de congestionamento da rede.

Por outro lado, apenas o controlador dsH2-AQM contou explicitamente em seu projeto com a influência do tráfego não adaptativo sobre os demais. Em tal projeto, o tráfego dessa natureza é expresso nas equações que descrevem o tráfego e o comportamento das filas nos roteadores. Apesar de na linearização das equações, sobre a qual se baseia o processo de síntese do controlador, o efeito do tráfego não adaptativo não ter sido considerado, sua influência é modelada na síntese do controlador, agora sob a forma de ruído. O dsH2-AQM foi projetado para obter um resultado ótimo global mesmo quando sob ruído de até 20% da capacidade do enlace-gargalo. Na prática, isso equivale a dizer que, mesmo sob efeito de fluxos não-adaptativos (UDP e outros protocolos) a 20% da capacidade do sistema, o controlador ainda estabiliza o sistema.

De forma sucinta, apenas o controlador dsH2-AQM foi projetado para suportar amostras de tráfego realistas para a classe de Serviço Assegurado de *DiffServ*. Nos testes a seguir, a fim de comparação, o controlador dsPI-AQM desenvolvido em [12] também é avaliado.

Experimentos 3 e 4

Nos experimentos 4 e 5, foi avaliado o comportamento dos controladores sob influência do tráfego não adaptativo UDP. Em todos os experimentos a seguir, foi utilizada a topologia descrita anteriormente.

Experimento 3

Neste experimento, o volume de dados gerado pelo quarto agregado é mantido constante enquanto as MGRs dos demais agregados são igualmente aumentadas em 20%, de forma a fazer com que a rede passe de superprovida a subprovida. Neste experimento, o desempenho dos controladores foi avaliado numa rede subprovida e sob influência de ruído gerado pelo tráfego UDP.

Experimento 4

Neste experimento, foi complementada a proposta do Experimento 3, realizando o aumento do ruído introduzido no sistema pelo tráfego UDP a um nível superior a 20%. Dessa forma, seria ultrapassado o limite estabelecido em sua modelagem, abaixo do qual o controlador garantiria desempenho ótimo. A proposta deste experimento é avaliar o comportamento do controlador sob outras situações além das quais para que fora projetado.

8.2 Métricas de desempenho

O tráfego agregado de cada roteador de entrada é composto por fluxos de longa duração (FTP) e de curta duração (HTTP).

São consideradas medidas que lidam tanto com o tráfego agregado quanto com as conexões individuais que o compõem.

Em relação ao roteador do enlace-gargalo, são coletadas a taxa de perda e tamanho médio da fila, considerando o tráfego total (agregado), sem distinção de protocolo, conexões individuais ou agregados.

Em relação às conexões individuais, são consideradas métricas como o *goodput* e a quantidade de *timeouts*. A ocorrência de *timeouts* nos fluxos TCP provoca uma grande variação nos níveis de atraso (*jitter*) por eles sofridos, o que pode comprometer a qualidade de serviço proporcionada ao tráfego e impossibilitar a garantia de banda. Sendo assim, a minimização do *jitter* é um fator importante nos ambientes de rede com suporte a Qualidade de Serviço.

Os gráficos para essas métricas são gerados em função da carga na rede, que é um dos parâmetros repassados ao gerador de tráfego TrafficGen em sua instanciação. Nesses experimentos, varia-se a carga, razão entre a taxa de emissão gerada e a capacidade do enlace-gargalo, entre 40% e 100%.

8.3 Scripts de simulação

A utilização do TrafficGen simplifica em muito a tarefa de simulação, uma vez que o estabelecimento e finalização de conexões é gerida automaticamente. Sendo assim, nos scripts de simulação, faz-se necessário apenas configurar a topologia da rede e as características gerais do tráfego gerado pelo TrafficGen, como descrito na Seção 8.1.

Como saída, o TrafficGen gera arquivos de log contendo todas as informações sobre as estatísticas de cada conexão estabelecida (medidas do ponto de vista do emissor). São exibidas, por exemplo, medidas como o momento de início e término da conexão, banda consumida, número de timeouts, goodput, RTT, tamanho da janela final etc.

Já as medidas que lidam com o tráfego agregado no roteador do enlace-gargalo devem ser coletadas separadamente e o código para coleta deve ser manualmente inserido nos scripts de simulação.

Para a exibição dos resultados, o conteúdo dos arquivos de log é filtrado usando ferramentas específicas para o escaneamento de padrões, como a linguagem de processamento AWK [2]. Obtidos os dados, sua representação gráfica é feita a partir do software Gnuplot [58].

8.4 Análise das simulações

Nesta seção, são apresentados os resultados das simulações que comparam o desempenho dos mecanismos de AQM dsH2, dsPI e RIO (RED), nos experimentos descritos na Seção 8.1.

8.4.1 Experimento 1

Na Figura 8.4.1, pode-se observar a variação do tamanho da fila em função da carga na rede. Nota-se que o comprimento médio da fila aumenta de acordo com a carga, o que não acontece com RIO (RED), que mantém um comprimento de fila praticamente constante à medida que a carga da rede varia. Isso acontece porque RIO tem como objetivo manter o comprimento da fila dentro de limiares constantes ao longo do tempo. Já dsH2 e dsPI permitem uma certa flutuação do comprimento da fila de acordo com a variação de carga na rede, visando mantê-la em níveis que possibilitem um melhor aproveitamento dos

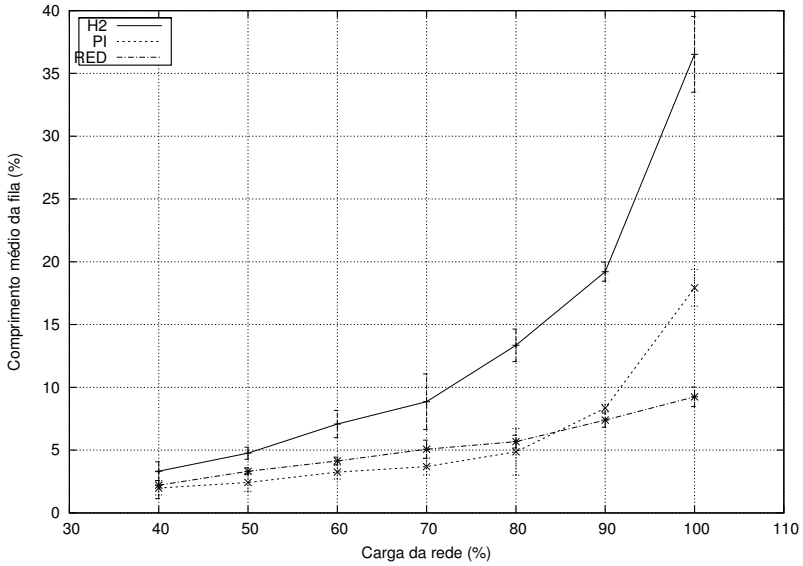

Figura 8.1: Comprimento da fila

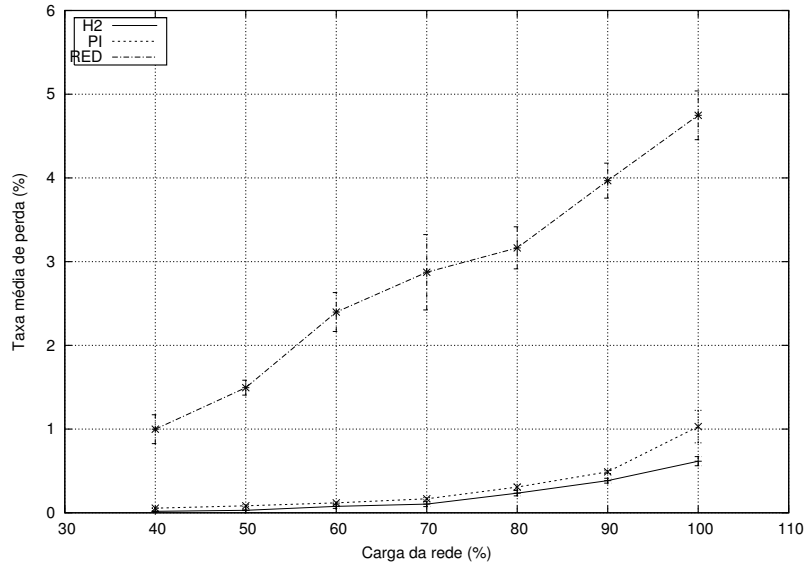

Figura 8.2: Taxa média de perda

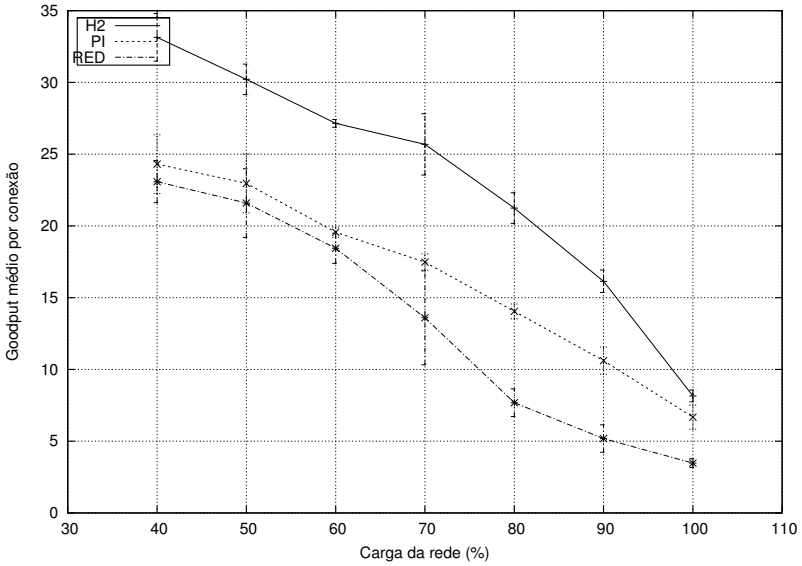

Figura 8.3: Goodput Médio por conexão

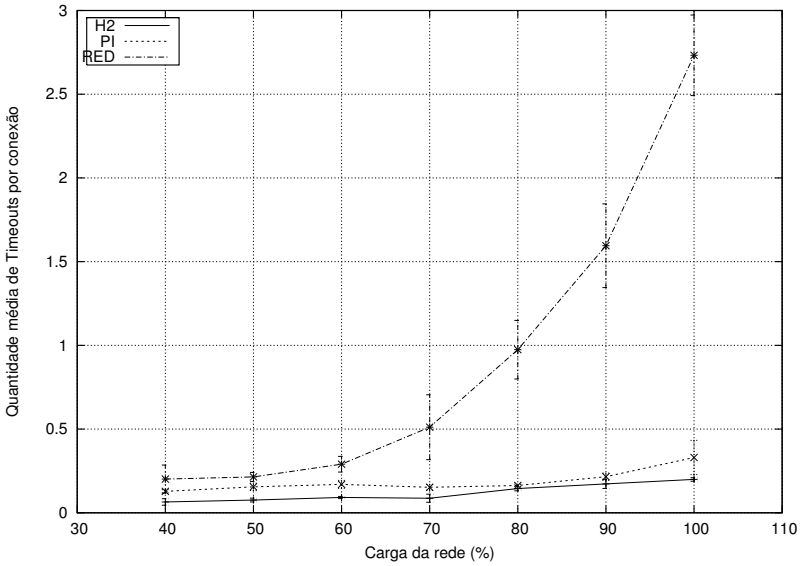

Figura 8.4: Número médio de *timeouts* por conexão

recursos disponíveis sem, no entanto, que isso implique em altos valores para as taxas de descarte, como pode-se constatar na Figura 8.2.

Em consequência disso, RIO subutiliza os recursos da rede, em termos de banda passante. Apesar de o uso de dsH2 sempre acarretar em um maior comprimento de fila, esse valor ainda é considerado baixo, correspondendo a, no máximo, 36% da capacidade total da fila.

Na Figura 8.2, observa-se a variação da taxa de perda nas filas em função da carga na rede. Nota-se a taxa de perda aumenta de acordo com a carga. Nota-se também que dsH2 consegue obter uma taxa de perda até 40% menor que dsPI, para carga total na rede, e de até 2800% com relação a RIO, para carga da rede em 70%. A alta taxa de perda experimentada por RIO provém do baixo grau de variação da fila por ele permitido. Por conta disso, sob alta carga na rede, a manutenção de baixos níveis de ocupação da fila é feita às custas do aumento da taxa de descarte. De forma oposta, dsH2 permite que o comprimento da fila se adeque à carga da rede, evita descartes desnecessários e melhora a utilização dos recursos da rede.

Na Figura 8.3, observa-se a variação do *goodput* obtido por cada conexão em função da carga na rede. Nota-se que o *goodput* diminui à medida que o congestionamento na rede aumenta. Por consequência de prover uma menor taxa de perda, o *goodput* obtido pelas conexões TCP quando usado o dsH2 chega a ser 37% maior que dsPI, para cargas de 40% do enlace e 43% maior que RIO, também para carga de 40% do enlace.

Já na Figura 8.4, nota-se que dsPI e dsH2 reduzem em muito o número de *timeouts* dos emissores TCP, com relação a RIO. Grande parte dos *timeouts* é provocada pela alta incidência de descartes sobre um mesmo fluxo de pacotes. Sendo assim, ao prover um menor nível de descarte de pacotes, dsH2 e dsPI conseguem também diminuir a ocorrência de *timeouts* sobre as conexões TCP. Para carga da rede em 100%, a melhoria de dsH2 em relação a dsPI chega a 10%. Uma consequência direta disso é que, ao se evitar a expiração dos emissores TCP, diminui-se o tempo total de transmissão dos arquivos através da rede.

8.4.2 Experimento 2

Na Figura 8.5, nota-se que os comprimentos médios da filas de dsH2 e dsPI mantém-se mais próximos, porém, ainda distantes do comprimento médio de fila proporcionado por RIO. Nessa figura, pode-se observar que dsH2 ainda proporciona um maior comprimento de fila, que pode chegar a 34% em relação a dsPI ou a 123% em relação a RIO, ambos sob carga total da rede. Como se trata de um ambiente subprovido, era realmente esperado que os comprimentos de fila aumentassem em relação aos do primeiro experimento, porém, o valor máximo de 57% de utilização ainda é considerado baixo para tais situações.

Na Figura 8.6, observa-se a variação da taxa de perda nas filas em função da carga

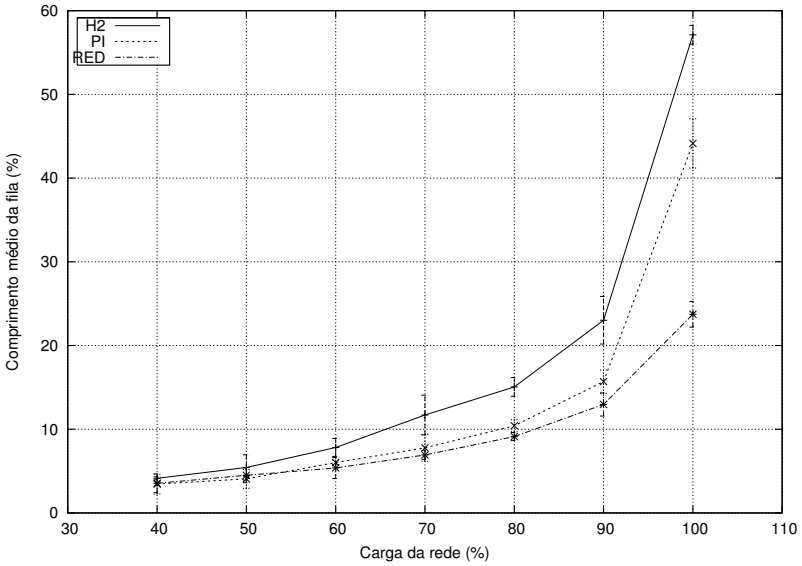

Figura 8.5: Comprimento da fila

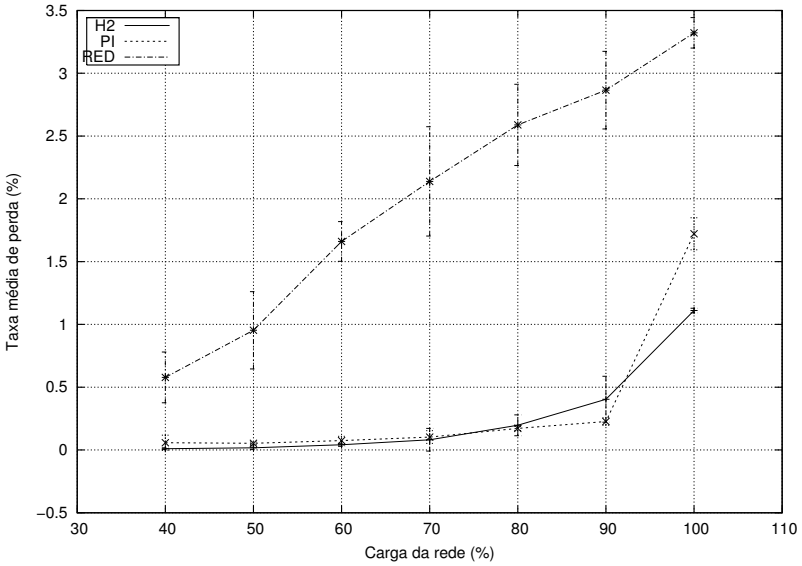

Figura 8.6: Taxa média de perda

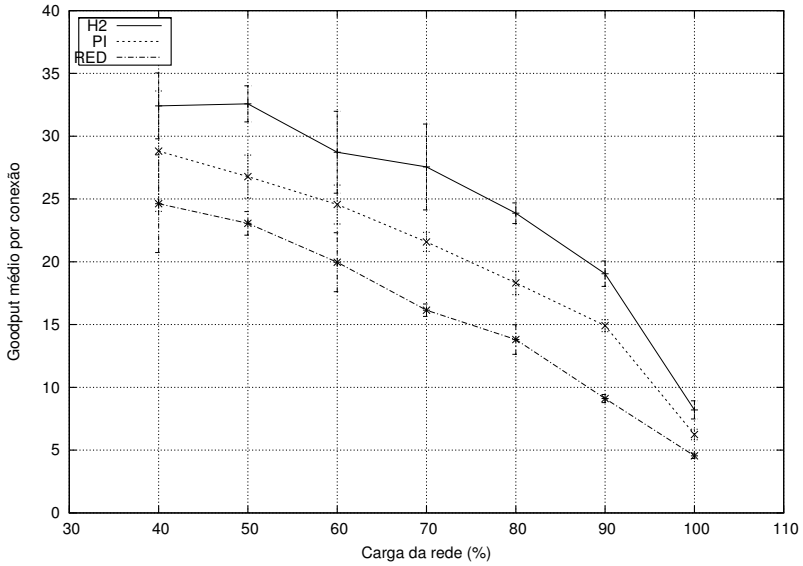

Figura 8.7: Goodput Médio por conexão

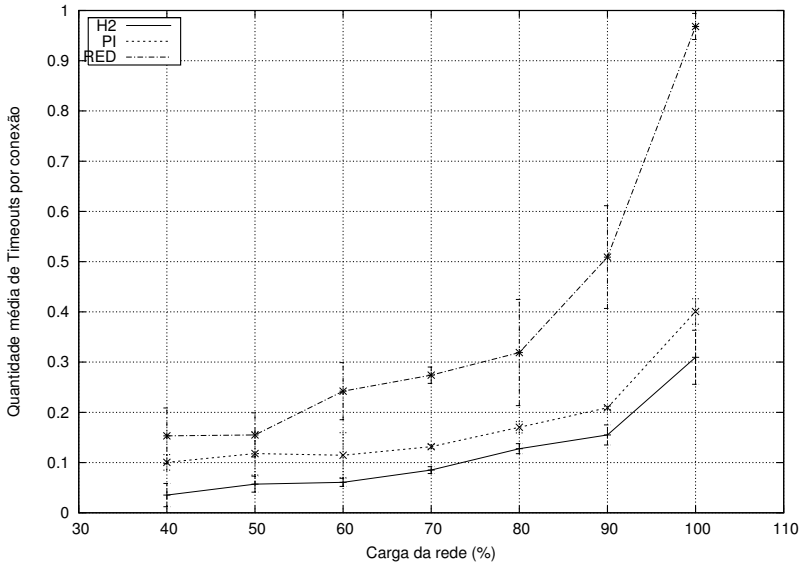

Figura 8.8: Número médio de *timeouts* por conexão

na rede. Nota-se que dsH2 consegue manter um nível de descarte mais baixo ou igual a dsPI e RIO mesmo sob condições de subprovisão. Para a carga máxima da rede, dsH2 consegue prover uma taxa de perda até 60% menor que dsPI e até 300% menor que RIO. Tal vantagem de dsH2 é obtida pela manutenção de um comprimento médio da fila alto, que torna possível a absorção dos fluxos *in*. Assim, evita-se que estes sejam descartados, mesmo em se considerando um ambiente de rede subprovido, o que é um dos objetivos fundamentais desse controlador. Ao evitar descartes desnecessários, dsH2 melhora a utilização de recursos da rede nesse cenário.

Na Figura 8.7, observa-se que o *goodput* obtido individualmente pelas conexões mantém-se muito próximo quando se considera ambiente subprovidos. dsH2 proporciona uma melhoria de cerca de 16% no *goodput* com relação a dsPI e de 40% em relação a RIO, para cargas entre 50 e 90% da rede.

Já na Figura 8.8, nota-se que dsPI e dsH2 reduzem em muito o número de *timeouts* dos emissores TCP, com relação a RIO. Além disso, para quaisquer cargas da rede, a melhoria de dsH2 em relação a dsPI gira em torno de 50 a 60%. Já em relação a RIO, esse ganho fica em torno de 250%. Dado que o ambiente de rede considerado não modela a ocorrência de perdas não causadas por descarte explícito de pacotes no roteador, a ocorrência de *timeouts* de dsPI e RIO é provocado exclusivamente por tais descartes. Sendo assim, o alto índice de *timeouts* por conexão reflete a penalização excessiva de determinados fluxos TCP, com relação ao descarte de pacotes.

8.4.3 Experimento 3

Na Figura 8.9, nota-se que os comprimentos médios da filas de RIO e dsPI mantém-se muito próximos, porém muito distantes do comprimento médio de fila proporcionado por dsH2, que chega a ser 75% mais alto para carga de 100% na rede. De forma geral, todos os AQMs proporcionaram comprimentos de fila notavelmente mais baixos, reflexo da inclusão de tráfego adicional UDP como ruído.

Na Figura 8.10, observa-se que as taxas médias de perda para dsH2 e dsPI são muito próximas nesse caso. Contudo, dsH2 provê uma distribuição de descartes mais distribuída, o que evita o acontecimento de *timeouts*, como comprovado na Figura 8.12. DsH2 e dsPI se distanciam em cerca de 250% com relação a RIO, que possui níveis de perda muito maiores.

Na Figura 8.11, observa-se que o *goodput* obtido individualmente pelas conexões quando se emprega dsH2 chega a ser 38% maior que o de dsPI e 78% para RIO, ambos para carga de 80% (na realidade, 100%, se for somada a quantidade de ruído inserida nas simulações).

Apesar de as taxas de perda de dsH2 e dsPI serem próximas, como visto na Figura 8.10,

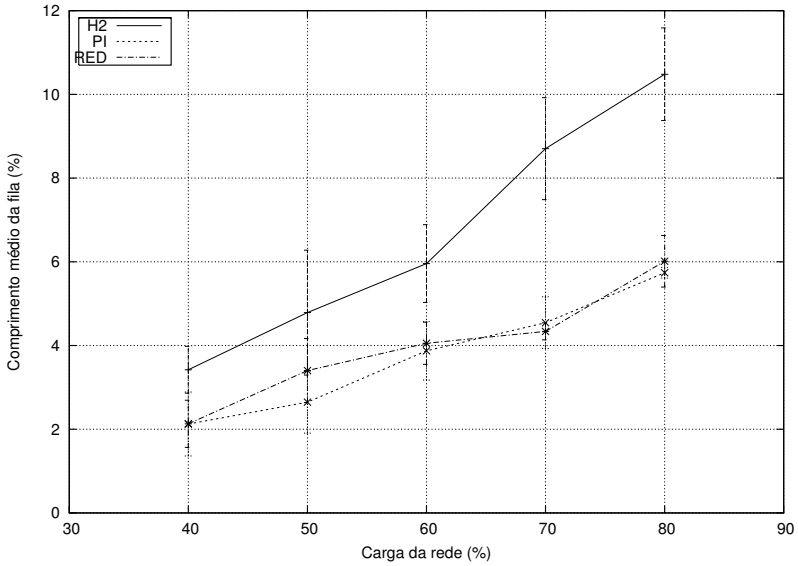

Figura 8.9: Comprimento da fila

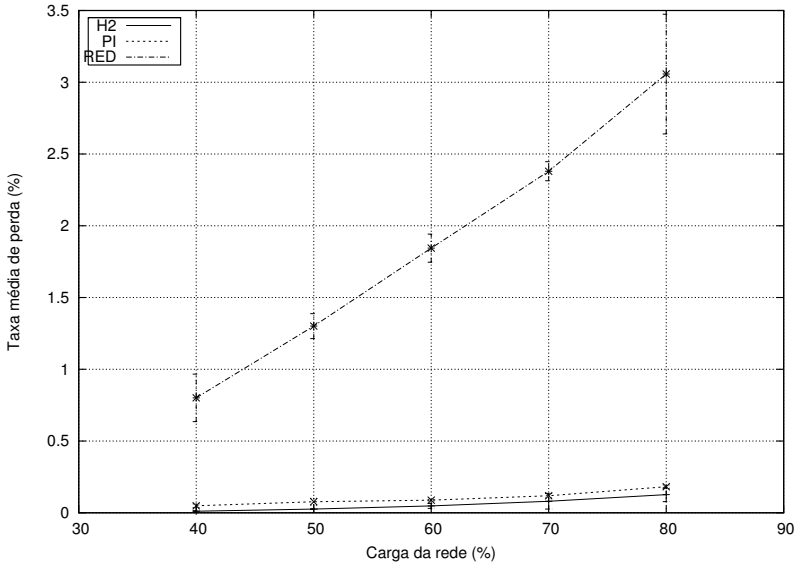

Figura 8.10: Taxa média de perda

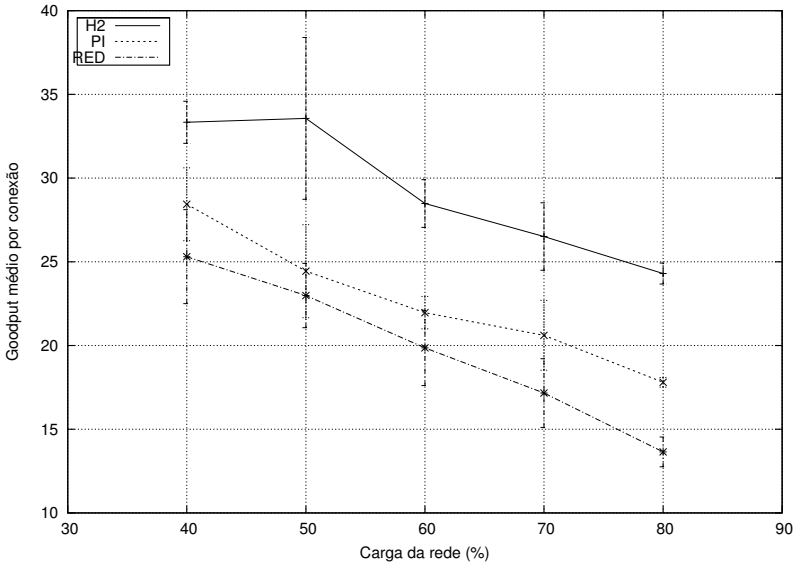

Figura 8.11: Goodput Médio por conexão

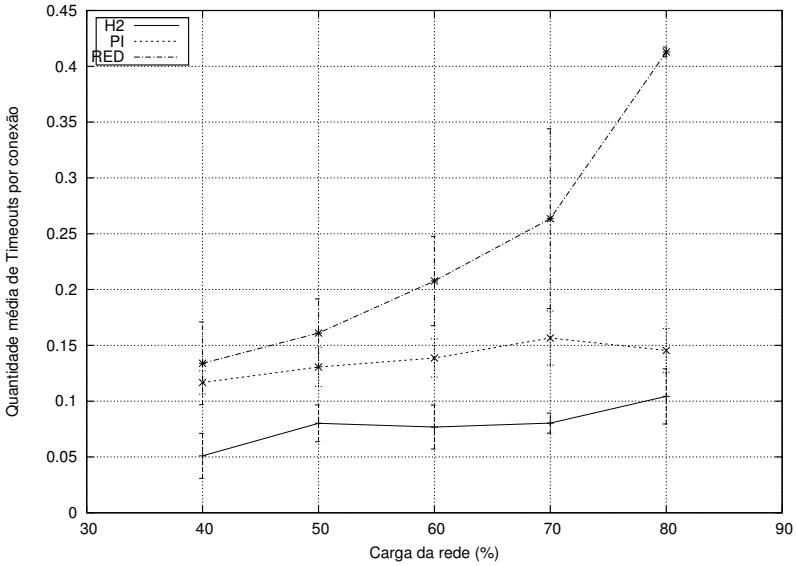

Figura 8.12: Número médio de *timeouts* por conexão

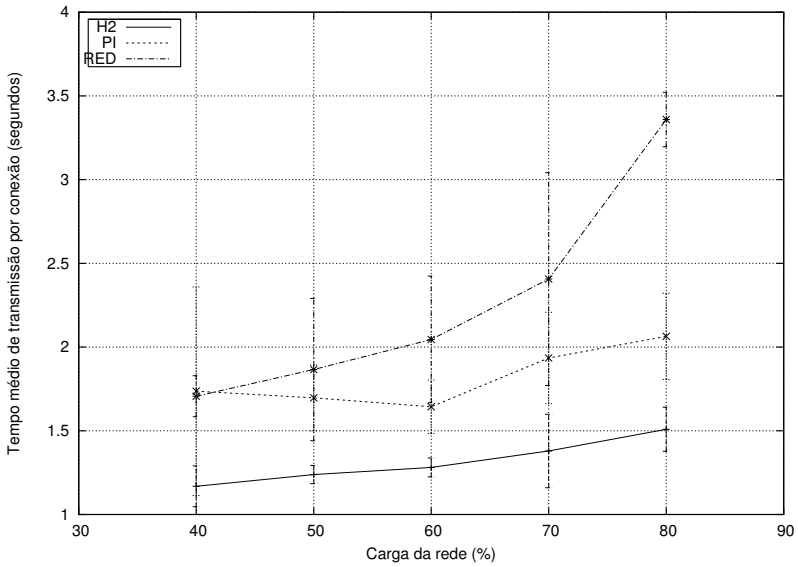

Figura 8.13: Tempo médio de transmissão por conexão

o cálculo do *goodput* considera ainda o tempo de transmissão da conexão pela rede. Por causa do menor número de *timeouts* (Figura 8.12), dsH2 obtém um menor tempo de transmissão (Figura 8.13).

8.4.4 Experimento 4

Na Figura 8.14, nota-se que o comprimento médio da fila de dsH2 chega a ser 100% maior que o de dsPI e 150% maior que o de RIO, para carga total na rede (70% de TCP e 30% de UDP). Contudo, pode-se reparar que esses valores representam uma porcentagem muito pequena da capacidade da fila, o que significa que o atraso inserido pelo enfileiramento dos pacotes sob a atuação dos três mecanismos de AQM é muito baixo diante do atraso de propagação da rede. Essa baixa utilização das filas é reflexo da inclusão de tráfego de ruído UDP em altas taxas (30%), o que provocou uma diminuição das taxas de transmissão dos fluxos TCP.

Na Figura 8.15, observa-se que a taxa média de perda para dsH2 e dsPI é extremamente próxima, ficando próxima a zero. Ambos, porém, se distanciam com relação a RIO, que possui níveis de perda muito maiores, chegando a cerca de 2% para carga total na rede.

Na Figura 8.16, observa-se que o *goodput* obtido individualmente pelas conexões quando se emprega dsH2 chega a ser 31% maior que o de dsPI e 52% para RIO, ambos para carga total da rede.

Apesar de a taxa de perda de dsH2 e dsPI ser próxima, como na simulação do cenário 3.1, o AQM dsH2 proporciona um ganho no *goodput* porque o cálculo desse valor considera

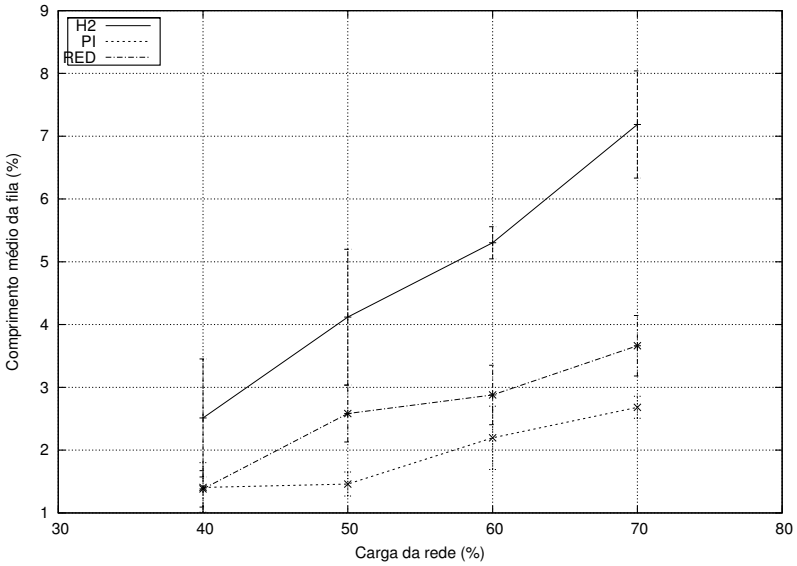

Figura 8.14: Comprimento da fila

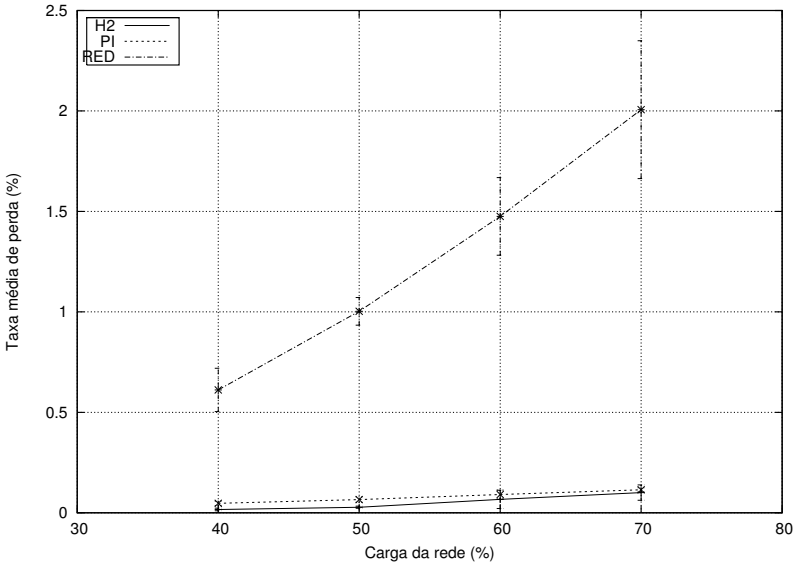

Figura 8.15: Taxa média de perda

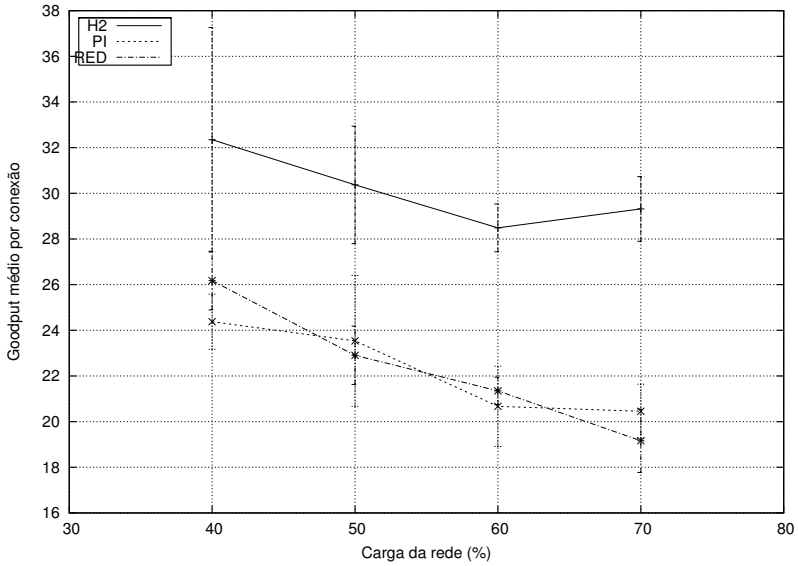

Figura 8.16: Goodput Médio por conexão

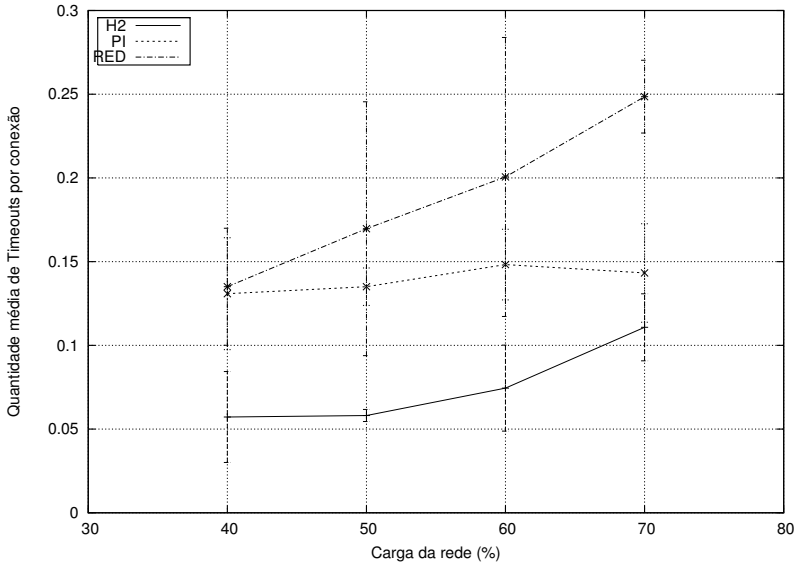

Figura 8.17: Número médio de *timeouts* por conexão

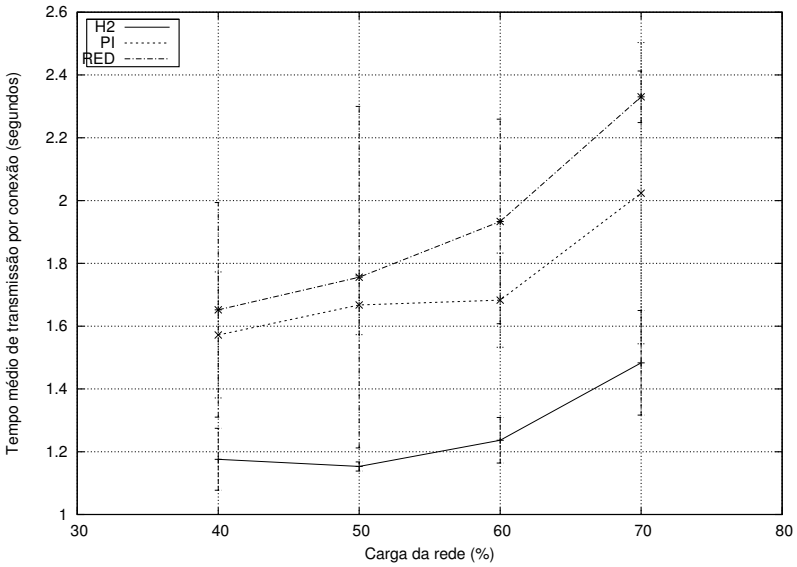

Figura 8.18: Tempo médio de transmissão por conexão

o tempo de transmissão da conexão pela rede. Da mesma forma que no caso anterior, por causa do menor número de *timeouts* (vide Figura 8.17), dsH2 obtém um menor tempo de transmissão (vide Figura 8.18. Assim, há uma melhora no *goodput* médio.

Capítulo 9

Conclusões e trabalhos futuros

Neste capítulo, são apresentadas as principais contribuições deste trabalho e apontados alguns possíveis trabalhos futuros.

9.1 Resultados e contribuições

Nesta dissertação, foi abordado o uso de mecanismos de AQM para a provisão de garantias de banda passante ao tráfego da classe AF-4 (Serviço Assegurado) de *DiffServ*. Em [12], foi proposta uma arquitetura, baseada em estratégias de marcação e diferenciação de pacotes, capaz de oferecer tais garantias. Essas tarefas são realizadas individualmente por dois controladores, um para regular a marcação (coloração) e outro para promover o gerenciamento ativo da filas nos roteadores de núcleo, proporcionando uma diferenciação de tratamento entre tais classes de pacotes e prevenindo a ocorrência de congestionamentos. Em [11], foram propostos o controlador ARM, para coloração, e o controlador dsPI-AQM, para o gerenciamento ativo de filas, ambos controladores *Proportional Integral*, desenvolvidos a partir da Teoria de Controle Moderno.

Neste trabalho, foi desenvolvido um controlador ótimo *non-overlapping* para gerenciamento ativo de filas na classe de Serviço Assegurado de *DiffServ*, de acordo com as especificações da arquitetura proposta. Utilizando uma modelagem do comportamento do tráfego mais precisa que a utilizada em [11], o controlador dsH2-AQM supera as limitações de desempenho de dsPI-AQM, provendo um maior *goodput*, menores taxas de perda e número de RTOs dos emissores TCP.

Foi desenvolvida uma modelagem que inclui detalhadamente o comportamento do tráfego da classe de Serviço Assegurado de *DiffServ*, incluindo o efeito de fluxos não-adaptativos, como UDP, sobre a atuação do controlador AQM. Dessa forma, ao contrário dos demais mecanismos de AQM propostos, dsH2-AQM foi o único mecanismo a incluir, desde a sua modelagem, a atuação de tráfego não-TCP sobre o sistema modelado. Essa

é, inclusive, uma necessidade apontada por estudos sobre as características do tráfego da Internet [28], nos quais esse tipo de tráfego chega a representar um terço do total. Isso faz com que a modelagem aqui introduzida seja mais realista em relação às demais.

Para a síntese do controlador dsH2-AQM, foi utilizada a Teoria de Controle Ótimo. Apesar de ter sido usada uma abordagem não-racional em seu desenvolvimento, o controlador obtido é racional, já que foi possível cancelar em sua síntese os termos referentes ao atraso. Isso configura o alcance da estratégia global ótima para solucionar o problema da minimização da norma H_2.

Para o desenvolvimento do controlador, foi investigado um ponto de equilíbrio para o sistema que apresentasse condições de rede realistas, baseadas em recentes medições sobre o tráfego da Internet, e favoráveis ao oferecimento de garantias de banda passante. O controlador dsPI-AQM foi desenvolvido sobre pontos de equilíbrio com parâmetros injustificados. Para efeito de uma comparação justa, esse controlador foi rederivado com base no mesmo ponto de equilíbrio. Esses controladores foram discretizados sob uma freqüência conveniente, permitindo que eles atuem de forma apropriada, não muito severa, mesmo sob condições de variação abruptas na rede.

Na avaliação experimental realizada, o desempenho de dsH2-AQM é comparado ao de dsPI-AQM e RIO sob as mesmas condições de rede. Nessas simulações, comprovou-se a eficiência de dsH2-AQM em prover altos valores de *goodput* e diminuição do número de RTOs dos emissores TCP. Inicialmente, foram realizados experimentos envolvendo tráfego exclusivamente adaptativo, ambiente para o qual os demais controladores foram projetados. Em tais ambientes, dsH2-AQM proporcionou *goodput* cerca de 40% maiores que dsPI-AQM e RIO. Por ter sido o único a modelar a influência de tráfego não-adaptativo sobre o sistema, dsH2-AQM apresentou melhor desempenho também em tais situações. Apesar de a modelagem de todos considerar apenas a existência de conexões TCP de longa duração, em todos os experimentos foi considerada também tráfego TCP de curta duração, como o tráfego web, que em ambientes de redes reais chega a representar cerca de dois terços de todo o tráfego TCP.

9.2 Trabalhos futuros

Apesar de os cenários de simulação aqui tratados considerarem amostras realistas de tráfego, simulações com cenários envolvendo múltiplos enlaces-gargalo e diferentes níveis de RTT entre os emissores podem trazer resultados interessantes. Com o mesmo objetivo, podem ainda ser consideradas simulações envolvendo pacotes IP com diferentes tamanhos.

Em trabalhos futuros, podem ser consideradas modelagens mais precisas sobre o comportamento dos emissores TCP, incorporando a influência de RTOs e a execução do algoritmo de Partida Lenta, nesta dissertação ignorado devido a sua baixa influência em

conexões de longa duração. Como amostras de tráfego realista consideram em grande parte o tráfego de curta duração, a influência do mecanismo de Partida Lenta na modelagem do controlador pode trazer resultados mais precisos em tais ambientes.

Com a sedimentação dos estudos sobre a arquitetura proposta para provisão de garantias mínimas de banda na arquitetura *DiffServ*, outros trabalhos podem ser desenvolvidos a fim de comparar o desempenho dos mecanismos de AQM, como dsH2-AQM, dsPI-AQM e RIO, quando sujeitos a mecanismos de Condicionamento de Tráfego alternativos ao ARM, como o VS-ACT [13], cuja adaptação às restrições *full color* de [12] ainda não foram estudadas pelos seus respectivos autores. Assim, pode-se verificar tanto a melhoria de desempenho local, na coloração de pacotes, oferecida por tais controladores quanto o desempenho global, oferecido pela utilização conjunta de várias combinações de diferentes mecanismos de CT e AQM.

Outra possibilidade seria a de se implementar os controladores AQM de forma adaptativa, de forma que estes tivessem sua sensibilidade ajustada de acordo com a intensidade do congestionamento, determinada pelo comprimento da fila. Isso permitiria aos controladores AQM uma ação mais adequada, nem excessivamente reativa nem passiva, de acordo com as variações de estado da rede.

Bibliografia

[1] D. Agrawal, F. Granelli, and N. L. S. da Fonseca. Integrated arm/aqm mechanisms based on pid controllers. In *IEEE International Conference on Communications (ICC), 2005*, volume 1, pages 6–10, May 2005.

[2] A. V. Aho, B. W. Kernighan, and P. J. Weinberger. The awk programming language. `http://cm.bell-labs.com/cm/cs/awkbook/index.html`, 2006.

[3] M. Allman, V. Paxson, and W. Stevens. TCP congestion control. RFC 2581, April 1999. URL: `http://www.ietf.org/rfc/rfc2581.txt`.

[4] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes un web client acess patterns: characteristics and caching implications. Technical Report 1998-23, Boston University, 4 1998.

[5] Shao Liu Basar and T. Srikant. Controlling the Internet: A survey and some new results. In *Proceedings of 42nd IEEE Conference on Decision and Control*, volume 3, pages 3048– 3057, 12 2003.

[6] Florian Baumgartner, Torsten Braun, and Pascal Habegger. Differentiated services: A new approach for quality of service in the internet. In *HPN*, pages 255–273, 1998.

[7] S. Blake, D. Black, M.Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, December 1998. URL: `http://www.ietf.org/rfc/rfc2475.txt`.

[8] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics Press, 1994.

[9] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. RFC 1633, June 1994. URL: `http://www.ietf.org/rfc/rfc1633.txt`.

[10] K. Cardoso and J. de Rezende. Http traffic modelling: development and application. In *International Telecommunications Symposium – ITS 2002, Natal, Brazil*, 2002.

[11] Y. Chait, C. V. Hollot, V. Misra, D. Towsley, H. Zhang, and J. C. S. Cui. Providing throughput differentiation for TCP flows using adaptative two-color marking and two-level AQM. In *INFOCOM'02, New York, NY, USA*, June 2002.

[12] Y. Chait, C. V. Hollot, V. Misra, D. Towsley, H. Zhang, and J. C. S. Cui. Throughput differentiation using coloring at the network edge and preferential marking at the core. *IEEE/ACM Transactions on Networking*, 13(4):743–754, August 2005.

[13] X. Chang and J. K. Muppala. On improving bandwidth assurance in AF-based diffserv networks using a control theoretic approach. *Computer Networks*, 49(6):816–839, 2005.

[14] Wu chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang S. Shin. Adaptive packet marking for providing differentiated services in the internet. Technical Report CSE-TR-347-97, 16 1997.

[15] D. D. Clark. The Design Philosophy of the DARPA Internet Protocols. In *sigcom88*, pages 106–114, aug 1988.

[16] D. D. Clark and W. Fang. Explicit allocation of best effort packet delivery service. *ACM Transactions on Networking*, 6(4):362–373, August 1998.

[17] B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An expedited forwarding PHB (per-hop behavior). RFC 3246, March 2002. URL: `http://www.ietf.org/rfc/rfc3246.txt`.

[18] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998. URL: `http://www.ietf.org/rfc/rfc2460.txt`.

[19] W. Feng. *Improving Internet congestion control and queue management algorithms*. PhD thesis, PhD thesis, University of Michigan, USA, 1999.

[20] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin. A self-configuring RED gateway. In *IEEE INFOCOM'99, New York, NY*, volume 3, pages 1320–1328, March 1999.

[21] Victor Firoiu and Marty Borden. A study of active queue management for congestion control. In *INFOCOM*, pages 1435–1444, 2000.

[22] S. Floyd. RED: Discussions of setting parameters, 1997.

[23] S. Floyd. RED: Optimum functions for computing the drop probability, 1997.

[24] S. Floyd. Recommendation on using the gentle variant of RED, 2000.

[25] S. Floyd. Ns: Network simulator. `http://www.isi.edu/nsnam/ns`, 2004.

[26] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[27] Sally Floyd. Metrics for the evaluation of congestion control mechanisms. IETF Draft draft-floyd-transport-metrics-00.txt, May 2005.

[28] M. Fomenkov, K. Keys, D. Moore, and K. Claffy. Longitudinal study of internet traffic in 1998-2003. In *AICPS, Cancun, Mexico*, January 2004.

[29] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, 3a edição edition, 1994.

[30] M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain. Effect of number of drop precedences in assured forwarding. In *GlobeCom'99, Rio de Janeiro, Brazil*, December 1999.

[31] J. Heinanen, T. Finland, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding PHB group. RFC 2597, June 1999. URL: `http://www.ietf.org/rfc/rfc2597.txt`.

[32] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong. A control theoretic analysis of RED. In *IEEE INFOCOM'01, Anchorage, Alaska, USA*, pages 1510–1519, 2001.

[33] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong. On designing improved controllers for aqm router supporting TCP flows. In *IEEE INFOCOM'01, Anchorage, Alaska, USA*, pages 1726–1734, 2001.

[34] IETF. Internet engineering task force. `http://www.ietf.org`, March 2005.

[35] ITU-T. "One-way transmission time". Recommendation G.114, September 2003. URL: `http://www.itu.int`.

[36] Ki Baek Kim, Ao Tang, and Steven H.Low. Design of AQM in supporting TCP based on the well-known AIMD model. In *Proceedings of IEEE Globecom 2003*, San Francisco, California, USA, 12 2003.

[37] Anurag Kumar. Comparative performance analysis of versions of TCP in a local network with a lossy link. *IEEE/ACM Transactions on Networking*, 6(4):485–498, 1998.

[38] K. R. Renjish Kumar, Akkihebbal L. Ananda, and Lillykutty Jacob. A memory-based approach for a tcp-friendly traffic conditioner in diffserv networks. In *ICNP*, pages 138–145, 2001.

[39] J. F. Kurose and K. W Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2004.

[40] P. Kuusela, P. Lassila, and J. Virtamo. Stability of TCP-RED congestion control. In *ITC-17, Salvador, Brazil*, pages 655–666, December 2001.

[41] M. M. E. Lima. *Projeto de Controladores Ótimos para Gerenciamento Ativo de Filas*. PhD thesis, PHD Thesis, Universidade Estadual de Campinas, Brazil, 2006.

[42] David G. Luenberger. *Introduction to Dynamic Systems: Theory, Models and Applications*. John Wiley & Sons, Stanford University, 1979.

[43] D. P. Farias M. C. de Oliveira and J. C. Geromel. LMI solver. `http://www.dt.fee.unicamp.br/~mauricio/software.html`, 2005.

[44] The MathWorks. Matlab 7. `http://www.mathworks.com/products/matlab/`, 2005.

[45] S. McCreary and K. Claffy. Trends in wide area IP traffic patterns - A view from ames Internet exchange. *Proceedings of the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling, Monterey, CA*, August 2000.

[46] V. Misra, W. B. Gong, and D. F. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *ACM SIGCOMM'00, Stockholm, Sweden*, pages 151–160, September 2000.

[47] Katsuhiko Ogata. *Engenharia de Controle Moderno*. LTC - Livros Técnicos e Científicos S.A., 3a edição edition, 2003.

[48] M. C. Oliveira and J. C. Geromel. Synthesis of non-rational controllers for linear delay systems. *Automatica*, 40(2):171–188, February 2004.

[49] T. Ott, J. Kemperman, and M. Mathis. The stationary behavior of ideal TCP congestion avoidance, 1996. URL: `ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps`.

[50] J. Postel. "Internet Protocol". RFC 791, September 1981. URL: `http://www.ietf.org/rfc/rfc0791.txt`.

[51] J. Postel. Transmission control protocol. RFC 793, September 1981. URL: `http://www.ietf.org/rfc/rfc0793.txt`.

[52] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, September 2001. URL: `http://www.ietf.org/rfc/rfc3168.txt`.

[53] J. Rezende. Trafficgen. `http://www.gta.ufrj.br/~rezende`, 2005.

[54] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu. On achievable service differentiation with token bucket marking for TCP. In *ACM SIGMETRICS'00, Santa Clara, CA, USA*, pages 23–33, June 2000.

[55] S. Shakkottai, R. Srikant, N. Brownlee an A. Broido, and kc Claffy. "The RTT distribution of TCP flows in the Internet and its impact on TCP based flow control". CAIDA paper, February 2004. URL: `http://www.caida.org/outreach/papers/2004/tr-2004-02/`.

[56] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. RFC 2212, September 1997. URL: `http://www.ietf.org/rfc/rfc2212.txt`.

[57] P. White. Rsvp and integrated services in the internet: A tutorial. *IEEE Communication Magazine*, pages 100–106, May 1997.

[58] T. Williams and C. Kelley. Gnuplot. `http://www.gnuplot.info/`, 2006.

[59] J. Wroclawski. Specification of the controlled-load network element service. RFC 2211, September 1997. URL: `http://www.ietf.org/rfc/rfc2211.txt`.

[60] J.K. XiaoLin Chang; Muppala. Adaptive marking threshold for improving bandwidth assurance in a differentiated services network. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 6, pages 3073–3077, 2003.

[61] Ikjun Yeom and A. L. Narasimha Reddy. Modeling TCP behavior in a differentiated services network. *IEEE ACM Transactions on Networking*, 9(1):31–46, 2001.