

# Análise de Protocolos TCP para Redes de Alta Velocidade

Este exemplar corresponde à redação da Dissertação apresentada para a Banca Examinadora antes da defesa da Dissertação.

Campinas, 10 de Dezembro de 2008.



Prof. Dr. Nelson Luis Saldanha da Fonseca  
(Orientador)

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

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Michel, Neila Fernanda

M582a Análise de Protocolos TCP para redes de alta velocidade/Neila  
Fernanda Michel -- Campinas, [S.P. :s.n.], 2008.

Orientador : Nelson Luís Saldanha da Fonseca

Dissertação (mestrado) - Universidade Estadual de Campinas,  
Instituto de Computação.

1. TCP/IP (Protocolo de rede de computação) - Análise. 2. Redes  
de alta velocidade. 3. Produto banda-atraso elevado. I. Fonseca, Nelson  
Luís Saldanha da. II. Universidade Estadual de Campinas. Instituto de  
Computação. III. Título.

Título em inglês: Analysis of TCP Protocols for High Speed Networks

Palavras-chave em inglês (Keywords): 1. TCP/IP (Computer Network Protocol) – Analysis.  
2. High Speed Networks. 3. High Bandwidth-Delay Product.

Área de concentração: Redes de computadores

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Nelson Luís Saldanha da Fonseca (IC-UNICAMP)  
Prof. Dr. Luís Henrique Maciel Kosmowski Costa (UFRJ)  
Prof. Dr. Ricardo de Oliveira Anido (IC-UNICAMP)

Data da defesa: 10/12/2008

Programa de Pós-Graduação: Mestrado em Ciência da Computação

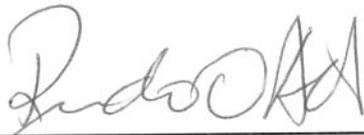
## TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 10 de dezembro de 2008, pela Banca examinadora composta pelos Professores Doutores:



---

**Prof. Dr. Luis Henrique Maciel Kosmalski Costa**  
UFRJ



---

**Prof. Dr. Ricardo de Oliveira Anido**  
IC/UNICAMP



---

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

# Análise de Protocolos TCP para Redes de Alta Velocidade

Neila Fernanda Michel<sup>1</sup>

Julho de 2009

**Banca Examinadora:**

- Prof. Dr. Nelson Luis Saldanha da Fonseca (Orientador)
- Prof. Dr. Ricardo de Oliveira Anido  
Unicamp - Instituto de Computação
- Prof. Dr. Luís Henrique Maciel Kosmowski Costa  
UFRJ
- Prof. Dr. Edmundo R. M. Madeira (Suplente)  
Unicamp - Instituto de Computação

---

<sup>1</sup>Suporte financeiro do CNPq, processo 134989/2005-8.

© Neila Fernanda Michel, 2009.  
Todos os direitos reservados.

# Resumo

O protocolo TCP Reno começou a operar em 01 de Janeiro de 1983. Desde sua concepção, houveram expressivos avanços nas tecnologias de transmissão de dados, como a introdução de enlaces ópticos e de satélites, que permitem altas taxas de transmissão.

O TCP Reno continua sendo o protocolo padrão na Internet. Entretanto existem inúmeras dificuldades na sua operação, que causam baixo desempenho em redes com produto banda-atraso elevado, devido ao mecanismo de atualização do tamanho da janela de congestionamento. Para superar suas deficiências, vários protocolos variantes do TCP Reno têm sido propostos com o objetivo de melhorar a utilização de recursos em redes de alta velocidade, e por esta razão são denominados “Protocolos para Redes de Alta Velocidade”.

Para que as proposições de variantes do TCP levem a protocolos que operem eficientemente em redes com alto valor do produto banda-atraso, é necessário que se avalie quais variações da dinâmica do crescimento da janela de transmissão são efetivas.

Nesta dissertação avalia-se um conjunto de protocolos variantes do TCP para redes de alta velocidade através de simulações. Diversas propriedades importantes para protocolos do tipo TCP são investigadas.

# Abstract

The importance of the Reno TCP protocol started operating in January 01<sup>st</sup>, 1983. Since this milestone, there were expressive advances on data transmission technologies, such as those of optical fibers and satellite channels which allow high transmission rates on long distances.

Reno TCP remains the Internet standard protocol. However, its operation presents several difficulties in networks with high bandwidth-delay product, producing poor performance due its window congestion mechanism. In order to overcome its deficiencies, several variants have been proposed with the aim of enhancing its utilization in high speed networks. These variants are generally called “High Speed Networks Protocols”.

Transmission characteristics, properties and performance evaluation are essential to the identification of problems and avoidance of poor performance.

In this dissertation a set of TCP variants protocols for high speed networks is evaluated. Several important properties of these protocols are investigated.

# Sumário

|  |            |
|--|------------|
| <b>Resumo</b>  | <b>vi</b>  |
| <b>Abstract</b>  | <b>vii</b> |
| <b>1 Introdução</b>  | <b>1</b>   |
| <b>2 Protocolos para transmissão de dados</b>                      | <b>5</b>   |
| 2.1 BIC TCP . . . . .  | 7          |
| 2.2 CUBIC TCP . . . . .  | 8          |
| 2.3 FAST TCP . . . . .   | 10         |
| 2.4 HighSpeed TCP . . . . .  | 11         |
| 2.5 H-TCP . . . . .  | 13         |
| 2.6 TCP Libra . . . . .  | 14         |
| 2.7 Scalable TCP . . . . .   | 16         |
| 2.8 Outras variantes do TCP . . . . .                              | 17         |
| 2.9 Classificação . . . . .  | 18         |
| <b>3 Trabalhos Relacionados</b>                                    | <b>20</b>  |
| <b>4 Comparação entre Protocolos para Redes de Alta Velocidade</b> | <b>24</b>  |
| 4.1 Topologia <i>Dumbbell</i> . . . . .                            | 26         |
| 4.1.1 Escalabilidade . . . . .                                     | 26         |
| 4.1.2 Compatibilidade com TCP Reno . . . . .                       | 33         |
| 4.1.3 Justiça . . . . .  | 34         |
| 4.1.4 Convergência . . . . .                                       | 39         |
| 4.1.5 Estabilidade . . . . .                                       | 40         |
| 4.1.6 Síntese: Topologia <i>Dumbbell</i> . . . . .                 | 42         |
| 4.2 Topologia <i>Parking-lot</i> . . . . .                         | 44         |
| 4.2.1 Escalabilidade . . . . .                                     | 44         |
| 4.2.2 Compatibilidade com TCP Reno . . . . .                       | 47         |

|          |   |           |
|----------|---|-----------|
| 4.2.3    | Justiça . . . . .   | 48        |
| 4.2.4    | Síntese: Topologia <i>Parking-lot</i> . . . . .                         | 51        |
| 4.3      | Análise comparativa do desempenho dos protocolos investigados . . . . . | 53        |
| <b>5</b> | <b>Considerações Finais e Conclusões</b>                                | <b>57</b> |
|          | <b>Bibliografia</b>   | <b>61</b> |

# Lista de Tabelas

|     |  |    |
|-----|--|----|
| 2.1 | Resumo das características dos protocolos variantes do TCP para redes de alta velocidade . . . . .                   | 19 |
| 4.1 | Variação na utilização do enlace para cada protocolo de alta velocidade. . .   | 32 |
| 4.2 | Justiça de RTT: Informações estatísticas . . . . .   | 37 |
| 4.3 | Vazão (Mbps) obtida pelos protocolos. . . . .  | 40 |
| 4.4 | Justiça intra-protocolos: Vazão média obtida por protocolo por RTT. . . .  | 50 |
| 4.5 | Vazão média obtida por protocolo por RTT. . . . .  | 52 |
| 4.6 | Notação utilizada para a classificar o desempenho dos protocolos em cada propriedade. . . . .                        | 53 |
| 4.7 | Classificação dos protocolos na topologia <i>Dumbbell</i> de acordo com o desempenho em cada propriedade. . . . .    | 54 |
| 4.8 | Classificação dos protocolos na topologia <i>Parking-lot</i> de acordo com o desempenho em cada propriedade. . . . . | 54 |

# Lista de Figuras

|      |  |    |
|------|--|----|
| 2.1  | Função de crescimento da janela do CUBIC [28]. . . . .   | 9  |
| 4.1  | Topologia <i>dumbbell</i> . Os nós indicados por círculos representam fontes e destinos do tráfego TCP principal; e os hexágonos, o tráfego <i>background</i> , cujo sentido de transmissão é indicado pelas setas pontilhadas. . . . .  | 24 |
| 4.2  | Topologia <i>parking lot</i> . Fontes e destinos do tráfego principal são representados por círculos, enquanto os nós envolvidos na transmissão do tráfego <i>background</i> , por hexágonos. As linhas pontilhadas indicam o caminho percorrido pelo tráfego <i>background</i> . . . . .  | 25 |
| 4.3  | Utilização em função do número de conexões para diferentes intensidades de tráfego <i>background</i> . Capacidade do enlace gargalo: <i>1Gbps</i> . Atraso de propagação: <i>100ms</i> . . . . .   | 27 |
| 4.4  | Utilização em função da capacidade do enlace para diferentes intensidades de tráfego <i>background</i> . Número de conexões: 20. Atraso de propagação no enlace gargalo: <i>100ms</i> . . . . .  | 28 |
| 4.5  | Estudo sobre a falta de escalabilidade dos protocolos. Em <i>a</i> ) fixou-se o número de conexões em $N = 100$ e o <i>buffer</i> em 2500 pacotes. Em <i>b</i> ) utilizou-se $N = 50$ conexões e <i>buffer</i> de 2500 pacotes. Em <i>c</i> ) mantiveram-se $N = 20$ conexões e o tamanho dos <i>buffers</i> foi aumentado para 20% do PBA. Todos com 20% de tráfego <i>background</i> . . . . . | 30 |
| 4.6  | Utilização em função do atraso para diferentes intensidades de tráfego <i>background</i> . Capacidade do enlace gargalo: <i>1Gbps</i> . Número de conexões: 20. . . . .  | 31 |
| 4.7  | Utilização do enlace alcançada pelo TCP Reno em função do número de conexões de protocolos de alta velocidade. . . . .   | 34 |
| 4.8  | Justiça intra-protocolos medido a partir do <i>Jain's index</i> . . . . .  | 36 |
| 4.9  | Índice de Jain para justiça inter-protocolos . . . . .   | 39 |
| 4.10 | Tempo de convergência para a justiça . . . . .   | 41 |
| 4.11 | Coefficiente de variação da vazão em função do tráfego <i>background</i> . . . . .   | 41 |

|      |   |    |
|------|---|----|
| 4.12 | Topologia <i>parking lot</i> . Fontes e destinos do tráfego principal são representados por círculos, enquanto os nós envolvidos na transmissão do tráfego <i>background</i> , por hexágonos. As linhas pontilhadas indicam o caminho percorrido pelo tráfego <i>background</i> . . . . . | 44 |
| 4.13 | Utilização em função do número de conexões - topologia <i>parking-lot</i> . . . .   | 45 |
| 4.14 | Utilização em função da capacidade do enlace - topologia <i>parking-lot</i> . . .   | 46 |
| 4.15 | Compatibilidade com TCP Reno sob diferentes intensidades de tráfego <i>background</i> . . . . .   | 48 |
| 4.16 | Justiça intra-protocolos . . . . .  | 49 |
| 4.17 | Justiça inter-protocolos. . . . .   | 51 |

# Lista de Acrônimos

|               |   |
|---------------|---|
| <b>ACK</b>    | Acknowledgement   |
| <b>AFRICA</b> | Adaptive and Fair Rapid Increase Congestion Avoidance         |
| <b>AIMD</b>   | Additive Increase Multiplicative Decrease                     |
| <b>AQM</b>    | Active Queue Management                                       |
| <b>BIC</b>    | Binary Increase Congestion Control Protocol                   |
| <b>BAIMD</b>  | Binary and Additive Increase Multiplicative Decrease          |
| <b>CUBIC</b>  | Cubic Congestion Control Protocol                             |
| <b>FAST</b>   | Fast Active Management Scalable Transmission Control Protocol |
| <b>FTP</b>    | File Transfer Protocol  |
| <b>HSTCP</b>  | HighSpeed Transmission Control Protocol                       |
| <b>HTCP</b>   | Hamilton Transmission Control Protocol                        |
| <b>IP</b>     | Internet Protocol   |
| <b>PBA</b>    | Produto Banda-Atraso  |
| <b>RTT</b>    | Round Trip Time   |
| <b>STCP</b>   | Scalable Transmission Control Protocol                        |
| <b>TCP</b>    | Transmission Control Protocol                                 |
| <b>UDP</b>    | User Datagram Protocol  |
| <b>XCP</b>    | Explicit Control Protocol                                     |

# Capítulo 1

## Introdução

As primeiras redes de computadores surgiram na década de 1960. Eram formadas por poucas máquinas que se comunicavam com o objetivo de expandir a quantidade de recursos disponíveis. Houve, no entanto, uma intensa popularização das redes de computadores e hoje tem-se uma grande rede global chamada Internet.

A Internet é um sistema complexo [58] e apresenta diversos fenômenos interessantes dependentes do seu estado. Quando há excesso de tráfego pode ocorrer congestionamento devido à insuficiência de recursos para acomodar o tráfego e ao desbalanceamento de carga entre os enlaces da rede. Como consequência, ocorrem descartes de pacotes, desperdício de recursos devido às retransmissões, redução na vazão e aumento do atraso fim-a-fim.

Os primeiros colapsos de congestionamento ocorreram nos anos 80 e deram início a uma busca por mecanismos de controle de transmissão de dados. Em 1988, Jacobson [56] propôs um protocolo para controle do congestionamento chamado *Transmission Control Protocol* (TCP), que ainda hoje é o protocolo de transporte padrão na Internet.

Concomitante à evolução da Internet ocorre um grande desenvolvimento das tecnologias de transmissão de dados. A difusão das redes sem fio e das fibras ópticas viabilizam altas capacidades de transmissão em longas distâncias. O aumento da banda e/ou do atraso levam ao aumento do produto banda-atraso (PBA), que indica o volume de dados que pode estar em trânsito em um determinado instante. Como a janela de congestionamento é proporcional ao PBA [7], o tamanho da janela deve ser muito grande para que os recursos disponíveis sejam totalmente utilizados.

O protocolo TCP é responsável por detectar e reagir às sobrecargas de tráfego, sendo a chave do sucesso operacional da Internet nas últimas décadas. Entretanto, em ambientes com PBA elevado, o TCP torna-se instável e seu desempenho insatisfatório, por ser demasiado conservativo e incrementar muito lentamente a janela de transmissão, dado que emprega o algoritmo de Incremento Aditivo Decremento Multiplicativo (AIMD) para alterar o tamanho da janela de transmissão. Todavia, esse padrão de crescimento da

janela limita, seriamente, a eficiência de operação do protocolo em enlaces de grande capacidade, tornando a camada de transporte o gargalo da rede. Por exemplo, para ocupar totalmente um canal de  $10Gbps$ , uma conexão TCP com pacotes de 1500 bytes e *Round Trip Time* (RTT) de 100ms precisaria de uma janela de congestionamento de  $W = 83.333$  pacotes e uma taxa de perda de no máximo 1 pacote a cada  $N = 5.000.000.000$  pacotes ( $N = W^2/1,5$ ), o que representa a perda de um pacote a cada 6000 segundos aproximadamente, o que é irrealista com as atuais tecnologias de transmissão [53]. Como conseqüências, o algoritmo não utiliza eficientemente a banda passante crescente e os fluxos com menores RTTs obtêm maiores quantidades de banda passante, provocando falta de equidade<sup>1</sup>. Além disso, pacotes perdidos em decorrência de erros provocados por interferências, desconexões ou mal funcionamento são erroneamente interpretados como sinais de congestionamento pelo protocolo TCP Reno[46].

O uso eficiente das tecnologias de transmissão que permitem altas taxas de transferência de dados depende do desenvolvimento de protocolos capazes de lidar com grandes quantidades de dados. Estes protocolos são designados Protocolos para Redes de Alta Velocidade.

O termo “velocidade” é adotado como sinônimo de taxa de transmissão e utilizado para expressar a quantidade de bits transmitidos por unidade de tempo. Já o conceito de alta velocidade é de difícil definição, pois varia com a evolução da tecnologia. Uma rede de alta velocidade consiste em grande largura de banda e baixa latência, além de lidar com PBA elevado. Protocolos designados a operar em redes de alta velocidade devem apresentar propriedades tais como:

- Escalabilidade: que representa o alcance de altas taxas de utilização do enlace independentemente da magnitude da largura de banda disponível (Kb/s, Mb/s, Gb/s, Tb/s, ...), permitindo grandes janelas de congestionamento com taxa de perda de pacotes realistas.
- Eficiência: significa a garantia de utilização eficiente dos recursos da rede, com a redução do tempo de subutilização de recursos. A eficiência é influenciada pela quantidade de fluxos existentes, tamanho das filas dos roteadores e taxa de perda de pacotes, e é avaliada pela capacidade de adaptação às mudanças ocorridas nas condições da rede.
- Compatibilidade: é a capacidade de ser justo com os protocolos pré-existentes (TCP Reno, TCP Newreno, TCP Sack), não utilizando uma quantidade de banda passante limitante aos fluxos destes protocolos.

---

<sup>1</sup>Equidade: Distribuição de recursos de forma justa entre conexões. Por exemplo, se há  $N$  conexões e a capacidade de enlace é  $C$ , então cada conexão deve receber  $\frac{C}{N}$  do total do enlace.

- Justiça: que significa garantir que todas as conexões concorrentes, independente do número e do RTT, consigam transmitir dados evitando que sofram inanição (*starvation*). Além disso, conexões devem coexistir em harmonia, compartilhando recursos de maneira igualitária.
- Prevenção de congestionamento: garantia de que o protocolo não leva a rede a um estado de colapso de congestionamento. Esta preocupação advém da natureza dos protocolos de alta velocidade serem mais agressivos que o protocolo TCP Reno. Prevenir o congestionamento está relacionado à capacidade de adaptação dos protocolos às mudanças ocorridas na rede.
- Estabilidade e convergência: o protocolo deve eliminar, tanto quanto possível, oscilações no tamanho da janela de congestionamento de forma a alcançar estabilidade. Além disso, deve ter a capacidade de convergir para um compartilhamento justo dos recursos entre todas as conexões.
- Fácil implantação: o protocolo deve ser tal que sua implantação seja facilitada, tornando factível sua implementação em ambientes reais.

Várias propostas têm sido desenvolvidas para superar as deficiências do protocolo TCP em redes de alta velocidade, através de alterações no algoritmo de controle de congestionamento. Algumas das propostas mais promissoras são TCP BIC [20], TCP CUBIC [28], FAST TCP [39], HSTCP [38], HTCP [32, 45], TCP LIBRA [48] e STCP [57]. Além destes, existem os protocolos TCP-Africa [37], *Compound TCP* [30, 31] e XCP [7]. Todos estes protocolos são denominados “variantes do TCP para redes de alta velocidade”, pois mantêm as funcionalidades básicas do TCP como estabelecimento de conexão e entrega confiável, enquanto modificam o algoritmo de controle de congestionamento para melhorar o desempenho em redes de alta velocidade.

O desenvolvimento de protocolos eficientes para transmissão de dados em alta velocidade é uma necessidade, entretanto, pouca atenção têm sido dada à avaliação do desempenho destes protocolos, o que é uma tarefa complexa. Uma das dificuldades principais é a inexistência de um conjunto padrão de testes capazes de determinar a eficiência de um protocolo, permitindo que diversas métricas e critérios sejam adotados levando a resultados possivelmente tendenciosos.

Neste trabalho, realiza-se uma ampla comparação entre os protocolos BIC, CUBIC, FAST, HSTCP, HTCP, LIBRA e STCP através de simulações, com o objetivo de determinar as características individuais de cada um, levando a conclusões sobre o desempenho destes protocolos. Propriedades essenciais como escalabilidade, justiça, estabilidade, convergência e compatibilidade com TCP Reno são investigadas, utilizando cenários realistas que obedecem às atuais recomendações [40, 54, 12] para análise de novos protocolos.

Esta dissertação está organizada da seguinte forma: No Capítulo 2 descreve-se o funcionamento do protocolo TCP Reno e apresentam-se suas deficiências em redes com PBA elevado. Além disso, os sete protocolos variantes do TCP para redes de alta velocidade são detalhados, incluindo as observações de desempenho realizadas pelos autores. No Capítulo 3 relatam-se alguns trabalhos comparativos previamente realizados. O Capítulo 4 é dividido em três partes. A primeira descreve o conjunto de experimentos e os resultados obtidos nas simulações utilizando a topologia *dumbbell*; a segunda parte descreve os resultados obtidos na topologia *parking lot*; e a última parte descreve a análise comparativa do desempenho dos protocolos. Para finalizar, no Capítulo 5 são descritas as conclusões e considerações finais relativas às investigações realizadas, além de apontamentos para trabalhos futuros.

## Capítulo 2

# Protocolos para transmissão de dados

O desenvolvimento de um protocolo de comunicação é composto, essencialmente, por três fases básicas: definição do problema, projeto do protocolo e implementação. Na primeira fase, são definidos os objetivos a serem alcançados e o ambiente no qual o protocolo deve funcionar. O projeto fornece a especificação detalhada do comportamento dos hospedeiros e de sua interação com outros dispositivos da rede. E a implementação é o conjunto de códigos específicos para um hardware e sistema operacional.

O *Internet Protocol* (IP) [3] é um protocolo da camada de rede usado entre *hosts* para encaminhamento de dados (datagramas). Qualquer rede conectada à Internet deve operar o protocolo IP e adaptar-se às suas convenções de nomeação e endereçamento. O modelo de serviço oferecido por este protocolo é o de melhor esforço, ou seja, não oferece garantia de entrega, entrega ordenada e integridade dos dados, sendo portanto, um protocolo não-confiável na entrega de dados. O Protocolo TCP atua na camada de transporte e amplia o serviço de entrega oferecido pelo IP fornecendo transferência confiável de dados a partir de técnicas de controle de fluxo, números de seqüência, reconhecimentos e temporizadores, o que garante a entrega correta e ordenada dos dados. Desta forma, o TCP converte um serviço não-confiável entre sistemas finais do IP em um serviço confiável de transporte de dados entre processos [25]. Além disso, o protocolo TCP possui controle de congestionamento que evita que conexões TCP abarquem enlaces e elementos comutadores com tráfego excessivo, preservando o bom funcionamento da rede.

O primeiro mecanismo de controle de congestionamento foi desenvolvido por Jacobson e Karels [56] e é denominado TCP Tahoe. *Slow-start* e *congestion avoidance* são seus algoritmos básicos. O *slow start* incrementa o tamanho da janela de congestionamento de um segmento a cada ACK enquanto o tamanho da janela for menor que um limiar. Nesta fase, a taxa de transmissão cresce exponencialmente. Quando a janela de transmissão

alcança o valor limiar ou ocorre perda de pacotes, inicia-se a fase de prevenção de congestionamento (*congestion avoidance*), que passa a incrementar a janela linearmente de um segmento por RTT até que ocorra um evento de congestionamento. Neste caso, a janela é ajustada para 1 segmento e o algoritmo *slow start* retoma o controle da transmissão.

O TCP Reno foi derivado do TCP Tahoe e introduziu o mecanismo de detecção de perdas através de reconhecimentos duplicados (*dupacks*). Desta forma, um pacote perdido pode ser detectado por duas maneiras: por *retransmission timeout* (RTO) ou *dupacks*. No primeiro caso, o fluxo de pacotes e reconhecimentos (ACK, do inglês, *acknowledgment*) permanecem estagnados, indicando congestionamento pesado, sendo necessário voltar a janela de congestionamento ao estado inicial. Caso novos reconhecimentos sejam recebidos, o que indica presença de fluxo na rede, a taxa de transmissão deve ser reduzida para não sobrecarregar o canal e nem torná-lo vazio. Este mecanismo é denominado *fast retransmit* e é seguido pelo algoritmo *fast recovery* que comanda a transmissão de novos segmentos até um ACK não duplicado ser recebido, fazendo com que o *congestion avoidance* seja executado e não o *slow start*, que passa a ser invocado somente na ocorrência de *timeout*. No caso de perda detectada por *dupacks*, o tráfego na rede continua operacional. Uma perda é detectada quando múltiplos reconhecimentos para um mesmo pacote é recebido, indicando que o pacote subsequente foi perdido.

Os quatro algoritmos utilizados pelo TCP Reno para controlar o congestionamento são *slow start*, *congestion avoidance*, *fast retransmit* e *fast recovery*. Estes algoritmos estão definidos na RFC 2581 [43].

A concepção do TCP Reno foi um grande avanço para o controle de congestionamento na Internet, porém apresenta deficiências. Quando ocorrem perdas em rajadas seu desempenho é degradado pois apenas a primeira perda é recuperada pelo *Fast Retransmit*, sendo as demais detectadas e recuperadas somente após a ocorrência de *timeout*, provocando redução drástica da janela [59] e aumento no atraso de transmissão. Outro fator é a instabilidade gerada pela tendência natural do TCP Reno em aumentar a quantidade de dados na rede até que o canal de transmissão não suporte a carga presente e exija a redução da janela. A oscilação causada pelo aumento e diminuição da janela leva a alternância de períodos de sobrecarga e subutilização dos recursos disponíveis provocando variações no tamanho das filas, RTTs e *jitter*. Existe ainda um outro problema, consequência do próprio mecanismo de controle, que incrementa a janela de apenas um segmento por RTT durante a fase de *congestion avoidance*. Isto permite a conexões com RTT menor incrementar sua janela de congestionamento mais frequentemente obtendo uma maior parte da banda disponível, além de deixar a banda subutilizada durante um grande número de RTTs após uma redução na taxa de transmissão.

As deficiências do TCP Reno são agravadas em redes com produto banda-atraso elevado, especialmente em ambientes com grande capacidade de transmissão, cada vez mais

comuns devido aos avanços nas tecnologias de transmissão de dados. O protocolo não consegue utilizar eficientemente a abundância de recursos disponíveis, sendo um limitador do desempenho deste protocolo em redes de alta velocidade. Em resposta ao desafio de encontrar mecanismos para transmissão de dados em alta velocidade, diversas variantes do TCP foram desenvolvidas. Estas variantes diferem na ação utilizada para definir a dinâmica de ajuste da janela de transmissão. Algumas propostas são baseadas em perdas, outras em atraso e outras são híbridas, considerando perdas e atraso conjuntamente. As abordagens baseadas em perdas aplicam mecanismos similares ao AIMD que tornam o crescimento da janela mais agressivo que o TCP Reno. Os métodos baseados em atraso identificam o congestionamento pela variação no RTT e reduzem a taxa de transmissão quando o RTT aumenta. Existem ainda variantes que propõem alterações no núcleo da rede (roteadores), que demandam um longo processo de difusão em ambientes reais. As principais variações estão descritas a seguir organizadas alfabeticamente.

## 2.1 BIC TCP

Em 2004, Rhee [20], propôs o protocolo *Binary Increase Congestion Control* (BIC) com o objetivo de corrigir a injustiça entre conexões com RTTs diferentes e ao mesmo tempo fornecer compatibilidade com TCP Reno e escalabilidade de banda. BIC parte do princípio de que um protocolo deve estimar, ativamente, a taxa de transmissão cujo valor esteja próximo do que provoca um evento de perda de pacotes. Para alcançar este objetivo, utiliza um algoritmo de busca binária combinada com incrementos aditivos para a atualização da janela, resultando em um complexo controle de congestionamento.

O protocolo BIC adapta a taxa de transmissão de acordo com o tamanho da janela. Na fase de incremento de busca binária (função logarítmica), a taxa de incremento é reduzida à medida que o tamanho da janela aproxima-se de um valor máximo (ponto de saturação). Como o número de pacotes perdidos é proporcional ao tamanho do último incremento antes da perda, a busca binária pode reduzir a quantidade de pacotes perdidos, evitando retransmissões e mantendo o tamanho da janela próximo do valor máximo.

Quando o tamanho da janela passa do tamanho máximo ( $cwnd_{max}$ ), o algoritmo de busca binária é substituído pelo *slow start* para sondar o novo tamanho máximo da janela, onde os incrementos ocorrem a cada RTT, em passos que seguem o padrão:

$$(cwnd_{max} + S_{min}), (cwnd_{max} + 2 * S_{min}), (cwnd_{max} + 4 * S_{min}), \dots, (cwnd_{max} + S_{max})$$

até que seja seguro incrementar a janela linearmente por  $S_{max}$ , ou seja, os algoritmos têm sua ordem de execução invertida.

Na ocorrência de perdas, que sinalizam eventos de congestionamento, a janela é reduzida por um fator multiplicativo  $\beta$ . Assim, quanto maior o tamanho da janela, maior

a redução. Caso a redução resulte em uma janela pequena, incrementos lineares por um grande fator ( $S_{max}$ ) permitem reduzir os períodos de subutilização de recursos. O incremento dependente apenas da diferença entre o tamanho das janelas atual e alvo permite aos fluxos convergirem para um compartilhamento justo. Fluxos com RTT curto podem alcançar o tamanho de janela alvo em períodos menores, mas não obtêm maiores porções de banda, mesmo em ambientes com perdas sincronizadas. Assim, a combinação das técnicas de incremento linear agressivo quando a janela é pequena e incremento logarítmico quando a janela é grande favorecem o compartilhamento justo da banda entre fluxos com diferentes RTTs.

A rápida convergência também está relacionada à combinação do incremento aditivo com busca binária. O incremento aditivo permite que janelas pequenas recuperem-se rapidamente enquanto o incremento logarítmico realiza um ajuste fino, com pequenas alterações na taxa de transmissão de forma a evitar perda de pacotes e, conseqüentemente, oscilações na taxa de transmissão.

Embora alcance boa escalabilidade, justiça e estabilidade em ambientes de alta velocidade, sua função de crescimento pode ser muito agressiva para o TCP, especialmente em redes de baixa velocidade. Além disso, as diferentes fases de controle da janela adicionam complexidade à análise do protocolo. A busca por uma nova função de crescimento da janela que permitisse preservar características como estabilidade e escalabilidade, simplificar o controle da janela, bem como melhorar a compatibilidade com o TCP resultou no protocolo CUBIC descrito na seção 2.2.

## 2.2 CUBIC TCP

O protocolo Cubic TCP [28] foi proposto com o objetivo de simplificar e melhorar o controle da janela introduzido pelo protocolo BIC. Cubic TCP adota uma função polinomial de terceira ordem e considera o tempo decorrido desde o último evento de congestionamento. CUBIC atualiza a janela de congestionamento de acordo com a seguinte função:

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

onde  $C$  é o fator de escala,  $t$  é o tempo decorrido desde o último evento de perda,  $cwnd_{max}$  é o tamanho da janela imediatamente anterior à última redução e  $K = \sqrt[3]{cwnd_{max}\beta/C}$ , sendo  $\beta$  o fator de decremento multiplicativo aplicado à janela a cada evento de perda, ou seja, a janela é reduzida para  $\beta * cwnd_{max}$ .

O crescimento da janela é rápido após uma redução da mesma, e à medida que a janela se aproxima de  $cwnd_{max}$  seu crescimento torna-se mais lento, chegando quase a ser nulo. Caso o tamanho da janela ultrapasse o valor de  $cwnd_{max}$ , CUBIC continua a sondar a banda disponível lentamente e acelera seu crescimento à medida que se afasta de  $cwnd_{max}$ .

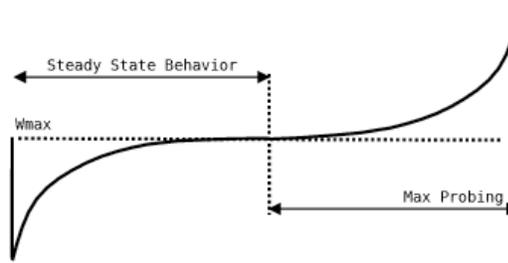


Figura 2.1: Função de crescimento da janela do CUBIC [28].

Este crescimento lento próximo de  $cwnd_{max}$  melhora a estabilidade do protocolo e aumenta a utilização da rede, enquanto o rápido crescimento quando o tamanho da janela é distante de  $cwnd_{max}$  assegura a escalabilidade do protocolo.

A função cúbica garante justiça intra-protocolo. Suponha que dois fluxos concorrentes sofram perdas sincronizadas e sejam reduzidos pelo mesmo fator multiplicativo  $\beta$ . O fluxo com maior  $cwnd_{max}$  terá maior redução e a função de crescimento permite ao fluxo com menor  $cwnd_{max}$  crescer mais lentamente ( $K$  aumenta com o aumento do valor de  $cwnd_{max}$ ). Assim, ambos os fluxos, eventualmente, convergem para o mesmo tamanho de janela [28].

A taxa de crescimento da janela é determinada pelo tempo decorrido,  $t$ . Isto assegura valores lineares de RTT, uma vez que quaisquer fluxos concorrentes com diferentes RTTs terão o mesmo período  $t$  após perdas de pacotes sincronizadas. Para melhorar a estabilidade e justiça, limitou-se o incremento da janela em  $S_{max}$  por segundo. Esta característica mantém o crescimento linear da janela enquanto estiver bem abaixo do valor de  $cwnd_{max}$ . O incremento linear do CUBIC é, portanto, dependente de estimativas realistas do tempo; um fluxo com RTT pequeno terá menos incrementos em um RTT, mas serão constantes no tempo [28], o que melhora a compatibilidade do protocolo, pois não privilegia fluxos em função da duração do RTT. Todavia, quando concorre com fluxos de outros protocolos que utilizam RTT para calcular o ajuste da janela, Cubic é pouco agressivo, podendo ser injustiçado.

Quando grandes janelas de congestionamento são alcançadas, pequenos ajustes são realizados, de forma a não provocar grandes saltos na taxa de transmissão, prevenindo perdas de pacotes, enquanto garante alta utilização dos enlaces.

O uso do intervalo de tempo entre eventos de perda ao invés de um contador de RTT (utilizado por BIC) no ajuste do tamanho da janela torna Cubic mais sensível a sessões concorrentes, particularmente em ambientes com RTTs curtos. A principal diferença entre os algoritmos BIC e CUBIC está na tentativa de CUBIC em reduzir alterações no

tamanho da janela quando se aproxima do tamanho da janela na ocasião da última perda [55].

## 2.3 FAST TCP

FAST TCP é uma abreviação para *Fast Active Management Scalable Transmission Control Protocol* [39]. Usa o atraso na fila como o principal fator no algoritmo de ajuste da janela de congestionamento e possui um mecanismo de controle composto por quatro componentes independentes: controle de dados, controle da janela, controle de rajada e estimativa.

O componente de estimativa calcula o atraso de enfileiramento e sinaliza perda de pacotes. Quando um reconhecimento positivo é recebido, calcula o RTT para o pacote e o utiliza para computar os RTTs mínimo e médio utilizados no controle da janela. Quando um reconhecimento negativo é recebido, gera uma indicação de perda para o controle de dados que seleciona o próximo conjunto de pacotes a serem enviados entre pacotes novos, retransmissão de pacotes considerados perdidos (reconhecimento negativo) e retransmissão de pacotes ainda não reconhecidos. O controle de rajada, por sua vez, decide se mais pacotes podem ser transmitidos e utiliza a técnica de  *pacing*  para eliminar grandes rajadas que podem criar longas filas e grandes perdas de pacotes [47].

A atualização da janela de congestionamento é baseada na diferença entre os RTTs atual e médio, tal que se o RTT for menor que o médio, a janela é incrementada; e se o RTT for maior, a janela é decrementada. A equação que governa o comportamento do protocolo é:

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

onde  $\gamma \in (0, 1]$  é uma constante, *baseRTT* é o RTT mínimo observado, *RTT* é o RTT médio e  $\alpha$  controla a justiça e o número de pacotes que cada fluxo tenta manter em equilíbrio nos  *buffers*  da rede, similar ao TCP Vegas [27], tal que:

$$\alpha(w, qdelay) = x$$

onde *qdelay* é o atraso médio fim-a-fim.

A janela de congestionamento é ajustada de acordo com uma estimativa da probabilidade de perda, a fim de que o tamanho da janela estabilize em torno do ponto de equilíbrio, que é alcançado quando um certo número de pacotes permanecem armazenados na rede. Uma das dificuldades encontradas por este protocolo é, justamente, estimar esta probabilidade de perda.

A utilização do atraso na fila como medida de congestionamento tem duas vantagens principais: o atraso nas filas pode ser melhor estimado do que a probabilidade de perda, além de fornecer mais informações; a dinâmica de atraso de filas relaciona-se diretamente com a capacidade da rede, estimando constantemente a banda disponível a partir de alterações no RTT. O atraso de propagação pode ser afetado por mudanças de roteamento e aumentos significativos podem degradar a vazão de fluxos FAST.

O protocolo FAST alcança justiça proporcional, utilizando somente a banda excedente; não é injusto com fontes TCP e não penaliza fontes com grandes atrasos de propagação pois o ajuste da janela é realizado de acordo com a quantidade de banda disponível, independente do RTT e da taxa de transmissão dos fluxos.

## 2.4 HighSpeed TCP

O protocolo HighSpeed TCP (HSTCP) [38] [10] foi proposto em 2002 como uma modificação do mecanismo de controle de congestionamento do TCP para melhorar o desempenho de conexões TCP com grandes janelas de congestionamento.

Este protocolo foi projetado para ter uma função de resposta diferente do TCP Reno em ambientes com taxa de perda de pacotes menores que  $10^{-3}$ . Desta forma, o comportamento do TCP Reno não é modificado em ambientes com congestionamento intenso, não aumentando o risco de colapso de congestionamento. Em ambientes com taxa de perdas de pacotes muito baixas (menores que  $10^{-3}$ ), o HSTCP apresenta uma função de resposta mais agressiva.

O protocolo HSTCP introduziu uma nova relação entre a janela de congestionamento média ( $cwnd$ ) e a taxa de perda de pacotes ( $p$ ) no estado estacionário. Sua função de resposta é especificada a partir de três parâmetros: *Low\_Window* ( $W_{low}$ ), *High\_Window* ( $W_{high}$ ) e *High\_P* ( $P_{high}$ ).  $W_{low}$  estabelece um ponto de transição e assegura compatibilidade. Se o tamanho da janela for menor que  $W_{low}$  utiliza a função TCP padrão, caso contrário adota uma função mais agressiva. A janela de congestionamento média ( $W_{high}$ ) e a taxa de exclusão de pacotes ( $P_{high}$ ) são utilizadas para especificar os limites da função HSTCP. Os valores padrão para  $W_{low}$ ,  $W_{high}$  e  $P_{high}$  são, tipicamente, 38, 83000 e  $10^{-7}$ . Estes parâmetros modificam os parâmetros de incremento e decremento da janela de congestionamento. Na fase de *congestion avoidance*, a janela é ajustada de acordo com as equações:

- Incremento:

$$cwnd \leftarrow cwnd + \frac{a(cwnd)}{cwnd} \quad (2.1)$$

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

- Decremento:

$$cwnd \leftarrow cwnd - b(cwnd) * cwnd \quad (2.2)$$

$$b(cwnd) = \frac{\log(cwnd) - \log(W_{low})}{\log(W_{high}) - \log(W_{low})} (b_{high} - 0.5) + 0.5$$

$$p(cwnd) = \exp\left(\frac{\log(cwnd) - \log(W_{low})}{\log(W_{high}) - \log(W_{low})} * (\log(P_{high}) - \log(P_{low})) + \log(P_{low})\right)$$

Quando o tamanho da janela é maior que  $W_{low}$  os valores dos parâmetros são alterados para  $a > 1$  e  $b < 0.5$ . Desta forma, o incremento da janela torna-se mais agressivo e o decremento mais suave que a função adotada no TCP Reno.

O HSTCP possui objetivos específicos: sustentar altas taxas de transmissão com taxas de perdas realistas, recuperar-se de múltiplos eventos de esgotamento do intervalo de temporização (*timeouts*) e períodos com pequenas janelas de congestionamento, sem incorrer em atrasos excessivos, bem como manter equidade de banda entre conexões. O primeiro objetivo é alcançado com o ajuste dos parâmetros de incremento e decremento realizados em função do tamanho corrente da janela, o que permite alcançar janelas de dezenas de milhares de segmentos sob taxas de perdas de pacotes realistas. Taxas de transmissão mais agressivas possibilitam que fluxos de melhor esforço iniciem com janelas maiores, favorecendo que grandes janelas sejam alcançadas ainda na fase *slow start*.

O protocolo HSTCP não responde prontamente a alterações na banda disponível. Quando a rede encontra-se em um estado de congestionamento severo, o decremento suave da janela faz com que fluxos com elevadas taxas de transmissão sofram múltiplas perdas e/ou *timeouts* para ajustar a janela às condições da rede. Caso as múltiplas reduções resultem em pequenas janelas de congestionamento, o incremento agressivo assegura que as pequenas janelas de congestionamento permaneçam durante alguns RTTs, abreviando o período de subutilização de recursos.

Pelo princípio da equidade, fluxos TCP deveriam obter a mesma quantidade de banda que fluxos HSTCP, porém quando a taxa de perda de pacotes é muito pequena o TCP não consegue adquirir maiores porções da banda, que permite que os fluxos HSTCP façam uso da banda excedente. Entretanto, quando a banda é altamente utilizada, os fluxos HSTCP

induzem um grande número de perdas que afetam os fluxos TCPs Reno, forçando-os a reduzir sua janela de congestionamento e, conseqüentemente, serão necessários mais ciclos de RTT para que obtenham a mesma largura de banda.

O ajuste da taxa de transmissão baseada no tamanho da janela pode provocar inanição de banda passante aos fluxos com longo RTT, pois os fluxos com RTTs menores incrementam suas janelas mais rapidamente obtendo, assim, grande parte da banda disponível.

À medida que as janelas de congestionamento aumentam, os incrementos tornam-se maiores e induzem perdas de pacotes para adequar a taxa de transmissão, isso contribui para a prevenção do colapso de congestionamento e provoca oscilações na taxa de transmissão, prejudicando a estabilidade do protocolo. A frequente super-estimação da banda dificulta o processo de convergência do protocolo.

## 2.5 H-TCP

O protocolo H-TCP [32] [45] adota um algoritmo de controle de congestionamento projetado para executar em redes convencionais, de alta velocidade e longas distâncias. Generaliza o algoritmo AIMD para permitir que o parâmetro de incremento varie em função do tempo decorrido ( $\Delta$ ) desde o último evento de congestionamento a partir de uma função  $\bar{\alpha}(\Delta)$ .

O algoritmo de controle de congestionamento do H-TCP obedece às seguintes regras:

$$\begin{aligned}
 \text{Ack} : & \begin{cases} \bar{\alpha} \leftarrow 2(1 - \beta)\bar{\alpha}(\Delta) \\ cwnd \leftarrow cwnd \frac{\bar{\alpha}}{cwnd} \end{cases} \\
 \text{Perda} : & \begin{cases} \beta \leftarrow \frac{RTT_{min}}{RTT_{max}}, \beta \in [0.5, 0.8] \\ cwnd \leftarrow \beta * cwnd \end{cases} \\
 \bar{\alpha} = & \begin{cases} 1 & \Delta \leq \Delta_L \\ 1 + 10(\Delta - \Delta_L) + 0.25(\Delta - \Delta_L)^2 & \Delta > \Delta_L \end{cases}
 \end{aligned}$$

Devido ao ajuste ser baseado no tempo decorrido desde o último evento de perda, os fluxos que estejam no modo de alta velocidade não têm privilégios sobre novos fluxos. Isto mantém simetria na rede e garante justiça com outras fontes H-TCP. Quando perdas são sincronizadas, os fluxos obtêm em média a mesma quantidade de banda, considerando a mesma probabilidade de perda.

O número de períodos de congestionamento até a convergência é determinado pelo fator de decremento  $\beta$ , que é baseado na variância do intervalo de RTT ( $RTT_{min}/RTT_{max}$ ). No TCP Reno,  $\beta = 0.5$  e leva aproximadamente 4 períodos para convergir. Para melhorar o tempo de resposta é preciso escolher uma função  $\bar{\alpha}(\Delta)$  tal que a duração dos períodos de congestionamento permaneça razoavelmente pequena enquanto o produto banda-atraso aumenta. A escolha de

$$\bar{\alpha}(\Delta) = 1 + 10(\Delta - \Delta_L) + 0.25(\Delta - \Delta_L)^2$$

e  $\Delta_L = 1s$  leva a um incremento polinomial na janela sobre um período de congestionamento e com o ajuste da polinomial controla-se a duração do período.

A compatibilidade com fluxos TCP é assegurada pelo ajuste do parâmetro de incremento

$$\bar{\alpha}(\Delta) = \begin{cases} 1 & \Delta \leq \Delta_L \\ \bar{\alpha}_H & \Delta > \Delta_L \end{cases}$$

onde  $\Delta_L$  é o limiar para troca entre os modos de operação TCP padrão/H-TCP.  $\bar{\alpha}$  indica a taxa de crescimento medida em pacotes/RTT e não em pacotes/segundo. A escolha da função  $\bar{\alpha}(\Delta)$  é governada pela taxa na qual a banda deve ser adquirida, quando  $\bar{\alpha}(\Delta) = 1$ , tem-se o modo TCP padrão, sendo portanto, compatível com TCP Reno em ambientes convencionais.

O parâmetro de incremento  $\alpha$  do AIMD tem impacto direto no comportamento da rede. Já mudanças no fator de decremento ( $\beta$ ) impactam a duração e o número de períodos de congestionamento antecedentes à convergência, bem como a vazão. Além disso, o valor de  $\beta$  pode provocar injustiça entre fluxos concorrentes, o que pode ser resolvido com o ajuste de  $\alpha_i = 2(1 - \beta_i)$ . A regra do produto banda-atraso requer dimensionamento correto dos *buffers* da rede enquanto que o *backoff* adaptativo requer que cada fonte ajuste seu fator de redução apropriadamente. A rápida convergência para o estado de equilíbrio é garantida se o maior fator de *backoff* na rede for pequeno [32].

Uma vez que os fluxos ajustam seu fator de *backoff* de acordo com nível de provisionamento do *buffer*, as filas somente esvaziam após um evento de *backoff* e o enlace continua a operar normalmente. Desta forma, ao invés de projetar *buffers* para acomodar o algoritmo TCP AIMD, modifica-se o algoritmo para acomodar os *buffers* na rede. Isto é particularmente relevante em ambientes de alta velocidade nos quais as filas são pequenas em relação ao produto banda-atraso.

## 2.6 TCP Libra

O protocolo TCP Libra [48] foi desenvolvido com o objetivo de obter o melhor equilíbrio entre utilização, justiça e compatibilidade com TCP New Reno. O algoritmo de ajuste da janela compensa as diferenças de RTT, levando a uma solução que fornece justiça entre fluxos TCP que compartilham o mesmo enlace gargalo independente do RTT e, desta forma, não é sensível ao comportamento auto-regulado <sup>1</sup>.

---

<sup>1</sup>Comportamento auto-regulado: mecanismos de controle de congestionamento baseados em perdas de pacotes detectam eventos de congestionamento e reduzem a taxa de transmissão. Ou seja, a taxa de transmissão é regulada automaticamente.

Os componentes principais são:

1. Estimativa: estima a capacidade do enlace gargalo no início de uma nova sessão.
2. Controle de Justiça: equaliza a vazão entre fluxos com RTTs heterogêneos.
3. Controle de Escalabilidade: recebe como entrada a capacidade do enlace gargalo e ajusta a amplitude do incremento da janela.
4. Controle de Estabilidade: certifica que, tomando como entrada o compartilhamento de ocupação do buffer, o protocolo opera na sua região estável.
5. Controle de Rajada: determina quando os pacotes devem ser enviados, certificando-se de não sobrecarregar a rede, prevenindo sincronização de perdas e múltiplas reduções da janela devido a *buffer overflow*.

O protocolo pertence à classe dos algoritmos AIMD. Reage à formação de filas diminuindo o incremento da janela à medida que a ocupação dos *buffers* aumenta. Quando perdas de pacotes são detectadas, a janela é reduzida por um fator multiplicativo variável. Seu algoritmo de controle de congestionamento é descrito pelas seguintes equações:

$$\begin{aligned} \text{RTT: } cwnd &\leftarrow cwnd + \frac{1}{cwnd} * \frac{\alpha T^2}{T+T_0} \\ \text{Perda: } cwnd &\leftarrow cwnd - \frac{T_1 cwnd}{2(T+T_0)} \\ \alpha &= k_1 * c * e^{-k_2 \frac{T-T_{min}}{T_{max}-T_{min}}} \end{aligned}$$

onde  $T$  é o RTT medido,  $T_0$ ,  $T_1$ ,  $k_1$  e  $k_2$  parâmetros fixos e  $c$  a capacidade do enlace. Cada um dos componentes pode ser mapeado no algoritmo, conforme descrito a seguir.

A escalabilidade é controlada pelo produto  $k_1 c$ , sendo  $k_1$  um parâmetro fixo que deve considerar o requisito de tempo de resposta do algoritmo e a execução no estado estável (*steady state*). Altos valores de  $k_1$  levam à instabilidade e baixos valores levam à baixa convergência (quanto maior  $k_1$  maior a agressividade,  $k_1$  deve ser selecionado juntamente com  $k_2$ ). Altos valores de  $c$  levam a problemas de instabilidade enquanto baixos valores levam a muito baixa convergência. Um valor para  $c$  não pode ser determinado a priori sem introduzir adaptabilidade nas condições da rede.

O controle de justiça é implementado pelos fatores de incremento ( $\frac{T^2}{T+T_0}$ ) e decremento ( $\frac{1}{T_0+T}$ ). O termo ( $\frac{1}{T_0+T}$ ) fornece o controle sobre a faixa de RTT para a qual as vazões serão equalizadas. Nos casos onde  $T \ll T_0$ , a influência de  $T$  sobre a porção de incremento é mínima então  $\frac{T^2}{T+T_0} \approx \frac{T^2}{T_0}$ . Em casos onde  $T$  é comparável a  $T_0$ ,  $\frac{T^2}{T+T_0} \approx \frac{T}{2}$ . Isto pode exigir um controle maior em casos de penalização de fluxos que excedem um certo RTT limite. Uma razão para isso é que um fluxos com RTT acima de um limiar podem ser interpretados como fluxos que estejam enfrentando grandes atrasos.

A prevenção de perdas sincronizadas e de múltiplas reduções nas janelas são resolvidas com a introdução de  *pacing*  aleatório, no qual os pacotes são transmitidos aleatoriamente no intervalo de um RTT, tornando possível obter melhores medidas de RTT.

O fator  $\alpha$  tem por objetivos aumentar a velocidade de convergência e alcançar escalabilidade, além de manter o comportamento do algoritmo estável. A escalabilidade é representada pelo fator  $k_1 * c$ , onde  $k_1$  é uma constante e  $c$  é a capacidade do enlace. A penalidade, projetada para fazer a taxa de incremento do algoritmo adaptativa ao estado de congestionamento na rede e manter a janela mais próximo possível de um valor máximo, é ajustada em função do RTT. Se  $k_1$  aumenta,  $k_2$  deve ser aumentado para manter a estabilidade, além disso, altos valores de  $c$  e  $\bar{T}$  exigem altos valores de  $k_2$ .

A vazão e a velocidade de convergência são ajustados pelo parâmetro  $T_0$ . Aumentar o valor deste parâmetro provoca redução na variância da vazão ao custo da velocidade de convergência. Logo, quando se aumenta o valor de  $T_0$  deve-se também aumentar  $k_1$ .

O controle de estabilidade é representado pelo termo  $e^{-k_2 \frac{T_n - T_{min}}{T_{max} - T_{min}}}$  que assegura que o algoritmo diminui a taxa de incremento da janela quando os enlaces tornam-se congestionados. Além disso, o termo  $\frac{T_n - T_{min}}{T_{max} - T_{min}}$  pode ser interpretado como o compartilhamento do buffer. O protocolo TCP Libra não penaliza fortemente a dinâmica da janela quando o buffer começa a encher, assim quando o RTT começa a aumentar, o incremento torna-se menor. Esta estratégia também permite compartilhamento justo com TCP uma vez que a reação à formação de filas não compromete fortemente o desempenho do TCP Libra quando este compete com TCP New Reno pela capacidade do enlace.

## 2.7 Scalable TCP

O protocolo  *Scalable TCP*  (STCP), proposto por Tom Kelly [57], consiste em modificar a forma de atualização da janela de congestionamento dos transmissores TCP, com o objetivo de obter desempenho melhor que o protocolo TCP Reno em redes de alta velocidade.

Os parâmetros de incremento e decremento são ajustados por fatores multiplicativos, paradigma MIMD. O algoritmo de controle de congestionamento funciona da seguinte forma: enquanto não houver sinal de congestionamento (indicado por perda de pacotes) cada reconhecimento recebido no intervalo de um RTT altera a janela do transmissor para  $cwnd \leftarrow cwnd + a$ , sendo  $a$  uma constante  $0 < a < 1$ . Quando houver indicação de congestionamento, a janela é ajustada para  $cwnd \leftarrow cwnd - \lceil b * cwnd \rceil$ , sendo  $b$  uma constante  $0 < b < 1$ . Os valores de  $a$  e  $b$  são ajustados respectivamente para 0.01 e 0.125 [57], considerando-se o impacto do STCP sobre o tamanho da janela, as propriedades de alocação de banda, variação na taxa de fluxo, propriedades de convergência e estabilidade.

No TCP Reno, após uma redução na taxa de transmissão, o tempo necessário para

alcançar o tamanho da janela anterior à redução é proporcional à janela e ao RTT da conexão. Os ajustes realizados pelo STCP em função dos parâmetros  $a$  e  $b$  fazem com que o tempo de recuperação seja proporcional somente ao RTT, respondendo eficientemente às alterações na disponibilidade de banda quando as janelas são grandes permitindo melhor desempenho em redes de alta velocidade. A agressividade obtida com o incremento multiplicativo garante que mesmo após uma redução drástica no tamanho da janela, o tempo de recuperação será curto, devido à rápida utilização dos recursos. Por outro lado, quando a banda está totalmente utilizada, o incremento multiplicativo provoca uma grande quantidade de perdas de pacotes devido aos constantes erros na estimativa da banda passante disponível.

Grandes janelas de congestionamento e elevadas taxas de exclusão de pacotes tornam o STCP compatível com TCP Reno, dadas às constantes reduções nas janelas de congestionamento e ao aumento do atraso fim-a-fim provocado pelas filas nos *buffers* dos roteadores. Entretanto, quando a taxa de exclusão é pequena, é injusto por utilizar um incremento extremamente agressivo comparado ao incremento aditivo dos fluxos TCP, o que leva à alocação de grande parte dos recursos, induzindo perdas que forçam os fluxos TCP a reduzirem suas taxas de transmissão.

Da mesma forma que protocolo HSTCP, descrito na seção 2.4, é injusto com fluxos com longo RTT, pois fluxos com RTT curtos incrementam sua janela muito mais rapidamente obtendo, assim, maior parte dos recursos disponíveis.

O processo de convergência de qualquer protocolo, idealmente, deveria ser instantâneo, porém a utilização de perda de pacotes como sinal de congestionamento, a necessidade de compatibilidade com os demais tráfegos que compartilham o canal e principalmente o mecanismo de controle de congestionamento dificultam esta tarefa. O STCP, devido ao uso de incrementos e decrementos multiplicativos, consegue altas taxas de utilização mas pouca convergência devido a quantidade e amplitude das oscilações.

## 2.8 Outras variantes do TCP

Existem outras variantes do protocolo TCP para redes de alta velocidade. Algumas propostas exigem modificações no núcleo da rede, o que não é uma característica desejável para a implantação efetiva em redes reais. Este é o caso do *eXplicit Congestion Protocol* (XCP) [11] [7] [5], que generaliza a proposta do *Explicit Congestion Notification* (ECN).

Outras propostas são derivadas de protocolos existentes, e possuem o objetivo de melhorar alguma característica específica, é o caso do protocolo TCP Africa [37], baseado no HSTCP. E ainda pode-se listar: YeAH TCP [14], TCP Hybla [36], TCP Fusion [24], Compound TCP [30] [31] e TCP SQRT [60].

## 2.9 Classificação

Os protocolos para redes de alta velocidade podem ser classificados de acordo com diferentes critérios, como o paradigma de atualização da janela, o tipo de evento monitorado ou o objetivo principal a ser alcançado. Na Tabela 2.1 apresenta-se, de forma resumida, as características intrínsecas de cada protocolo.

| Protocolo | Objetivo  | Função [50]     | Mecanismo                    | Evento        | Ação                                      |
|-----------|---|-----------------|------------------------------|---------------|---|
| Reno      | Controlar a quantidade de dados trafegando na rede                    | convexa         | AIMD(1, 0.5)                 | ack           | $w = w + 1/w$                             |
|           |   |                 |                              | perda         | $w = w - w/2$                             |
| BIC       | Minimizar oscilações quando altas taxas de transmissão são alcançadas | côncavo-convexa | BAIMD                        | ack (binário) | $w = w + (w_{prevMax} - w)/2$             |
|           |   |                 |                              | ack (aditivo) | $w = w + step_{max}$                      |
|           |   |                 |                              | perda         | $w = w - w.\beta$                         |
| CUBIC     | Simplificar o mecanismo proposto por BIC                              | côncavo-convexa | Função cúbica                | ack           | $w = C(t - K)^3 + w_{max}$                |
|           |   |                 |                              | perda         | $w = w - w.\beta$                         |
| FAST      | Prover um estado de equilíbrio  | N/A             | Variação no RTT              | RTT           | $w = w(RTT_{base}/RTT) + \alpha$          |
| HSTCP     | Melhorar a utilização de enlaces de grande capacidade                 | convexa         | AIMD( $a(w), b(w)$ )         | ack           | $w = w + a(w)/w$                          |
|           |   |                 |                              | perda         | $w = w - b(w)w$                           |
| HTCP      | Justiça entre fluxos  | convexa         | AIMD-like( $\alpha, \beta$ ) | ack           | $\alpha = 2(1 - \beta).(T^2/2 + 10T + 1)$ |
|           |   |                 |                              | perda         | $\beta = RTT_{min}/RTT_{max}$             |
| LIBRA     | Justiça entre fluxos com diferentes RTTs                              | convexa         | AIMD-like( $\alpha, 0.5$ )   | ack           | $w = w + (\alpha.T^2)/(w(T + T_0))$       |
|           |   |                 |                              | perda         | $w = w - (w.T_1)/(2(T + T_0))$            |
| STCP      | Minimizar o tempo de recuperação após perdas                          | convexa         | MIMD(1/100, 1/8)             | ack           | $w = w + 1/100$                           |
|           |   |                 |                              | perda         | $w = w - w/8$                             |

Tabela 2.1: Resumo das características dos protocolos variantes do TCP para redes de alta velocidade

## Capítulo 3

# Trabalhos Relacionados

Os protocolos variantes do TCP para redes de alta velocidade têm o objetivo comum de lidar com elevadas taxas de transmissão. Para tal, adotam diferentes mecanismos e políticas para atualização da janela de congestionamento, que podem, em alguns casos, ser adotadas em conjunto [23]. A avaliação de desempenho destas novas propostas de protocolos é necessária para a identificação das qualidades e deficiências destes mecanismos. No entanto, uma das dificuldades encontradas para tal avaliação é a inexistência de um conjunto padrão de testes capazes de determinar a eficiência de um protocolo. Observa-se que diferentes estudos adotaram diferentes métricas e critérios, levando, possivelmente, a resultados que não podem ser generalizados.

Existe atualmente uma força-tarefa no sentido de estabelecer um *benchmark* que possa ser utilizado para avaliar as diferenças entre os vários algoritmos de controle de congestionamento, estabelecendo métricas de desempenho para diferentes implementações e que objetiva também servir como instrumento em simulações e experimentos.

Sally Floyd [54] propôs um *framework* para avaliação de algoritmos de controle de congestionamento que inclui várias métricas como vazão, taxa de perda de pacotes, atraso e justiça, e também diferentes ambientes de rede. Wei *et al.* [40] propuseram, paralelamente, um conjunto de *benchmarks* que permite avaliações de desempenho comparativas entre protocolos. Este *benchmark* é composto por vários cenários, incluindo distribuições de *heavy-tailed* para tamanhos de arquivos, bem como estabelece valores típicos para atraso de propagação. Andrew *et al.* [51] propuseram recentemente um conjunto de cenários para avaliação de protocolos que incluem distribuições de tráfego e atrasos de propagação, topologias, e métodos de avaliação.

Trabalhos de avaliação de desempenho de novos protocolos têm sido realizados através de técnicas de simulação ou experimentos reais. Entretanto, o que se observa é que cada autor seleciona um conjunto de testes, muitas vezes incompletos, e que não levam em consideração parâmetros importantes que podem influenciar o desempenho dos protocolos,

tais como a composição do tráfego *background*.

Ha et al. [52] demonstraram que os protocolos para redes de alta velocidade exibem comportamentos diferentes quando avaliados com e sem tráfego *background*. Os autores afirmam que avaliar o desempenho dos protocolos sem tráfego *background* não leva a conclusões corretas acerca do comportamento destes.

Os algoritmos de controle de congestionamento são sensíveis a variáveis como intensidade e composição do tráfego *background*, atraso de propagação e topologia, portanto a avaliação de desempenho deve explorar sistematicamente o espaço de parâmetros e os vários aspectos do protocolo. De acordo com [29], a inserção do tráfego *background* é importante nos modelos de simulação de rede por diversas razões:

1. O comportamento agregado do tráfego *background* permite flutuações na dinâmica das filas e nos padrões de perda de pacotes.
2. Ambientes de rede sem quaisquer aleatoriedades na chegada e atraso de pacotes são altamente suscetíveis aos efeitos de fase [41] [13]. Um tráfego *background* composto por diversos padrões de chegadas de pacotes e atrasos reduz a probabilidade de sincronização de fluxos TCP.
3. O núcleo da rede permite um alto grau de multiplexação estatística. Alguns protocolos assumem que seus fluxos executam neste tipo de ambiente, por exemplo, HSTCP e STCP que confiam na multiplexação para rápida convergência.

Em consequência, a avaliação de desempenho de protocolos de rede com pouco ou nenhum tráfego reverso e/ou *background* não é suficiente para delinear o comportamento provável do protocolo quando este estiver sendo executado na Internet [29].

Bulot et al. [33] realizaram comparações experimentais entre os protocolos TCP NewReno e as variantes para alta velocidade BIC, FAST, HSTCP, STCP, HTCP e HSTCP-LP em redes de produção de alta velocidade. Este estudo reporta que o TCP NewReno apresenta instabilidade e baixo desempenho enquanto a maioria dos protocolos de alta velocidade alcançam melhorias significativas sobre o NewReno. Seus resultados são bastante motivadores, entretanto os experimentos foram realizados sem um controle adequado sobre o tráfego *background*, composto somente por UDP. Além disso, não foram consideradas diferentes composições de tráfego *background* que podem influenciar no comportamento dos protocolos.

Li et al. [35] realizaram experimentos com os protocolos BIC, FAST, HSTCP, HTCP e STCP em uma rede de laboratório. Notaram que todos os protocolos, exceto HTCP, geram injustiça entre os fluxos, e enfatizaram o bom desempenho do HTCP. Os autores afirmam que os protocolos foram testados na presença de tráfego WEB, porém somente

resultados sem tráfego *background* foram descritos sob a justificativa que os resultados obtidos foram similares.

Destaca-se o trabalho de Nabeshima e Yata [26], que avaliou, através de simulações, algumas propriedades de um subconjunto (BIC, HSTCP, HTCP e STCP) dos protocolos de alta velocidade. Utilizando uma topologia *dumbbell* limitada e uma composição minimalista de tráfego *background* relatam resultados interessantes acerca da escalabilidade, justiça de RTT e compatibilidade dos protocolos.

Além das avaliações realizadas entre protocolos, existem avaliações específicas para cada um deles. Em [16] é apresentada uma avaliação experimental do protocolo CUBIC. Os autores utilizaram uma topologia *dumbbell* e basearam os testes em [35]. Seus resultados apontam que CUBIC apresenta lenta convergência para a justiça entre fluxos; quando o produto banda-atraso for maior que 5000 pacotes, CUBIC retorna à função de crescimento linear, tornando-se semelhante ao TCP padrão; além disso, afirmam que CUBIC exibe injustiça de RTT, fortemente dependente do instante em que as conexões iniciam.

Mascolo [15] investigou o efeito do tráfego reverso nas propriedades de justiça e estabilidade dos protocolos TCP NewReno, STCP, HSTCP, HTCP, BIC e FAST empregando tráfego WEB como tráfego *background*. Seus resultados apontam que os protocolos BIC e HTCP alcançam um compartilhamento justo da banda entre fluxos com mesmo RTT, enquanto os demais protocolos não. Por outro lado, FAST e HTCP foram os únicos protocolos a apresentar justiça entre fluxos com diferentes RTTs. Além disso, constatou-se que todos os protocolos avaliados ocupam rapidamente a banda disponível exibindo notáveis oscilações na janela de congestionamento na presença de tráfego reverso. Observa-se que todos os protocolos sofrem em algum cenário específico.

Chen e Bensaou [6] investigaram o problema da injustiça entre protocolos de alta velocidade. Consideraram cenários de simulação com baixo nível de multiplexação estatística, muitos enlaces gargalos e utilizaram as políticas de gerenciamento ativo de filas e *Drop Tail*. Seus resultados indicam que com filas *Drop Tail* a injustiça entre fluxos é alta mesmo quando apresentam RTTs semelhantes, apontando que esse comportamento é devido às perdas sincronizadas. Além disso, defendem que o uso de políticas de AQM seria uma forma de atenuar tal injustiça.

Estudos recentes [50, 34] têm mostrado que o formato da curva de crescimento da janela dos protocolos pode impactar a estabilidade, promovendo oscilações e possivelmente *overflow* nas filas dos roteadores. De acordo com tais estudos, protocolos cujas funções de crescimento iniciam com uma função côncava e tornam-se convexas próximas do tamanho máximo alcançado no último período de congestionamento tendem a apresentar menores taxas de variação. Desta forma, os protocolos BIC e CUBIC teriam vantagens sobre protocolos que exibem curvas convexas, como H-TCP, HSTCP e STCP.

Outros trabalhos que merecem destaque são [18, 9], pois avaliam o impacto de perdas sincronizadas e do tamanho dos *buffers* em versões *high speed* do TCP.

A contribuição deste trabalho é a execução de um conjunto complexo de simulações para avaliar características de escalabilidade, justiça, compatibilidade, estabilidade e convergência dos principais protocolos para redes de alta velocidade que leve à conclusões mais precisas acerca do desempenho destes protocolos em relação às encontradas na literatura.

O capítulo seguinte é dedicado à descrição dos cenários e experimentos de simulação utilizados para a avaliação de desempenho dos protocolos BIC, CUBIC, FAST, HSTCP, HTCP, LIBRA e STCP avaliados neste trabalho.

## Capítulo 4

# Comparação entre Protocolos para Redes de Alta Velocidade

As atuais recomendações para avaliação de desempenho de novos protocolos de transporte [40, 54, 12] sugerem diversas configurações para os cenários de simulação. Tais configurações não se restringem a topologias, mas estendem-se a outros aspectos como atraso de propagação, quantidade de fontes e largura de banda. Baseado nestas recomendações foi derivado o conjunto de experimentos utilizado para analisar as propriedades de escalabilidade, estabilidade, justiça, compatibilidade e convergência dos protocolos BIC, CUBIC, FAST, HSTCP, HTCP, LIBRA e STCP para redes de alta velocidade.

Foi utilizado o software *Network Simulator 2* [2] versão 2.29 como ferramenta de simulação. Adotaram-se duas topologias, uma simples e outra mais complexa para capturar um grande número de características dos protocolos a serem avaliados. As topologias são:

- *Dumbbell*: Topologia bastante simplificada composta por um único enlace gargalo. A Figura 4.1 mostra a disposição das fontes ( $F_i$ ) e destinos ( $D_i$ ) do tráfego principal e *background* utilizados. Segundo Floyd [13], esta topologia é suficiente para investigar muitas propriedades de tráfego.

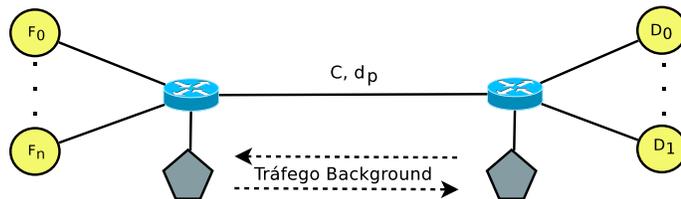


Figura 4.1: Topologia *dumbbell*. Os nós indicados por círculos representam fontes e destinos do tráfego TCP principal; e os hexágonos, o tráfego *background*, cujo sentido de transmissão é indicado pelas setas pontilhadas.

- *Parking lot*: Topologia mais complexa composta por múltiplos enlaces gargalos que podem ser caracterizados com diferentes capacidades e atrasos de propagação. Além disso, os fluxos podem percorrer caminhos com diferentes números de saltos, dependendo da sua configuração. Esta topologia é uma simplificação para a representação de conexões com diferentes RTTs, números de saltos e tráfego cruzado entre clientes e servidores [49]. A Figura 4.2 exemplifica a topologia.

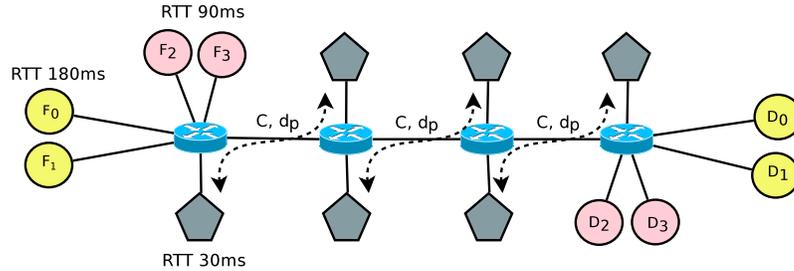


Figura 4.2: Topologia *parking lot*. Fontes e destinos do tráfego principal são representados por círculos, enquanto os nós envolvidos na transmissão do tráfego *background*, por hexágonos. As linhas pontilhadas indicam o caminho percorrido pelo tráfego *background*.

Para tornar os cenários mais próximos de uma representação real incluímos tráfego *background*, responsável por ocupar uma porcentagem da capacidade do enlace nos sentidos direto e reverso da transmissão e composto por 20% de tráfego UDP CBR, 24% FTP e 56% WEB [29]. Ao longo das simulações variou-se a intensidade deste tráfego entre 20%, 50% e 80% da capacidade do enlace gargalo. Para a geração do tráfego *background* utilizou-se o gerador *traffic-gen* [22].

Para evitar os efeitos de sincronização [41, 13], comuns em ambientes de simulação, foram acrescentados pequenos ajustes aleatórios no tempo de início das fontes TCP e no atraso de propagação dos enlaces. Fixou-se o tempo total de simulação de cada experimento em  $T = 600$  segundos, tempo suficiente para observar os fenômenos que ocorrem na rede; e o tamanho do *buffer* de cada enlace em 2500 pacotes de 1500 bytes, estando em conformidade com tamanho de *buffers* reais [1, 40]. A política de fila *DropTail* foi adotada para que o desempenho dos protocolos não fosse influenciado pelas políticas de *Active Queue Management* (AQM).

A versão 2.29 simulador NS implementa dois dos protocolos avaliados neste trabalho, são eles: STCP e HSTCP. Os demais protocolos (BIC, CUBIC, FAST, HTCP e LIBRA) foram implementados para tornar possível a execução das simulações. Nenhum parâmetro específico da configuração dos protocolos foi alterado, sendo preservados os parâmetros originais.

O simulador NS é baseado em eventos, tal que cada pacote enviado/recebido/perdido gera um evento. Desta forma, o número de eventos aumenta exponencialmente com a capacidade do enlace e impacta direta e negativamente no tempo de simulação. O grande número de simulações, aproximadamente 1250, somadas ao longo tempo de simulação, devido a utilização de enlaces de grande capacidade, foram os limitadores para que cada experimento fosse realizado uma única vez, sem repetição.

Este capítulo foi dividido em três partes: a primeira é dedicada a apresentar os resultados obtidos utilizando a topologia *dumbbell*; a segunda concentra os resultados obtidos a partir da topologia *parking lot*; e na última é realizada a análise comparativa do desempenho dos protocolos em cada topologia.

## 4.1 Topologia *Dumbbell*

Nesta primeira parte são detalhados os experimentos e resultados obtidos através de simulações utilizando a topologia *dumbbell*, composta basicamente por  $N$  pares de fontes e destinos de tráfego TCP, conectados por um único enlace.

Nas próximas subseções descrevem-se as propriedades investigadas juntamente com uma discussão acerca dos resultados obtidos.

### 4.1.1 Escalabilidade

A Internet é um sistema distribuído de larga escala, composto de muitas entidades complexas que interagem entre si, como tráfego, topologia, roteamento e endereçamento [44]. A contínua expansão da Internet faz da escalabilidade uma das propriedades mais importantes para um protocolo designado a operar em ambientes de alta velocidade.

Segundo Wei [40], estes protocolos devem ser escaláveis com o número de conexões, o atraso de propagação e com a capacidade do enlace. Investigou-se a escalabilidade dos protocolos avaliados a partir destes três aspectos.

- Escalabilidade em função do número de conexões: para cada protocolo, variou-se o número de conexões concorrentes no intervalo  $2^i, i = 1..8$  para cada intensidade de tráfego *background*. O objetivo deste cenário é verificar se as conexões são capazes ocupar um enlace de *1Gbps* independente da quantidade de conexões.
- Escalabilidade em função do atraso de propagação: para cada protocolo, variou-se o atraso de propagação do enlace gargalo entre *10ms* (Estado), *50ms* (País), *100ms* (Trans-Atlântico) e *200ms* (Trans-Pacífico) para cada intensidade de tráfego *background*. O objetivo deste cenário é verificar se 20 conexões alcançam alta utilização do enlace independente do RTT experimentado pelos fluxos.

- Escalabilidade em função da capacidade do enlace: para cada protocolo, variou-se a capacidade do enlace gargalo utilizando os valores de  $155Mbps$  (OC-03),  $622Mbps$  (OC-12),  $1Gbps$  (Gigabit Ethernet),  $2.5Gbps$  (OC-48),  $5Gbps$  e  $10Gbps$  (OC-192) para cada intensidade de tráfego *background*. O objetivo deste cenário é verificar se 20 conexões são capazes de utilizar todo o enlace disponível.

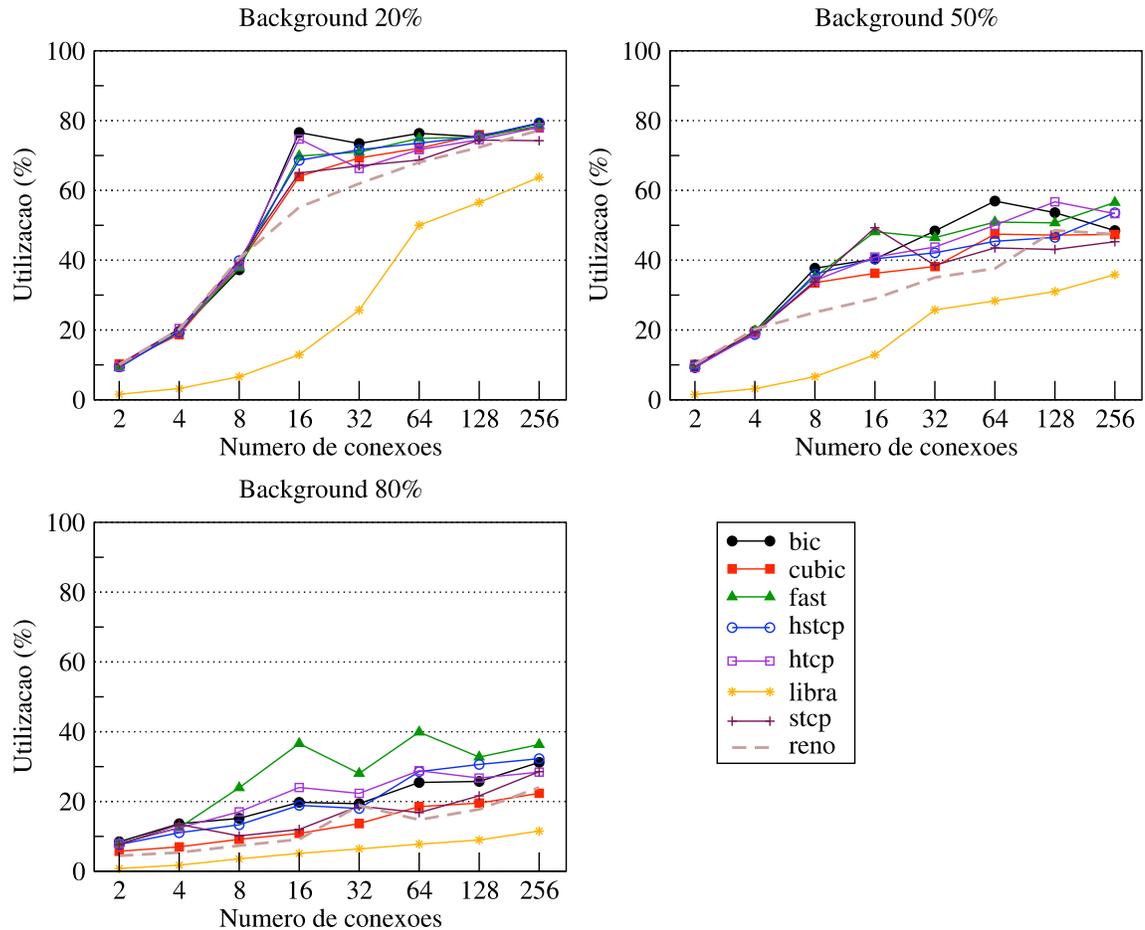


Figura 4.3: Utilização em função do número de conexões para diferentes intensidades de tráfego *background*. Capacidade do enlace gargalo:  $1Gbps$ . Atraso de propagação:  $100ms$ .

A Figura 4.3 é composta por três gráficos que apresentam os resultados obtidos para a escalabilidade em função do número de conexões para as diferentes intensidades de tráfego *background*.

Nestes experimentos, utilizou-se um enlace gargalo de  $1Gbps$  com atraso de propagação de  $100ms$ . Pode-se observar que em um cenário com 20% de tráfego *background*, são

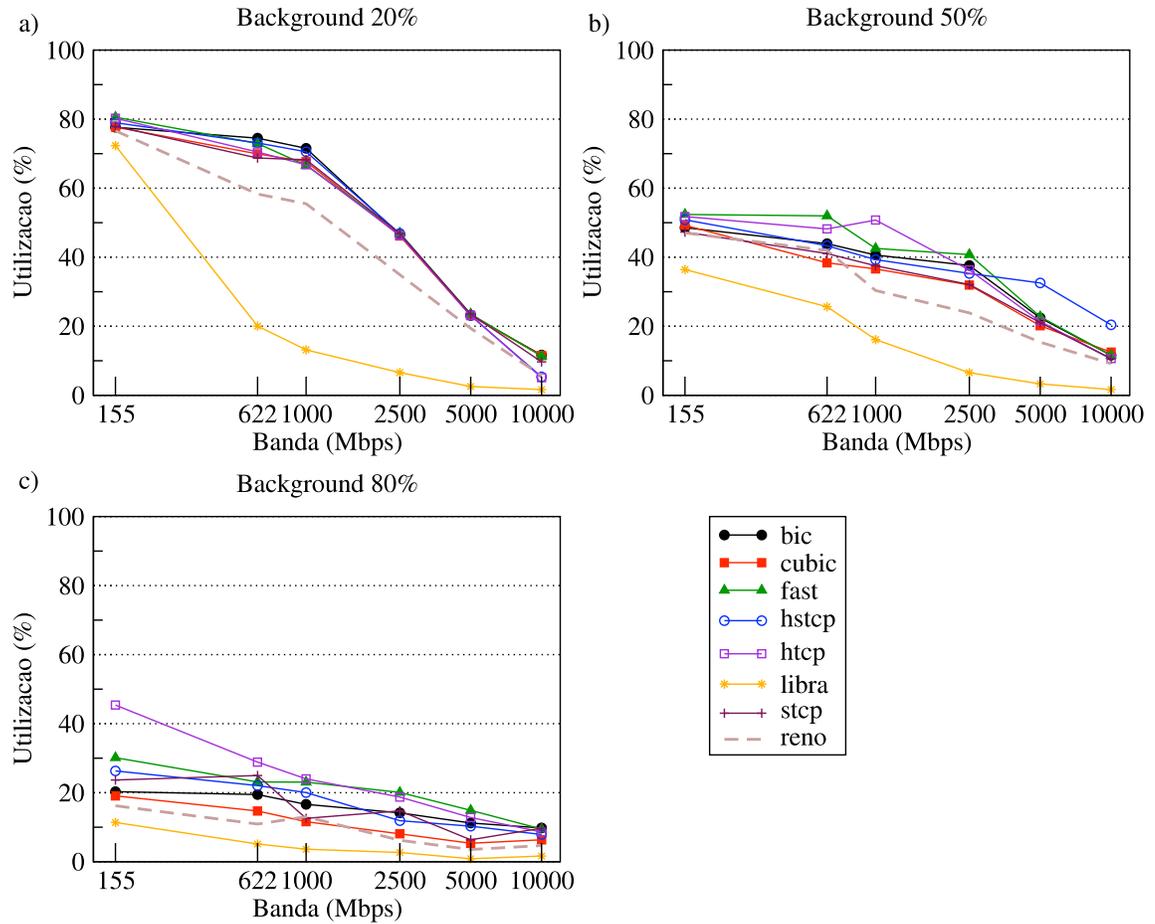


Figura 4.4: Utilização em função da capacidade do enlace para diferentes intensidades de tráfego *background*. Número de conexões: 20. Atraso de propagação no enlace gargalo: 100ms.

necessárias apenas 16 conexões de um protocolo para ocupar o enlace disponível, e incrementar o número de conexões não altera a utilização do enlace. Entretanto, quando a intensidade do tráfego *background* é de 50% ou 80% da capacidade do enlace gargalo, aumentar a quantidade de conexões contribui para melhorar incrementalmente a utilização.

Observa-se também que mesmo com tráfego *background* de 50%, os protocolos FAST, BIC, HTCP e HSTCP conseguem utilizar mais de 50% do enlace quando o número de conexões é superior a 64. O mesmo comportamento pode ser observado no cenário com intenso tráfego *background*, porém neste caso todos os protocolos, exceto LIBRA, alcançam utilização acima de 20%. É interessante notar que os protocolos disputam o enlace com o tráfego *background*, podendo subtrair alguma parcela da banda a este destinada.

Neste conjunto de experimentos, os protocolos FAST e HTCP apresentaram melhores resultados em relação aos demais protocolos. Por outro lado, o protocolo LIBRA mostrou-se pouco escalável, exigindo uma quantidade de conexões extremamente superior aos demais protocolos para atingir o mesmo nível de utilização do enlace. Por exemplo, com 50% de tráfego *background* LIBRA precisou de  $N = 256$  conexões para atingir o nível de utilização alcançado pelos outros protocolos com apenas  $N = 8$  conexões.

A escalabilidade com a capacidade do enlace é apresentada na Figura 4.4. Nestes experimentos fixou-se em 20 o número de conexões para cada protocolo e o atraso de propagação no enlace gargalo em  $100ms$ . Pode-se observar que, com o conjunto de parâmetros utilizados, os protocolos conseguem utilizar eficientemente a banda disponível em enlaces de até  $1Gbps$ . Acima desta capacidade a utilização torna-se gradativamente menor. Este comportamento não determina falta de escalabilidade, uma vez que pode ser originado por dois fatores:

- Quantidade de conexões insuficiente. No experimento anterior foi verificado que 16 conexões seriam capazes de tomar um enlace de  $1Gbps$ , entretanto esse número pode não ser suficiente para ocupar enlaces de maiores capacidades.

Para investigar este caso, repetiu-se a execução das simulações aumentando a quantidade de conexões de  $N = 20$  para  $N = 50$  e  $N = 100$ . Devido ao longo tempo necessário para as simulações, utilizou-se somente o cenário com 20% de tráfego *background*. Foi verificada uma expressiva melhoria na utilização dos enlaces conforme gráficos (a) e (b) da Figura 4.5. Por exemplo, para o protocolo BIC, com enlace de  $C = 5Gbps$  e  $N = 50$  conexões, a utilização subiu de 22% para aproximadamente 57% e com  $N = 100$  conexões, a utilização ficou em torno de 78%.

- *Buffers* pequenos para absorver as rajadas. Segundo [19] para garantir alta utilização do enlace, os *buffers* deveriam ser de 20% do produto banda-atraso. Para enlaces de  $5Gbps$  e  $10Gbps$  com  $RTT = 200ms$  o tamanho dos *buffers* deveria ser de 16666 e 33333 pacotes, respectivamente.

Neste caso, repetiram-se as simulações utilizando *buffers* de 20% do produto banda-atraso. O resultado pode ser observado no gráfico c) da Figura 4.5 que indica que este ajuste de *buffer* não é suficiente para alterar significativamente a utilização do enlace.

Nestes experimentos o protocolo LIBRA apresentou o menor desempenho em escalabilidade enquanto que HTCP e FAST alcançaram melhores níveis de utilização em relação aos demais, quando a intensidade do tráfego *background* é de 50% ou 80% da capacidade do enlace.

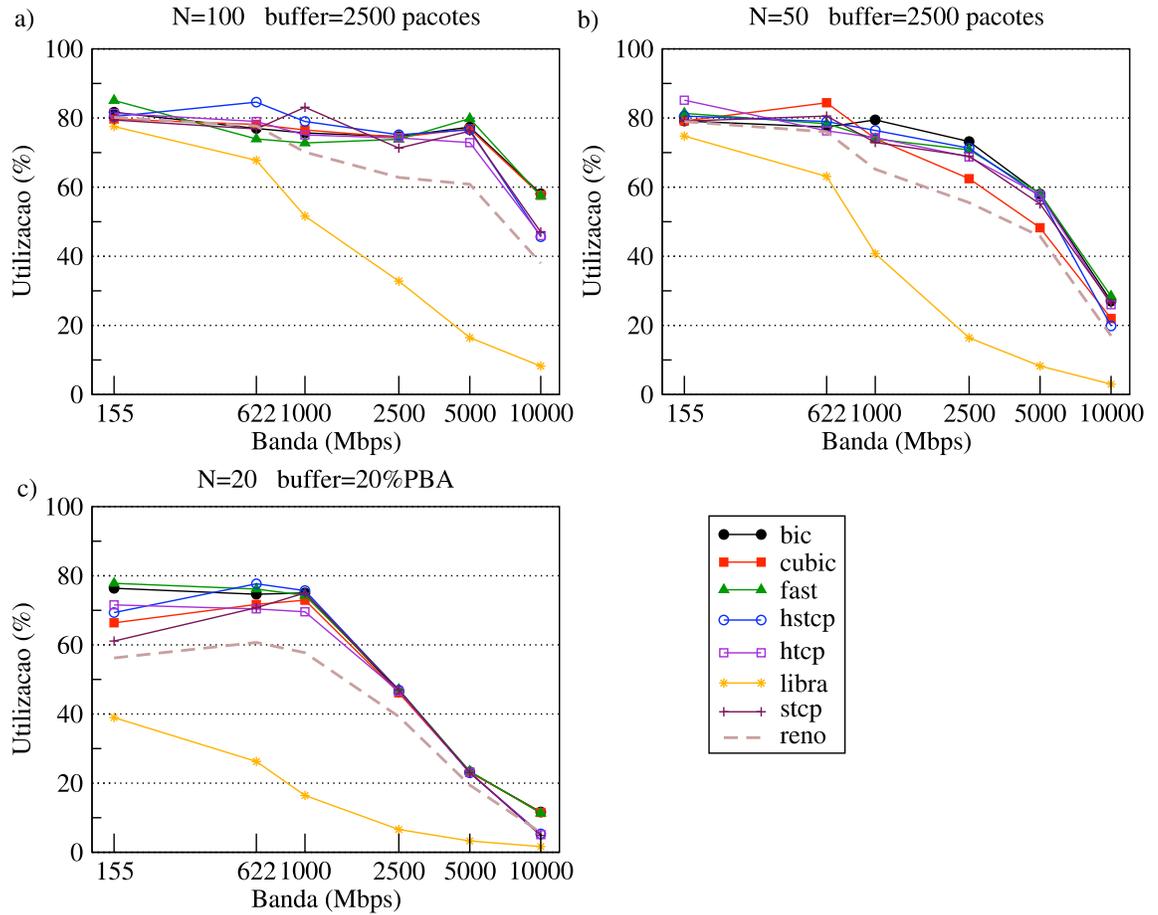


Figura 4.5: Estudo sobre a falta de escalabilidade dos protocolos. Em *a*) fixou-se o número de conexões em  $N = 100$  e o *buffer* em 2500 pacotes. Em *b*) utilizou-se  $N = 50$  conexões e *buffer* de 2500 pacotes. Em *c*) mantiveram-se  $N = 20$  conexões e o tamanho dos *buffers* foi aumentado para 20% do PBA. Todos com 20% de tráfego *background*.

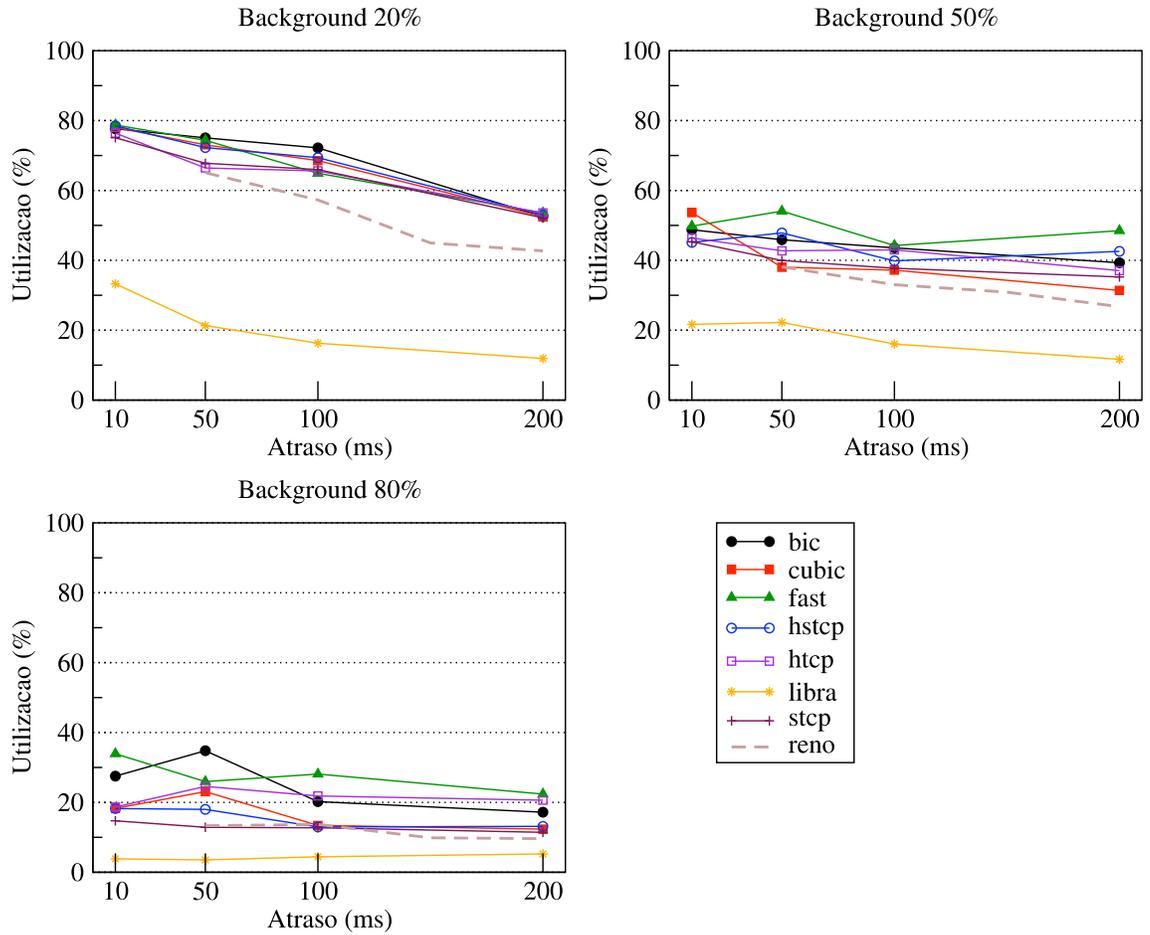


Figura 4.6: Utilização em função do atraso para diferentes intensidades de tráfego *background*. Capacidade do enlace gargalo:  $1Gbps$ . Número de conexões: 20.

A escalabilidade em função do atraso de propagação é apresentada na Figura 4.6. Nestas simulações mantiveram-se fixos o número de conexões para cada protocolo em  $N = 20$  e a capacidade do enlace gargalo em  $C = 1Gbps$ .

Verifica-se que a utilização do enlace diminui proporcionalmente ao aumento do atraso de propagação, independente da intensidade do tráfego *background*. Entretanto a presença do tráfego *background* influencia na variação da utilização do enlace. A Tabela 4.1 mostra que quanto maior a intensidade do tráfego *background* menor é a variação na utilização do enlace, considerando os valores de utilização obtidos para atrasos de propagação de  $10ms$  e  $200ms$ . Isto ocorre porque, com 20% da capacidade do enlace ocupada por tráfego *background*, o intervalo entre as atualizações da janela de congestionamento não é impactado pelo atraso nas filas provocado por enlaces sobrecarregados, mas essencialmente

pelo atraso de propagação.

|       | Tráfego <i>background</i> |        |        |
|-------|---------------------------|--------|--------|
|       | 20%                       | 50%    | 80%    |
| BIC   | 25.14%                    | 9.45%  | 10.30% |
| CUBIC | 25.49%                    | 22.26% | 5.97%  |
| FAST  | 25.16%                    | 1.29%  | 11.54% |
| HSTCP | 25.46%                    | 2.58%  | 5.11%  |
| HTCP  | 22.82%                    | 9.35%  | 1.95%  |
| LIBRA | 21.35%                    | 10.00% | 1.40%  |
| STCP  | 22.91%                    | 10.09% | 3.31%  |

Tabela 4.1: Variação na utilização do enlace para cada protocolo em relação às diferentes intensidades de tráfego *background*. Diferenças entre a utilização obtida para atrasos de 10ms e 200ms.

Todos os protocolos obtiveram desempenho semelhante para o cenário com tráfego *background* de 20%, exceto o protocolo LIBRA, que alcança o menor nível de utilização nos três cenários analisados. Pode-se destacar os protocolos HSTCP e BIC pois conseguem utilizar o enlace com maior eficiência do que os protocolos STCP e HTCP. Com 50% de tráfego *background* o protocolo FAST obteve o melhor desempenho, seguido pelos protocolos HSTCP e BIC. No cenário com tráfego intenso, o protocolo FAST obteve melhor desempenho em relação aos demais protocolos, alcançando utilização sempre acima de 20% do enlace. Os fluxos do protocolo BIC também alcançam utilização acima ou próxima de 20%, mostrando-se eficiente na alocação de banda para enlaces altamente utilizados.

Em resumo, a investigação dos aspectos da propriedade de escalabilidade leva a concluir que o protocolo LIBRA não apresenta escalabilidade desejável, obtendo desempenho inferior ao protocolo TCP Reno. Entretanto, vale ressaltar que o objetivo principal do protocolo LIBRA é alcançar justiça de RTT [48]. O protocolo FAST apresentou bom desempenho na escalabilidade em função do atraso de propagação para intensidades de tráfego *background* de 50% e 80%. HTCP obteve melhor utilização que os demais protocolos nos experimentos de escalabilidade em relação à capacidade do enlace. Em relação à escalabilidade em função do número de conexões pode-se destacar o desempenho do protocolo FAST, que alcançou o melhor nível de utilização no cenário com 80% de tráfego *background*. Além disso, BIC, HTCP e HSTCP também atingem bons níveis de utilização neste aspecto.

A exceção do LIBRA, todos os protocolos avaliados caracterizam-se por apresentar melhor escalabilidade que TCP Reno. Esta observação confirma que estes protocolos

conseguem melhorar a utilização do enlace em relação ao TCP Reno.

### 4.1.2 Compatibilidade com TCP Reno

Esta propriedade caracteriza a compatibilidade dos protocolos de transmissão de dados em alta velocidade com o TCP Reno. Um protocolo é considerado compatível com TCP Reno se os fluxos TCP Reno conseguem alocar a mesma quantidade de banda que alocariam caso não estivessem compartilhando o enlace, tal que os fluxos TCP Reno não sejam penalizados pela presença de fluxos de outros protocolos.

Para avaliar a propriedade de compatibilidade dos protocolos definiu-se um cenário composto por  $N = 20$  conexões TCP Reno, enlace gargalo de  $C = 1Gbps$  e atraso de propagação de  $100ms$ , e fixou-se o tempo total de simulação em  $t = 600$  segundos para garantir que os fluxos alcancem o estado estável. Variou-se o número de conexões dos protocolos de alta velocidade de 0 a 40, utilizando diferentes intensidades de tráfego *background*. Os resultados são apresentados na Figura 4.7 onde cada gráfico mostra a porcentagem do enlace utilizada pelas conexões TCP Reno em função do número de conexões dos protocolos de alta velocidade.

Pode-se observar nos três gráficos que quando conexões TCP Reno executam sozinhas o enlace não é totalmente utilizado. Isto confirma a afirmação [35] de que o TCP Reno não é eficiente na alocação de banda em enlaces de grande capacidade.

O protocolo que apresentou melhor compatibilidade com o TCP Reno foi o TCP LIBRA. Tal comportamento está intrinsecamente relacionado à sua natureza pouco escalável, visando justiça entre fluxos. A equalização da vazão entre os fluxos com RTTs heterogêneos e a utilização de *padding* aleatório contribuem para a estimativa do RTT e, por consequência, para um ajuste apropriado da janela.

Outro protocolo que se destaca nesta propriedade é CUBIC, que tem entre seus objetivos, melhorar a compatibilidade em relação ao seu antecessor, BIC. O incremento linear da janela dependente do tempo real contribui para melhorar a compatibilidade do protocolo pois não privilegia fluxos com diferentes RTTs. Além disso, o decremento proporcional ao tamanho da janela minimiza a disparidade entre taxas de transmissão.

Por outro lado, os protocolos que mais impactam o desempenho do protocolo TCP Reno são as variantes FAST e HTCP. FAST estima a banda disponível a partir de amostras de RTT e tende a utilizar somente a banda excedente. Entretanto, após perdas, por convergir rapidamente ao seu ponto de equilíbrio, obtém uma grande porção da banda.

O ajuste da janela de congestionamento do HTCP depende do tempo decorrido desde o último evento de congestionamento, isso significa que quanto maior a disponibilidade de banda, maior será a taxa de incremento. Quando o tamanho da janela aproxima-se do valor máximo suportado pela rede, incrementos acentuados podem levar à instabilidade

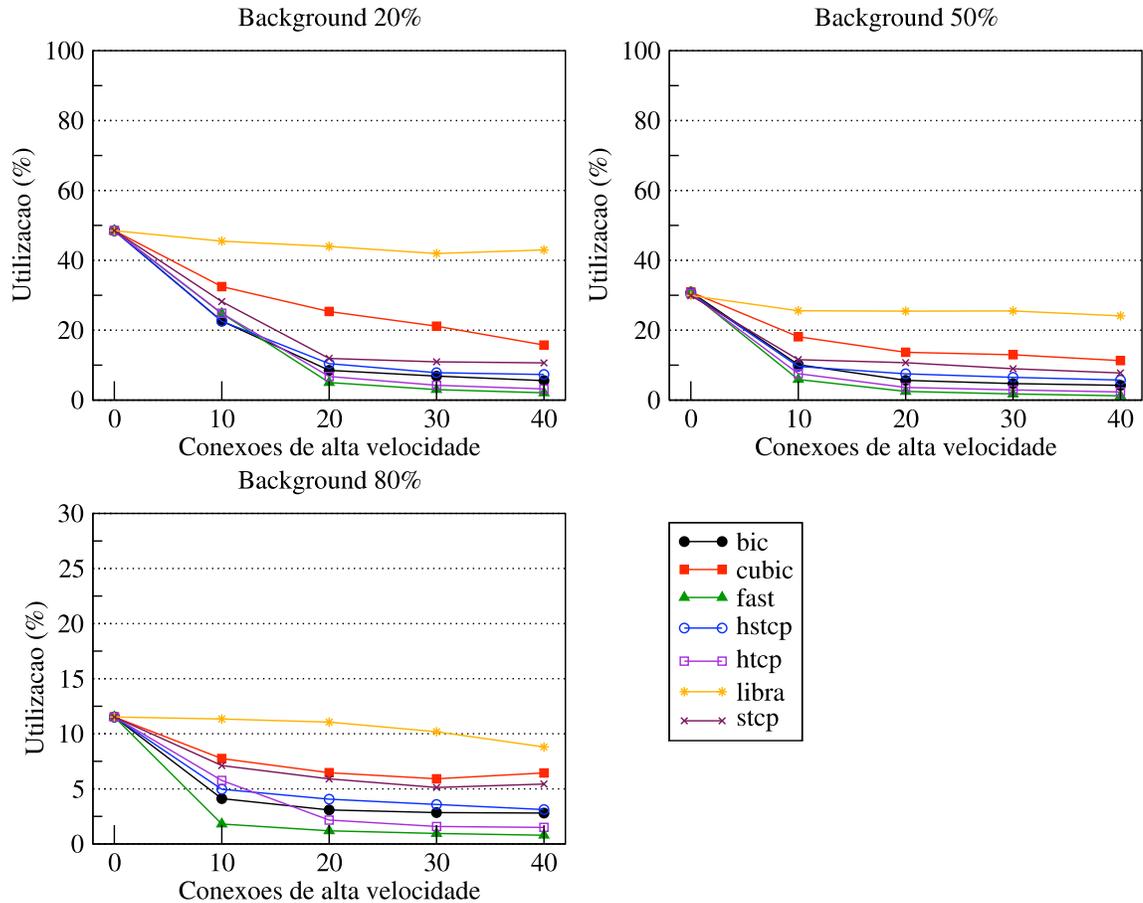


Figura 4.7: Utilização do enlace alcançada pelo TCP Reno em função do número de conexões de protocolos de alta velocidade.

e também à injustiça entre fluxos.

Ao comparar os experimentos de escalabilidade e compatibilidade observa-se que quanto mais escalável for um protocolo menos compatível com o TCP Reno este será.

### 4.1.3 Justiça

A propriedade de justiça permite aferir como os fluxos compartilham a banda disponível. Pode ser subdividida em justiça intra-protocolos, medida entre fluxos do mesmo protocolo e justiça inter-protocolos, medida entre fluxos de diferentes protocolos.

Um protocolo que apresenta justiça garante que nenhuma conexão que participa do compartilhamento do enlace sofrerá de inanição (*starvation*), ou seja, garante que as conexões obterão uma taxa de transmissão satisfatória.

Para investigar a justiça dos protocolos ajustaram-se a capacidade do enlace gargalo para  $C = 1Gbps$ , o atraso de propagação para  $100ms$  e manteve-se em  $N = 20$  o número de conexões. Este cenário foi especificado para atender os aspectos de justiça analisados:

- Justiça intra-protocolos
  - a. Entre fluxos que iniciam no mesmo instante e possuem diferentes RTTs. Este cenário denomina-se Justiça de RTT. Variou-se o atraso de propagação das conexões uniformemente entre  $d_p = 10ms$  e  $d_p = 105ms$ .
  - b. Entre fluxos que iniciam em tempos diferentes e possuem RTTs semelhantes. Variou-se o tempo de início das conexões uniformemente entre  $t = 0$  e  $t = 475$  segundos, de forma que as conexões fossem iniciadas a cada  $\delta = 25$  segundos.
  - c. Entre fluxos que iniciam no mesmo instante e possuem RTTs semelhantes.
- Justiça inter-protocolos: fluxos de diferentes protocolos que iniciam no mesmo instante e possuem RTTs semelhantes. Neste cenário utilizaram-se duas conexões de cada protocolo, somando 14 no total.

A fim de medir a justiça dos protocolos, utilizou-se o *Jain's index* [17]. A equação 4.1 expressa o cálculo deste índice, onde  $n$  é o número de fluxos do tráfego agregado  $j$  e  $v_j$  representa a vazão individual de cada fluxo.

$$\text{índice de Jain} = \frac{(\sum_{j=1}^n (v_j))^2}{n * \sum_{j=1}^n (v_j)^2} \quad (4.1)$$

O índice de Jain varia no intervalo de 0 a 1, e quanto mais próximo de 1 maior é a equidade entre os fluxos. É independente de métrica e escala, e reflete qualquer mudança na alocação da banda entre os fluxos.

### Justiça Intra-protocolos

A Figura 4.8 apresenta três gráficos do índice de Jain em função da intensidade do tráfego *background*. Em (a) apresenta-se a justiça entre fluxos com diferentes RTT; em (b), entre fluxos com diferentes tempos de início; e em (c), entre fluxos com mesmo RTT e instante inicial.

A partir do gráfico (a) observa-se que todos os protocolos apresentam bons índices de justiça intra-protocolos, pois o menor índice de Jain medido é próximo de 0.8. Ainda assim, o protocolo LIBRA é nitidamente o que apresenta melhor justiça de RTT. Isso se deve ao objetivo do projeto deste protocolo que é fornecer justiça de RTT aos fluxos através do componente de controle de justiça, que impõe menor penalidade para fluxos

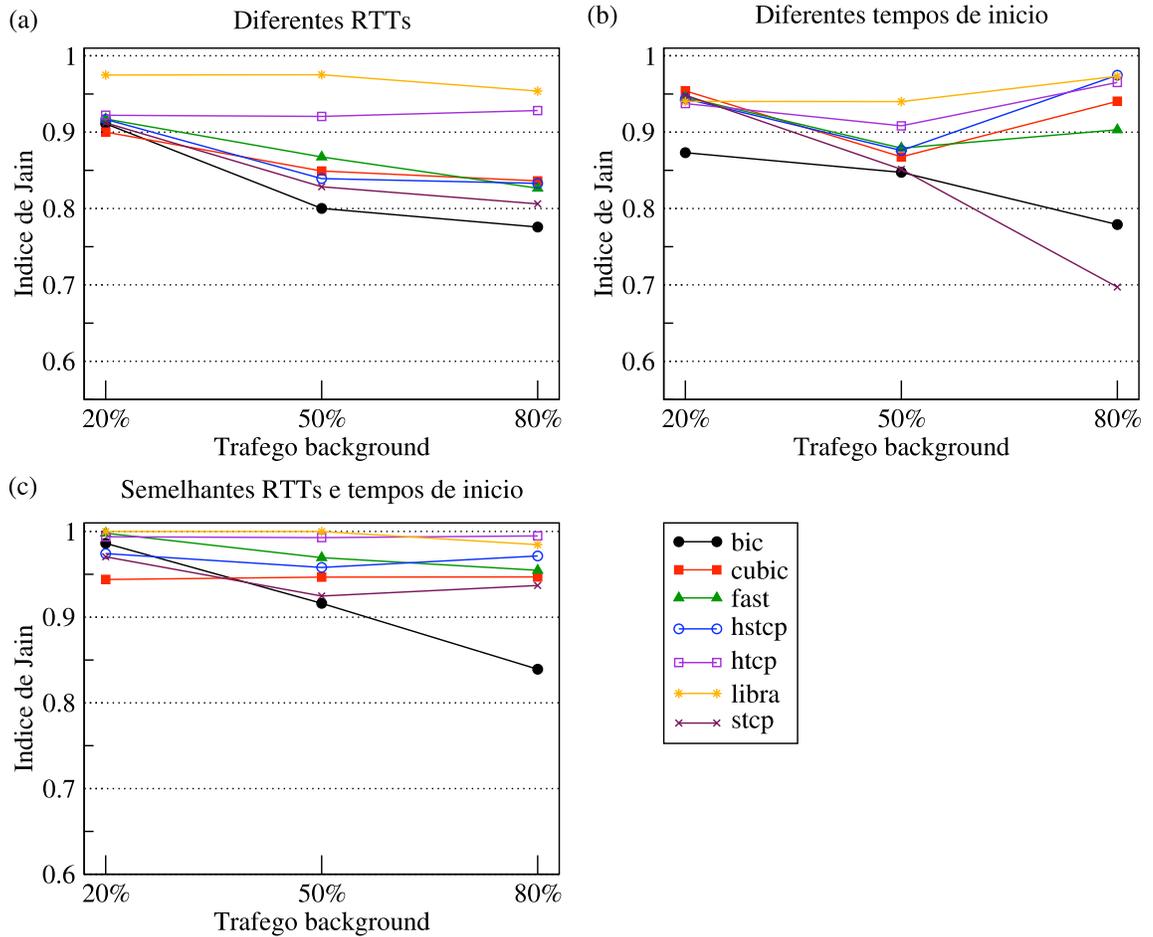


Figura 4.8: Justiça intra-protocolos medido utilizando-se o *Jain's index*. Em (a) o índice de justiça obtido por fluxos de diferentes RTT; em (b) o índice de justiça obtido por fluxos com diferentes tempos de início; e em (c) o índice de justiça entre fluxos com mesmo RTT e instante inicial.

que excedem um certo RTT limite, sendo considerados fluxos que apresentam problemas no caminho.

Destaca-se, também, o comportamento do protocolo HTCP, que mantém o índice de justiça independente da intensidade do tráfego *background*. Além disso, observa-se que os protocolos menos justos em relação ao RTT são BIC e STCP.

A injustiça de RTT é uma característica conhecida do protocolo STCP, pois seu mecanismo de atualização da janela independe do RTT. Por esta razão, os fluxos com RTT menor incrementam a janela, multiplicativamente, com frequência muito maior que fluxos com RTTs longos, alcançando maiores porções da banda.

Em [16] relata-se que CUBIC apresenta injustiça de RTT fortemente dependente do tempo de início das conexões. Entretanto, os experimentos demonstram que CUBIC não é tão injusto em relação ao RTT quanto mencionado. Contudo, esta divergência pode ser decorrente das diferenças de ambiente, tais como a composição do tráfego *background*.

FAST apresentou melhor índice de justiça de RTT que os protocolos HSTCP, STCP e BIC. Isto é condizente com a observação feita em [39].

|       | <i>background</i> 20% |           |        | <i>background</i> 50% |           |        | <i>background</i> 80% |           |        |
|-------|-----------------------|-----------|--------|-----------------------|-----------|--------|-----------------------|-----------|--------|
|       | $\bar{x}$             | <i>cv</i> | Jain   | $\bar{x}$             | <i>cv</i> | Jain   | $\bar{x}$             | <i>cv</i> | Jain   |
| BIC   | 26.979                | 31.238    | 0.9110 | 14.546                | 49.977    | 0.8001 | 6.067                 | 53.764    | 0.7757 |
| CUBIC | 27.262                | 33.322    | 0.9000 | 13.819                | 42.161    | 0.8490 | 4.338                 | 44.267    | 0.8361 |
| FAST  | 28.217                | 30.037    | 0.9172 | 16.265                | 39.084    | 0.8674 | 10.102                | 45.801    | 0.8266 |
| HSTCP | 29.551                | 30.130    | 0.9167 | 16.723                | 43.776    | 0.8391 | 5.064                 | 44.780    | 0.8329 |
| HTCP  | 27.711                | 29.058    | 0.9221 | 17.524                | 29.369    | 0.9205 | 8.333                 | 27.791    | 0.9282 |
| LIBRA | 6.3915                | 16.077    | 0.9748 | 6.3668                | 15.942    | 0.9752 | 1.969                 | 22.022    | 0.9537 |
| STCP  | 28.248                | 31.033    | 0.9121 | 14.409                | 45.501    | 0.8284 | 4.646                 | 49.035    | 0.8061 |

Tabela 4.2: Justiça de RTT: Informações estatísticas de média ( $\bar{x}$ ) em Mbps, coeficiente de variação (*cv*) em porcentagem e índice de Jain para diferentes intensidades de tráfego *background*.

A Tabela 4.2 sintetiza, para cada protocolo, as informações de vazão média ( $\bar{x}$ ) obtida em *Mbps*, o coeficiente de variação (*cv*) da vazão dos fluxos e o índice de Jain. O coeficiente de variação indica o grau de dispersão da vazão obtida pelos fluxos em relação ao valor médio. Expressa a variabilidade dos dados sem a influência da ordem de grandeza da variável e é definido pelo quociente entre o desvio padrão e a média:

$$cv = \frac{s}{\bar{x}} \quad (4.2)$$

Quanto maior o valor de *cv* maior a injustiça entre os fluxos. O que é confirmado pelo

cálculo do índice de Jain. Na Tabela 4.2 pode-se observar os baixos valores de  $cv$  para o protocolo LIBRA e os altos coeficientes obtidos pelo BIC.

O gráfico (b) da Figura 4.8 mostra que o tempo de início da transmissão das fontes e a intensidade do tráfego *background* podem influenciar o processo de compartilhamento da banda dos protocolos. Assim como no gráfico (a), os índices de justiça dos protocolos mantiveram-se elevados, indicando que os fluxos adaptam suas taxas de transmissão quando novos fluxos entram na rede, convergindo para o compartilhamento da banda. É notável, todavia, a injustiça entre os fluxos dos protocolos STCP e BIC, quando a rede está altamente utilizada. Esta tendência é devido ao mecanismo agressivo de incremento da janela destes protocolos, o que leva à instabilidade na taxa de transmissão.

O gráfico (c) mostra que os protocolos que melhor apresentam justiça intra-protocolos são LIBRA, HTCP e FAST. Os dois últimos destacaram-se na propriedade de escalabilidade em função do número de conexões (Figura 4.3). Por outro lado, os protocolos CUBIC e STCP, que obtiveram desempenho inferior aos demais no mesmo aspecto de escalabilidade, apresentaram menor justiça intra-protocolos. Isto indica que quanto mais justo um protocolo, maior a probabilidade deste protocolo alcançar alta utilização. É claro que outros fatores influenciam na utilização do enlace. Por exemplo: LIBRA é um protocolo que apresenta justiça intra-protocolos porém seu desempenho em escalabilidade é inferior aos demais, pois precisa de uma quantidade maior de conexões para melhorar a utilização; BIC é escalável com o número de conexões entretanto suas conexões são injustas entre si.

### Justiça Inter-protocolos

A avaliação da justiça entre diferentes protocolos para redes de alta velocidade tem o objetivo de mensurar possíveis impactos provenientes da interação destes protocolos.

A Figura 4.9 apresenta o índice de Jain medido entre a vazão obtida pelos fluxos de cada protocolo em função das diferentes intensidades de tráfego *background*. Nela é possível observar que a injustiça entre os protocolos é diretamente proporcional à intensidade de tráfego na rede. A vazão obtida por protocolo é apresentada na Tabela 4.3. Nesta tabela, identifica-se com <sup>(+)</sup> os protocolos que alcançaram maior vazão, como BIC, FAST e HTCP; e com <sup>(-)</sup> os que obtiveram vazões menores, tais como STCP, LIBRA e CUBIC. Observa-se que LIBRA e CUBIC são inferiores ao protocolo TCP Reno na obtenção de banda independente da intensidade de tráfego. Isto indica que estes protocolos são demasiado conservadores e não teriam bom desempenho quando coexistindo com outros protocolos para redes de alta velocidade. Além disso, o baixo desempenho do protocolo LIBRA contribui para reduzir o índice de justiça em aproximadamente 10%, como é mostrado na última linha da tabela e na linha tracejada do gráfico, nos quais desprezam-se os valores de vazão obtidos pelo protocolo LIBRA.

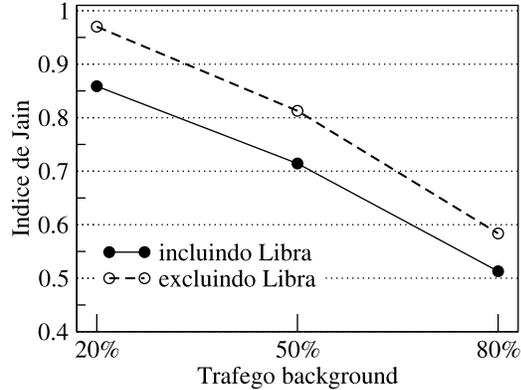


Figura 4.9: Índice de Jain para justiça inter-protocolos em função da intensidade do tráfego *background*.

#### 4.1.4 Convergência

Define-se tempo de convergência entre múltiplos fluxos como o tempo decorrido a partir de um compartilhamento de banda não justo até o alcance de um estado de justiça. Um compartilhamento ideal é aquele que divide os recursos igualmente entre todas as conexões, obtido matematicamente por:

$$\frac{1}{N} \left( C - \frac{C * bg}{100} \right)$$

onde  $N$  é o número de conexões,  $C$  é a capacidade do enlace e  $bg$  é a porcentagem de tráfego *background*.

Para avaliar a propriedade de convergência dos protocolos fixou-se a capacidade do enlace gargalo em  $C = 1Gbps$  com atraso de propagação de  $100ms$ . Utilizaram-se 20 conexões de forma que as primeiras 19 iniciam aleatoriamente entre  $t = 0$  e  $t = 10$  segundos e a última inicia no tempo  $t = 150$ . O objetivo é verificar quanto tempo é necessário para que a última conexão iniciada alcance sua porção justa do enlace.

Nesta propriedade não se obteve resultados para o protocolo LIBRA, pois a falta de escalabilidade deste protocolo limita a obtenção da banda, tornando o estudo desta propriedade não factível, a menos que um número muito maior de conexões fossem utilizadas para este protocolo.

Monitorou-se a quantidade de bits transmitidos pela última conexão iniciada, em intervalos de 1 segundo e calculou-se o tempo decorrido desde o seu início até alcançar a vazão correspondente a sua porção justa. A Figura 4.10 resume o tempo de convergência observado por protocolo.

De acordo com a Figura 4.10 os protocolos BIC e CUBIC precisam de um tempo para convergência maior que os protocolos FAST, HSTCP e HTCP. A lenta convergência de

|                 | Vazão (Mbits)         |                       |                       |
|-----------------|-----------------------|-----------------------|-----------------------|
|                 | <i>background</i> 20% | <i>background</i> 50% | <i>background</i> 80% |
| BIC             | 98.88 <sup>(+)</sup>  | 81.76 <sup>(+)</sup>  | 20.78                 |
| CUBIC           | 51.89 <sup>(-)</sup>  | 16.64                 | 5.216                 |
| FAST            | 98.90 <sup>(+)</sup>  | 77.37 <sup>(+)</sup>  | 58.20 <sup>(+)</sup>  |
| HSTCP           | 96.82                 | 50.45                 | 9.332                 |
| HTCP            | 96.26                 | 67.54                 | 37.45 <sup>(+)</sup>  |
| LIBRA           | 3.745 <sup>(-)</sup>  | 0.802 <sup>(-)</sup>  | 0.346 <sup>(-)</sup>  |
| RENO            | 94.83                 | 60.67                 | 17.81                 |
| STCP            | 92.34                 | 16.26 <sup>(-)</sup>  | 4.935 <sup>(-)</sup>  |
| índice de Jain  | 0.763                 | 0.638                 | 0.464                 |
| índice de Jain* | 0.970                 | 0.817                 | 0.594                 |

Tabela 4.3: Vazão agregada obtida pelos protocolos na simulação do cenário de justiça inter-protocolos. Os valores de índice de Jain\*, excluem a vazão obtida pelo protocolo LIBRA.

CUBIC é também relatada na avaliação experimental deste protocolo [16]. O protocolo STCP levou um longo tempo para convergir no cenário com baixa intensidade de tráfego *background*, porém converge rapidamente em enlaces altamente utilizados. Essa dificuldade de convergência é devido ao paradigma MIMD de atualização da janela, que garante alta utilização mas provoca instabilidade e baixa convergência [56].

Em geral, os protocolos convergem mais rápido com tráfego *background* de 80%. Isto porque a instabilidade gerada pelo tráfego intenso facilita as conexões obterem a sua porção justa do enlace. Todavia, alcançar rapidamente o compartilhamento justo da rede não assegura que este compartilhamento será mantido ao longo do tempo.

#### 4.1.5 Estabilidade

Estabilidade é a capacidade de um sistema em manter inalterável suas propriedades e desempenho durante um certo tempo. No contexto de redes, isto implica nos fluxos alcançarem o estado estável, do inglês *steady state*, e permanecer nele.

Para avaliar a propriedade de estabilidade dos protocolos, fixaram-se a capacidade do enlace gargalo em  $C = 1Gbps$  com atraso de propagação de  $100ms$  e  $N = 20$  conexões. O objetivo foi verificar se os algoritmos de controle de congestionamento adotados pelos protocolos de alta velocidade são capazes de conduzir seus fluxos à estabilidade na taxa de transmissão. Para isso, calculou-se o coeficiente de variação ( $cv$ ) da vazão do tráfego agregado. Em geral, valores de  $cv$  de até 25% são considerados baixos.

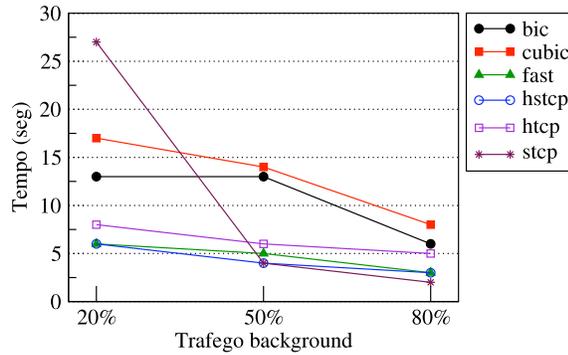


Figura 4.10: Tempo de convergência para a justiça

A Figura 4.11 mostra o coeficiente de variação da vazão do tráfego agregado em função da intensidade do tráfego *background* para cada protocolo.

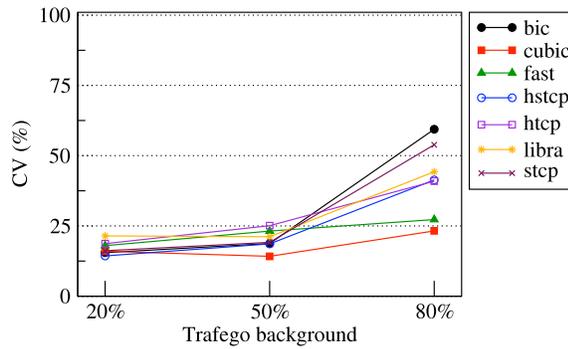


Figura 4.11: Coeficiente de variação da vazão do tráfego agregado em função da intensidade de tráfego *background*

Observa-se que em cenários de baixa e média intensidades de tráfego *background* a variação de *cv* é pequena, indicando que os protocolos alcançam estabilidade na taxa de transmissão. Por outro lado, quando o enlace está altamente utilizado os protocolos tendem à instabilidade e é possível distinguir três níveis:

1. Baixa instabilidade: CUBIC e FAST. Protocolos que mantiveram-se próximo do limiar de 25%, sustentando baixa variabilidade na vazão independente da intensidade de tráfego. Estes protocolos atualizam a janela cautelosamente quando alcançam grandes janelas, não induzindo perdas para controlar a taxa de transmissão.
2. Média instabilidade: LIBRA, HSTCP e HTCP. Protocolos que obtiveram *cv* próximo de 40%.

3. Alta instabilidade: BIC e STCP, obtendo  $cv$  acima de 50%. Ambos possuem mecanismos de incremento da janela bastante agressivos, por isso tornam-se instáveis quando o enlace está altamente utilizado.

A estabilidade é a característica mais marcante do protocolo FAST. Seu algoritmo de atualização da janela usa uma estimativa da probabilidade de perda na tentativa de estabilizar em torno de um ponto de equilíbrio, eliminando oscilações. A estratégia utilizada pelo protocolo CUBIC para alcançar estabilidade é limitar o incremento de grandes janelas, mantendo-o linear no tempo. Isto previne perdas em grandes rajadas e mantém alta utilização. Em contrapartida, o paradigma MIMD adotado por STCP tem tendência natural à instabilidade [56].

BIC e CUBIC, embora tenham curvas de crescimento da janela semelhantes [28], apresentam comportamentos bastante distintos em relação à estabilidade em enlaces altamente utilizados.

#### 4.1.6 Síntese: Topologia *Dumbbell*

Neste capítulo, foram especificados cenários utilizando-se uma topologia *dumbbell* para analisar as propriedades de escalabilidade, justiça, convergência, compatibilidade e estabilidade de protocolos variantes do TCP para redes de alta velocidade. Os protocolos avaliados foram: BIC, CUBIC, FAST, HSTCP, HTCP, LIBRA e STCP. Nesta seção, sintetizam-se os resultados obtidos:

##### ◇ BIC

Este protocolo mostrou-se escalável nos três aspectos investigados; apresentou baixa compatibilidade com TCP Reno; entre os protocolos avaliados alcançou o menor índice de justiça intra-protocolos nas três diferentes avaliações, especialmente entre conexões com diferentes tempos de início, para enlaces com tráfego *background* acima de 20%; possui tempo de convergência de aproximadamente 12 segundos e em enlaces altamente utilizados torna-se o mais instável dos protocolos, atingindo um coeficiente de variação da vazão de 59%.

##### ◇ CUBIC

Não obteve destaque nos aspectos de escalabilidade, porém apresentou boa compatibilidade com o TCP Reno, sendo inferior apenas ao protocolo LIBRA; alcança justiça de RTT e justiça entre fluxos que iniciam em diferentes instantes; exibiu estabilidade indiferente à intensidade de tráfego. Por outro lado, possui o maior tempo para convergência entre os protocolos investigados, variando de 17 segundos (*background* de 20%) a 12 segundos (*background* 80%).

## ◇ FAST

Protocolo escalável com o número de conexões, com o atraso de propagação e com a capacidade do enlace. Apresenta baixa compatibilidade com TCP Reno, impedindo que fluxos deste alcancem mais de 5% do enlace disponível quando o número de conexões FAST é igual ou maior ao número de conexões TCP Reno; seus fluxos são capazes de alcançar justiça de RTT e compartilhamento justo da banda entre conexões que iniciam em diferentes instantes; converge rapidamente e é estável independente da intensidade do tráfego *background*.

## ◇ HSTCP

Apresenta escalabilidade e, assim como outros protocolos agressivos no incremento da janela, não mostrou-se compatível com TCP Reno. Eventualmente alcança justiça entre fluxos com diferentes RTTs e diferentes tempos de início. Converte rapidamente independente da intensidade do tráfego *background* e pode apresentar instabilidade em enlaces altamente utilizados.

## ◇ HTCP

Protocolo escalável com a capacidade do enlace e pouco compatível com TCP Reno. Apresenta índice de justiça de RTT acima de 0.9 para qualquer intensidade de tráfego *background*, além de alcançar um dos melhores índices de justiça entre fluxos que iniciam em diferentes instantes. Possui curto tempo de convergência, demorando no máximo 8 segundos para o caso de enlace com 20% de tráfego *background*. Quanto à estabilidade, assemelha-se ao protocolo HSTCP, sendo instável em enlaces altamente utilizados.

## ◇ LIBRA

A escalabilidade é um dos seus pontos fracos, tendo desempenho inferior ao protocolo TCP Reno; apresenta alta compatibilidade com TCP Reno e excelente índice de justiça de RTT, acima de 0.95, independente da intensidade de tráfego *background*. Obteve o melhor indicador de justiça entre fluxos que iniciam em tempos diferentes. O tempo de convergência é desconhecido, pois sua falta de escalabilidade inviabiliza o estudo desta propriedade. É estável em cenários com tráfego *background* de até 50%.

## ◇ STCP

É escalável e não compatível com TCP Reno, embora nesta última seja inferior apenas aos protocolos LIBRA e CUBIC. Alcança, eventualmente, justiça de RTT e não atinge um compartilhamento justo da banda entre fluxos que iniciam em diferentes tempos para enlaces altamente utilizados. Seu comportamento de convergência é indeterminado, podendo convergir e divergir, ou mesmo nunca convergir. Alcança

estabilidade em enlaces com média-baixa utilização e torna-se altamente instável em enlaces sobrecarregados.

## 4.2 Topologia *Parking-lot*

Nesta seção descrevem-se os resultados dos experimentos otidos através de simulações utilizando a topologia *parking lot*, caracterizada pela presença de múltiplos enlaces gargalos. A Figura 4.12, retirada de [48], ilustra a topologia composta por fontes de tráfego principal e *background*, representadas, respectivamente, por círculos e hexágonos.

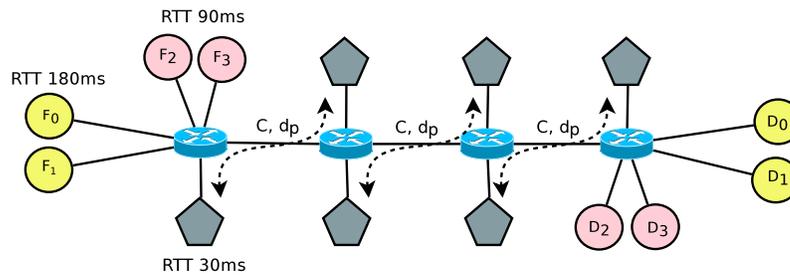


Figura 4.12: Topologia *parking lot*. Fontes e destinos do tráfego principal são representados por círculos, enquanto os nós envolvidos na transmissão do tráfego *background*, por hexágonos. As linhas pontilhadas indicam o caminho percorrido pelo tráfego *background*.

Neste tipo de topologia são comuns a presença de fontes com diferentes RTTs e de caminhos com diferentes números de saltos. Em todos os experimentos, 50% das fontes de tráfego TCP principal possuem  $RTT = 90ms$ , e o restante,  $RTT = 180ms$ , enquanto as fontes de tráfego *background* possuem  $RTT = 30ms$  com a mesma composição utilizada na topologia *dumbbell*: 20% UDP CBR, 24% FTP e 56% WEB. Fixaram-se o tempo total de simulação em  $T = 600$  segundos e os *buffers* em 2500 pacotes.

Foram investigadas as propriedades de escalabilidade em função da capacidade do enlace e do número de conexões; justiça intra e inter-protocolos; e compatibilidade com TCP Reno. Os resultados obtidos são descritos nas próximas subseções.

### 4.2.1 Escalabilidade

A propriedade de escalabilidade foi investigada sob dois aspectos: em função do número de conexões e em função da capacidade do enlace. Para investigar o primeiro aspecto, variou-se a quantidade de conexões em  $2^i$ ,  $i = 2..8$  e manteve-se fixa a capacidade dos enlaces gargalos em  $C = 1Gbps$ .

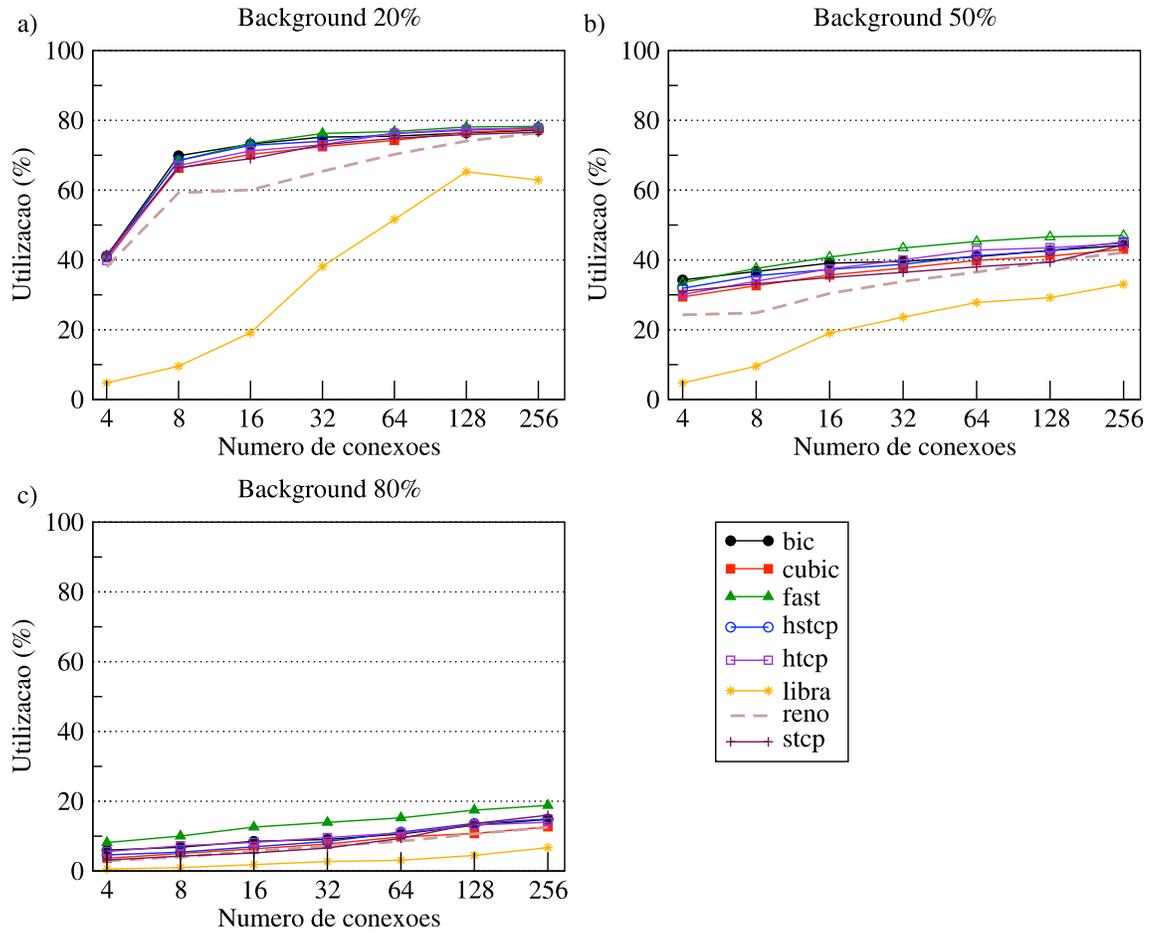


Figura 4.13: Utilização em função do número de conexões para diferentes intensidades de tráfego *background*. Capacidade dos enlaces:  $C = 1\text{Gbps}$ .

A Figura 4.13 apresenta a utilização do enlace em função do número de conexões para diferentes intensidades de tráfego *background*. No gráfico (a) pode-se observar que as mesmas 16 conexões necessárias para ocupar um único enlace gargalo (topologia *dumbbell*) também garantem alta utilização na topologia *parking lot*. Porém, aumentar o número de conexões concorrentes contribui para melhorar a utilização do enlace, que aproxima-se da capacidade disponível com 128 conexões. Este mesmo comportamento pode ser observado nos cenários com maior intensidade de tráfego *background*, apresentados nos gráficos (b) e (c). Por esta razão, empregou-se  $N = 100$  conexões no estudo da escalabilidade em função da capacidade dos enlaces.

Conclui-se que os protocolos avaliados, com exceção do LIBRA, são escaláveis com o número de conexões, pois não há queda na utilização do enlace mesmo na presença

de um grande número de conexões. Assim como observado na topologia *dumbbell* o protocolo LIBRA é o menos escalável em função do número de conexões, apresentando desempenho inferior ao TCP Reno. Entretanto, a utilização do enlace aumenta com o número de conexões. Isto sugere que LIBRA é capaz de ocupar enlaces de grande capacidade desde que um grande número de conexões esteja disponível. Os protocolos FAST e HTCP mostraram-se ligeiramente superiores aos demais protocolos neste aspecto de escalabilidade, independente da intensidade do tráfego *background*, sugerindo que estes protocolos compartilham o enlace de forma eficiente, com justiça entre as conexões.

A Figura 4.14 apresenta os resultados obtidos para a escalabilidade em função da capacidade do enlace para diferentes cargas de tráfego *background*. Para este cenário, utilizou-se  $N = 100$  conexões e variou-se a capacidade dos enlaces gargalo de  $C = 155\text{Mb}$  a  $10\text{Gb}$ .

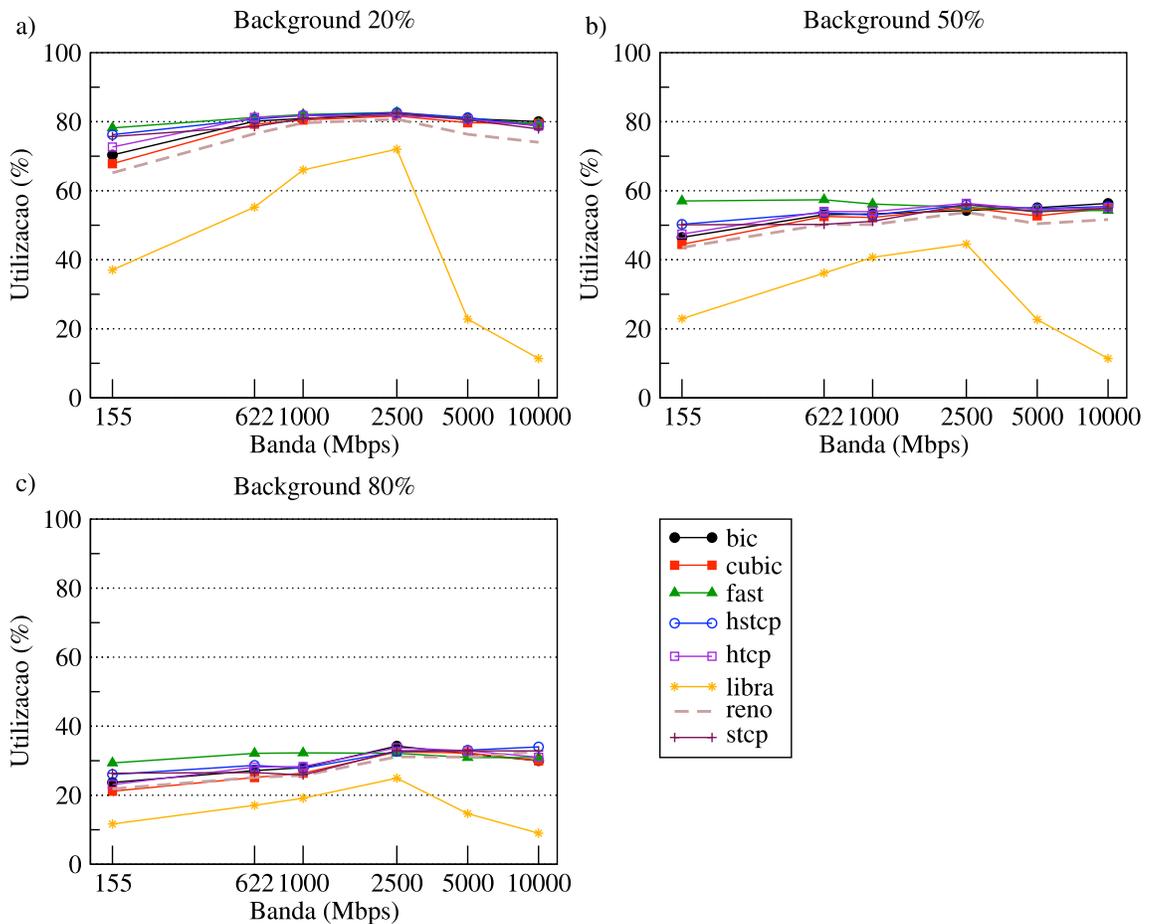


Figura 4.14: Utilização em função da capacidade do enlace para diferentes intensidades de tráfego *background*. Número de conexões:  $N = 100$ .

Observa-se nos três gráficos que todos os protocolos, a exceção do LIBRA, alcançam utilização acima do TCP Reno independentemente da capacidade dos enlaces ou da intensidade do tráfego *background*. Além disso, em enlaces de  $C = 155\text{Mb}$  existe uma dispersão na porcentagem de utilização. Este comportamento ocorre devido as constantes perdas de pacotes provocadas pelo grande número de conexões, e assim os protocolos mais agressivos aumentam suas janelas mais rapidamente, reduzindo o tempo de subutilização, enquanto protocolos um pouco mais conservadores acabam por ter uma utilização menor.

Quando a porcentagem de tráfego *background* é de 50% ou 80% os protocolos são capazes de alocar uma parcela da banda maior que a esperada, isto é, mais de 50% do enlace no cenário com 50% de tráfego *background* e mais de 20% no cenário com 80% de tráfego *background* mesmo em enlaces de grande capacidade ( $C \geq 2500\text{Mb}$ ). Isto indica que os protocolos são competitivos e disputam o enlace com outros fluxos.

Destaca-se o bom desempenho do FAST que obteve utilização acima dos demais protocolos em enlaces de menor capacidade (155Mb e 622Mb), enquanto CUBIC obteve menor utilização. Em enlaces de maior capacidade, todos os protocolos apresentaram desempenho semelhante.

### 4.2.2 Compatibilidade com TCP Reno

A compatibilidade dos protocolos em relação ao TCP Reno foi avaliada utilizando-se uma composição de conexões do protocolo TCP Reno com protocolos de alta velocidade, sob diferentes intensidades de tráfego *background*.

Empregou-se  $N = 20$  conexões do protocolo TCP Reno e variou-se a quantidade de conexões de outros protocolos de 0 a 40 de forma que 50% das conexões TCP Reno e 50% das conexões do TCP variante possuíssem  $RTT = 90\text{ms}$ , e as demais  $RTT = 180\text{ms}$ .

Os resultados obtidos são apresentados na Figura 4.15, onde cada gráfico mostra a porcentagem do enlace utilizada pelas conexões TCP Reno em função do número de conexões dos protocolos de alta velocidade.

Neste experimento observa-se um comportamento semelhante ao obtido na topologia *dumbbell*. O protocolo LIBRA mostrou-se extremamente compatível com o TCP Reno, seguido pelos protocolos CUBIC e STCP; enquanto que os protocolos FAST e HTCP mostraram-se pouco compatíveis, sugerindo novamente que os protocolos mais escaláveis são menos compatíveis com o TCP Reno. Uma observação interessante é que na topologia *parking lot* ocorre um ganho significativo na utilização do enlace, comparada a topologia *dumbbell*, de aproximadamente 10, 15 e 20 pontos percentuais para os cenários com tráfego *background* de 80%, 50% e 20%, respectivamente.

A seqüência dos gráficos chama a atenção para a tendência dos protocolos CUBIC e STCP. À medida que o tráfego *background* torna-se mais intenso, CUBIC torna-se

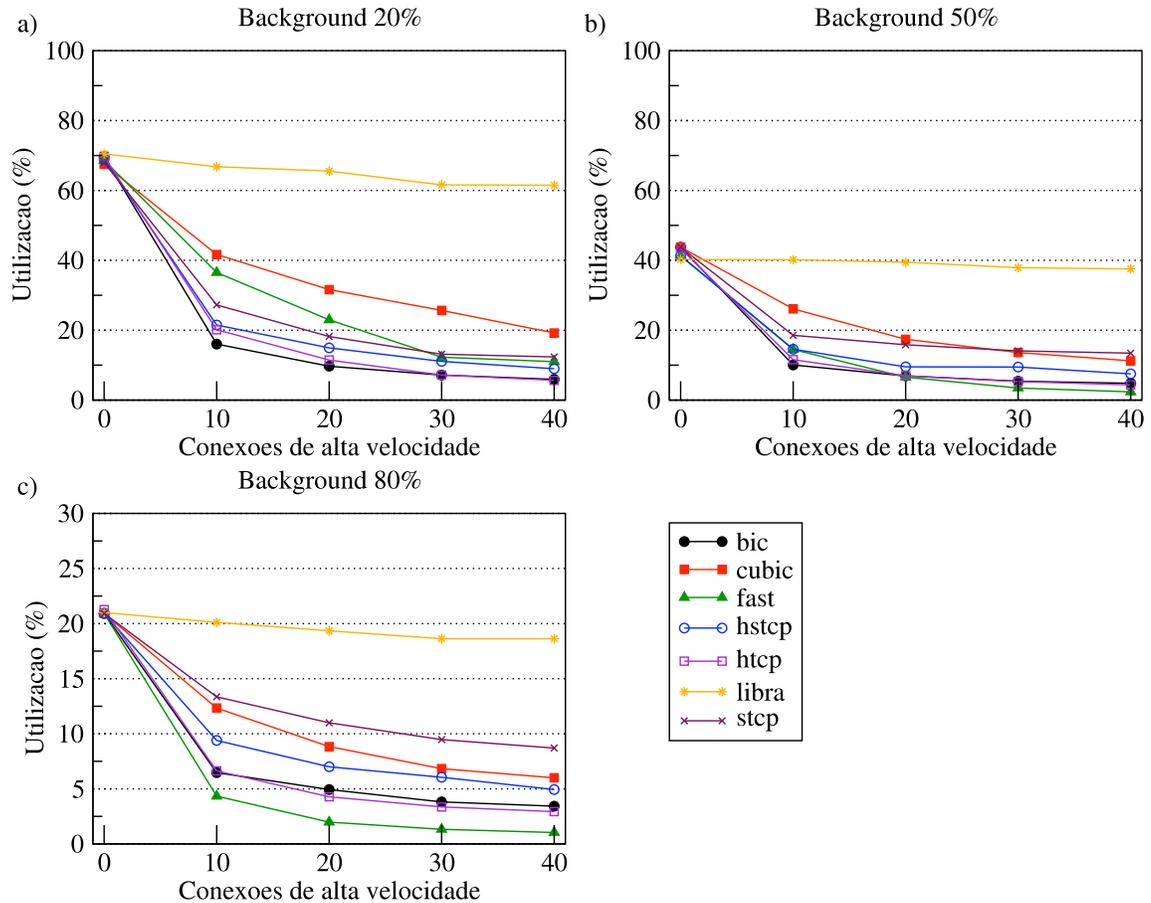


Figura 4.15: Compatibilidade com TCP Reno em função do número de conexões de alta velocidade sob diferentes intensidades de tráfego *background* utilizando a topologia *parking-lot*.

menos compatível que STCP, esta transição ocorre quando 50% do enlace é ocupado por tráfego *background*. Além disso, pode-se observar que a compatibilidade do STCP melhora à medida que o enlace torna-se mais sobrecarregado. Isto porque, o paradigma MIMD adotado por STCP induz grande quantidade de perdas, o que mantém as janelas menores, contribuindo para a compatibilidade [57].

### 4.2.3 Justiça

Investigou-se a propriedade de justiça observando os aspectos intra e inter-protocolos. Na análise da justiça intra-protocolos, fixaram-se a capacidade dos enlaces gargalo em  $C = 1Gbps$  e o número de conexões em  $N = 20$  e variou-se a intensidade do tráfego

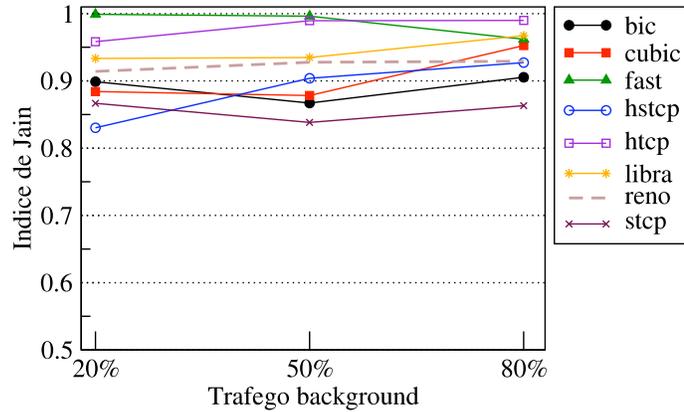


Figura 4.16: Justiça intra-protocolos. Índice de Jain em função da intensidade do tráfego *background*. Parâmetros utilizados:  $N = 20$  conexões,  $C = 1$  Gbps.

*background*. Calculou-se a vazão média alcançada por conexão e o índice de Jain. A Figura 4.16 apresenta os resultados obtidos.

Neste gráfico observa-se que os protocolos FAST, HTCP e LIBRA apresentam índices de justiça intra-protocolos acima de 0.9, independente da intensidade de tráfego *background*, indicando que tais protocolos alcançam um compartilhamento justo da banda disponível entre conexões com diferentes RTTs. FAST e HTCP destacaram-se também no aspecto de escalabilidade em função do número de conexões, sendo um indício de que protocolos escaláveis que compartilham o enlace de forma justa entre seus fluxos, conseguem melhores índices de utilização.

No outro extremo, HSTCP e STCP mostraram-se os protocolos menos justos em relação a conexões com diferentes RTTs. No caso do protocolo HSTCP, o índice de justiça aumenta à medida que o enlace torna-se mais sobrecarregado, enquanto que STCP penaliza conexões com maior RTT. A injustiça de RTT verificada por estes dois protocolos é também relatada em [57, 10].

A Tabela 4.4 resume a vazão média (*Mbps*) obtida por grupo de conexões. Pode-se observar que as colunas referentes às conexões com menor RTT (*90ms*) alcançam maiores porções do enlace. Há uma tendência de melhoria da justiça de RTT entre as conexões à medida que a intensidade do tráfego *background* aumenta. Isso se deve ao maior enfileiramento de pacotes nos *buffers* dos roteadores, que equilibra a disparidade dos RTTs.

É interessante notar que HTCP apresenta uma tendência diferente dos demais protocolos. O ajuste da janela em função do tempo garante a justiça, permitindo até mesmo que conexões com maior RTT obtenham maior vazão que conexões com menor RTT, que pode ser observado nas colunas referentes ao tráfego *background* de 50% e 80%. Esta

|       | Tráfego <i>background</i> |        |        |        |       |       |
|-------|---------------------------|--------|--------|--------|-------|-------|
|       | 20%                       |        | 50%    |        | 80%   |       |
|       | 90ms                      | 180ms  | 90ms   | 180ms  | 90ms  | 180ms |
| BIC   | 50.04                     | 27.12  | 33.90  | 15.54  | 3.44  | 2.72  |
| CUBIC | 46.88                     | 27.17  | 29.92  | 15.10  | 4.24  | 2.75  |
| FAST  | 40.27                     | 37.90  | 27.11  | 24.21  | 8.44  | 5.62  |
| HSTCP | 54.75                     | 22.81  | 30.02  | 17.73  | 13.58 | 7.90  |
| HTCP  | 44.69                     | 33.37  | 22.72  | 25.67  | 3.921 | 4.47  |
| LIBRA | 14.52                     | 8.32   | 14.47  | 8.34   | 7.04  | 5.07  |
| STCP  | 43.54                     | 32.17  | 37.92  | 21.92  | 13.50 | 6.61  |
| Média | 42.103                    | 26.982 | 28.013 | 18.365 | 7.742 | 5.024 |

Tabela 4.4: Vazão média (*Mbps*) obtida por grupo de conexões por protocolo.

análise da justiça intra-protocolos do HTCP confirma os resultados apresentados em [32].

### Justiça Inter-Protocolos

Para investigar a justiça inter-protocolos foram adotadas duas conexões de cada protocolo de alta velocidade, sendo uma com RTT de 90ms e outra com RTT de 180ms. Calculou-se o índice de Jain e a vazão obtida por protocolo.

A Figura 4.17 (a) apresenta o índice de Jain obtido pelas conexões dos diferentes protocolos. Observa-se que sob tráfego *background* de 20% as conexões com RTT de 180ms são mais justas entre si do que conexões com RTT de 90ms. A Tabela 4.5 explica esse comportamento detalhando a vazão média (em *Mbps*) obtida pelas conexões de cada protocolo. Pode-se observar que os protocolos que alcançam maior vazão média (BIC, HSTCP e STCP) não levam o RTT em consideração para o ajuste da janela de congestionamento. Desta forma, são mais agressivos na alocação de banda, principalmente para as conexões com menor RTT, que atualizam a janela de transmissão com maior frequência. Nos demais protocolos, é menor a discrepância entre as vazões obtidas, e por consequência, é maior a injustiça entre conexões de menor RTT.

Com tráfego *background* de 50% e 80% ocorre o contrário, sendo as conexões com menor RTT mais justas entre si. O aumento do tráfego contribui para que ocorram mais perdas e maior enfileiramento de pacotes nos *buffers* fazendo com que as conexões mais agressivas reduzam sua janela mais vezes, diminuindo, assim, a vazão média. Desta forma, a diferença na vazão obtida entre os protocolos é minimizada, aumentando a justiça.

No cenário com tráfego *background* de 80% as conexões do protocolo FAST são responsáveis pela queda no índice de Jain, pois, devido a sua escalabilidade, alcançam uma

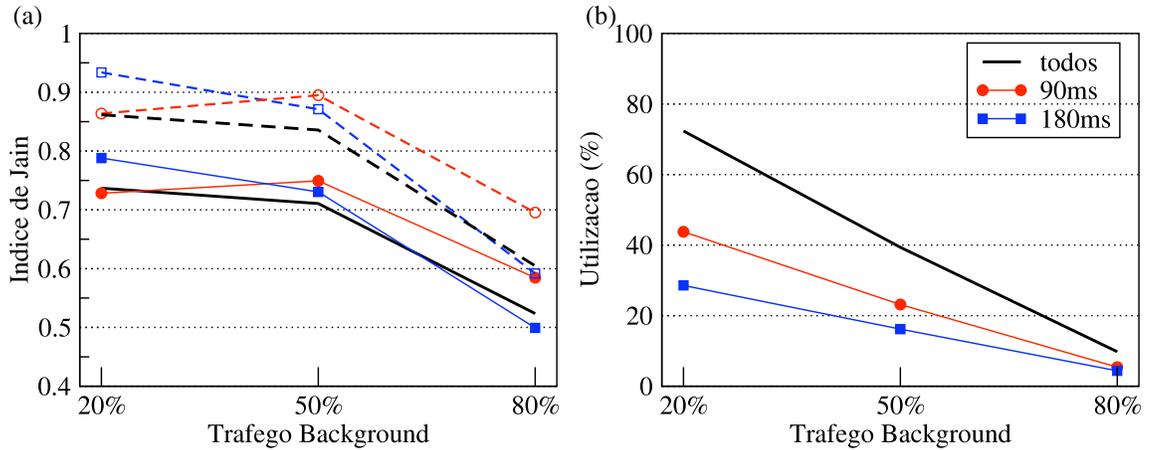


Figura 4.17: Justiça inter-protocolos. Em (a) o índice de Jain entre fluxos com diferentes RTTs. As linhas tracejadas excluem o protocolo LIBRA. Em (b) a porcentagem de utilização do enlace obtida pelas conexões de diferentes RTTs. Parâmetros utilizados:  $C = 1Gbps$ ;  $N = 14$  conexões, sendo duas por protocolo, uma com  $RTT = 90ms$  e outra com  $RTT = 180ms$ .

porção do enlace consideravelmente maior que as conexões dos outros protocolos. Além disso, os protocolos BIC e HSTCP mantiveram uma grande diferença de vazão entre suas conexões, diminuindo a justiça inter-protocolos para as conexões de  $RTT$  de  $180ms$ .

Olhando para o conjunto completo de conexões é possível verificar que os protocolos não alcançam um compartilhamento justo da banda disponível em ambientes bastante heterogêneos, com protocolos e  $RTT$  diferentes. Entretanto, devido a baixa escalabilidade do LIBRA, o índice de justiça foi recalculado excluindo-se a vazão obtida por ele e observou-se um ganho significativo na justiça inter-protocolos. O resultado é apresentado nas linhas tracejadas do gráfico (a).

A Figura 4.17 (b) apresenta a utilização do enlace em função da intensidade do tráfego *background*. Observa-se que, em geral, as conexões com menor  $RTT$  ( $90ms$ ) alcançam maior vazão que as conexões com maior  $RTT$  ( $180ms$ ). De acordo com a Tabela 4.5 os protocolos BIC, HSTCP e HTCP foram os três que ocuparam a maior parte da banda disponível, enquanto LIBRA e CUBIC tomaram menor parcela do enlace. Este mesmo comportamento foi observado no experimento análogo realizado na topologia *dumbbell*.

#### 4.2.4 Síntese: Topologia *Parking-lot*

Neste capítulo foram analisadas as propriedades de escalabilidade, justiça e compatibilidade de protocolos variantes do TCP para redes de alta velocidade, em uma topologia *parking-lot*. Nesta seção sintetizam-se os resultados obtidos.

|       | Tráfego <i>background</i> |        |        |        |        |        |
|-------|---------------------------|--------|--------|--------|--------|--------|
|       | 20%                       |        | 50%    |        | 80%    |        |
|       | 90ms                      | 180ms  | 90ms   | 180ms  | 90ms   | 180ms  |
| BIC   | 98.060                    | 60.220 | 56.338 | 38.803 | 11.083 | 5.8370 |
| CUBIC | 27.026                    | 25.292 | 14.738 | 9.4354 | 3.5046 | 2.5001 |
| FAST  | 47.840                    | 42.093 | 44.570 | 39.177 | 22.736 | 21.274 |
| HSTCP | 93.817                    | 56.933 | 47.492 | 25.279 | 4.7649 | 2.6487 |
| HTCP  | 82.559                    | 52.517 | 52.830 | 38.158 | 9.4433 | 9.5773 |
| LIBRA | 2.0796                    | 1.5350 | 0.5082 | 0.4766 | 0.2290 | 0.2612 |
| STCP  | 86.322                    | 47.185 | 15.264 | 10.577 | 2.4034 | 1.6861 |
| Média | 62.529                    | 40.825 | 33.105 | 23.129 | 7.737  | 6.254  |

Tabela 4.5: Vazão média (Mbps) obtida por grupo de conexões por protocolos. O primeiro grupo é composto pelas conexões com longo RTT (180ms) enquanto o segundo é composto pelas conexões com RTT de 90ms.

◇ BIC

Protocolo escalável em função da capacidade do enlace e do número de conexões; mostrou-se pouco compatível com o TCP Reno, especialmente no cenário com 20% de tráfego *background*; apresenta um dos menores índices de justiça intra-protocolos entre os protocolos avaliados, sendo superior somente ao STCP.

◇ CUBIC

Escalável em relação aos dois aspectos de escalabilidade investigados, porém, produziu a menor utilização nos enlaces de menor capacidade (155Mbps e 622Mbps), exceto pelo o protocolo LIBRA; é um dos protocolos mais compatíveis com o TCP Reno, sendo inferior apenas ao LIBRA e ao STCP, em enlaces sobrecarregados; quanto à justiça intra-protocolos, possui desempenho semelhante ao BIC, mostrando-se mais justo em enlaces com tráfego *background* intenso.

◇ FAST

Escalável em função do número de conexões e da capacidade do enlace; produziu a melhor utilização nos enlaces de menor capacidade (155Mb e 622Mb) indiferente à intensidade de tráfego *background*; à medida que o enlace torna-se mais sobrecarregado, menor é sua compatibilidade com TCP Reno; é um dos protocolos que geram melhores índices de justiça intra-protocolos.

◇ HSTCP

Protocolo escalável nos dois aspectos investigados: capacidade do enlace e número

de conexões; não mostrou-se compatível com o protocolo TCP Reno; não produz justiça intra-protocolos em enlaces com 20% de tráfego *background*, tornando-se mais justo em enlaces sobrecarregados.

◇ HTCP

Apresentou escalabilidade em ambos os aspectos investigados, entretanto, pouca compatibilidade com TCP Reno, sendo um dos protocolos menos compatíveis; é um dos protocolos com maior índice de justiça intra-protocolos.

◇ LIBRA

Baixo desempenho nos aspectos de escalabilidade, porém extremamente compatível com o TCP Reno; apresenta índice de justiça intra-protocolos acima de 0.9 independente da intensidade de tráfego *background*.

◇ STCP

Assim como a maioria dos protocolos, apresentou bom desempenho em ambos os aspectos de escalabilidade; sua compatibilidade com TCP Reno aumenta à medida que o enlace torna-se sobrecarregado, sendo o protocolo mais compatível (depois do LIBRA) em cenários com 80% de tráfego *background*; por outro lado, as conexões deste protocolo apresentaram o maior índice de injustiça intra-protocolos.

### 4.3 Análise comparativa do desempenho dos protocolos investigados

Nas seções anteriores, foi descrito o desempenho dos protocolos, sob a perspectiva das propriedades investigadas: escalabilidade, estabilidade, convergência, justiça e compatibilidade com o TCP Reno. Desta análise, foi possível identificar pontos fortes e fracos de cada protocolo. Nesta seção apresenta-se uma análise de desempenho comparativa dos protocolos investigados, destacando-se tendências e interrelações entre as propriedades.

Adotou-se uma notação para classificar o desempenho dos protocolos para cada propriedade investigada baseado nos resultados obtidos nas simulações para as duas topologias. Essa notação é descrita na Tabela 4.6.

| Muito Bom | Bom | Regular | Ruim |
|-----------|-----|---------|------|
| ★★        | ★   | ◇       | ■    |

Tabela 4.6: Notação utilizada para a classificar o desempenho dos protocolos em cada propriedade.

O critério de classificação adotado considera o desempenho do protocolo em relação aos demais.

Uma avaliação Muito Bom (★★) indica que o protocolo obteve destaque positivo na propriedade em ambos os critérios. Bom (★) indica que o protocolo alcançou bom desempenho na propriedade, entretanto, não está entre os protocolos de maior destaque. Regular (◆) indica que o protocolo obteve um resultado satisfatório, porém sem destaque. Finalmente, classifica-se como Ruim (■) os protocolos que apresentaram desempenho inferior aos demais nas propriedades. Vale ressaltar que, sendo esta uma classificação comparativa, a atribuição de Ruim a alguma propriedade não declara que o protocolo é ruim, mas que não obteve bom desempenho comparado aos demais protocolos investigados.

Inicialmente foram compilados os resultados obtidos nas seções 4.1 e 4.2. Cada protocolo foi classificado de acordo com a notação previamente definida. As Tabelas 4.7 e 4.8 resumem o histórico de desempenho dos protocolos para as duas topologias utilizadas.

| Protocolo | Escalabilidade | Compatibilidade | Justiça | Convergência            | Estabilidade |
|-----------|----------------|-----------------|---------|-------------------------|--------------|
| BIC       | ★              | ◆               | ■       | ■                       | ◆            |
| CUBIC     | ◆              | ★               | ◆       | ■                       | ★★           |
| FAST      | ★★             | ■               | ★       | ★★                      | ★★           |
| HSTCP     | ★              | ◆               | ◆       | ★★                      | ★            |
| HTCP      | ★★             | ■               | ★★      | ◆                       | ★            |
| LIBRA     | ■              | ★★              | ★★      | indefinido <sup>1</sup> | ★            |
| STCP      | ◆              | ★               | ◆       | ◆                       | ◆            |

Tabela 4.7: Classificação dos protocolos na topologia *Dumbbell* de acordo com o desempenho em cada propriedade.

| Protocolo | Escalabilidade | Compatibilidade | Justiça |
|-----------|----------------|-----------------|---------|
| BIC       | ★              | ◆               | ◆       |
| CUBIC     | ◆              | ★               | ◆       |
| FAST      | ★★             | ■               | ★★      |
| HSTCP     | ★              | ◆               | ◆       |
| HTCP      | ★★             | ■               | ★★      |
| LIBRA     | ■              | ★★              | ★       |
| STCP      | ★              | ★               | ■       |

Tabela 4.8: Classificação dos protocolos na topologia *Parking-lot* de acordo com o desempenho em cada propriedade.

Analisando as Tabelas 4.7 e 4.8, constatam-se semelhanças nos resultados obtidos. Isto deve-se a modelagem utilizada na construção da topologia *Dumbbell* que considerou aspectos tais como a composição complexa e controlada do tráfego *background* e a capacidade dos enlaces. A topologia *Dumbbell* utilizada representa uma simplificação da *parking-lot* com apenas um enlace gargalo. Portanto, conclui-se que a utilização de uma topologia *dumbbell* com hipóteses realistas é suficiente para evidenciar muitas características interessantes dos protocolos.

Na Tabela 4.7, verifica-se que as colunas referentes a escalabilidade e compatibilidade são complementares, indicando que os protocolos mais escaláveis são menos compatíveis com TCP Reno. Este comportamento era esperado, uma vez que protocolos com funções de crescimento mais agressivas tendem a ser mais injustos com TCP Reno.

Observa-se, também, que existe uma tendência dos protocolos mais justos serem mais escaláveis. A única exceção é o protocolo LIBRA, que, embora seja justo não alcança escalabilidade por ser pouco agressivo no incremento da janela. Ressalta-se que a classificação “ruim” atribuída ao BIC é devido ao fato deste ter apresentado o menor índice de justiça em relação aos demais protocolos.

Nota-se que a convergência é influenciada pela justiça, o que é intuitivo, pois a justiça depende dos fluxos alcançarem o compartilhamento da banda. Além disso, protocolos com menor tempo de convergência tendem a apresentar melhor estabilidade. No caso dos protocolos BIC e CUBIC, embora suas funções côncavo-convexas favoreçam a estabilidade [50], contribuem para aumentar o tempo de convergência, pois tendem a ter baixa variação no tamanho da janela quando estão próximas ao ponto de inflexão.

Dado que os protocolos foram projetados para operar em ambientes com grande disponibilidade de recursos, deve-se dedicar uma atenção especial a propriedade de escalabilidade. Nesta propriedade, os protocolos que obtiveram melhores resultados foram FAST e HTCP. Embora sejam essencialmente diferentes, o fator que possuem em comum e que os diferencia dos demais é a utilização de informações dinâmicas do ambiente no ajuste da janela de congestionamento, como o RTT e o tempo decorrido desde a última perda, no caso do HTCP.

Quanto à estabilidade, destaca-se o desempenho dos protocolos CUBIC e BIC. Embora BIC tenha sido classificado como regular, devido a instabilidade apresentada em enlaces altamente utilizados, em cenários com menor intensidade de tráfego *background* apresenta comportamento semelhante ao CUBIC. Segundo [50], protocolos que exibem funções com formato côncavo-convexa tendem a obter menores variações nas taxas, favorecendo a estabilidade.

Outro protocolo de destaque é FAST, que tem a estabilidade como objetivo maior. Para tanto, procura equilibrar o fluxo de pacotes de forma a controlar o tamanho das filas e, conseqüentemente, reduzir oscilações no tamanho da janela. É notável o bom

desempenho do protocolo FAST, sendo observado em diversos trabalhos [47, 8].

Constata-se, também, que os resultados obtidos no presente trabalho são muito semelhantes aos obtidos em [26], especialmente quanto as propriedades de escalabilidade, justiça e convergência. Algumas divergências de resultados são decorrentes das diferenças de cenários utilizados, como a composição do tráfego *background*, o número de fontes e o tamanho dos *buffers*. Além disso, o trabalho realizado em [26] não avalia protocolos importantes como FAST e CUBIC. Resultados semelhantes também foram encontrados em [48].

Por outro lado, observam-se discrepâncias entre os resultados de simulação apresentados neste trabalho e os obtidos experimentalmente em [35]. Embora exista uma concordância quanto aos resultados de escalabilidade e convergência, existe divergência nos resultados de justiça. Embora resultados experimentais sejam confiáveis, ressaltam-se importantes diferenças entre os cenários utilizados: a largura de banda utilizada nos seus experimentos é significativamente menor à capacidade dos enlaces utilizados nas simulações deste trabalho e presença ou composição do tráfego *background* não é mencionada.

De acordo com [35], o tempo de convergência dos protocolos tende a ser menor na presença de tráfego *background* do que na ausência. Os resultados obtidos sugerem que a intensidade do tráfego também contribui para o tempo de convergência. Este comportamento não surpreende, pois o tempo de convergência é medido desde o instante em que o novo fluxo entra na rede, até alcançar sua porção justa do enlace, e quanto menor a banda disponível, menor o tamanho da janela a ser atingida.

Após a análise do desempenho dos protocolos nas propriedades investigadas, sugere-se uma classificação quanto a sua efetividade. Destaca-se o protocolo FAST como o mais efetivo para operar em ambientes de alta velocidade, acompanhado pelos protocolos HTCP e HSTCP. Na seqüência, indicam-se os protocolos CUBIC e STCP; e finalmente BIC e LIBRA. Embora o protocolo LIBRA tenha obtido um desempenho mediano, foi severamente penalizado devido ao seu baixo desempenho na propriedade de escalabilidade, considerada um dos principais objetivos para um protocolo de alta velocidade.

# Capítulo 5

## Considerações Finais e Conclusões

O protocolo TCP Reno foi concebido em uma época na qual a capacidade de transmissão dos meios físicos era limitada. Desde então, houve avanços significativos nas tecnologias de transmissão, permitindo altas velocidades e longo alcance. Por outro lado, o protocolo TCP Reno não evoluiu na mesma proporção, não sendo eficiente na utilização de recursos nestes ambientes.

Para solucionar as dificuldades de operação do TCP Reno em redes com produto banda-atraso elevado, diversos protocolos variantes do TCP foram propostos. Dentre eles, os mais promissores e que têm despertado substancial interesse [35] são BIC, CUBIC, FAST, HSTCP, HTCP, LIBRA e STCP.

Nesta dissertação foram investigadas diversas propriedades essenciais de protocolos para redes de alta velocidade, utilizando cenários de simulação que obedecem as recomendações [40, 54, 12] para análise de protocolos. As propriedades investigadas foram escalabilidade, justiça, estabilidade, convergência e compatibilidade com TCP Reno, utilizando as topologias *dumbbell* e *parking lot*.

Verificou-se que poucas conexões são suficientes para garantir alta utilização em enlaces de *1Gbps*. Para obter pequenos ganhos na utilização com taxas superiores de transmissão é necessário um incremento exponencial no número de conexões. Os protocolos que se destacaram neste aspecto de escalabilidade, em ambas as topologias, foram FAST e HTCP; os protocolos LIBRA e CUBIC obtiveram menor desempenho. Verificou-se, também, que a utilização de enlaces de grande capacidade depende de um grande número de conexões. Os protocolos mais escaláveis neste aspecto foram HTCP e FAST. O baixo desempenho em escalabilidade do protocolo LIBRA não é, necessariamente, uma deficiência, uma vez que este protocolo conseguiria alcançar melhor utilização da banda caso um número superior de conexões fossem utilizadas. Além disso, o principal objetivo deste protocolo é alcançar justiça.

Na investigação da propriedade de compatibilidade verificou-se que o protocolo TCP

Reno, executando sem interferência de fluxos de alta velocidade, não é capaz de ocupar toda a capacidade disponível. Na topologia *dumbbell*, a utilização alcançada pelas conexões TCP Reno foi inferior à obtida na topologia *parking lot*. Esta tendência pode ser explicada pela diferença de RTT entre as topologias: na *dumbbell*, o RTT é de 200ms enquanto que o maior RTT da topologia *parking lot* é de 180ms. Quanto menor o RTT, maior a vazão. Além disso, o protocolo LIBRA apresentou excelente compatibilidade com TCP Reno, seguido pelos protocolos CUBIC e STCP. Por outro lado, os protocolos menos compatíveis com TCP foram FAST e HTCP que são os protocolos mais escaláveis; indicando que quanto maior a escalabilidade de um protocolo menor é sua compatibilidade com TCP Reno.

A análise da justiça revelou que, em geral, os protocolos alcançam bons índices de justiça intra-protocolos. Na topologia *parking lot*, este índice tornou-se um pouco menor para alguns protocolos, devido à presença de RTTs diferentes entre as conexões, o que permite as com menor RTT alcançarem maiores porções da banda, prejudicando a justiça global entre os fluxos. Em ambas as topologias, os protocolos que obtiveram melhores índices de justiça intra-protocolos foram LIBRA, FAST e HTCP, independentemente da intensidade do tráfego *background*. Os protocolos que apresentaram menores índices foram: BIC, no cenário com topologia *dumbbell* e intenso tráfego *background*; HSTCP, na topologia *parking lot* com 20% de tráfego *background* e STCP, no mesmo cenário, com 50% e 80% de tráfego *background*. Observou-se que a propriedade de justiça intra-protocolos relaciona-se com a escalabilidade em função do número de conexões, pois os protocolos que alcançaram melhor utilização do enlace naquele aspecto, obtiveram melhores índices de justiça intra-protocolos.

A intensidade do tráfego *background* influencia a justiça entre fluxos com diferentes RTTs e entre fluxos que iniciam em diferentes instantes, provocando alterações nos índices de justiça da maioria dos protocolos. LIBRA e HTCP apresentaram os melhores índices de justiça nestes aspectos, enquanto BIC e STCP atingiram os menores.

Os indicadores de justiça inter-protocolos mostram a tendência de que a justiça é inversamente proporcional à intensidade do tráfego *background*. Os índices obtidos em ambas as topologias divergiram apenas no cenário com 20% de tráfego *background*, sendo menor o índice da topologia *parking lot*. Este comportamento era esperado, uma vez que a presença de grupos de conexões com diferentes RTTs causa desequilíbrio na vazão obtida, principalmente quando existe grande disponibilidade de banda, como é o caso do cenário com 20% de tráfego *background*. Além disso, protocolos agressivos que não consideram estimativas de RTT para o ajuste da janela agravam esta disparidade, provocando maior injustiça entre os protocolos nesta topologia.

A análise da convergência revelou que BIC e CUBIC são os protocolos que levam maior tempo para que um novo fluxo que entra na rede alcance sua porção justa do enlace.

O protocolo STCP apresentou um comportamento de convergência complexo, devido à utilização do paradigma MIMD, consumindo um longo tempo em enlaces com 20% de tráfego *background* e convergindo rapidamente em enlaces sobrecarregados. Por outro lado, os protocolos FAST e HSTCP apresentaram os menores tempos de convergência independente da intensidade de tráfego *background*.

Protocolos estáveis têm uma menor probabilidade de causar colapsos de congestionamento. Constatou-se que a estabilidade é influenciada pelo tráfego *background* e, em geral, à medida que o tráfego é intensificado, a estabilidade dos protocolos diminui. Contudo, FAST e CUBIC sustentaram seus níveis de estabilidade, enquanto que BIC e STCP mostraram-se instáveis em enlaces altamente utilizados.

A partir das análises realizadas, percebe-se que os protocolos FAST, HTCP e LIBRA apresentaram melhor desempenho em um número maior de propriedades que os demais. Por outro lado, os protocolos que obtiveram maior número de qualificações negativas foram BIC e STCP.

A partir do desempenho geral dos protocolos, destacam-se as seguintes tendências:

- A justiça favorece a escalabilidade;
- Escalabilidade e compatibilidade com TCP Reno são complementares;
- Mecanismos que apresentam funções côncavo-convexas possuem melhor estabilidade.

Observando os protocolos mais efetivos em termos das características avaliadas (FAST e HTCP), percebe-se que estes guardam um “histórico” de eventos, como alteração de RTT e distância entre períodos de congestionamento, e usam estas informações do ambiente para ajustar a taxa de transmissão; o protocolo CUBIC também faz uso inteligente das informações que captura da rede no seu algoritmo de controle de congestionamento. Isto sugere fortemente que os protocolos mais promissores são os que conseguem extrair informações dinâmicas da rede para controlar o envio de pacotes.

O protocolo de alta velocidade padrão do *kernel* Linux atualmente é o CUBIC [21], que recentemente foi alterado para promover melhorias na convergência e na escalabilidade [4], o que é muito interessante, uma vez que indicadores destas deficiências foram observados nos resultados das simulações.

Salienta-se que o desempenho do protocolo LIBRA foi insuficiente apenas na propriedade de escalabilidade. Embora esta propriedade seja essencial, não o torna indigno de consideração. Seu projeto modular permite identificar as variáveis que influenciam em cada propriedade, e é possível que a escalabilidade possa ser melhorada através de um ajuste nas constantes  $k_1$  e  $k_2$ .

Por fim, conclui-se que nenhum dos protocolos analisados atende completamente aos requisitos básicos inerentes às redes de alta velocidade. Para que as futuras gerações de protocolos possam atendê-los melhor, recomenda-se que os protocolos sejam desenvolvidos tendo em mente fatores como considerar o *feedback* dado pelo ambiente, que indica variações na dinâmica da rede; possuir mecanismos de atualização da janela que não sejam demasiadamente agressivos, mas que permitam ao protocolo ser escalável; ser pró-ativo, antecipando possíveis tendências da rede; e garantir um melhor aproveitamento dos recursos disponíveis.

Adicionalmente, observa-se que a semelhança nos resultados obtidos nas duas topologias sugere que o uso de uma topologia mais simples, como a topologia *dumbbell*, com hipóteses realistas é suficiente para evidenciar características importantes dos protocolos.

Como trabalhos futuros sugere-se uma investigação minuciosa dos algoritmos de controle de congestionamento dos protocolos mais promissores, a fim de identificar oportunidades de melhoria e possivelmente o desenvolvimento de um novo protocolo que atenda as necessidades das redes de alta velocidade na sua totalidade. Além disso, sugere-se a investigação de desempenho destes protocolos sob diferentes políticas de fila como *Random Early Detection* (RED) [42], recomendada pelo IETF, em ambientes reais.

# Referências Bibliográficas

- [1] Buffers, queues and thresholds on catalyst 6500 ethernet modules. Disponível em: <http://www.cisco.com/application/pdf/en/us/guest/products/ps708/c1244/ccmigration09186a0080131086.pdf>.
- [2] Network simulator. <http://www.isi.edu/nsnam/ns>.
- [3] Internet protocol. Defense Advanced Research Projects Agency, Setembro 1981.
- [4] BIC and CUBIC. <http://netsrv.csc.ncsu.edu/twiki/bin/view/Main/BIC>, Março 2008.
- [5] Steven H. Low; Lachlan L. H. Andrew e Bartek P. Wydrowski. Understanding xcp: Equilibrium and fairness. *IEEE Infocom*, 2:1025–1036, Março 2005.
- [6] Shan Chen e Brahim Bensaou. Can high-speed networks survive with drop tail queues management? *Computer Networks*, 51(7):1763–1776, Maio 2007.
- [7] Dina Katabi; Mark Handley e Charlie Rohrs. Congestion control for high bandwidth-delay product networks. *ACM SIGCOMM*, 32(4):89–102, Agosto 2002.
- [8] Yi-Cheng Chan; Chia-Liang Lin e Cheng-Yuan HO. Quick vegas: Improving performance of TCP vegas for high bandwidth-delay product networks. *IECE Transaction Communication*, E91-B(4), 2008.
- [9] Sofiane Hassayoun e David Ros. Loss synchronization and router buffer sizing with high-speed versions of TCP. *IEEE INFOCOM High-Speed Networks Workshop*, Abril 2008.
- [10] Evandro Souza e Deb Agarwal. A highspeed TCP study: Characteristics and deployment issues. Technical Report LBNL-53215, Lawrence Berkeley National Lab, 2003.
- [11] Aaron Falk e Dina Katabi. Specification for the explicit control protocol (XCP). Internet-Draft, Outubro 2004.

- [12] Sally Floyd e E. Kohler. Tools for the evaluation of simulation and testbed scenarios. Internet-Draft: draft-irtf-tmrg-tools-03, Dezembro 2006.
- [13] Sally Floyd e Eddie Kohler. Internet research needs better models. In *ACM SIGCOMM Computer Communication Review*, volume 33, pages 29–34, New York, NY, USA, Janeiro 2003. ACM Press.
- [14] Andrea Baiocchi; Angelo Castellani e Francesco Vacirca. YeAH-TCP: Yet another highspeed TCP. *Proc. of PFLDNET*, Fevereiro 2007.
- [15] Saverio Mascolo e Francesco Vacirca. The effect of reverse traffic on the performance of new TCP congestion control algorithms. In *Proc. of PFLDNET*, Fevereiro 2006.
- [16] D. J. Leith; R. N. Shorten e G. McCullagh. Experimental evaluation of Cubic-TCP. Whitepaper, Hamilton Institute, 2007.
- [17] R. Jain; D.M. Chiu e H. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared systems. Technical Report DEC-TR-301, Digital Equipment Corporation, 1984.
- [18] Eui-Nam Hu e Hyunseng Choo. Performance enhancement of TCP in high-speed networks. *Information Sciences: an International Journal*, 178(2):352–362, 2008.
- [19] Dhiman Barman; Georgios Smaragdakis e Ibrahim Matta. The effect of router buffer size on highspeed TCP performance. In *Proc of IEEE Globecom'04*, pages 1807–1810, 2004.
- [20] Lisong Xu; Khaled Harfoush e Injong Rhee. Binary increase congestion control (BIC) for fast long-distance networks. In *IEEE Infocom*, volume 4, pages 2514–2524, Março 2004.
- [21] Sangtae Ha e Injong Rhee. Hybrid slow start for high-bandwidth and long-distance networks. 2008.
- [22] K. Cardoso e J. de Rezende. HTTP traffic modeling: Development and application. *IEEE International Telecommunications Symposium– ITS2002*, 2002.
- [23] Michele C. Weigle; Pankaj Sharma e Jesse R. Freeman IV. Performance of competing high-speed TCP flows. In *Networking*, pages 476–487, Coimbra, Portugal, Maio 2006.
- [24] Kazumi Kaneko; Tomoki Fujikawa; Zhou Su e Jiro Katto. TCP-fusion: A hybrid congestion control algorithm for high speed networks. *Proc. of PFLDNET*, Fevereiro 2007.

- [25] James F. Kurose e Keith W. Ross. *Redes de Computadores e a Internet: uma nova abordagem*. Addison Wesley, São Paulo, 2003.
- [26] Masayoshi Nabeshima e Kouji Yata. Performance evaluation and comparison of transport protocols for fast long-distance networks. *IECE Transaction Communication*, 89(4), Abril 2006.
- [27] Lawrence S. Brakmo; Sean W. O'Malley e Larry L. Peterson. TCP vegas: New techniques for congestion detection and avoidance. In *SIGCOMM*, pages 24–35, 1994.
- [28] Injong Rhee e Lisong Xu. CUBIC: A new TCP-friendly high-speed TCP variant. In *Proc. of PFLDNET*, Lyon, France, Fevereiro 2005.
- [29] Sangtae Ha; Long Le; Injong Rhee e Lisong Xu. Impact of background traffic on performance of high-speed TCP variant protocols. *Computer Networks*, 51(7):1748–1762, 2007.
- [30] Kun Tan; Jingmin Song; Qian Zhang e Murari Sridharan. A compound TCP approach for high-speed and long distance networks. Technical Report MSR-TR-2005-86, Microsoft Research, 2005.
- [31] Kun Tan; Jingmin Song; Qian Zhang e Murari Sridharan. Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks. In *PDFDnet2006*, 2006.
- [32] D. Leith e R. Shorten. H-TCP: TCP for high-speed and long-distance networks. In *Proc. of PFLDNET*, Argonne, Fevereiro 2004.
- [33] Hadrien Bulot; R. Les Cottrell e Richard Hughes-Jones. Evaluation of advanced TCP stacks on fast long-distance production networks. *J. Grid Comput.*, 1(4):345–359, 2003.
- [34] Douglas Leith e Robert N. Shorten. Next generation TCP: Open questions. *Proc. of PFLDNET*, 2008.
- [35] Yee-Ting Li; Douglas Leith e Robert N. Shorten. Experimental evaluation of TCP protocols for high-speed networks. *IEEE / ACM Transactions on Networking*, 15(5):1109–1122, Outubro 2008.
- [36] Carlo Caini e Rosario Firrincieli. TCP hybla: a TCP enhancement for heterogeneous networks. *International Journal of Satellite Communications and Networking*, 22:547–566, 2004.

- [37] Ryan King; Richard Baraniuk e Rudolf Riedi. TCP-Africa: An adaptive and fair rapid increase rule for scalable TCP. In *IEEE Infocom*, volume 3, pages 1838–1848, Miami, Florida, Março 2005.
- [38] Sally Floyd; Sylvia Ratnasamy e Scott Shenker. Modifying TCP’s congestion control for high speeds. Internet-Draft, Maio 2002.
- [39] Cheng Jin; David X. Wei e Steven H. Low. FAST TCP: Motivation, architecture, algorithms, performance. In *IEEE Infocom*, Março 2004.
- [40] David X. Wei; Pei Cao e Steven H. Low. Time for a TCP benchmark suite? Disponível em: <http://www.cs.caltech.edu/weixl/research/technical/benchmark/summary.ps>, Agosto 2005.
- [41] Sally Floyd e Van Jacobson. On traffic phase effects in packet-switched gateways. In *ACM SIGCOMM Computer Communication Review*, volume 21, pages 115–156, New York, NY, USA, Abril 1991. ACM Press.
- [42] Sally Floyd e Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transaction Networking*, 1(4):397–413, 1993.
- [43] M. Allman; V. Paxson e W. Stevens. TCP congestion control. Network Working Group, Abril 1999.
- [44] Matthew Roughan; Steve Uhlig e Walter Willinger. Internet scalability: Properties and evolution. *IEEE Network*, Março 2008.
- [45] Douglas Leith; R.N. Shorten e Y. Lee. H-TCP: A framework for congestion control in high-speed and long-distance networks. Technical report, Hamilton Institute, 2005.
- [46] Aaron Falk; Ted Faber et al. Transport protocols for high performance. *Communications of the ACM*, 46(11):42–49, Novembro 2003.
- [47] Cheng Jin; David X. Wei; Steven H. Low et al. FAST TCP: From theory to experiments. *IEEE Network*, 19(1):4–11, Fevereiro 2004.
- [48] G. Marfia; C. Palazzi; G. Pau et al. TCP-Libra: Exploring RTT fairness for TCP. Technical Report TR-050037, UCLA-CSD, 2005.
- [49] Gustavo Marfia; Claudio E. Palazzi et al. TCP Libra: balancing flows over heterogeneous propagations scenarios. In *SigMetrics*, 2006.

- [50] H. Cai; D. Eun; I. Rhee et al. Stochastic ordering for Internet congestion control and its applications. *INFOCOM 2007*, pages 910 – 918, 2007.
- [51] L. Andrew; I. Rhee et al. Towards a common TCP evaluation suite. Proc. of PFLD-NET, Dezembro 2008.
- [52] Sangtae Ha; Yusung Kim; Long Le et al. A step toward realistic performance evaluation of high-speed TCP variants. In *PDFDnet2006*, Nara, Japan, Fevereiro 2006.
- [53] S. Floyd. Limited slow-start for TCP with large congestion windows. Network Working Group, Março 2004.
- [54] Sally Floyd. Metrics for the evaluation of congestion control mechanisms. Internet-Draft: draft-irtf-tmrg-metrics-07, Fevereiro 2007.
- [55] Geoff Huston. Gigabit TCP. *The Internet Protocol Journal*, 9(2), Junho 2006.
- [56] Van Jacobson. Congestion avoidance and control. *ACM Computer Communication Review*, 18(4):314–329, 1988.
- [57] Tom Kelly. Scalable TCP: Improving performance in highspeed wide area networks. Technical Report 2, CERN-IT Division, Abril 2003.
- [58] Neila Fernanda Michel. Simulação de fluxo de dados em redes de escala livre. Technical report, UNIOESTE, Cascavel, Brasil, Dezembro 2004.
- [59] L. A. Rodrigues. Uma ferramenta para comparação dos mecanismos de controle de congestionamento em redes TCP/IP utilizando o ns (network simulator). Master's thesis, UNIOESTE - Universidade Estadual do Oeste do Paraná, Cascavel, Brazil, 2003.
- [60] Tomoya Hatano; Masahiko Fukuhara; Hiroshi Shigeno and Ken-Ichi Okada. TCP-friendly SQRT TCP for high speed networks. *Proc. of APSITT*, pages 455–460, Novembro 2003.