

## Uma variação do protocolo DCCP para Redes de Alta Velocidade

**Carlos Augusto Froidi**

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

Campinas, 11 de Novembro de 2011.



Prof. Dr. Nelson Luis Saldanha da Fonseca

Instituto de Computação, Unicamp

(Orientador)

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

FICHA CATALOGRÁFICA ELABORADA POR  
MARIA FABIANA BEZERRA MÜLLER - CRB8/6162  
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E  
COMPUTAÇÃO CIENTÍFICA - UNICAMP

|       |  |
|-------|--|
| F925v | <p>Froldi, Carlos Augusto, 1979-<br/>Uma variação do protocolo DCCP para redes de alta<br/>velocidade / Carlos Augusto Froldi. - Campinas, SP :<br/>[s.n.], 2011.</p> <p>Orientador: Nelson Luis Saldanha da Fonseca.<br/>Dissertação (mestrado) – Universidade Estadual de<br/>Campinas, Instituto de Computação.</p> <p>1. Sistemas multimídia. 2. DCCP (Protocolo de rede<br/>de computação. 3. Redes de alta velocidade.<br/>4. Protocolos de transporte. 5. Redes de computadores.<br/>I. Fonseca, Nelson Luis Saldanha da, 1963-.<br/>II. Universidade Estadual de Campinas. Instituto de<br/>Computação. III. Título.</p> |
|-------|--|

Informações para Biblioteca Digital

**Título em inglês:** A DCCP variant for high speed networks

**Palavras-chave em inglês:**

Multimedia systems

Datagram Congestion Control Protocol

Transport protocols

Computer networks

**Área de concentração:** Ciência da Computação

**Titulação:** Mestre em Ciência da Computação

**Banca examinadora:**

Nelson Luis Saldanha da Fonseca [Orientador]

Edmundo Roberto Mauro Madeira

André Costa Drummond

**Data da defesa:** 11-11-2011

**Programa de Pós-Graduação:** Ciência da Computação

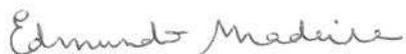
## TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 11 de novembro de 2011, pela Banca examinadora composta pelos Professores Doutores:



---

**Prof. Dr. André Costa Drummond**  
DCC / UnB



---

**Prof. Dr. Edmundo Roberto Mauro Madeira**  
IC / UNICAMP



---

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

# Uma variação do protocolo DCCP para Redes de Alta Velocidade

**Carlos Augusto Froldi**

Novembro de 2011

## **Banca Examinadora:**

- Prof. Dr. Nelson Luis Saldanha da Fonseca  
Instituto de Computação, Unicamp (Orientador)
- Prof. Dr. Edmundo Roberto Mauro Madeira  
Instituto de Computação, Unicamp
- Prof. Dr. André Costa Drummond  
Departamento de Ciência da Computação, Universidade de Brasília
- Prof. Dr. Daniel Macêdo Batista  
Instituto de Matemática e Estatística, Universidade de São Paulo (Suplente)
- Profa. Dra. Islene Calciolari Garcia  
Instituto de Computação, Unicamp (Suplente)

# Resumo

Os protocolos da camada de transporte na Internet, TCP e UDP, não oferecem serviços para transmissão eficiente de fluxos multimídia, porém, este último é adotado com maior frequência por essas aplicações. Uma proposta de um novo protocolo da camada de transporte, chamado DCCP, foi elaborada para atender a demanda das aplicações multimídia e substituir o protocolo UDP. A presente dissertação propõe uma variante deste protocolo, chamada Fast DCCP, para operar de maneira eficiente em redes de alta velocidade. A nova variante foi avaliada através de experimentos de simulação, utilizando o NS-2 e experimentos de medição, utilizando o sistema operacional Linux.

# Abstract

The Internet transport layer protocols, TCP and UDP, do not provide efficient transport service for multimedia streams. UDP is usually used for these applications, due to its low overhead. A new transport layer protocol, called DCCP, was proposed to meet the demand of multimedia applications, aiming at replacing the UDP protocol. This dissertation will propose a variant for this protocol, called Fast DCCP, for operating efficiently on high-speed networks. It was evaluated by simulation using the NS-2 network simulator and measurements of the FAST DCCP protocol operation in the Linux operating system.

# Agradecimentos

Gostaria de agradecer em primeiro lugar a Deus por mais esta conquista, suas bênçãos e graças desde sempre me capacitam a fazer aquilo que está além de minhas habilidades. Agradeço também aos meus saudosos pais, Marco Antonio Frolidi e Salma Bernardino da Costa Frolidi, por tudo o que fizeram por mim e por nunca medir esforços para prover aos seus filhos aquilo que julgavam ser primordial e digno para suas vidas, mais do que justo dedicar esta conquista a eles. Agradeço a minha esposa, Fabiane Filetti Frolidi, pelo apoio incondicional, pelo carinho e pela compreensão durante todo o processo de desenvolvimento do trabalho de mestrado, de outra forma teria sido impossível chegar ao final do mesmo.

Gostaria de agradecer aos amigos e funcionários do IC que me apoiaram durante os trabalhos no mestrado: Jerônimo, Furlan, Éric, Miguel, Edelson, Daniel Capeleto, Flávio, Ademílson, Michel e Fernandinho. Agradecimentos especiais ao William, Roland e a Camila pelo apoio e pela inestimável ajuda com os trabalhos.

Agradecimentos especiais aos amigos com quem tive o prazer de conviver de maneira mais próxima e que sempre me ajudaram muito na realização dos trabalhos (além de serem grandes amigos): André Drummond, Daniel Batista, Jorge Lima, Diego Solarte, Joana Gonzales Malaverri, Luiz Bittencourt, Ariel Vargas, Ricardo dos Santos, Fábio Bezerra, André Atanásio, Evandro Bracht, Rodrigo Minetto, Juliana Santi, Juliana Freitag, Fabiano Costa, Cleo Bila, Gustavo Figueiredo, Cláudio Carvalho, Daniel Manzato, Neumar Malheiros e Rafael Pasquini. A todos os amigos do LRC, com quem tive o prazer de conviver durante os últimos 5 anos. Agradecimentos especiais ao Daniel Batista, André Drummond e Jorge Lima por terem me ajudado de maneira mais próxima, tirando dúvidas do latex, ajudando a resolver bugs do NS-2, ajuda para elaborar scripts para processar resultados, etc, etc, etc.

Gostaria de agradecer aos professores Flávio Miyazawa e Maria Cecília Baranauskas, que foram coordenadores da CPG, pelo apoio e incentivo durante o meu ingresso no programa de pós-graduação do IC. Gostaria de agradecer aos professores com os quais tive prazer de ter aula: Prof. Nelson Fonseca, Prof. Edmundo Madeira, Profa. Maria Beatriz Toledo, Profa. Cecília Rubira e Profa. Anamaria Gomide. Agradecimentos

especiais ao Prof. Ricardo Dahab e a Profa. Christiane Campos pelo apoio incondicional durante os trabalhos do mestrado, o carinho e apoio que vocês demonstraram por mim foi essencial para realização dos trabalhos. Agradecimentos aos professores que apoiaram e deram uma importante contribuição durante o período em que estive no mestrado: Prof. Jorge Stolfi, Prof. Paulo Centoducatte, Prof. Neucimar Leite, Prof. Pedro Rezende, Prof. Alexandre Falcão, Prof. Cid Souza, Profa. Célia Mello, Prof. Luiz Buzato e prof. Zanoni Dias. Gostaria de agradecer ao Prof. Edmundo por ter me encorajado a ingressar no mestrado e por estar sempre a disposição para ajudar. Agradeço de maneira especial ao meu orientador, Prof. Nelson Fonseca, pelo convite para ser seu aluno e pelo voto de confiança, seus ensinamentos e seu apoio contribuíram muito para minha capacitação e irão ajudar muito na sequência da minha vida profissional.

Gostaria de agradecer a todos os meus amigos e familiares que me deram apoio incondicional para realização deste projeto.

*A memória dos meus pais, Marco Antonio Frolidi e Salma Bernardino da Costa Frolidi e  
a minha esposa Fabiane Filetti.*

# Acrônimos

**ACK** Acknowledgement

**AIMD** Additive Increase / Multiplicative Decrease

**BIC TCP** Binary Increase Congestion Transmission Control Protocol

**CBR** Constant Bit Rate

**CCID** Congestion Control Identifier

**DCCP** Datagram Congestion Control Protocol

**ECN** Explicit Congestion Notification

**FAST TCP** Fast Active Queue Management Scalable Transmission Control Protocol

**FTP** File Transfer Protocol

**HSTP** High-Speed Transport Protocol

**HTCP** High-Speed Transport Control Protocol

**IETF** Internet Engineering Task Force

**LDA** Loss-Delay Based Adaption Algorithm

**LPR** Linear Proportional Response

**MIMD** Multiplicative Increase / Multiplicative Decrease

**MTCP** Multicast Transmission Control Protocol

**NACK** Negative Acknowledgement

**NCA** Nominee-Based Congestion Avoidance

**PGMCC** Pragmatic General Multicast Congestion Control

**RAP** Rate Adaptation Protocol  
**RLA** Random Listening Algorithm  
**RTCP** Real-Time Transport Control Protocol  
**RTO** Retransmission Timeout  
**RTT** Round Trip Time  
**SACK** Selective Acknowledgment  
**STCP** Scalable Transmission Control Protocol  
**TCP** Transmission Control Protocol  
**TEAR** TCP Emulation at Receivers  
**TFRC** TCP-Friendly Rate Control  
**UDP** User Datagram Protocol

# Sumário

|   |            |
|---|------------|
| <b>Resumo</b>   | <b>v</b>   |
| <b>Abstract</b>   | <b>vi</b>  |
| <b>Agradecimentos</b>   | <b>vii</b> |
| <b>Acrônimos</b>  | <b>x</b>   |
| <b>1 Introdução</b>   | <b>1</b>   |
| 1.1 Contribuições da Dissertação . . . . .  | 2          |
| 1.2 Organização da Dissertação . . . . .  | 3          |
| 1.3 Publicações realizadas . . . . .  | 3          |
| <b>2 Mecanismos de Controle de Congestionamento</b>   | <b>4</b>   |
| 2.1 Controle de Congestionamento Amigável ao TCP . . . . .  | 4          |
| 2.2 Esquemas de Controle de Congestionamento baseados em Janela ou Taxa de Transmissão . . . . .                  | 5          |
| 2.3 Protocolos de Controle de Congestionamento Baseados em Taxa . . . . .   | 6          |
| 2.3.1 Rate Adaption Protocol (RAP) . . . . .  | 6          |
| 2.3.2 Loss-Delay Based Adaption Algorithm (LDA+) . . . . .  | 6          |
| 2.3.3 TCP-Friendly Rate Control Protocol (TFRC) . . . . .   | 6          |
| 2.3.4 TCP Emulation at Receivers (TEAR) . . . . .   | 7          |
| 2.4 Protocolos de Controle de Congestionamento Baseados em Janela . . . . .                                       | 7          |
| 2.4.1 Random Listening Algorithm (RLA) e Linear Proportional Response (LPR) . . . . .                             | 7          |
| 2.4.2 Multicast TCP (MTCP) . . . . .  | 8          |
| 2.4.3 Nominee-Based Congestion Avoidance (NCA) e Pragmatic General Multicast Congestion Control (PGMCC) . . . . . | 9          |

|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>Protocolos para Redes de Alta Velocidade</b>  | <b>10</b> |
| 3.1      | CUBIC TCP . . . . .  | 10        |
| 3.2      | FAST TCP . . . . .   | 11        |
| 3.3      | Scalable TCP (STCP) . . . . .  | 11        |
| 3.4      | Avaliação de Desempenho das variações do protocolo TCP para Redes de Alta Velocidade . . . . . | 12        |
| <b>4</b> | <b>Protocolo DCCP</b>  | <b>13</b> |
| 4.1      | Conexões DCCP . . . . .  | 13        |
| 4.2      | Formato dos Pacotes . . . . .  | 15        |
| 4.3      | Controle de Congestionamento TCP-Like (CCID 2) . . . . .                                       | 17        |
| 4.4      | Controle de Congestionamento TFRC (CCID 3) . . . . .   | 17        |
| 4.4.1    | Protocolo de Envio de Dados do TFRC . . . . .  | 18        |
| 4.4.2    | Cálculo da taxa de eventos de perda do TFRC . . . . .  | 21        |
| <b>5</b> | <b>Avaliação do protocolo DCCP em Redes de Alta Velocidade</b>                                 | <b>22</b> |
| 5.1      | Avaliação de desempenho . . . . .  | 23        |
| 5.1.1    | Escalabilidade . . . . .   | 24        |
| 5.1.2    | Justiça . . . . .  | 26        |
| 5.1.3    | Convergência . . . . .   | 27        |
| 5.1.4    | Compatibilidade com o TCP-Reno . . . . .   | 27        |
| 5.1.5    | Limitações conhecidas do TFRC . . . . .  | 28        |
| 5.1.6    | Resumo conclusivo . . . . .  | 29        |
| <b>6</b> | <b>Variação do protocolo DCCP</b>  | <b>35</b> |
| 6.1      | Propriedades Avaliadas . . . . .   | 37        |
| 6.2      | Avaliação de desempenho . . . . .  | 37        |
| 6.2.1    | Escalabilidade . . . . .   | 39        |
| 6.2.2    | Justiça . . . . .  | 41        |
| 6.2.3    | Convergência . . . . .   | 41        |
| 6.2.4    | Compatibilidade com o TCP-Reno . . . . .   | 42        |
| 6.2.5    | Resumo Conclusivo . . . . .  | 43        |
| <b>7</b> | <b>Conclusões e Trabalhos Futuros</b>  | <b>47</b> |
|          | <b>Bibliografia</b>  | <b>49</b> |

# Lista de Tabelas

|     |  |    |
|-----|--|----|
| 5.1 | Taxa de utilização obtida pelos fluxos TCP (NS-2)  | 28 |
| 5.2 | Taxa de utilização obtida pelos fluxos TCP (Linux) | 28 |
| 6.1 | Taxa de utilização obtida pelos fluxos TCP (NS-2)  | 43 |
| 6.2 | Taxa de utilização obtida pelos fluxos TCP (Linux) | 43 |

# Lista de Figuras

|      |  |    |
|------|--|----|
| 4.1  | Conexão DCCP . . . . .   | 15 |
| 4.2  | Cabeçalho do pacote DCCP . . . . .                                     | 16 |
| 4.3  | Pacote DCCP de Dados . . . . .   | 16 |
| 4.4  | Pacote DCCP de Confirmação . . . . .                                   | 16 |
| 5.1  | Topologia Dumbbell . . . . .   | 24 |
| 5.2  | Utilização do enlace em função da variação da banda (NS-2) . . . . .   | 25 |
| 5.3  | Utilização do enlace em função da variação da banda (Linux) . . . . .  | 26 |
| 5.4  | Utilização do enlace em função do número de conexões (NS-2) . . . . .  | 31 |
| 5.5  | Utilização do enlace em função do número de conexões (Linux) . . . . . | 31 |
| 5.6  | Primeiro Cenário de Justiça . . . . .                                  | 32 |
| 5.7  | Segundo Cenário de Justiça . . . . .                                   | 32 |
| 5.8  | Resultados da avaliação da convergência . . . . .                      | 33 |
| 5.9  | DCCP/TFRC x TCP-Reno (Vazão) . . . . .                                 | 33 |
| 5.10 | DCCP/TFRC x TCP-Reno (Perdas) . . . . .                                | 34 |
| 6.1  | Utilização do enlace em função da variação da banda (NS-2) . . . . .   | 39 |
| 6.2  | Utilização do enlace em função da variação da banda (Linux) . . . . .  | 40 |
| 6.3  | Utilização do enlace em função do número de conexões (NS-2) . . . . .  | 44 |
| 6.4  | Utilização do enlace em função do número de conexões (Linux) . . . . . | 44 |
| 6.5  | Primeiro Cenário de Justiça . . . . .                                  | 45 |
| 6.6  | Segundo Cenário de Justiça . . . . .                                   | 45 |
| 6.7  | Resultados da avaliação da convergência . . . . .                      | 46 |

# Capítulo 1

## Introdução

Nos últimos anos, houve um aumento considerável do número de aplicações multimídia na Internet, que são sensíveis ao atraso e que tem requisitos mínimos de taxas de transmissão. É extremamente desafiador fornecer a qualidade de serviço necessária a estas aplicações utilizando o tipo de serviço *Best Effort*, apesar do aumento significativo de banda nos enlaces da Internet. Dos protocolos, atualmente, empregados na camada de transporte, *Transmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP), este último tem sido adotado com mais frequência, uma vez que as aplicações multimídia são sensíveis ao atraso e a banda passante.

O aumento da utilização do protocolo UDP pelas aplicações multimídia pode colocar em risco o bom funcionamento da Internet, sobretudo em redes de alta velocidade, uma vez que este protocolo não efetua controle de congestionamento. Por outro lado, algumas funcionalidades do protocolo TCP, como por exemplo, o conceito de conexão, são interessantes para alguns tipos de aplicações multimídia. A proposta, realizada por um grupo de trabalho do IETF, de um novo protocolo para a camada de transporte, o *Datagram Congestion Control Protocol* (DCCP) [10], oferece o estabelecimento de serviços orientados a conexão para envio não confiável de fluxo de dados e controle de congestionamento. O protocolo fornece dois mecanismos distintos de controle de congestionamento, sendo ambos amigáveis ao protocolo TCP. Um dos controles, o *TCPLike* [8], é similar ao controle de janela implementado no TCP e o outro, o *TCP-Friendly Rate Control* (TFRC) [7] implementa o controle baseado em taxas. Para que este novo protocolo possa ser introduzido à Internet, para transmissão de fluxos multimídia, é necessário conhecer suas potencialidades e limitações.

Apesar de avaliações preliminares do protocolo DCCP terem sido realizadas em [34], [46], [18], [17], nenhuma delas avaliou o protocolo em redes de alta velocidade que tem grande disponibilidade de banda passante. O foco destes trabalhos foi a avaliação do protocolo DCCP para operação em redes sem fio e a compatibilidade do mesmo com o

protocolo TCP-Reno em tais redes ou em redes de baixa velocidade.

No entanto, os requisitos de mecanismo de controle de congestionamento em redes de alta velocidade diferem, significativamente, daqueles de redes de baixa velocidade. Como por exemplo, o protocolo TCP, amplamente utilizado na Internet, não consegue em redes de alta velocidade obter a banda passante disponível, em intervalo de tempo razoável [27], tornando-se o gargalo em transferência de dados nestas redes. Tal ineficiência motivou a proposição de uma série de variações do protocolo TCP, tais como *FAST TCP* [25] e *CUBIC TCP* [20]. No entanto, estas variações devem propor escalabilidade com novas taxas de transmissão assim como assegurar outras propriedades, tais como justiça interprotocolo e compatibilidade com o protocolo TCP-Reno, a versão mais popular na Internet.

O objetivo desta dissertação é avaliar o protocolo DCCP, quando este opera em redes de alta velocidade, para verificar suas potencialidades e limitações, a fim de propor possíveis mudanças no protocolo, de forma a garantir a operação eficiente deste em tais redes.

## 1.1 Contribuições da Dissertação

A presente dissertação contribui com uma avaliação de desempenho do funcionamento do protocolo DCCP, quando este opera com o controle de congestionamento TFRC, em redes de alta velocidade. Tal avaliação é de suma importância dado que o mecanismo de controle de congestionamento do DCCP foi projetado para ser compatível com o do TCP-Reno e, conseqüentemente, pode ter as mesmas ineficiências da operação deste último em redes de alta velocidade. A avaliação realizada permitiu verificar deficiências do protocolo DCCP, o que confirma as hipóteses de que o TFRC apresenta deficiências também observadas no mecanismo de controle de congestionamento do TCP-Reno.

Nesta avaliação, utilizou-se tanto simulação, tendo como ferramenta o simulador NS-2 [35], quanto medições, utilizando a implementação do protocolo DCCP presente no sistema operacional Linux. Diferentes cenários foram elaborados para a avaliação da escalabilidade, da convergência, da justiça e da compatibilidade com o TCP-Reno. Todos os cenários seguem as recomendações para análise dos mecanismos de controle de congestionamento em novos protocolos propostas em [9] e [1].

Outra importante contribuição desta dissertação é a proposta de uma variação do protocolo DCCP e do mecanismo de controle de congestionamento TFRC, intitulada *Fast DCCP*, que visa sanar as ineficiências observadas na operação do protocolo DCCP em redes de alta velocidade, contribuindo, assim, para sua adoção na Internet e o seu uso alternativo ao protocolo UDP.

## 1.2 Organização da Dissertação

Os próximos capítulos estão organizados da seguinte forma No capítulo 2, discorre-se sobre alguns mecanismos de controle de congestionamento existentes. No capítulo 3, apresentam-se protocolos que empregam mecanismos de congestionamento desenvolvidos para operar em redes de alta velocidade. No capítulo 4, descreve-se o protocolo DCCP, seu funcionamento básico e seus mecanismos de controle de congestionamento, em especial o TFRC. No capítulo 5, relata-se a avaliação do protocolo DCCP e do mecanismo de controle de congestionamento TFRC, quando este opera em redes de alta velocidade. No capítulo 6, introduz-se uma variação do protocolo DCCP, o protocolo *Fast DCCP*, assim como uma avaliação do mesmo, quando este opera em redes de alta velocidade. No capítulo 7, conclui-se a Dissertação, apresenta-se um resumo dos resultados obtidos e indicam-se possíveis pontos a serem abordados como trabalhos futuros.

## 1.3 Publicações realizadas

A partir dos resultados obtidos através da avaliação do protocolo DCCP em redes de alta velocidade foram publicados os seguintes artigos:

1 - C. Frolidi, N. Fonseca, C. Papotti, D. Manzato; Performance Evaluation of the DCCP Protocol in High-Speed Networks. In: IEEE CAMAD 2010 - International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks. [16]

2 - C. Frolidi, N. Fonseca, C. Papotti; Avaliação de desempenho do protocolo DCCP para redes de alta velocidade. In: Wperformance 2010 - Workshop em Desempenho de Sistemas Computacionais e de Comunicação. [14]

Com relação a proposta do *Fast DCCP* e sua avaliação em redes de alta velocidade, foram publicados os seguintes artigos:

3 - C. Frolidi, N. Fonseca, C. Papotti; Fast DCCP: Uma variante do protocolo DCCP para redes de alta velocidades. In: SBRC 2011 - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos [15].

4 - C. Frolidi, N. Fonseca; A DCCP Variant for High Speed Networks In: LatinCom 2011 - IEEE Latin-American Conference on Communications [13].

## Capítulo 2

# Mecanismos de Controle de Congestionamento

Na Internet, as perdas de pacotes ocorrem em geral devido ao congestionamento na rede. Fluxos TCP apresentam um mecanismo de controle de congestionamento fim-a-fim que reduz sua taxa de transmissão assim que o congestionamento é detectado.

É cada vez mais comum a presença de fluxos que transmitem áudio e vídeo e que buscam atingir taxas estáveis de transmissão maiores que as do TCP. Tais fluxos, porém, não efetuam controle de congestionamento para garantir uma justa distribuição de banda passante. Por essa razão, é necessário que se definam mecanismos e regras de adaptação de taxa para o tráfego gerado por protocolos diferentes do TCP que sejam compatíveis com o mecanismo de controle de congestionamento TCP. Essas regras de adaptação de taxa transformam os mecanismos amigáveis ao TCP.

Preende-se apresentar, neste capítulo, alguns mecanismos de controle de congestionamento que são amigáveis ao TCP.

### 2.1 Controle de Congestionamento Amigável ao TCP

Fluxos que não são gerados pelo protocolo TCP são definidos como amigáveis ao TCP quando sua vazão em um longo período não excede a vazão de uma conexão TCP sob as mesmas condições [6]. Uma definição mais restritiva para protocolos amigáveis ao TCP é [51]:

- Para Unicast: Um fluxo unicast é considerado amigável ao TCP quando ele não reduz a obtenção de banda passante de qualquer fluxo TCP coexistente mais do que um outro fluxo TCP reduziria sob as mesmas condições da rede.

- Para Multicast: Um fluxo multicast é definido como amigável ao TCP quando para cada par transmissor-receptor, o fluxo multicast tem a propriedade de ser um fluxo amigável ao TCP.

## 2.2 Esquemas de Controle de Congestionamento baseados em Janela ou Taxa de Transmissão

Algoritmos baseados em janela usam uma janela de congestionamento no transmissor ou no receptor. Semelhantemente ao TCP, cada pacote transmitido consome um slot na janela de congestionamento, enquanto que o reconhecimento de um pacote recebido libera um slot. O transmissor é liberado para enviar pacotes apenas quando um slot livre estiver disponível. Aumenta-se o tamanho da janela de congestionamento, na ausência da indicação de congestionamento, e diminui-se o tamanho na ocorrência de congestionamento.

Controle de congestionamento baseado em taxa adapta dinamicamente a taxa de transmissão, de acordo com feedback da rede que indique congestionamento. O controle de congestionamento baseado em taxa pode ser implementado utilizando um método de aumento aditivo e diminuição multiplicativa (AIMD) simples ou utilizando uma equação matemática para definir a variação da taxa. Métodos AIMD simples imitam o comportamento do controle de congestionamento do TCP. Isso resulta em uma taxa que demonstra o comportamento típico de dente de serra presente no TCP, fazendo com que métodos AIMD simples sejam inadequados para fluxos contínuos de mídia. Tal controle de congestionamento utiliza um modelo matemático que emula a vazão de fluxo TCP, como apresentado em [37], ao invés de um mecanismo AIMD similar ao do TCP.

Ao adaptar a taxa de envio a média da vazão do TCP, o controle de congestionamento pode produzir mudanças de taxa muito mais suaves e que se aproximam melhor da taxa ideal. Tais esquemas não imitam a taxa de envio TCP em curtos períodos, mas ainda são amigáveis ao TCP em longos períodos de tempo. No entanto, estes mecanismos de controle de congestionamento não reconstróem o controle de congestionamento do TCP e deve-se, portanto, investigar se os mecanismos de ajuste de taxa garantem uma competição justa com o TCP e outros fluxos na competição por banda passante.

## 2.3 Protocolos de Controle de Congestionamento Baseados em Taxa

Serão descritos a seguir algumas propostas de controle de congestionamento baseadas em taxa.

### 2.3.1 Rate Adaption Protocol (RAP)

Utiliza um método AIMD simples para fluxo unicast [38]. Cada pacote é reconhecido pelo receptor, e esses ACKs são utilizados para detectar perdas de pacotes e inferir o RTT. Caso seja detectada uma perda, o transmissor reduz pela metade a taxa de envio; caso contrário o transmissor aumenta sua taxa em 1 pacote/RTT. Ajustes finos baseados em RTT's médios resultam em taxas de envio mais suaves.

### 2.3.2 Loss-Delay Based Adaption Algorithm (LDA+)

Utiliza mecanismo de retroalimentação do protocolo RTCP para efetuar o controle de sua taxa de envio [45]. É um AIMD no qual os fatores de aumento e diminuição são ajustados dinamicamente de acordo com as condições da rede. O tamanho da largura de banda que representa o gargalo da rede é estimado, e a quantidade de aumento aditivo é definida como a mínima entre três fatores independentes: i) fluxos com uma pequena largura de banda podem aumentar sua taxa mais rapidamente do que fluxos com larguras de banda maiores; ii) fluxos não excedem o gargalo da largura de banda estimada; iii) fluxos não aumentam sua largura de banda mais rápido do que uma conexão TCP.

Caso receptores informem a ocorrência de perdas, a taxa de envio é diminuída proporcionalmente à taxa de perda. Além disso, diminui-se a taxa de envio no máximo a um valor equivalente à taxa dada pelo modelo TCP.

### 2.3.3 TCP-Friendly Rate Control Protocol (TFRC)

Evoluí a partir do TFRCP [36]. Este protocolo ajusta sua taxa de envio baseado na complexa equação do TCP [11], porém utiliza métodos mais sofisticados para reunir os parâmetros necessários para o ajuste. Várias estimativas da taxa de erro são utilizadas. Utiliza-se o método do intervalo da média de erros, que melhor representa os requerimentos. A taxa de erro é medida em termos de intervalos de erro. Calcula-se a média de um certo número de intervalos de perda, usando decremento de forma que antigos intervalos de perda tenham um menor peso na média. A taxa de erro é calculada como o inverso do tamanho da média dos intervalos de perda. Mecanismos adicionais são providos para

impedir que reações abruptas a eventos de perda simples aconteçam e para garantir que a taxa de perda se adapte rapidamente a longos intervalos sem a ocorrência de perdas.

### 2.3.4 TCP Emulation at Receivers (TEAR)

É um protocolo híbrido que combina aspectos dos controles de congestionamento baseados em taxa e baseados em janela [40]. O receptor calcula uma taxa de recepção que é enviada de volta para o transmissor, que então ajusta a taxa de envio. O receptor mantém uma janela de congestionamento que é modificada similarmente ao modelo TCP. Um receptor TEAR tenta determinar a partir dos pacotes que chegam quando o TCP irá aumentar ou diminuir o tamanho da janela de congestionamento. Crescimento aditivo e redução de janela causados por três ACKs duplicados são fáceis de emular. No entanto, devido a falta de ACKs, timeouts podem ser estimados apenas de forma aproximada. Em contraste com TCP o protocolo TEAR não usa a janela de congestionamento diretamente para determinar a quantidade de dados a ser enviada, mas calcula a taxa de envio TCP correspondente. Para prevenir o comportamento de dente-de-serra, TEAR calcula a média da taxa sobre um período, que é definida como o tempo entre eventos de redução de taxa consecutivos. Para evitar futuras mudanças de taxa desnecessárias causadas por ruído no padrão de perdas, uma taxa é determinada, utilizando-se uma média balanceada sobre um certo número de períodos para a taxa final. Este valor é, então, informado para o transmissor que ajusta, então, a taxa de envio.

## 2.4 Protocolos de Controle de Congestionamento Baseados em Janela

Serão descritos a seguir algumas propostas de controle de congestionamento baseadas em janela.

### 2.4.1 Random Listening Algorithm (RLA) e Linear Proportional Response (LPR)

Estende o ACK Seletivo (SACK) do TCP pela introdução de alguns avanços para o multicast [49]. Para cada receptor, o transmissor multicast armazena o RTT e a probabilidade de congestionamento medida. Uma perda é detectada pelo transmissor via a identificação de ACKs descontínuos ou através de temporização. Baseado nas indicações de perda, um número  $n$  de receptores com uma grande probabilidade de congestionamento é procurado. Se for detectado o congestionamento, a janela é dividida pela metade nos seguintes casos: i)

se o corte anterior da janela foi feito há muito tempo; ii) se um número aleatório uniforme gerado for menor ou igual a  $1/n$ .

Quando um pacote é reconhecido por todos os receptores, a janela de congestionamento  $cwnd$  é incrementada por  $1/cwnd$ , como no TCP. Um esquema de retransmissão do tipo TCP com recuperação rápida também está incluído no RLA.

O LPR [4] é um esquema de filtro indicador de probabilidade de perda que avança sobre o mecanismo RLA. A probabilidade com que uma fonte multicast reduz sua janela de congestionamento é proporcional a probabilidade de perda no receptor. O esquema LPR alcança melhor justiça em sessões multicast no sentido de sessões unicast concorrentes do que o esquema de indicação de ajuste de janela do RLA. Resultado de simulação indicam que a combinação com o mecanismo de ajuste de janela do RLA possibilita o LPR alcançar boas características amigáveis ao TCP.

## 2.4.2 Multicast TCP (MTCP)

É um protocolo multicast confiável que agrupa os participantes de uma sessão em uma estrutura de árvore lógica na qual a raiz desta árvore é o transmissor de dados [39]. Um pai na estrutura de árvore armazena um pacote recebido até a recepção da confirmação de todos os seus filhos. Tendo recebido um pacote, um filho (que também pode ser um pai para outros participantes) envia um ACK para seus pais usando unicast. Para controlar o congestionamento, MTCP requer que cada pai mantenha dois valores: uma janela de congestionamento e uma janela de transito. O tamanho da janela de congestionamento é gerenciado, de forma similar ao do TCP, incluindo partida lenta e prevenção de congestionamento. As maiores diferenças em relação ao TCP são: a janela de congestionamento é incrementada apenas quando ACKs são recebidos de todos os filhos. Um pacote é imediatamente (re)transmitido para um filho caso ele indique via um ACK negativo (NACK) que ainda não recebeu o pacote. O tamanho da janela de congestionamento é dividida ao meio quando qualquer filho informar três NACKs consecutivos, ou na ocorrência de um timeout pelo fato de um filho não ter reconhecido nenhum pacote.

A janela de transito mantém controle dos dados que um filho não tenha reconhecido. Com cada ACK, um nó pai envia um sumário do congestionamento para seus próprios pais. Este sumário contém o mínimo dos tamanhos de sua própria janela de congestionamento mais aqueles informados pelos seus filhos. O transmissor pode então enviar a diferença entre o tamanho mínimo da janela de congestionamento e o tamanho máximo da janela de transito. O maior problema no MTCP é sua complexidade e a necessidade de configuração de uma estrutura de árvore na qual cada nó tem que efetuar o armazenamento de pacotes, reparo, e o monitoramento de congestionamento.

### 2.4.3 Nominee-Based Congestion Avoidance (NCA) e Pragmatic General Multicast Congestion Control (PGMCC)

São duas abordagens de controle de congestionamento com a mesma ideia fundamental [26] [43]: selecionam, como um representante de grupo, o receptor com a pior conexão de rede. Este receptor reconhece todos os pacotes recebidos e, desse modo, permite ao transmissor utilizar um algoritmo de controle de congestionamento do estilo TCP. Nessa abordagem, o controle de congestionamento e o reparo de pacotes são tratados independentemente. Logo, esta abordagem pode ser utilizada em combinação com um grande número de mecanismos que estabelecem confiabilidade, assim como para transmissão não confiável de dados. O aspecto mais desafiador do NCA e PGMCC é como selecionar o representante de grupo.

Em ambos os algoritmos cada receptor calcula a taxa de dados na qual é possível receber utilizando uma fórmula TCP simples. Esta fórmula leva em conta o RTT e a taxa de perda experimentada pelo receptor. A informação sobre a taxa aceitável é informada de volta ao transmissor via piggybacking em NACKs (PGMCC) ou acumulada em uma estrutura de roteadores em árvore que sempre encaminham a informação do participante com a menor taxa de dados aceitável (NCA). Com estas informações o transmissor seleciona como representante o participante com a menor taxa aceitável e utiliza um mecanismo de controle de congestionamento do tipo TCP para este participante.

# Capítulo 3

## Protocolos para Redes de Alta Velocidade

O protocolo TCP é ineficiente quando opera em redes de alta velocidade [27]; consequentemente, várias propostas de variações deste protocolo foram desenvolvidas, visando sanar as deficiências observadas. Estas propostas introduzem mudanças no mecanismo de controle de congestionamento e tratam, em sua maioria, da obtenção de escalabilidade da taxa de transmissão para utilizar a alta disponibilidade de recursos. Procuram, também, realizar diminuições menos drásticas da taxa de transmissão, quando congestionamento é observado.

Dentre as variações propostas destacam-se: BIC TCP [52], CUBIC TCP [41], FAST TCP [25], TCP Libra [32], HSTP [12], HTCP [30] e STCP [28]. Neste capítulo, pretende-se apresentar algumas destas variações, enfatizando-se os mecanismos de congestionamento.

### 3.1 CUBIC TCP

O CUBIC TCP garante escalabilidade de banda em enlaces de alta velocidade. Faz uso de uma função polinomial de terceira ordem, levando em conta o histórico de perdas da conexão vigente, para escalar a janela de maneira eficiente. A atualização da janela é dada por [41]:

$$W_{cubic} = C(t - K)^3 + W_{max}$$

onde  $C$  é o fator de escala,  $t$  é o tempo decorrido desde o último evento de perda,  $W_{max}$  é o tamanho máximo da janela antes do evento de perda e  $K = \sqrt[3]{W_{max} * \beta / C}$ , onde  $\beta$  é o fator de decremento multiplicativo da janela, aplicado a cada evento de perda.

Após a ocorrência de um evento de perda, o protocolo é capaz de garantir um crescimento rápido da janela e a medida que esta se aproxima de  $W_{max}$ , o crescimento torna-se

mais lento, visando garantir a estabilidade do protocolo através do uso eficiente dos recursos e da prevenção de perdas de pacotes.

## 3.2 FAST TCP

O Fast Active Management Scalable Transmission Control Protocol (FAST TCP) [25] utiliza o atraso de propagação, considerando RTT e atrasos no enfileiramento de pacotes, como principal fator para realização do ajuste da janela de congestionamento. O protocolo possui, em seu mecanismo de controle de congestionamento, quatro componentes independentes: controle de dados, controle de janela, controle de rajada e estimador. O controle de dados determina que pacotes serão transmitidos, o controle de janela determina quantos pacotes serão transmitidos e o controle de rajada determina quando transmitir estes pacotes. Todas estas decisões são feitas com base nas informações fornecidas por um estimador que calcula o atraso de propagação e sinaliza os eventos de perda de pacotes.

Quando um reconhecimento é recebido, calcula-se o RTT para o pacote transmitido e este valor é utilizado para calcular o RTT mínimo e médio observados na conexão. A atualização da janela de transmissão é baseada na diferença do RTT atual, que foi observado durante o envio do último pacote, e do RTT médio. Caso o valor do RTT atual seja menor do que o RTT médio, incrementa-se a janela, caso contrário, decrementa-se a janela. A equação que controla a atualização da janela é [25]:

$$W_{i+1} = \min(2W, (1 - \theta)W + \theta((baseRTT/RTT) * W + \alpha))$$

onde  $\theta \in [0, 1]$ , *baseRTT* é o RTT mínimo observado na conexão, *RTT* é o RTT médio observado na conexão e  $\alpha$  é uma variável positiva para manutenção do equilíbrio dos fluxos do protocolo [25].

## 3.3 Scalable TCP (STCP)

O protocolo STCP [28] utiliza um mecanismo de controle de congestionamento do tipo MIMD (Multiplicative increase, multiplicative decrease) para atualizar o tamanho da janela de transmissão. Enquanto não ocorrerem perdas de pacotes na conexão, cada reconhecimento recebido altera a janela de transmissão para  $W_{i+1} = W + \alpha$ , sendo  $\alpha$  uma constante, de forma que  $0 < \alpha < 1$ . Quando houver uma perda, a janela de transmissão é atualizada para  $W_{i+1} = W - (\beta * W)$ , sendo  $\beta$  uma constante, de forma que  $0 < \beta < 1$  [28]. O protocolo STCP garante que o tempo de recuperação do tamanho da janela, definido pelo tempo em que a janela leva para retornar ao tamanho anterior à uma ocorrência de perda e o decremento da mesma seja proporcional ao RTT, o que difere do protocolo TCP-Reno, cujo tempo é proporcional ao tamanho da janela (anterior a ocorrência da

perda) e ao RTT. Este cálculo garante um crescimento eficiente da janela ao aumento de disponibilidade de banda nos enlaces de alta velocidade. A agressividade obtida com o incremento multiplicativo garante que após uma redução no tamanho da janela o tempo de recuperação será curto, permitindo que o protocolo faça uso dos recursos disponíveis de maneira eficiente.

### **3.4 Avaliação de Desempenho das variações do protocolo TCP para Redes de Alta Velocidade**

Vários trabalhos [5], [29], [33] e [44], foram elaborados para avaliar o desempenho das variações do protocolo TCP para redes de alta velocidade e diversos deles avaliam o uso eficiente dos recursos disponíveis e propriedades como justiça, convergência e compatibilidade com o protocolo TCP-Reno. Estas avaliações evidenciaram uma série de questões destes protocolos, tais como: instabilidade na promoção de justiça, problemas de lentidão no tempo de convergência e problemas de compatibilidade com o TCP-Reno. Tais questões continuam em aberto mesmo nos protocolos que conseguem promover escalabilidade frente ao aumento da disponibilidade de banda passante.

# Capítulo 4

## Protocolo DCCP

O Datagram Congestion Control Protocol (DCCP) foi concebido para facilitar a implementação de aplicações multimídia na Internet, tais como *streaming* e voz sobre IP. O protocolo DCCP oferece controle de congestionamento e reconhecimento da chegada de pacotes (ACK) às aplicações que não podem ter o tempo de entrega de pacotes aumentado pelo *overhead* introduzido pelos mecanismos de entrega confiável do protocolo TCP. As características principais do protocolo DCCP são [10]: fluxo de datagramas orientado à conexão sem entrega confiável e com confirmação de entrega, *handshake* para estabelecimento e término confiáveis da conexão, negociação confiável de parâmetros a serem utilizados durante a conexão e mecanismos de controle de congestionamento amigáveis ao TCP, que podem incorporar o mecanismo *Explicit Congestion Notification* (ECN).

Cada conexão DCCP ocorre entre dois hospedeiros, um dos *hosts* é denominado servidor (DCCP socket opera no modo passivo, durante estabelecimento da conexão) e o outro cliente (DCCP socket opera no modo ativo durante estabelecimento da conexão). Dado que o DCCP é um protocolo *full-duplex*, os dados podem fluir em ambas ou em apenas uma direção. O protocolo define como *half-connection*, o fluxo de dados em uma direção, assim como o fluxo de ACKs correspondentes no sentido oposto. Cada *half-connection* tem seu controle de congestionamento gerenciado de maneira independente. Durante o processo de estabelecimento de conexão, são negociados entre os dois *hosts*, cliente e servidor, os mecanismos de controle de congestionamento a serem utilizados e seus parâmetros, assim como, outras funcionalidades providas pelo protocolo.

### 4.1 Conexões DCCP

O protocolo DCCP possui 9 tipos de pacotes distintos [10]: DCCP-Request, DCCP-Response, DCCP-Data, DCCP-Ack, DCCP-DataAck, DCCPCloseReq, DCCP-Close, DCCP-Reset e DCCP-Move.

O ciclo de uma conexão DCCP, incluindo a utilização dos pacotes listados acima, é descrito abaixo [10]:

1. O host cliente envia um pacote DCCP-Request, especificando as portas do cliente e do servidor, o tipo de serviço requisitado e qualquer outra opção adicional do protocolo, incluindo o mecanismo de controle de congestionamento que o cliente deseja que seja provido pelo host servidor.
2. O host servidor envia um pacote DCCP-Response ao host cliente indicando sua disponibilidade para a conexão requisitada. Este pacote informa quais opções requisitadas foram aceitas pelo host servidor e inicia a negociação de outras opções se for necessário.
3. O host cliente envia um pacote DCCP-Ack ao host servidor confirmando o recebimento do pacote DCCP-Response. Dependendo do pacote enviado pelo host servidor pode ser necessário continuar o processo de negociação de opções para a conexão.
4. São trocados pacotes, entre os hosts cliente e servidor, DCCP-Ack, tantos quanto forem necessários, para finalizar o processo de negociação da conexão. Na transmissão do ack final, o host cliente pode enviar um pedido de um protocolo da camada de aplicação, gerando um pacote DCCP-DataAck.
5. Os hosts, cliente e servidor, trocam pacotes DCCP-Data, para o envio de dados e pacotes DCCP-Ack para confirmação do recebimento destes dados. Opcionalmente, pacotes DCCP-DataAck podem ser enviados, para o caso de pacotes que contenham dados e confirmação de recebimento.
6. O host servidor envia um pacote DCCP-CloseRequest solicitando o encerramento da conexão.
7. O host cliente envia um pacote DCCP-Close confirmando o encerramento da conexão.
8. O host servidor envia um pacote DCCP-Reset, com o campo Reason contendo a informação "Closed" e reinicializa o estado da conexão.
9. O host cliente recebe o pacote DCCP-Reset e espera por um período antes de encerrar a conexão, para que possa receber algum pacote restante da conexão.

A Figura 4.1, ilustra uma conexão DCCP, efetuadas pelos hosts DCCP A (cliente) e DCCP B (servidor).

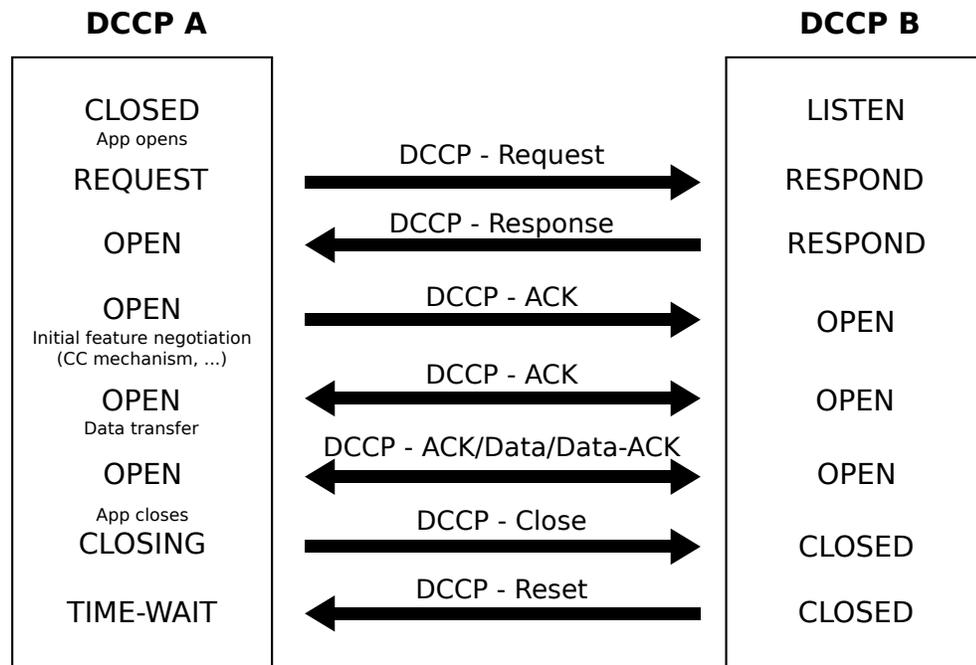


Figura 4.1: Conexão DCCP

## 4.2 Formato dos Pacotes

Todos os pacotes DCCP possuem um cabeçalho com os seguintes campos:

- Source Port: Porta de origem
- Dest Port: Porta de destino
- Type: tipo de mensagem enviada, podendo conter os seguintes valores: 0-DCCP-Request packet, 1-DCCP-Response packet, 2-DCCP-Data packet, 3-DCCP-Ack packet, 4-DCCP-DataAck packet, 5-DCCP-CloseReq packet, 6-DCCP-Close packet, 7-DCCP-Reset packet, 8-DCCP-Move packet.
- CCVal: tipo de controle de congestionamento (CCID) a ser utilizado.
- Sequence number: número de sequência.
- Data Offset: offset do início do cabeçalho ao início do payload do pacote.
- NDP: número de pacotes que não contém dados.
- Cslen: indica a abrangência do checksum do pacote.

- CheckSum: informações para verificação de consistência do pacote.

A Figura 4.2 ilustra o cabeçalho de um pacote DCCP.

|             |       |                 |           |  |
|-------------|-------|-----------------|-----------|--|
| Source Port |       |                 | Dest Port |  |
| Type        | CCval | Sequence Number |           |  |
| Data Offset | # NDP | Cslen           | Checksum  |  |

Figura 4.2: Cabeçalho do pacote DCCP

O payload de uma conexão DCCP é enviado em pacotes do tipo DCCP-Data ou DCCP-DataAck, enquanto pacotes DCCP-Ack não possuem payload. A Figura 4.3 ilustra o formato dos pacotes DCCP-Data. A Figura 4.4 ilustra o formato dos pacotes DCCP-Ack.

|  |             |
|--|-------------|
| Generic DCCP Header (12 bytes) with Type=2 (DCCP Data) |             |
| Options  | / [padding] |
| Data<br>...  |             |

Figura 4.3: Pacote DCCP de Dados

|   |                        |
|---|------------------------|
| Generic DCCP Header (12 bytes) with Type=3 (DCCP-Ack) |                        |
| Reserved  | Acknowledgement Number |
| Options   | / [padding]            |

Figura 4.4: Pacote DCCP de Confirmação

Os mecanismos de controle de congestionamento que podem ser usados no DCCP são definidos através de um CCID (*Congestion Control Identifier*) e podem ser identificados por um número de 0 a 255. Os CCIDs definidos, inicialmente, para o DCCP, de acordo com

[10], são: 0 – Reservado, 1 – Controle de congestionamento não especificado baseado no usuário, 2 – Controle de congestionamento *TCP-Like*, 3 – Controle de congestionamento TFRC.

## 4.3 Controle de Congestionamento TCP-Like (CCID 2)

O controle de congestionamento TCP-Like (CCID 2) é recomendado para aplicações que desejam transmitir o máximo de dados possível em um curto período de tempo e não são muito sensíveis a mudanças abruptas na taxa de transmissão. Um exemplo de aplicação que utilizaria o CCID 2 são aplicações do tipo streaming, pois utilizam um buffer dos dados no receptor, que atrasa o início da execução do playback, tornando este tipo de aplicação menos sensível à mudanças abruptas na taxa de transmissão.

Os mecanismos de controle de congestionamento do TCP-Like são muito similares aos mecanismos empregados no TCP, incluindo partida lenta, janela de congestionamento, timeouts, etc. Algumas diferenças são:

O DCCP é um protocolo do tipo datagrama, portanto alguns parâmetros que são contabilizados em bytes no TCP, como a janela de congestionamento (*cwnd*), são contabilizados em pacotes no DCCP.

Por ser um protocolo que não possui entrega confiável, o DCCP nunca efetua a retransmissão de pacotes, utiliza as informações obtidas pelo controle de congestionamento para redimensionar a taxa de envio de pacotes subsequentes.

## 4.4 Controle de Congestionamento TFRC (CCID 3)

O controle de congestionamento TFRC (CCID 3) é recomendado para aplicações que precisam transmitir a taxas de dados constantes, pois são sensíveis à mudanças abruptas na taxa de envio. Um exemplo de aplicação que utilizaria o CCID 3 é voz sobre IP em tempo real. No TFRC, o receptor efetua medições de perdas de pacotes e retorna esta informação ao emissor para medição do *round-trip time* (RTT). O retorno das perdas enviado pelo receptor e o RTT estimado são utilizados para atualizar a taxa de transmissão do emissor, calculada através da equação de vazão, que é uma versão simplificada da equação de vazão do TCP-Reno, definida por:

$$X = \frac{s}{R * \text{sqrt}(2 * b * p/3) + (tRTO * (3 * \text{sqrt}(3 * b * p/8) * p * (1 + 32 * p^2)))} \quad (4.1)$$

onde:  $X$  é a taxa de transmissão em bytes/segundo,  $s$  é o tamanho do pacote em bytes,  $R$  é o RTT em segundos,  $p$  é a taxa de ocorrência de perdas, valor entre 0 e 1, que expressa o número de eventos de perda em relação a fração de pacotes transmitidos,  $tRTO$  é o valor do *timeout* de retransmissão do TCP em segundos e  $b$  é igual a 1 e representa o número máximo de pacotes confirmados por um simples pacote de confirmação. Embora o protocolo DCCP não efetue a retransmissão de pacotes perdidos, o *timeout* de retransmissão é considerado para indicar a ocorrência de perda de pacotes.

O emissor envia pacotes ao receptor através de um fluxo de dados com taxa de transmissão controlada. A medida que pacotes de confirmação de envio são recebidos (ou não) ocorrem mudanças na taxa de transmissão. Um pacote de confirmação enviado pelo receptor contém as seguintes informações: o *timestamp* do último pacote de dados recebidos, o tempo decorrido entre o recebimento do último pacote e a geração do pacote de confirmação e a taxa estimada com a qual o receptor recebeu os dados após o envio do último pacote de confirmação.

#### 4.4.1 Protocolo de Envio de Dados do TFRC

O emissor envia pacotes ao receptor através de um fluxo de dados com taxa de transmissão controlada. À medida que pacotes de confirmação de envio são recebidos, mudanças são observadas na taxa de transmissão. Um pacote de confirmação enviado pelo emissor contém as seguintes informações:

O *timestamp* do último pacote de dados recebido (*trecvdata*). O tempo decorrido entre o recebimento do último pacote e a geração do pacote de confirmação (*tdelay*). A taxa estimada com a qual o receptor recebeu os dados após o envio do último pacote de confirmação ( $Xrecv$ ), esta estimativa é calculada através da soma dos pacotes recebidos do emissor.

Durante a inicialização do protocolo de envio no emissor o valor de  $X$  é 1 pacote/segundo. Um temporizador para o recebimento de confirmação de pacotes enviados, definido por *nofeedback*, é inicializado com o valor igual a 2 segundos. O comportamento de  $X$  será alterado, da seguinte maneira [7]:

- Quando um pacote de confirmação é recebido, a taxa de envio é atualizada conforme o algoritmo a seguir:

---

**ALGORITHM *Atualiza Taxa***

---

1.  $Rsample = (tnow - trecvdata) - tdelay;$
2. **If**  $ACKRecv = 0;$
3.  $R = Rsample;$
4. **Else**  $R = q * R + (1 - q) * Rsample;$
5. **Endif**
6.  $TRTO = 4 * R;$
7. **If**  $p > 0;$
8.  $X = \max(\min(X_{calc}, 2 * X_{recv}), s / tmbi);$
9. **Else;**
10. **If**  $tnow - tld \geq R;$
11.  $X = \max(\min(2 * X, 2 * X_{recv}), s / R);$
12. **Endif**
13. **Endif**
14.  $nofeedback = \max(4 * R, 2 * s / X)$

---

O atraso de propagação para conexão vigente é atualizado, considerando o atraso médio de propagação ( $R$ ) e o atraso observado na última interação entre emissor e receptor ( $Rsample$ ) (linhas de 1 a 4). O valor do *timeout* de retransmissão ( $TRTO$ ) é atualizado considerando o novo valor médio do atraso de propagação (linha 6).

A taxa de envio é atualizada, caso erros tenham ocorrido na conexão ( $p > 0$ ), considerando o mínimo valor entre o valor da taxa de envio calculado através da equação de vazão (Equação 4.1) e o dobro da taxa de envio estimada pelo receptor ( $X_{recv}$ ), e o valor máximo entre este e o envio de um pacote a cada 64s (linhas 7 e 8). Caso erros não tenham ocorrido, a taxa de envio será atualizada considerando o mínimo valor entre o dobro do valor da taxa de envio ( $2 * X$ ) e o dobro da taxa de envio estimada pelo receptor ( $X_{recv}$ ), e o valor máximo entre este e o envio de um pacote pelo atraso médio ( $s / R$ ) (linhas 9, 10 e 11).

- Quando ocorre a expiração do temporizador para recebimento de confirmação, a taxa de envio é atualizada conforme o algoritmo a seguir:

---

**ALGORITHM *Atualiza Taxa Erro***

---

1.  $X = X/2$ ;
2. **If**  $X_{calc} > 2 * X_{recv}$ ;
3.  $X_{recv} = \max(X_{recv}/2, s/(2 * tmbi))$ ;
4. **Else**  $X_{recv} = X_{calc}/4$ ;
5. **Endif**
6. **If**  $p > 0$ ;
7.  $X = \max(\min(X_{calc}, 2 * X_{recv}), s/tmbi)$ ;
8. **Else**;
9. **If**  $tnow-tld \geq R$ ;
10.  $X = \max(\min(2 * X, 2 * X_{recv}), s/R)$ ;
11. **Endif**
12. **Endif**
13.  $nofeedback = \max(4 * R, 2 * s/X)$

---

A taxa de envio ( $X$ ) é reduzida pela metade em caso de ocorrência de perdas. Um novo valor para taxa de envio estimada pelo receptor ( $X_{recv}$ ) é calculado (linhas 2 a 5). O restante do algoritmo executa os mesmos passos descritos para o algoritmo anterior.

### 4.4.2 Cálculo da taxa de eventos de perda do TFRC

Calcular a taxa de eventos de perda ( $p$ ), de maneira eficiente, é de fundamental importância para o TFRC. O receptor efetua este cálculo através de medições que consideram os pacotes enviados e os pacotes recebidos. Todo pacote transmitido possui um número de sequência que é incrementado de uma unidade, para cada pacote enviado. O receptor mantém uma estrutura para armazenar um histórico dos pacotes recebidos e dos pacotes não-recebidos. Um pacote perdido é detectado quando são recebidos três pacotes com número de sequência superior a um reconhecimento não-recebido. Para efetuar o cálculo da taxa de eventos de perda é preciso calcular o intervalo médio de perdas. Isto é feito considerando os  $n$  eventos de perda mais recentes e atribuindo pesos a cada um deles. A fórmula para calcular os pesos destes eventos, assim como o intervalo médio de perdas, é definida em [7].

---

**ALGORITHM *Calcula Perda***

---

1. **If**  $i < n/2$ ;
2.  $w_i = 1$ ;
3. **Else**  $w_i = 1 - (i - (n/2 - 1))/(n/2 + 1)$ ;
4. **Endif**
5.  $Itot0 = 0$ ;  $Itot1 = 0$ ;  $Wtot = 0$ ;
6. **For**  $i = 0$  to  $n - 1$ ;
- do**
7.  $Itot0 = Itot0 + (I_i * w_i)$ ;
8.  $Wtot = Wtot + w_i$ ;
9. **Endfor**
10. **For**  $i = 1$  to  $n$ ;
- do**
11.  $Itot1 = Itot1 + (I_i * w_{(i - 1)})$ ;
12. **Endfor**
13.  $Itot = \max(Itot0, Itot1)$ ;
14.  $I_{mean} = Itot/Wtot$ ;
15.  $p = 1/I_{mean}$

---

O número de ocorrência de perdas é definido por  $n$ , pesos são atribuídos a estas perdas, sendo o peso das perdas mais recentes mais significativo do que as perdas anteriores (linhas de 1 a 12). A razão entre os eventos de perdas e o peso das mesmas é calculado e este valor é utilizado para calcular a taxa de eventos de perda (linhas 12 a 15).

## Capítulo 5

# Avaliação do protocolo DCCP em Redes de Alta Velocidade

Para verificar a eficiência da operação do protocolo DCCP, em redes de alta velocidade, as seguintes métricas foram utilizadas: escalabilidade, justiça, convergência e compatibilidade com o protocolo TCP-Reno, [9]. A seguir, define-se cada uma destas métricas e sua relevância.

A escalabilidade é a capacidade do protocolo em se adaptar ao crescimento da disponibilidade de recursos de uma rede, sendo fundamental para um protocolo operar nas redes de alta velocidade. Os experimentos avaliam a escalabilidade em função da capacidade disponível no enlace. Para tal, aumenta-se a capacidade de transmissão dos enlaces e avalia-se a habilidade do protocolo de utilizar, de forma eficiente, a banda disponível. Os experimentos investigaram, também, a escalabilidade, em função do número de conexões. Para se fazer essa avaliação, aumenta-se o número de conexões do protocolo e verifica-se a habilidade do protocolo de manter a utilização do enlace estável e justa entre as conexões.

O critério de justiça pode ser avaliado pela capacidade do protocolo em garantir divisão igualitária de banda passante entre as conexões. É importante que um protocolo apresente justiça para garantir que as conexões que compartilham o enlace possam aumentar a sua taxa de transmissão de maneira eficiente. Quantifica-se o grau de justiça através do coeficiente *Jain Fairness Index* [23], que varia entre 0 e 1. Valores mais próximos de 0 indicam injustiça e valores mais próximos de 1 um sistema justo.

A convergência à justiça é o tempo necessário para que se obtenha divisão igualitária da banda passante. É importante que um protocolo apresente convergência para garantir que conexões recentemente iniciadas possam competir de maneira justa, pela banda passante, com conexões iniciadas há mais tempo.

A compatibilidade com o TCP-Reno é definida como a capacidade dos fluxos de dados do protocolo coexistirem de maneira justa com os fluxos de dados do protocolo TCP,

garantindo que ambos os fluxos possam disputar de maneira equilibrada o uso da banda passante.

## 5.1 Avaliação de desempenho

Para avaliação do protocolo DCCP foram realizados experimentos de simulação e medição. Os cenários e métricas utilizados para avaliação do protocolo DCCP em redes de alta velocidade foram derivados a partir do proposto em [9]. Todos os experimentos utilizam a topologia *Dumbbell* (Figura 5.1), que contém um único enlace gargalo constituído por dois roteadores por onde passam conexões. O número de conexões varia de acordo com os experimentos realizados.

Nos experimentos de simulação foi utilizado o simulador de redes NS-2 [35], versão 2.31. Para os experimentos de medição foi utilizado o sistema operacional Linux, com o kernel da versão 2.6.20, que possui o protocolo DCCP implementado e operacional. Tanto a implementação do protocolo DCCP no NS-2, quanto no Linux, seguem as especificações definidas em [10].

Nos experimentos de simulação com o NS-2, os dois roteadores utilizam gerenciamento de fila *droptail* e possuem *buffer* de 2500 pacotes, valor este semelhante ao tamanho de buffers em roteadores reais [50]. Para evitar impactos relacionados à sincronização das conexões nas simulações, os instantes dos inícios das transmissões e os atrasos de propagação para cada uma das conexões foram gerados aleatoriamente. Os instantes dos inícios variaram entre 1 e 10s e o atraso de propagação entre 5 e 15ms. Tráfego *background* foi introduzido, visando refletir composição de tráfego em ambiente de redes reais, conforme proposto em [19]. O tráfego *background* utiliza 20%, 50% ou 80% da capacidade do enlace gargalo, no sentido reverso, de acordo com a distribuição [3]: 20% de tráfego UDP, 56% de tráfego Web e 24% de tráfego FTP. O tráfego transportado pelas conexões foi do tipo CBR, com a taxa de transmissão de cada conexão ajustada para que a mesma faça uso de toda a capacidade de transmissão do enlace gargalo estipulado no cenário do experimento.

Nos experimentos de medição, foram utilizados dois computadores com interfaces ethernet com capacidade de 1Gbps, interligados através de um *switch router* com capacidade de 1Gbps por porta, sendo que cada uma das portas do *switch* estava com uma rota estática habilitada, emulando dois roteadores, dado que as portas possuem um processador de pacotes e filas independentes. Todas as métricas de distribuição de tráfego *background* e limitações da capacidade do enlace físico foram obtidas através do uso da ferramenta *tc*, *Traffic Control* [47], e para o envio de dados das conexões DCCP e TCP foi utilizado a ferramenta *Iperf* [22], sempre com transmissão de tráfego CBR, com tempos para os instantes dos inícios variando entre 1 e 10s. As únicas métricas não coletadas

nos experimentos de medição dizem respeito aos tempos de atraso de propagação, que dependem das características físicas e de operação da rede.

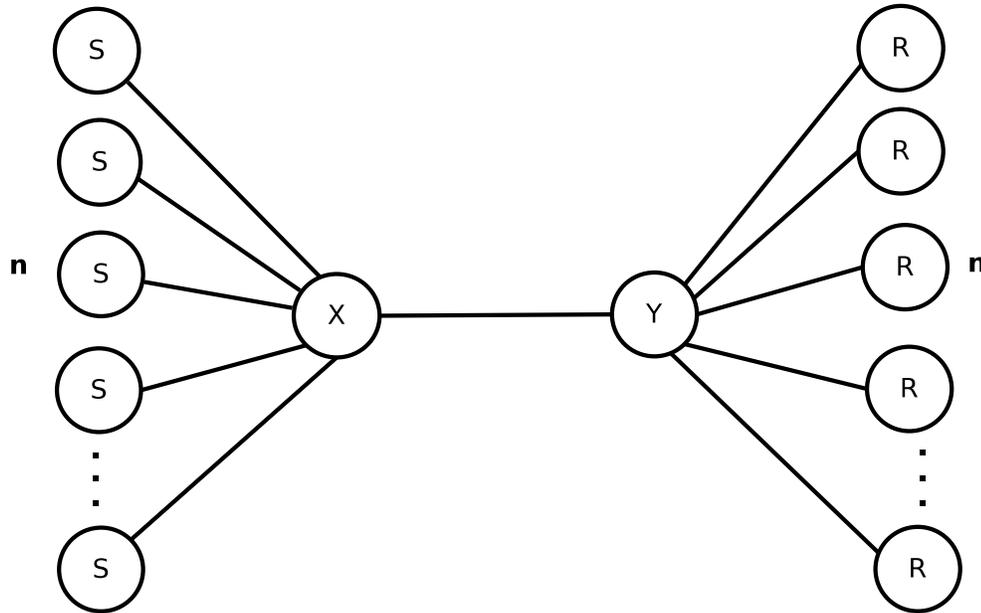


Figura 5.1: Topologia Dumbbell

Os gráficos apresentados mostram o valor médio obtido a partir da execução de 6 replicações para cada experimento. Para o cálculo do intervalo de confiança, foram utilizados o método de replicações independente com nível de confiança de 95%. O tempo de execução definido para todos os experimentos foi de 900s.

### 5.1.1 Escalabilidade

Para avaliar a escalabilidade do protocolo DCCP foram elaborados dois cenários distintos: um para avaliar a escalabilidade em função do aumento da banda passante dos enlaces e outro em função do número de conexões. Para avaliar a escalabilidade em função da capacidade de transmissão do enlace gargalo, variou-se a capacidade no seguinte conjunto: [155Mbps, 622Mbps, 1Gbps]. O atraso do enlace foi fixado em 100ms, nos experimentos de simulação, e foram utilizadas 16 conexões DCCP. A Figura 5.2 e a Figura 5.3 apresentam, respectivamente, os resultados obtidos para os experimentos de simulação e de medição. Em ambos os gráficos pode-se perceber comportamento semelhante, com valores um pouco superiores para os experimentos de medição, o que se deve ao fato de que, nos mesmos, o atraso de propagação é inferior ao valor de 100ms estipulado nos experimentos de simulação. A medida que o tráfego *background* fica mais intenso, os valores em ambos os experimentos ficam mais próximos. As 16 conexões DCCP utilizam; eficientemente, o

enlace (utilização superior a 0.9) para enlaces com capacidade igual a 155 Mbps. Com o aumento da capacidade do enlace, nos experimentos de medição, a utilização é reduzida de maneira significativa, chegando a ser 0.6, para a capacidade de 1Gbps (Figura 5.3). Pode-se concluir que o protocolo não é escalável em relação ao aumento de disponibilidade de banda. Esta ineficiência deve-se a ausência de um mecanismo para estimar a disponibilidade de banda passante na implementação do TFRC, a fim de que se tenha o uso eficiente da largura de banda disponível.

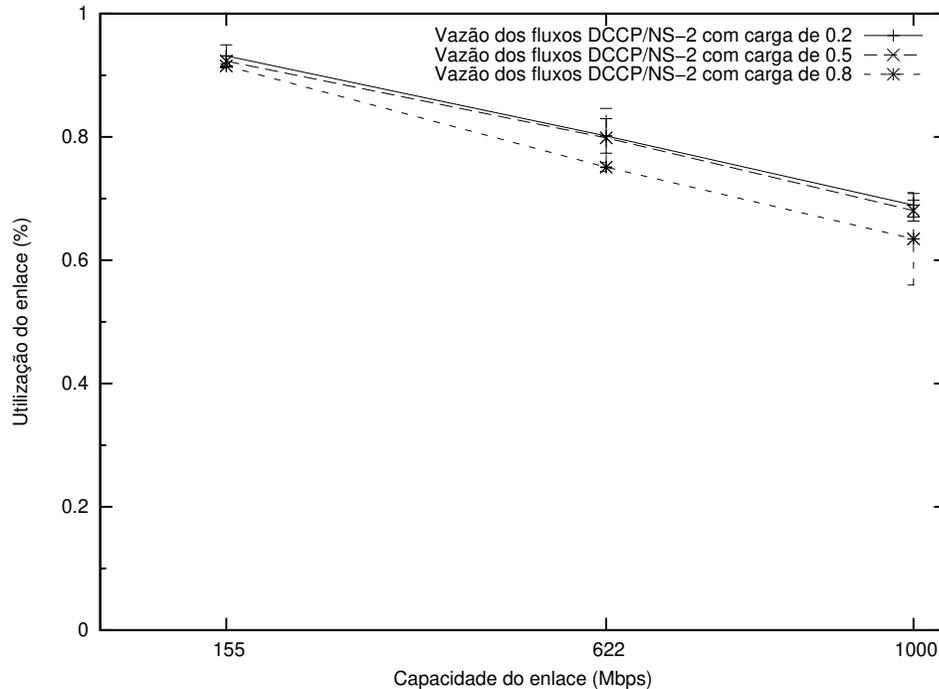


Figura 5.2: Utilização do enlace em função da variação da banda (NS-2)

Para avaliar a escalabilidade em função do número de conexões, variou-se o número de conexões DCCP entre 2 e 32 ( $2^1, 2^2, 2^3, \dots, 2^4$ ), para os experimentos de medição, e variou-se entre 2 e 256 ( $2^1, 2^2, 2^3, \dots, 2^8$ ), nos experimentos de simulação. A diferença na variação do número de conexões entre os experimentos deve-se às limitações de hardware presente nos dispositivos utilizados para realização dos experimentos de medição. A capacidade de transmissão do enlace foi fixado em 1Gbps e o atraso de propagação em 100ms, no caso dos experimentos de simulação. A Figura 5.4 e a Figura 5.5 apresentam, respectivamente, os resultados da avaliação da escalabilidade em função da variação do número de conexões, para os experimentos de simulação e de medição. Pode-se observar valores um pouco superiores para os experimentos de medição, devido ao fato de que, nos mesmos, o atraso de propagação é inferior ao valor de 100ms estipulado nos experimentos de simulação. A medida que o tráfego *background* fica mais intenso, os valores em ambos

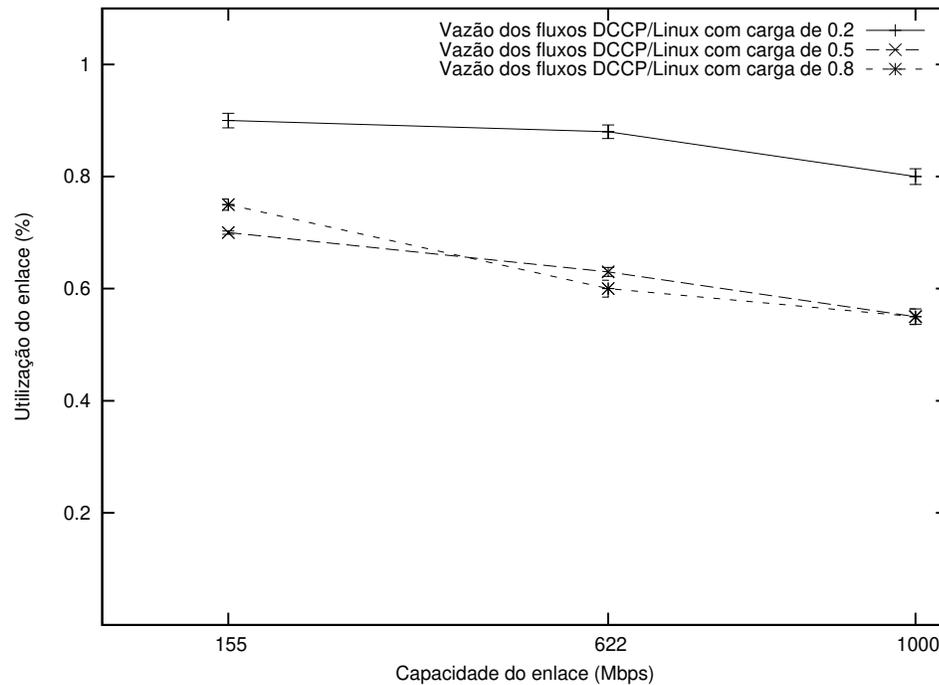


Figura 5.3: Utilização do enlace em função da variação da banda (Linux)

os experimentos ficam mais próximos. Nota-se que a medida que o número de conexões aumenta, nos experimentos de simulação, a utilização do enlace permanece estável e o tráfego *background* implica em mudanças discretas na utilização. O protocolo DCCP opera de maneira escalável e eficiente em relação ao aumento do número de conexões, o que demonstra que o mecanismo de controle de congestionamento empregado pelo TFRC garante a escalabilidade em função do número de conexões.

### 5.1.2 Justiça

Para avaliar a justiça do protocolo foram definidos dois cenários de simulação, ambos com 4 conexões DCCP cuja transmissão atravessa o enlace gargalo com atraso de propagação de 100ms. Nos experimentos de simulação a capacidade do enlace é de 1Gbps. No primeiro cenário, os fluxos DCCP são iniciados, no mesmo instante de tempo (no instante 1s), enquanto que no segundo cenário os fluxos DCCP são iniciados, em instantes de tempo distintos (aleatoriamente entre 1s e 10s). A Figura 5.6 apresenta os resultados da avaliação de justiça para o primeiro cenário proposto, tanto para os experimentos de simulação quanto para os de medição. Nota-se, em ambos os casos, que independentemente da intensidade do tráfego *background* os valores do *Jain Fairness Index* ficam próximos a 0.9, que indica que o protocolo produz justiça, ou seja, o DCCP provê distribuição de

banda equilibrada entre os fluxos que competem pelo enlace. A Figura 5.7 apresenta os resultados da avaliação de justiça para o segundo cenário, tanto para os experimentos de simulação quanto os de medição. Os valores são similares aos obtidos no cenário anterior, reforçando a propriedade de justiça existente no protocolo.

### 5.1.3 Convergência

Para avaliar a convergência do protocolo nos experimentos de simulação, foram realizados experimentos com o enlace gargalo com capacidade de 1Gbps, duas conexões DCCP e o atraso de propagação no enlace gargalo em 10ms. A segunda conexão DCCP inicia a transmissão no instante 50s, tempo suficiente para a primeira conexão alocar toda a banda disponível do enlace gargalo. A Figura 5.8 apresenta os resultados da avaliação da convergência, tanto para os experimentos de simulação quanto para os de medição, para diferentes intensidades de tráfego *background*. O tempo de convergência é alto, em ambos os casos, apresentando valores entre 90s e 120s, que são, tipicamente, muito superiores ao tempo de duração médio de uma conexão na Internet. Pode-se dizer que o mecanismo de controle de congestionamento implementado pelo TFRC, mesmo produzindo altos graus de justiça, não consegue prover uma resposta rápida à divisão igualitária da banda da rede entre fluxos que competem pelo enlace. Enfatiza-se que estudos anteriores, descritos em [2] e [48], apontam o problema do alto tempo de resposta do mecanismo de congestionamento do TFRC, mais lento do que o observado no TCP-Reno.

### 5.1.4 Compatibilidade com o TCP-Reno

Para avaliar a compatibilidade do protocolo DCCP com o protocolo TCP-Reno, foram realizados experimentos com dois cenários distintos. Em um deles, dois fluxos TCP-Reno, transmitindo tráfego FTP, competem pela banda do enlace gargalo e no outro um fluxo DCCP e um fluxo TCP-Reno, com o mesmo tráfego FTP do cenário anterior, competem pelo enlace gargalo. A capacidade do enlace usada é 1Gbps, o tempo de propagação foi de 10ms, para os experimentos de simulação. O objetivo destes experimentos é avaliar se o TCP-Reno obtém a mesma banda do enlace que seria obtida se o mesmo não estivesse concorrendo com o protocolo DCCP. Na Tabela 5.1, são apresentados os resultados da avaliação de compatibilidade do DCCP/TFRC com o TCP-Reno, derivados nos experimentos de simulação. A utilização do enlace para TCP-1 apresenta os valores obtidos pelo fluxo TCP, quando este concorre com outro fluxo TCP, enquanto que os resultados TCP-2 apresentam os valores obtidos pelo fluxo TCP, quando este concorre com um fluxo DCCP. Os diferentes valores de utilização observadas em ambos os fluxos TCP sugerem a incompatibilidade entre o protocolo DCCP, quando este emprega o mecanismo de controle

|                          |       | Tráfego Background (%) |        |        |
|--------------------------|-------|------------------------|--------|--------|
|                          |       | 0.2                    | 0.5    | 0.8    |
| Utilização do enlace (%) | TCP-1 | 0.0152                 | 0.0124 | 0.0061 |
|                          | TCP-2 | 0.0043                 | 0.0031 | 0.0012 |

Tabela 5.1: Taxa de utilização obtida pelos fluxos TCP (NS-2)

|                          |       | Tráfego Background (%) |        |        |
|--------------------------|-------|------------------------|--------|--------|
|                          |       | 0.2                    | 0.5    | 0.8    |
| Utilização do enlace (%) | TCP-1 | 0.3120                 | 0.2210 | 0.1830 |
|                          | TCP-2 | 0.1225                 | 0.0885 | 0.0523 |

Tabela 5.2: Taxa de utilização obtida pelos fluxos TCP (Linux)

TFRC, com o protocolo TCP-Reno, uma vez que o fluxo TCP-2 obtém uma utilização inferior a obtida pelo fluxo TCP-1.

Na Tabela 5.2, são apresentados os resultados da avaliação de compatibilidade do DCCP/TFRC com o TCP-Reno, para os experimentos de medição. A utilização do enlace para TCP-1 apresenta os valores obtidos pelo fluxo TCP, quando este concorre com outro fluxo TCP, enquanto que os resultados TCP-2 apresentam os valores obtidos pelo fluxo TCP, quando este concorre com um fluxo DCCP. Os diferentes valores de utilização sugerem a incompatibilidade entre o protocolo DCCP, quando este emprega o mecanismo de controle TFRC, com o protocolo TCP-Reno, dado que no fluxo TCP-2 a utilização é inferior a obtida pelo fluxo TCP-1. As diferenças entre as utilizações obtidas pelos fluxos TCP sugerem a incompatibilidade destes com os fluxos DCCP/TFRC.

### 5.1.5 Limitações conhecidas do TFRC

Este último grupo de experimentos realizados, apenas através de experimentos de simulação, devido às limitações de reprodução destes em cenário experimental, objetiva verificar se as limitações do TFRC em redes convencionais, apontadas em [42], são observadas também em redes de alta velocidade.

Em [42], foram apontadas algumas limitações do TFRC verificadas a partir da execução de experimentos, entre elas as diferenças de vazão obtidas por fluxos TFRC e TCP concorrentes. Esta diferença foi atribuída principalmente ao cálculo do RTO por estes protocolos, que ocasiona ocorrências de eventos de perda de pacotes distintas entre os fluxos e, conseqüentemente, taxas de transmissão distintas. Vale ressaltar que o RTO calculado pelo

TCP-Reno é o valor máximo entre  $1s$  e  $RTT + 4D$ , onde  $D$  representa a variação do RTT e o RTO calculado pelo TFRC é  $4 * RTT$ . Para atrasos de propagação muito pequenos, o valor do RTO estimado pelo TFRC tende a ser menor do que o do TCP. A medida que o atraso de propagação cresce, observa-se a ocorrência inversa. Nos resultados relatados em [42], os fluxos TFRC e TCP apresentam um desbalanceamento nas taxas de transmissão obtidas, sendo que os fluxos TFRC possuem uma taxa de transmissão menor e uma taxa de eventos de perdas maior do que os fluxos TCP.

Adotou-se um conjunto de experimentos para redes de alta velocidade, derivado de [42], com o objetivo de avaliar se os pontos observados são recorrentes nessas redes. Os cenários são baseados na topologia *Dumbbell* com dois roteadores com fila *droptail*, tamanho do *buffer* de 2500 pacotes, 4 fluxos TCP e 4 fluxos DCCP, transmitindo tráfego CBR. Além disto, utilizou-se o tempo total de simulação de 900s, a capacidade do enlace de 1Gbps, o atraso de propagação no enlace gargalo de 100ms, tráfego *background* reverso de 20% e o tráfego *background* direto constituído por um número de conexões TCP de longa duração (variando entre 50 e 400), que transmitem tráfego FTP. As Figuras 5.9 e 5.10 apresentam os resultados da taxa de transmissão média obtida pelos fluxos e a taxa de eventos de perdas. Diferentemente do que foi observado em [42], os fluxos DCCP/TFRC obtiveram uma taxa de transmissão superior aos fluxos TCP, mesmo com uma taxa de perdas de pacotes superior a estes. Isto reforça o desbalanceamento da utilização obtida pelos fluxos. As diferenças entre os resultados aqui apresentados e os resultados apresentados em [42] devem-se às limitações do TCP-Reno em fazer uso eficiente dos recursos em enlaces de alta velocidade e ao fato de que o protocolo DCCP/TFRC, mesmo possuindo deficiências de escalabilidade, em função do aumento da banda do enlace (como visto anteriormente), possui desempenho melhor do que a do TCP-Reno nos enlaces de alta velocidade.

### 5.1.6 Resumo conclusivo

Os resultados da avaliação de escalabilidade em função da variação da capacidade da banda do enlace apontam para a ineficiência do protocolo DCCP em obter o uso eficiente dos recursos disponíveis no enlace gargalo. Os resultados da avaliação de escalabilidade em função do número de conexões demonstram que o protocolo DCCP é eficiente na manutenção do equilíbrio e da justiça entre fluxos que competem pelo enlace gargalo, em função do número de fluxos. Os resultados da avaliação de convergência indicam que o protocolo DCCP possui convergência lenta, para um estado de equilíbrio e justiça entre fluxos que disputam o enlace gargalo. Os valores obtidos foram, em média, próximos a 100s. Os resultados da avaliação de compatibilidade entre os protocolos DCCP e TCP-Reno indicam, também, deficiências do protocolo DCCP que acarretam o desbalanceamento nas taxas de transmissão obtidas pelos fluxos de ambos os protocolos que disputam o enlace gargalo.

Os resultados da avaliação de justiça demonstram a eficiência do protocolo DCCP em garantir que diferentes fluxos DCCP possam competir, de maneira equilibrada e justa, pelos recursos do enlace gargalo.

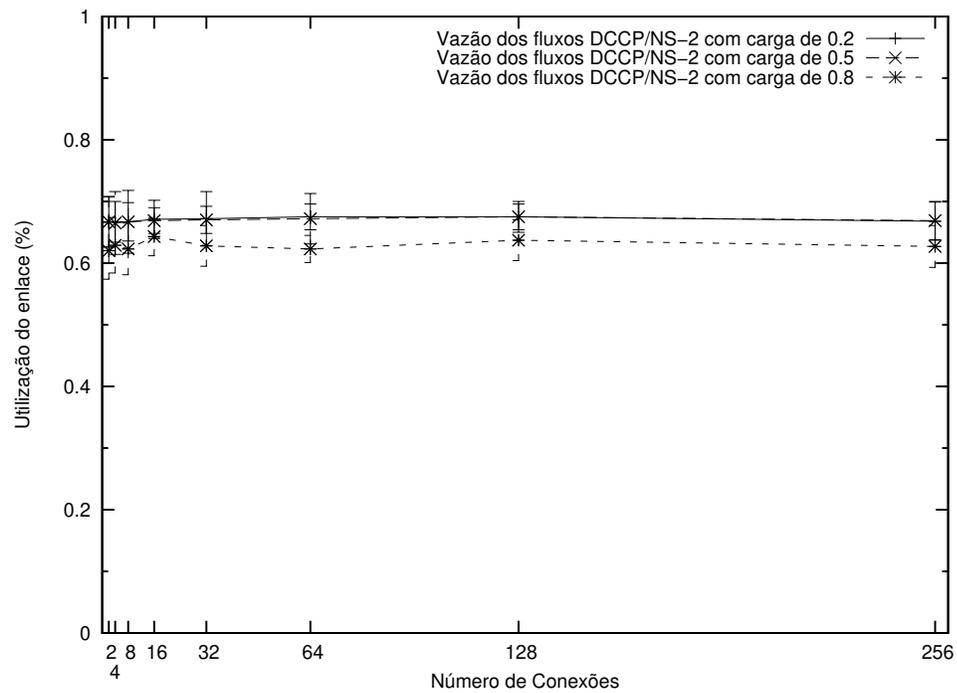


Figura 5.4: Utilização do enlace em função do número de conexões (NS-2)

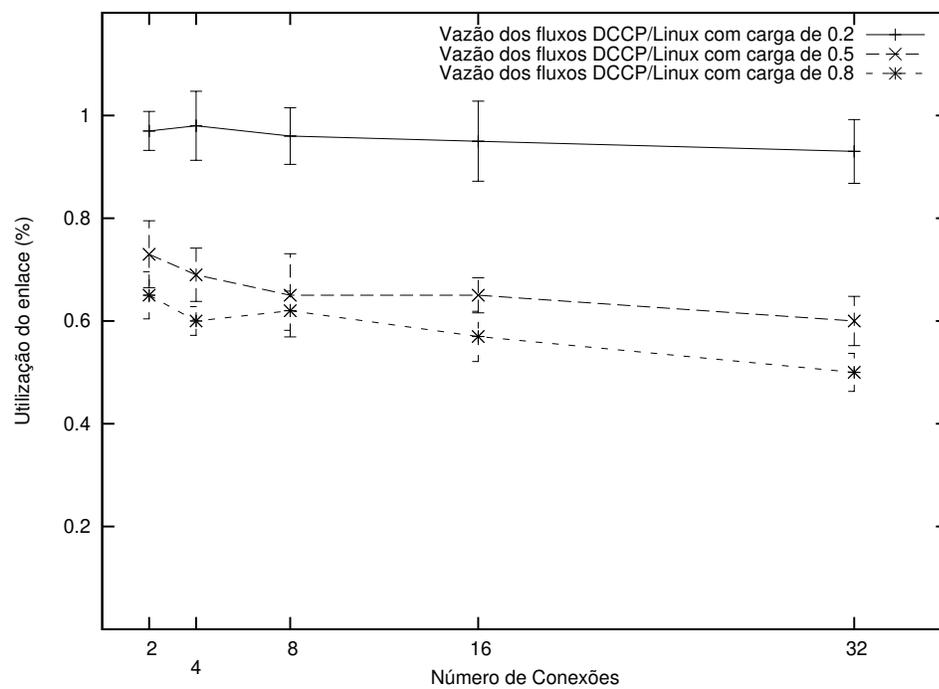


Figura 5.5: Utilização do enlace em função do número de conexões (Linux)

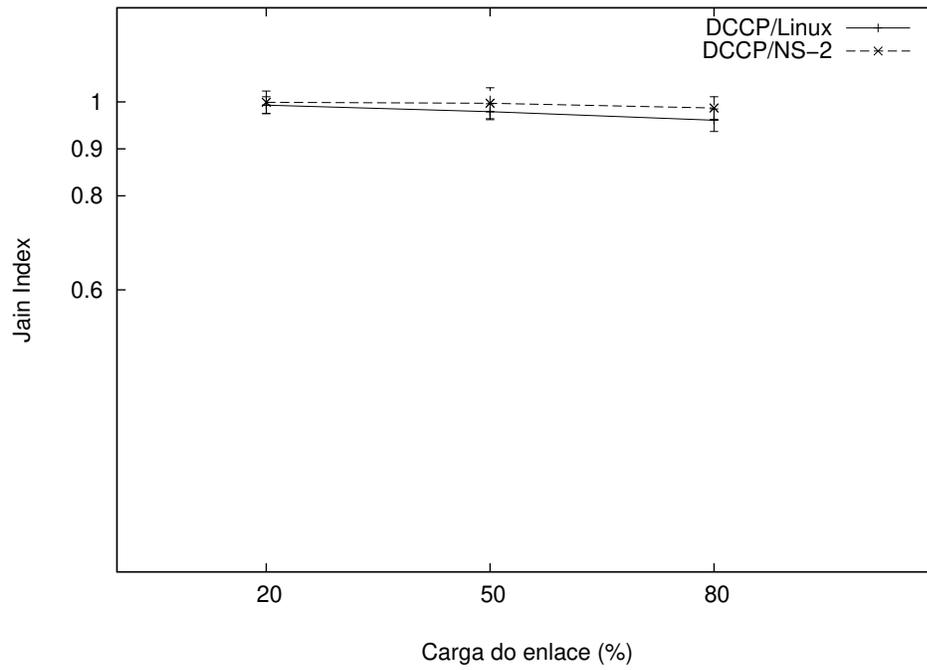


Figura 5.6: Primeiro Cenário de Justiça

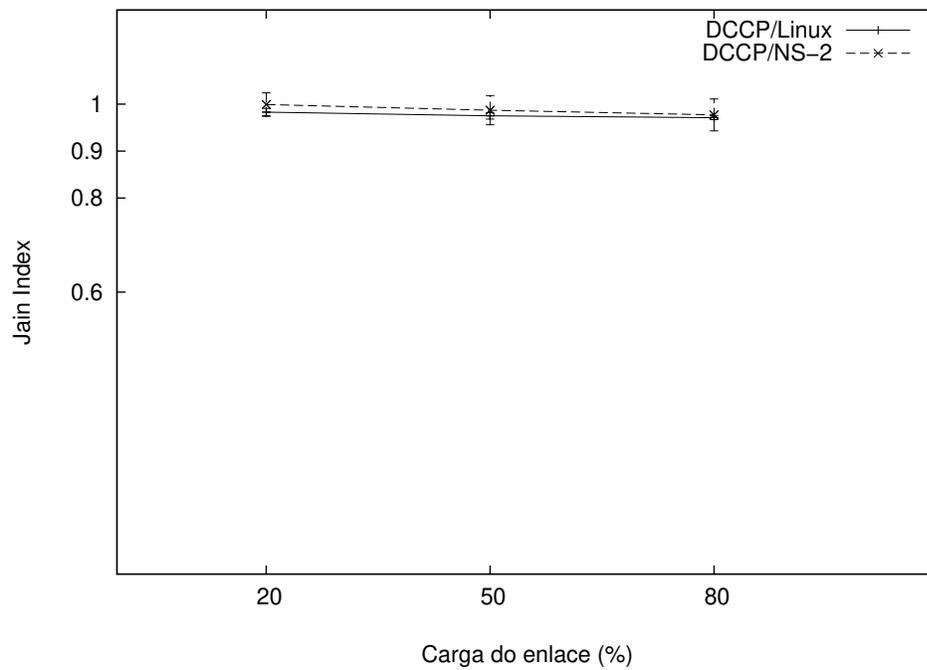


Figura 5.7: Segundo Cenário de Justiça

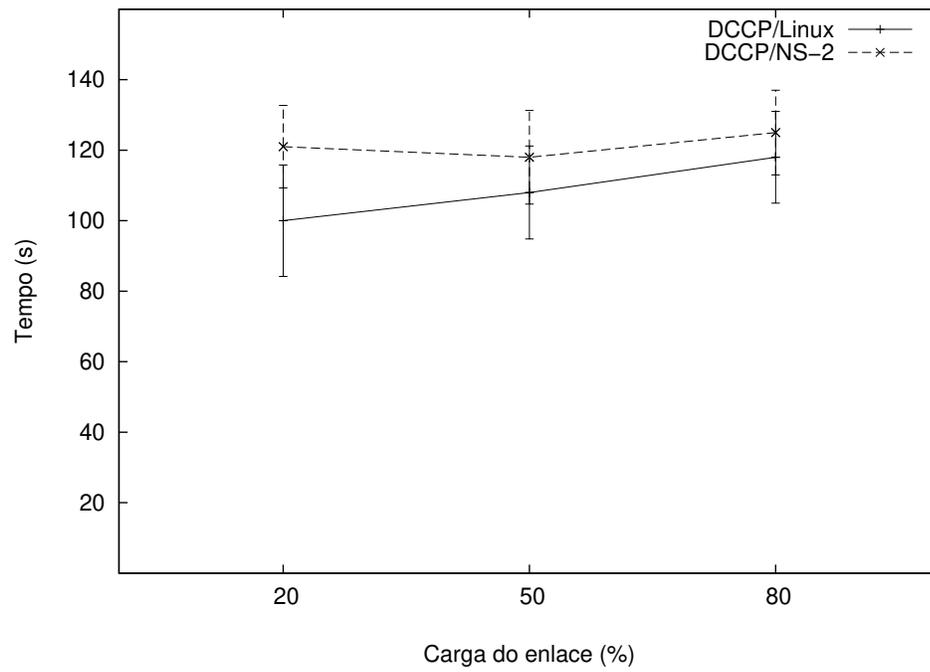


Figura 5.8: Resultados da avaliação da convergência

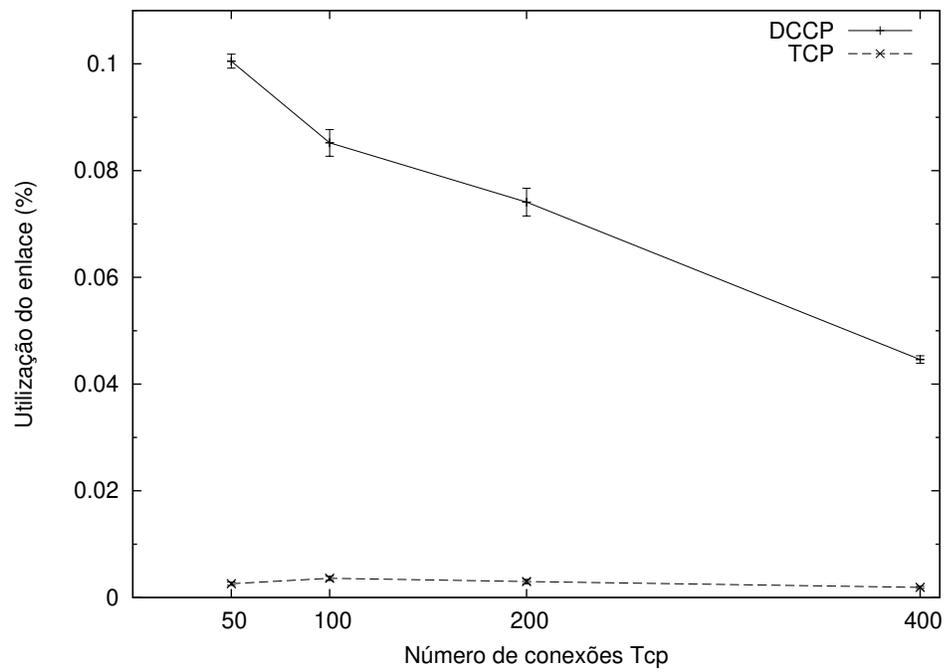


Figura 5.9: DCCP/TFRC x TCP-Reno (Vazão)

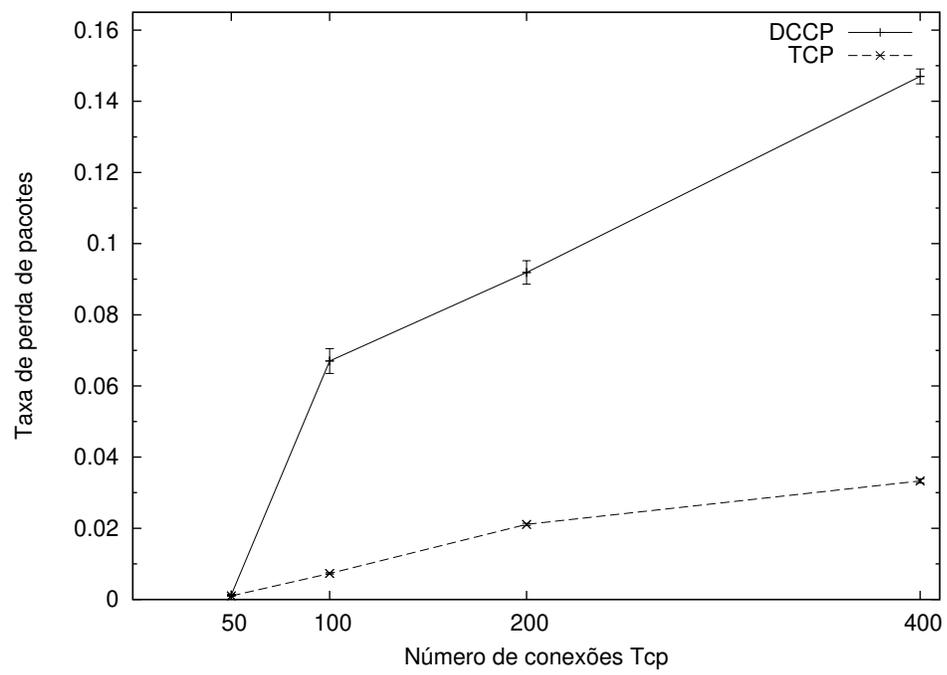


Figura 5.10: DCCP/TFRC x TCP-Reno (Perdas)

# Capítulo 6

## Variação do protocolo DCCP

O presente capítulo introduz uma variação do protocolo DCCP para redes de alta velocidade, chamada *Fast DCCP*, que visa sanar alguns dos problemas observados na avaliação do protocolo DCCP apresentada no capítulo 5. O protocolo *Fast DCCP*, tem como objetivo melhorar as deficiências do protocolo DCCP, a fim de promover escalabilidade em função do aumento da capacidade de banda passante no enlace.

O protocolo *Fast DCCP* propõe duas mudanças no mecanismo de controle de congestionamento TFRC, do protocolo DCCP. A primeira alteração é a inclusão de um fator de correção no cálculo da taxa de transmissão, que permite ao TFRC aumentar esta taxa de maneira mais eficiente. A segunda alteração trata da modificação da função de atualização da taxa de transmissão do TFRC, quando perdas são detectadas.

O fator de correção proposto para o mecanismo TFRC do protocolo *Fast DCCP* é calculado através do uso de amostragens da variação de atraso, que são utilizadas para realização do cálculo da nova taxa de transmissão. O novo mecanismo proposto, embora seja distinto, foi inspirado em dois mecanismos existentes no protocolo *FAST TCP*, o estimador de banda e o mecanismo para o acréscimo da janela de transmissão.

Conexões do protocolo DCCP contabilizam o valor de duas variáveis: o atraso médio de propagação para a conexão ( $R$ ) e o atraso atual ( $R_{sample}$ ), que considera o atraso de propagação verificado na última interação entre emissor e receptor. O cálculo desses valores, definidos em [7], é realizado de acordo com as expressões abaixo:

$$R_{i+1} = (0.9 * R_i) + (1-0.9) * R_{sample} \quad (6.1)$$

$$R_{sample} = (t_{now} - t_{recvdata}) - t_{delay} \quad (6.2)$$

onde  $t_{now}$  é o momento em que o pacote é recebido pelo emissor,  $t_{recvdata}$  é o timestamp do último pacote de dados recebido pelo receptor e  $t_{delay}$  é o tempo decorrido

entre o recebimento do último pacote e a geração do pacote de confirmação observado no receptor.

O fator de correção da taxa de transmissão, denotado por  $\alpha$ , proposto para o protocolo *Fast DCCP*, considera a razão entre o atraso médio para a conexão ( $R$ ) e o atraso atual ( $R_{sample}$ ), o que possibilita acréscimos multiplicativos da taxa de transmissão, ao invés de aditivos como no DCCP, permitindo que se alcance maior escalabilidade em função do aumento da banda. Valores de  $\alpha$  superiores a 1, indicam a existência de recursos na rede, tendo em vista que o atraso de propagação atual ( $R_{sample}$ ) é inferior ao atraso médio de propagação para a conexão ( $R$ ).

O fator de correção é utilizado para calcular um novo valor para a taxa de transmissão ( $X_{corrigido}$ ) do protocolo *Fast DCCP*, através de:

$$X_{corrigido} = ((X * \alpha) + X); \quad (6.3)$$

onde  $X$  é o valor atual da taxa de transmissão. De acordo com [7], o mecanismo TFRC do protocolo DCCP efetua o cálculo da taxa de transmissão, quando não há ocorrência de perdas, da seguinte maneira:

$$X = \max(\min(2 * X, 2 * X_{recv}), s/R); \quad (6.4)$$

onde  $X_{recv}$  é a taxa de transmissão informada pelo receptor e  $s$  é o pacote definido para conexão DCCP, conforme descrito em [7]. O valor da taxa de transmissão do protocolo *Fast DCCP* considera o valor máximo entre as taxas calculadas pelo TFRC do protocolo DCCP ( $2 * X$  e  $2 * X_{recv}$ ) e a taxa calculada através do fator de correção ( $X_{corrigido}$ ), conforme a equação:

$$X = \max(\max(X_{corrigido}, 2 * X, 2 * X_{recv}), s/R); \quad (6.5)$$

Assim, sempre que houver disponibilidade de recursos na rede ( $\alpha > 1$ ), o valor para a taxa de transmissão calculada com o uso do fator de correção ( $X_{corrigido}$ ) será superior aos demais valores das taxas calculados pelo TFRC, o que permite que o protocolo *Fast DCCP* utilize de maneira eficiente os recursos disponíveis.

A falta de escalabilidade do protocolo DCCP em função do aumento de banda nos enlaces reportada no capítulo 5, além de evidenciar a necessidade de um estimador de banda para que se possa avaliar o quanto a taxa de transmissão pode crescer, permitiu também verificar que a capacidade de crescimento da taxa de transmissão do protocolo DCCP é significamente impactada pela ocorrência de perdas na conexão, pois sempre que uma perda é detectada o emissor DCCP reduz a sua taxa de transmissão de acordo com:

$$X = \max(\min(X_{calc}, 2 * X_{recv}), s/t_{mbi}); \quad (6.6)$$

onde  $X_{calc}$  é a taxa de transmissão calculada pela Equação 4.1 e  $s/t_{mbi}$  representa um pacote ( $s$ ) enviado a cada 64 segundos (valor da constante  $t_{mbi}$  definido em [7]).

O outro aspecto determinante para a escalabilidade é a reação frente a ocorrência de perdas. O mecanismo de controle de congestionamento TFRC foi concebido para ser amigável ao protocolo TCP-Reno. Consequentemente, diminui, de maneira considerável, sua taxa de transmissão na existência de congestionamento, indicada pela ocorrência de perdas. Assim, espera-se que fluxos TCP-Reno e DCCP possam competir em igualdade de condições pelos recursos disponíveis no enlace. Quando um protocolo opera em redes de alta velocidade, a redução acentuada da taxa de transmissão leva à ineficiência em se utilizar a banda disponível dado o longo tempo necessário para se retornar o valor da taxa de transmissão no momento da perda.

A segunda alteração proposta para o mecanismo TFRC modifica o cálculo da taxa de transmissão descrita na Equação 6.6, a fim de evitar reduções abruptas da taxa de transmissão. O cálculo da taxa de transmissão, proposto para o protocolo *Fast DCCP*, quando ocorrências de perda são observadas, é definido por:

$$X = \max(\max(X_{calc}, 2 * X_{recv}), s/t_{mbi}); \quad (6.7)$$

A taxa de transmissão, portanto, será atualizada com o valor máximo e não com o valor mínimo, dentre os valores calculados, pelo TFRC, para nova taxa de transmissão ( $X_{calc}$  ou  $2 * X_{recv}$ ). O novo mecanismo de controle de congestionamento disponível no protocolo *Fast DCCP*, contendo as alterações descritas nesta seção, será chamado de *Fast TFRC*.

## 6.1 Propriedades Avaliadas

Para avaliar a eficiência do protocolo *Fast DCCP*, quando este opera em redes de alta velocidade, métricas específicas foram medidas, tanto nos experimentos de simulação quanto nos de medição. Os resultados obtidos foram comparados com os resultados obtidos pelo protocolo DCCP original. São elas: escalabilidade, justiça, convergência e compatibilidade com o protocolo TCP-Reno. A avaliação destas métricas são recomendadas em [9].

## 6.2 Avaliação de desempenho

Os cenários e métricas utilizados para avaliação do protocolo *Fast DCCP* em redes de alta velocidade foram derivados a partir do proposto em [9]. Todos os experimentos utilizam a topologia *Dumbbell*, que contém um único enlace gargalo constituído por dois roteadores

por onde passam conexões. O número de conexões varia, de acordo com os experimentos realizados.

Nos experimentos de simulação foi utilizado o simulador de redes NS-2 [35], versão 2.31. Para os experimentos de medição foi utilizado o sistema operacional Linux, com o kernel da versão 2.6.20. O módulo do DCCP contido no NS-2 e a implementação do protocolo DCCP contida no sistema operacional Linux foram alterados, para corresponder a variação proposta, o protocolo *Fast DCCP*.

Nos experimentos de simulação com o NS-2, os dois roteadores utilizam gerenciamento de fila *droptail* e possuem *buffer* de 2500 pacotes, valor este semelhante ao tamanho de buffers em roteadores reais [50]. Para evitar impactos relacionados à sincronização das conexões nas simulações, os instantes dos inícios das transmissões e os atrasos de propagação para cada uma das conexões foram gerados aleatoriamente. Os instantes dos inícios variaram entre 1 e 10s e o atraso de propagação entre 5 e 15ms. Tráfego *background* foi introduzido, visando refletir composição de tráfego em ambiente de redes reais, conforme proposto em [19]. O tráfego *background* utiliza 20%, 50% ou 80% da capacidade do enlace gargalo, no sentido reverso, de acordo com a distribuição [3]: 20% de tráfego UDP, 56% de tráfego Web e 24% de tráfego FTP. O tráfego utilizado pelas conexões foi do tipo CBR, com a taxa de transmissão de cada conexão ajustada para que a mesma faça uso de toda a capacidade de transmissão do enlace gargalo estipulado no cenário do experimento.

Nos experimentos de medição, foram utilizados dois computadores com interfaces ethernet com capacidade de 1Gbps, interligados através de um *switch router* com capacidade de 1Gbps por porta, sendo que cada uma das portas do *switch* estava com uma rota estática habilitada, emulando dois roteadores, já que as portas possuem um processador de pacotes e filas independentes. Todas as métricas de distribuição de tráfego *background* e limitações da capacidade do enlace físico foram obtidas através do uso da ferramenta *tc*, *Traffic Control* [47], e para o envio de dados das conexões *Fast DCCP* e TCP foi utilizado a ferramenta *Iperf* [22], sempre com transmissão de tráfego CBR, com tempos para os instantes dos inícios variando entre 1 e 10s.

Os gráficos apresentados mostram o valor médio obtido à partir da execução de 6 replicações para cada experimento. Para o cálculo do intervalo de confiança, foram utilizados o método de replicações independente com nível de confiança de 95%. O tempo de execução definido para todos os experimentos foi de 600s. Os resultados da avaliação do protocolo DCCP, obtidos anteriormente no capítulo 5 serão comentados, para que se possa comparar com os da avaliação do protocolo *Fast DCCP*.

### 6.2.1 Escalabilidade

Para avaliar a escalabilidade do protocolo *Fast DCCP* foram elaborados dois cenários distintos: um para avaliar a escalabilidade em função do aumento da banda passante dos enlaces e outro em função do número de conexões. Para avaliar a escalabilidade em função da capacidade de transmissão do enlace gargalo, variou-se a capacidade no seguinte conjunto: [155Mbps, 622Mbps, 1Gbps]. O atraso do enlace foi fixado em 100ms e o número de conexões *Fast DCCP* utilizado foi 16. A Figura 6.1 e a Figura 6.2 apresentam, respectivamente, os resultados obtidos para os experimentos de simulação e de medição. Em ambos os gráficos, pode-se perceber um comportamento semelhante. As 16 conexões *Fast DCCP* utilizam eficientemente o enlace (utilização superior a 0.9) para todas as capacidades de enlaces avaliadas. Analisando os resultados obtidos para a avaliação do protocolo DCCP do capítulo 5, pode-se observar que a medida que a capacidade do enlace aumenta, a utilização é reduzida de maneira significativa, chegando a ser 0.6, para a capacidade de 1Gbps nos experimentos de medição, enquanto que para o protocolo *Fast DCCP* esta utilização é superior a 0.9 em todos os cenários. Pode-se concluir que o protocolo *Fast DCCP* é escalável em função do aumento de disponibilidade de banda. O fator de correção para o cálculo da taxa de transmissão empregado pelo mecanismo *Fast TFRC* do *Fast DCCP* permite que a banda disponível seja utilizada de maneira eficiente.

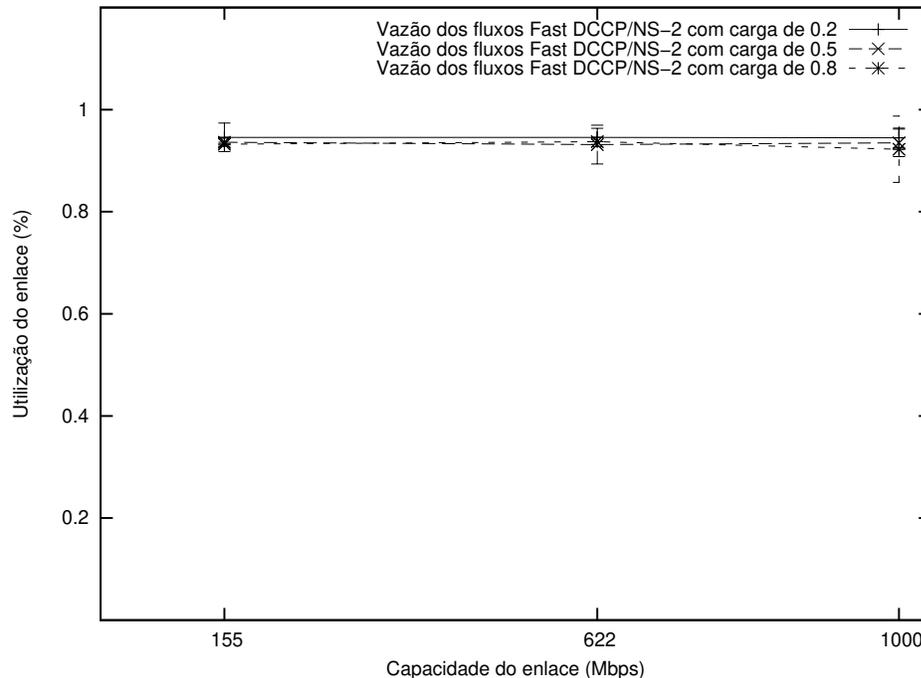


Figura 6.1: Utilização do enlace em função da variação da banda (NS-2)

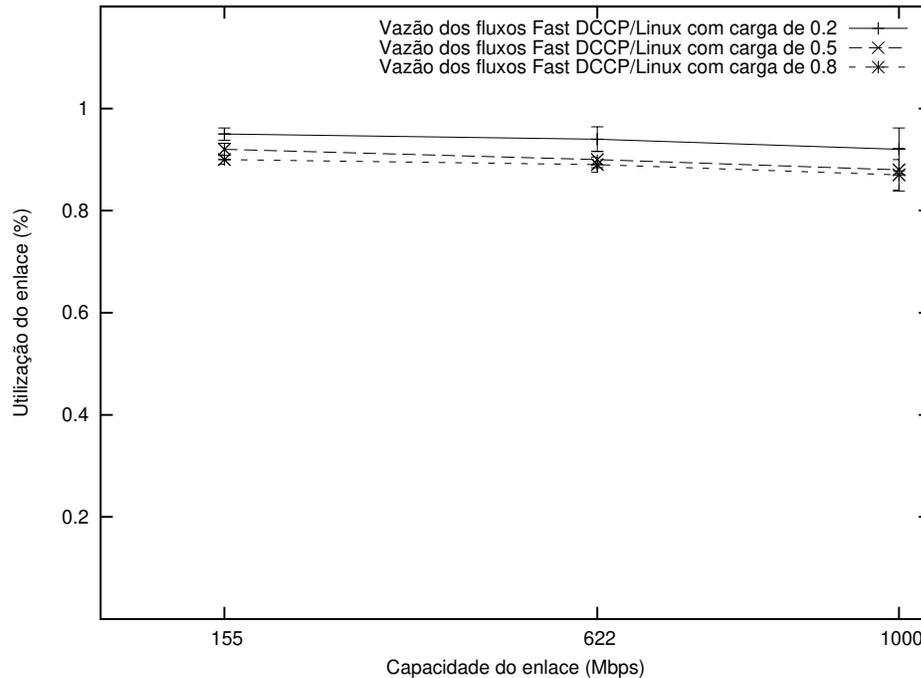


Figura 6.2: Utilização do enlace em função da variação da banda (Linux)

Para avaliar a escalabilidade em função do número de conexões, foram realizados experimentos com o número de conexões *Fast DCCP* variando entre 2 e 32 ( $2^1, 2^2, 2^3, \dots, 2^4$ ), para os experimentos de medição, e variando entre 2 e 256 ( $2^1, 2^2, 2^3, \dots, 2^8$ ), para os experimentos de simulação. A diferença na variação do número de conexões entre os experimentos deve-se às limitações de hardware presente nos dispositivos utilizados para realização dos experimentos de medição. A capacidade de transmissão do enlace foi fixado em 1Gbps e o atraso de propagação em 100ms. A Figura 6.3 e a Figura 6.4 apresentam, respectivamente, os resultados da avaliação da escalabilidade em função da variação do número de conexões, para os experimentos de simulação e de medição. Nota-se que na medida que o número de conexões aumenta, nos experimentos de simulação, a utilização do enlace permanece estável e o tráfego *background* implica em apenas mudanças discretas na utilização. Analisando os resultados obtidos para a avaliação do protocolo DCCP no capítulo 5, nota-se que valores semelhantes aos do protocolo *Fast DCCP* foram obtidos, quanto ao aumento do número de conexões. A única diferença que se nota entre os experimentos do protocolo DCCP e o protocolo *Fast DCCP*, é que para este último a utilização do enlace é maior. O protocolo *Fast DCCP* opera de maneira escalável e eficiente em relação ao aumento do número de conexões, demonstrando que o mecanismo de controle de congestionamento empregado pelo *Fast TFRC* garante a escalabilidade em função do número de conexões.

### 6.2.2 Justiça

Para avaliar a justiça do *Fast DCCP* foram definidos dois cenários de simulação, ambos com 4 conexões *Fast DCCP* transmitindo no enlace gargalo, atraso de propagação de 100ms e capacidade do enlace de 1Gbps. No primeiro cenário, os fluxos *Fast DCCP* são iniciados, no mesmo instante de tempo (no instante 1s), enquanto que no segundo cenário os fluxos *Fast DCCP* são iniciados, em instantes de tempo distintos (aleatoriamente entre 1s e 10s).

A Figura 6.5 apresenta os resultados da avaliação de justiça para o primeiro cenário proposto, tanto para os experimentos de simulação quanto para os de medição. Nota-se, em ambos os casos, independentemente da intensidade do tráfego *background*, os valores do *Jain Fairness Index* ficaram próximos a 0.9, o que indica que o protocolo produz justiça, ou seja, o *Fast DCCP* provê distribuição de banda equilibrada entre os fluxos que competem pelo enlace. A Figura 6.6 apresenta os resultados da avaliação de justiça para o segundo cenário, tanto para os experimentos de simulação quanto os de medição. Os valores são similares aos obtidos no cenário anterior, reforçando a propriedade de justiça existente no protocolo. Analisando os resultados obtidos para a avaliação do protocolo DCCP no capítulo 5, pode-se observar semelhança entre estes gráficos e os obtidos na avaliação do protocolo *Fast DCCP*, o que indica que o *Fast DCCP* possui as propriedades de justiça observadas no protocolo DCCP.

### 6.2.3 Convergência

Para avaliar a convergência do protocolo *Fast DCCP*, foram realizados experimentos com a seguinte configuração: enlace gargalo fixado em 1Gbps, duas conexões *Fast DCCP*, o atraso de propagação no enlace gargalo em 100ms. A segunda conexão *Fast DCCP* inicia a transmissão no instante 50s, tempo suficiente para a primeira conexão alocar toda a banda disponível do enlace gargalo. A Figura 6.7 apresenta os resultados da avaliação da convergência, tanto para os experimentos de simulação quanto para os de medição, para diferentes intensidades de tráfego *background*. O tempo de convergência é alto, em ambos os casos, apresentando valores entre 100s e 135s, que são, tipicamente, muito superiores ao tempo de duração médio de uma conexão na Internet. Quando se compara estes resultados com os resultados obtidos na avaliação do protocolo DCCP no capítulo 5, nota-se valores similares no tempo de convergência; que para o protocolo DCCP estes variam entre 90s e 120s. Pode-se dizer que o mecanismo de controle de congestionamento implementado pelo *Fast TFRC*, mesmo produzindo altos graus de justiça, não consegue prover uma resposta rápida a divisão igualitária da banda da rede entre fluxos que competem pelo enlace. Enfatiza-se que estudos anteriores, [2] e [48], apontam o problema do tempo de resposta alto do mecanismo de congestionamento provido pelo TFRC. A variação do

protocolo DCCP, o protocolo *Fast DCCP*, apresenta as mesmas deficiências, uma vez que seu mecanismo de controle de congestionamento, o *Fast TFRC*, é uma variação do mecanismo TFRC. Nota-se que os valores do tempo de convergência observados para o protocolo *Fast DCCP* são, em média, cerca de 7% maiores do que os valores observados para o protocolo DCCP. Isto se deve ao fato da variação do TFRC, chamada *Fast TFRC*, presente no protocolo *Fast DCCP* possuir um mecanismo mais agressivo para aumento da taxa de transmissão e um mecanismo mais tolerante para diminuição da taxa de transmissão.

#### 6.2.4 Compatibilidade com o TCP-Reno

Para avaliar a compatibilidade do protocolo *Fast DCCP* com o protocolo TCP-Reno, foram realizados experimentos com dois cenários distintos. Em um deles, dois fluxos TCP-Reno, transmitindo tráfego FTP competem pela banda do enlace gargalo e no outro um fluxo *Fast DCCP* e um fluxo TCP-Reno, com o mesmo tráfego FTP do experimento anterior, competem pelo enlace gargalo. A capacidade do enlace usada foi 1Gbps, o tempo de propagação foi de 10ms. O objetivo destes experimentos é avaliar se o TCP-Reno obtém a mesma banda do enlace que seria obtida se o mesmo não estivesse concorrendo com o protocolo *Fast DCCP*. Na Tabela 6.1, são apresentados os resultados da avaliação de compatibilidade do *Fast DCCP* com o TCP-Reno, derivados nos experimentos de simulação. A utilização do enlace para TCP-1 apresenta os valores obtidos pelo fluxo TCP, quando este concorre com outro fluxo TCP, enquanto que os resultados TCP-2 apresentam os valores obtidos pelo fluxo TCP, quando este concorre com um fluxo *Fast DCCP*. Os diferentes valores de utilização observadas em ambos os fluxos TCP sugerem a incompatibilidade entre o protocolo *Fast DCCP* com o protocolo TCP-Reno, uma vez que o fluxo TCP-2 obtém uma utilização inferior a obtida pelo fluxo TCP-1.

Na Tabela 6.2, são apresentados os resultados da avaliação de compatibilidade do *Fast DCCP* com o TCP-Reno, para os experimentos de medição. A utilização do enlace para TCP-1 apresenta os valores obtidos pelo fluxo TCP, quando este concorre com outro fluxo TCP, enquanto que os resultados TCP-2 apresentam os valores obtidos pelo fluxo TCP, quando este concorre com um fluxo *Fast DCCP*. Os diferentes valores de utilização sugerem a incompatibilidade entre o protocolo *Fast DCCP* com o protocolo TCP-Reno, dado que a utilização do fluxo TCP-2 é inferior a obtida pelo fluxo TCP-1.

Analisando os resultados obtidos para a avaliação do protocolo DCCP no capítulo 5, pode-se observar valores similares para a avaliação de compatibilidade entre o protocolo DCCP e o protocolo TCP-Reno. Nota-se a mesma incompatibilidade entre os fluxos de ambos os protocolos. Tal incompatibilidade já havia sido descrita em [42]. Nota-se, no entanto, que a incompatibilidade entre o protocolo TCP-Reno e o protocolo *Fast*

|                          |       | Tráfego Background (%) |        |        |
|--------------------------|-------|------------------------|--------|--------|
|                          |       | 0.2                    | 0.5    | 0.8    |
| Utilização do enlace (%) | TCP-1 | 0.0152                 | 0.0124 | 0.0061 |
|                          | TCP-2 | 0.0022                 | 0.0017 | 0.0009 |

Tabela 6.1: Taxa de utilização obtida pelos fluxos TCP (NS-2)

|                          |       | Tráfego Background (%) |        |        |
|--------------------------|-------|------------------------|--------|--------|
|                          |       | 0.2                    | 0.5    | 0.8    |
| Utilização do enlace (%) | TCP-1 | 0.3120                 | 0.2210 | 0.1830 |
|                          | TCP-2 | 0.0820                 | 0.0545 | 0.0433 |

Tabela 6.2: Taxa de utilização obtida pelos fluxos TCP (Linux)

*DCCP* tornou-se mais acentuada, uma vez que os mecanismos para crescimento da taxa de banda deste protocolo são mais agressivos do que os mecanismos empregados pelo protocolo *DCCP*. Vale ressaltar que outros protocolos para redes de alta velocidade, como por exemplo *FAST TCP*, apresentam incompatibilidades similares quando seus fluxos disputam por recursos com fluxos do protocolo TCP-Reno, [24], [31] e [21].

### 6.2.5 Resumo Conclusivo

Os resultados da avaliação de escalabilidade em função da variação da capacidade da banda do enlace apontam para a eficiência do protocolo *Fast DCCP* em obter o uso eficiente dos recursos disponíveis no enlace gargalo. Os resultados da avaliação de escalabilidade em função do número de conexões demonstram que o protocolo *Fast DCCP* é eficiente na manutenção da justiça entre fluxos que competem pelo enlace gargalo, em função do número de fluxos. Os resultados da avaliação de convergência indicam que o protocolo *Fast DCCP*, a exemplo do protocolo *DCCP*, possui convergência lenta, para um estado de equilíbrio e justiça entre fluxos que disputam o enlace gargalo, com valores, em média, próximos a 115s.

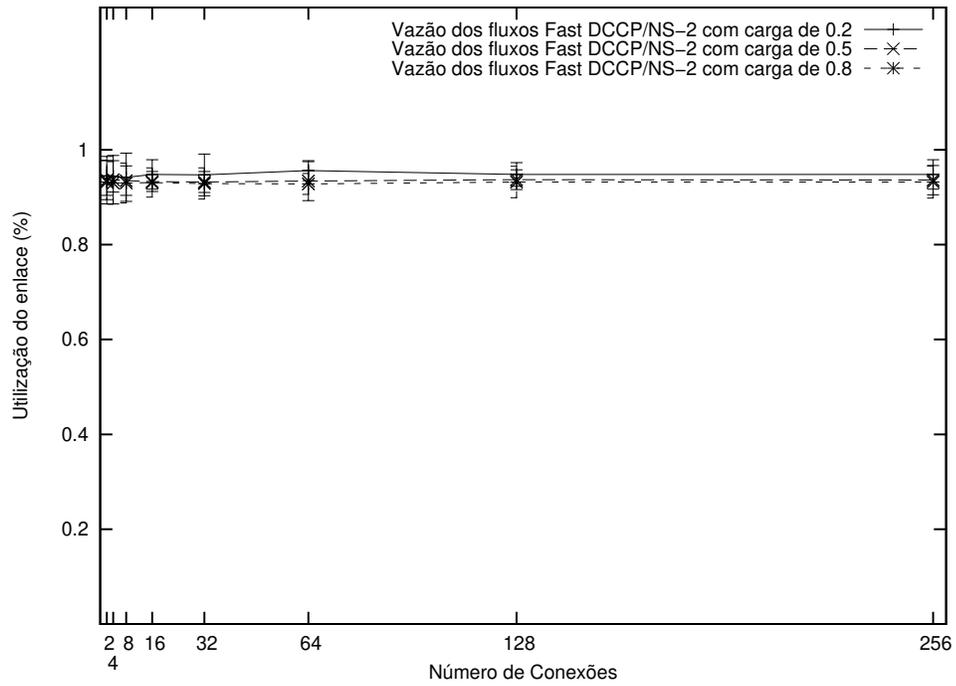


Figura 6.3: Utilização do enlace em função do número de conexões (NS-2)

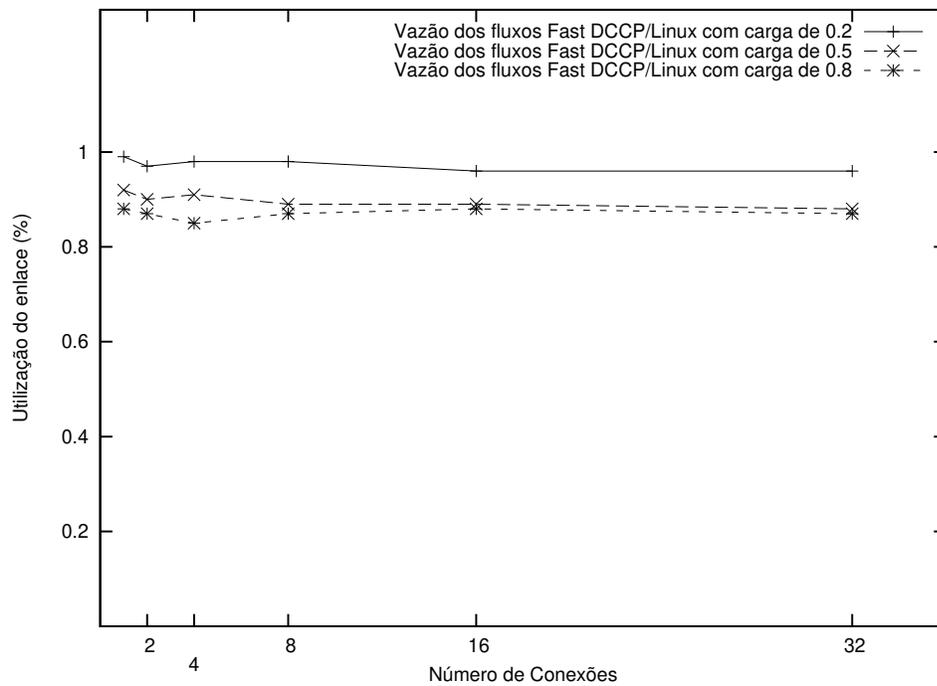


Figura 6.4: Utilização do enlace em função do número de conexões (Linux)

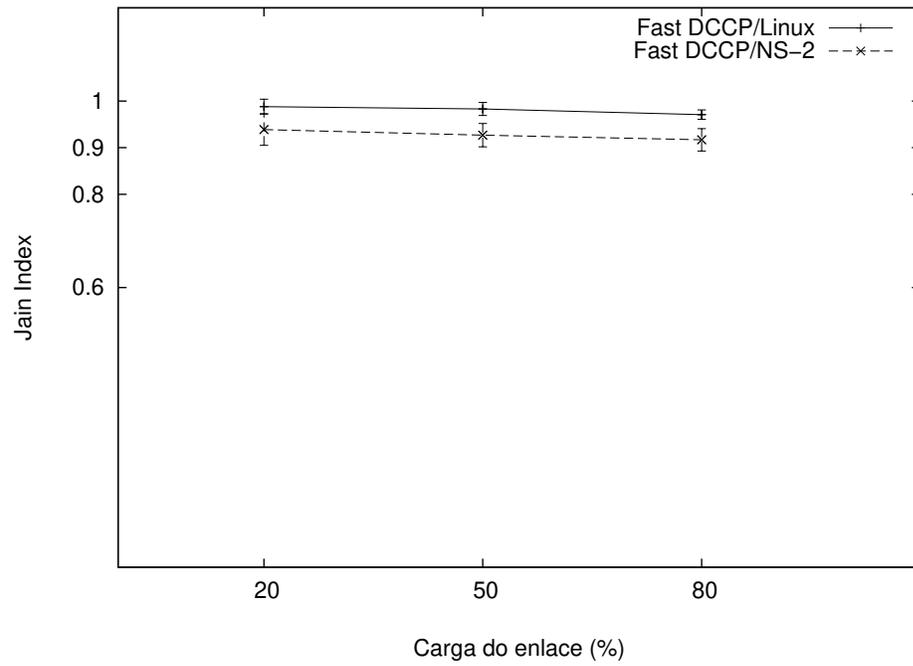


Figura 6.5: Primeiro Cenário de Justiça

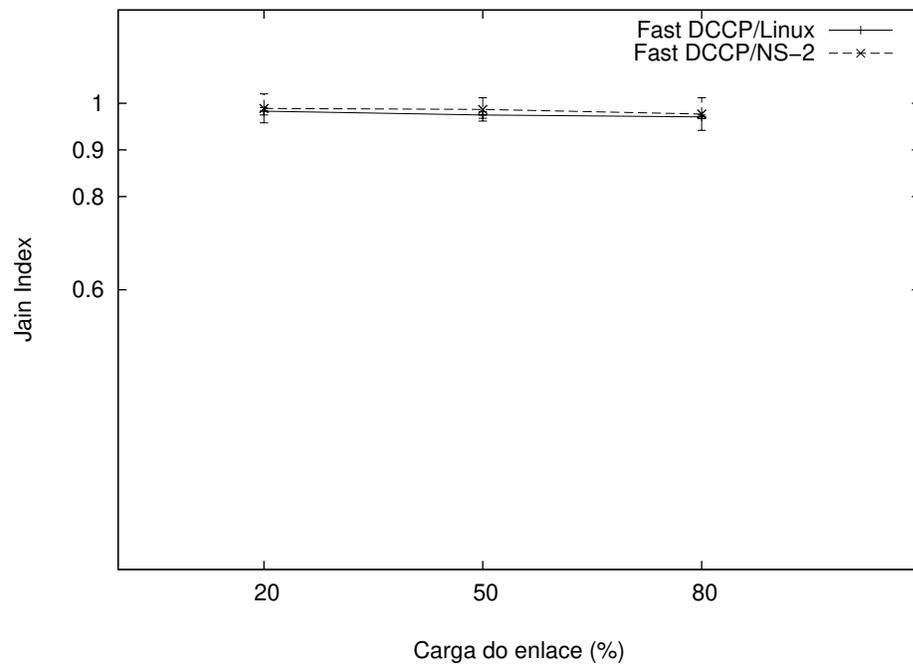


Figura 6.6: Segundo Cenário de Justiça

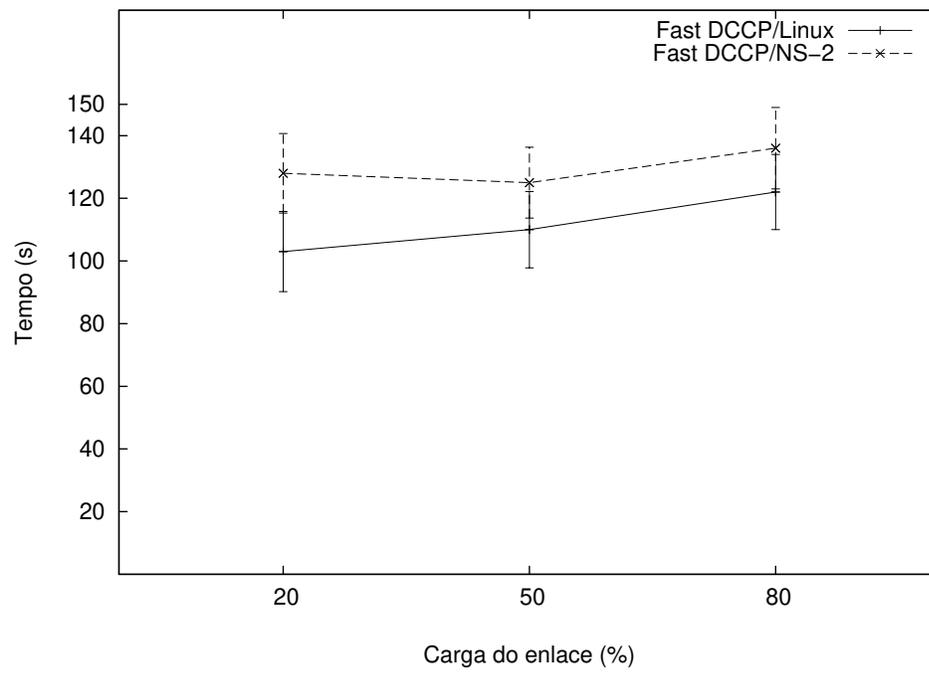


Figura 6.7: Resultados da avaliação da convergência

# Capítulo 7

## Conclusões e Trabalhos Futuros

Esta Dissertação apresentou uma avaliação do protocolo DCCP, quando este opera em redes de alta velocidade. Com o resultado da avaliação de desempenho foi possível identificar algumas deficiências do protocolo DCCP, com o controle de congestionamento TFRC, em redes de alta velocidade.

A avaliação da escalabilidade em função do aumento de disponibilidade de banda, mostrou a deficiência do protocolo em fazer uso dos recursos disponíveis, dada a ausência de um mecanismo, que atue em conjunto com o TFRC, para estimar a disponibilidade de banda, a fim de que se possa garantir um aumento da taxa de transmissão dos fluxos.

Em resposta a esta deficiência foi proposta uma variação do protocolo DCCP, *Fast DCCP*, que objetiva garantir o uso eficiente dos recursos disponíveis nos enlaces de alta velocidade. Com o resultado da avaliação de desempenho foi possível comprovar a eficiência do protocolo *Fast DCCP* em resolver a limitação de escalabilidade em função do aumento de banda dos enlaces, característica mandatória para o sucesso do protocolo em redes de alta velocidade. Fica evidente que o mecanismo que estima banda, baseado nos tempos de atraso amostrados em um conexão *Fast DCCP* vigente, contribui para o uso mais eficiente do enlace.

Os resultados dos outros pontos avaliados para o protocolo *Fast DCCP* no capítulo 6 mostram a semelhança destes com os resultados obtidos na avaliação do protocolo DCCP no capítulo 5.

Os resultados da avaliação de compatibilidade entre os protocolos *Fast DCCP* e TCP-Reno indicam o desbalanceamento nas taxas de transmissão obtidas pelos fluxos de ambos os protocolos que disputam o enlace de gargalo. O desbalanceamento já havia sido reportado no capítulo 5, quando fluxos DCCP competiam com fluxos TCP-Reno, porém a intensidade do desbalanceamento ficou mais acentuada quando da utilização do protocolo *Fast DCCP*, pois este emprega mecanismos mais agressivos para aumento da taxa de transmissão.

Os resultados da avaliação de justiça demonstram a eficiência do protocolo *Fast DCCP* em garantir que diferentes fluxos *Fast DCCP* possam competir, de maneira equilibrada e justa, pelos recursos do enlace de gargalo.

Os resultados da avaliação da convergência apontam para a ineficiência do protocolo em convergir para uma distribuição justa de banda passante entre os fluxos concorrentes. Para acelerar o tempo de convergência, é necessário avaliar de maneira mais detalhada as deficiências descritas em [2] e [48], a fim de propor melhorias no TFRC que permitam tempos de convergência entre fluxos concorrentes mais razoáveis, assim como um tempo de resposta ao congestionamento mais eficiente. Como trabalho futuro, sugere-se investigar com mais detalhes os pontos que envolvem o tempo de convergência do protocolo, de forma a avaliar potenciais alterações no protocolo *Fast DCCP*.

Foi possível constatar que o *Fast DCCP* foi bem-sucedido em sanar a deficiência de escalabilidade em função do aumento de banda passante, sem degradar outras métricas observadas no protocolo DCCP, tais como: justiça, escalabilidade em função do número de conexões e convergência.

# Referências Bibliográficas

- [1] L. Andrew, C. Marcondes, L. Dunn, R. Guillier, W. Gang, L. Eggert, S. Ha, and I. Rhee. Towards a common tcp evaluation suite. In *Proceedings of PFLDnet08*, 2008.
- [2] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic behavior of slowly-responsive congestion control algorithms. In *Proceedings of ACM SIGCOMM'01*, pages 263–274, San Diego,CA, 2001.
- [3] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of ACM SIGMETRICS'98*, pages 151–160, 1998.
- [4] S. Bhattacharyya, D. Towsley, and J. Kurose. A novel loss indication filtering approach for multicast congestion control. In *Journal of Computer Communications, Special Issue on Multicast*, 2000.
- [5] H. Bulot, R. Les Cottrell, and R. Hughes-Jones. Evaluation of advanced tcp stacks on fast long-distance production networks. *Grid Comput*, 4(1):345–359, 2003.
- [6] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE Transactions on Network*, 7(4):458–472, 1999.
- [7] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Tcp friendly rate control (tfrc): Protocol specification. In *RFC 3448*, 2006.
- [8] S. Floyd and E. Kohler. Profile for datagram congestion control protocol (dccc) - congestion control id 2: Tcp-like congestion control. In *RFC 4341*, 2006.
- [9] S. Floyd and E. Kohler. Tools for the evaluation of simulation and testbed scenarios. In *Internet Draft: draft-irtf-tmrg-tools-05*, 2006.
- [10] S. Floyd, E. Kohler, and M. Handley. Designing dccc: Congestion control without reliability. In *Proceedings of ACM SIGCOMM'06*, pages 27–38, Pisa, Itália, 2006.

- [11] S. Floyd, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *Proceedings of ACM SIGCOMM'00*, pages 43–56, 2000.
- [12] S. Floyd, S. Ratnasamy, and S. Shenker. Modifying tcp's congestion control for high speeds. In *Internet-Draft*, 2006.
- [13] C. Froldi and N. Fonseca. A dccp variant for high speed networks. In *IEEE 3rd Latin-American Conference on Communications 2011*, pages 1–6, Belém, Brasil, 2011.
- [14] C. Froldi, N. Fonseca, and C. Papotti. Avaliação de desempenho do protocolo dccp para redes de alta velocidade. In *XXX CSBC*, pages 1831–1844, Belo Horizonte, Brasil, 2010.
- [15] C. Froldi, N. Fonseca, and C. Papotti. Fast dccp: Uma variante do protocolo dccp para redes de alta velocidade. In *XXIX SBRC*, pages 761–774, Campo Grande, Brasil, 2011.
- [16] C. Froldi, N. Fonseca, C. Papotti, and D. Manzato. Performance evaluation of the dccp protocol in high-speed networks. In *Proceedings of IEEE CAMAD 2010*, pages 41–46, Miami, USA, 2010.
- [17] G.Sawar, E. Lochin, and R. Boreli. Experimental performance of dccp over live satellite and long range wireless links. In *Proceedings of IEEE ICSIT'07*, Sydney, Australia, 2007.
- [18] X. Gu, P. Di, and L. Wolf. Performance evaluation of dccp: A focus on smoothness and tcp-friendliness. In *Annals of Telecommunications Journal*, 61, pages 191–216, 2006.
- [19] S. Ha, Y. Kim, L. Le, I. Rhee, and L. Xu. A step toward realistic performance evaluation of high-speed tcp variants. In *Proceedings of PFLDnet06*, 2006.
- [20] S. Ha, I. Rhee, and L. Xu. Cubic:a new tcp-friendly high-speed tcp variant. In *Proceedings of ACM SIGOPS*, pages 64–74, 2008.
- [21] T. Hatano, H. Shigeno, and K. Okada. Tcp-friendly congestion control for highspeed network. In *Proceedings of IEEE SAINT'07*, pages 1–10, 2007.
- [22] <http://www.noc.ucf.edu/tools/iperf>, 2010.
- [23] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley & Sons, 1991.

- [24] H. Jamal and K. Sultan. Performance analysis of tcp congestion control algorithms. In *INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS*, pages 30–38, 2008.
- [25] C. Jin, D. Wei, and S. Low. Fast tcp: Motivation, architecture, algorithms, performance. In *Proceedings of IEEE/ACM Transactions on Network 2006*, pages 1246–1259, 2004.
- [26] S. Kasera, S. Bhattacharyya, M. Keaton, D. Kiwior, S. Zabele, J. Kurose, and D. Towsley. Scalable fair reliable multicast using active services. *IEEE Network*, 14(1):48–57, 2000.
- [27] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *Proceedings of ACM SIGCOMM'02*, pages 19–23, 2002.
- [28] T. Kelly. Scalable tcp: Improving performance in highspeed wide area networks. In *Technical Report 2, CERN-IT Division*, 2003.
- [29] D. Leith, Y. Li, and R. Shorten. Experimental evaluation of tcp protocols for high-speed networks. *IEEE / ACM Transactions on Networking*, 15(5):1109–1122, 2008.
- [30] D. Leith and R. Shorten. H-tcp: Tcp for high-speed and long-distance networks. In *Proceedings of PFLDnet04*, 2004.
- [31] Y. Li, D. Leith, and R. Shorten. Experimental evaluation of tcp protocols for high-speed networks. In *ACM TRANSACTIONS ON NETWORKING, Vol. 15*, pages 1109–1122, 2007.
- [32] G. Marfia, C. Palazzi, G. Pau, M. Gerla, and M. Roccetti. Tcp-libra: Exploring rtt fairness for tcp. In *Technical Report TR-050037, UCLA-CSD*, 2005.
- [33] M. Nabeshima and K. Yata. Performance evaluation and comparison of transport protocols for fast long-distance networks. *IECE Transaction Communication*, 89(4):1273–1283, 2006.
- [34] P. Navaratnam, N. Akhtar, and R. Tafazolli. On the performance of dccp in wireless mesh networks. In *Proceedings of ACM MobiWac'06*, pages 144–147, 2006.
- [35] <http://www.isi.edu/nsnam/ns>, 2010.
- [36] J. Padhye, D. Kurose, and R. Towsley. A model based tcp-friendly rate control protocol. In *Proceedings of NOSSDAV'99*, 1999.

- [37] Jitendra Padhye, Victor Firoiu, and Donald Towsley. A survey on tcp-friendly congestion control. *IEEE/ACM Transactions on Network*, 8(2):133–145, 2000.
- [38] R. Rejaie, M. Handley, and D. Estrin. Rap: An end-to-end rate-based congestion control mechanism for realtime streams in the internet. In *Proceedings of IEEE INFOCOM*, pages 1337–1345, 1999.
- [39] I. Rhee, N. Balaguru, and G. Roukas. Mtcp: Scalable tcp-like congestion control for reliable multicast. In *Proceedings of IEEE INFOCOM*, pages 1265–1273, 1999.
- [40] I. Rhee, V. Ozdemir, and Y. Yi. Tear: Tcp emulation at receivers - flow control for multimedia streaming. In *Technical Reports, Dept. of Computer Science, NCSU*, 2000.
- [41] I. Rhee and L. Xu. Cubic: A new tcp-friendly high-speed tcp variant. In *Proceedings of PFLDnet05*, 2005.
- [42] Injong Rhee and Lisong Xu. Limitations of equation-based congestion control. *IEEE Transactions on Network*, 15(4):852–865, 2007.
- [43] L. Rizzo. Pgmcc: A tcp-friendly single-rate multicast congestion control. In *Proceedings of ACM SIGCOMM'00*, pages 17–28, 2000.
- [44] R. Shorten, D. Leith, and G. McCullagh. Experimental evaluation of cubic-tcp. In *Whitepaper, Hamilton Institute*, 2007.
- [45] D. Sisalem and A. Wolisz. Lda+ tcp-friendly adaptation: A measurement and comparison study. In *Proceedings of NOSSDAV'00*, 2000.
- [46] S. Takeuchi, H. Koga, K. Iida, Y. Kadobayashi, and S. Yamaguchi. Performance evaluations of dcep for bursty traffic in real-time applications. In *Proceedings of IEEE SAINT'05*, pages 142–149, 2005.
- [47] <http://tldp.org/howto/traffic-control-howto/index.html>, 2010.
- [48] M. Vojnovic and J. Le Boudec. On the long run behavior of equation-based rate control. In *Proceedings of ACM SIGCOMM'02*, pages 103–116, 2003.
- [49] H. Wang and M. Schwartz. Achieving bounded fairness for multicast and tcp traffic in the internet. In *Proceedings of ACM SIGCOMM'98*, pages 81–92, 1998.
- [50] David X. Wei, Pei Cao, and Steven H. Low. Time for a tcp benchmark suite? <http://www.cs.caltech.edu/~weixl/research/technical/benchmark/summary.ps>, 2005.

- [51] Joerg Widmer, Robert Denda, and Martin Mauve. A survey on tcp-friendly congestion control. *IEEE Network*, 15:28–37, 2001.
- [52] L. Xu, K. Harfoush, and I. Rhee. Binary increase congestion control (bic) for fast long-distance networks. In *Proceedings of IEEE INFOCOM*, pages 2514–2524, 2004.