

# Uma investigação sobre a escalabilidade de variantes do protocolo TCP para redes de alta velocidade

Neila Fernanda Michel<sup>1</sup>, Nelson L. S. Fonseca<sup>1</sup>

<sup>1</sup>Instituto de Computação – Universidade Estadual de Campinas  
Caixa Postal 6176 – 13084-971 – Campinas – SP – Brasil

fernanda.michel@students.ic.unicamp.br, nfonseca@ic.unicamp.br

**Abstract.** *The use of TCP Reno in networks with high bandwidth-delay product has several drawbacks, specially due to the low growth of the congestion window during congestion avoidance periods which causes underutilization of network resources. Several proposals for overcoming the challenge of using the available bandwidth of high capacity links have been made. This paper investigates the scalability of seven of these proposals: BIC, CUBIC, FAST, HSTCP, HTCP, Libra and STCP. Simulation was conducted according to recommended procedures for evaluating this type of protocols.*

**Resumo.** *O protocolo TCP Reno apresenta sérias limitações em redes com produto banda-atraso elevado como lento crescimento da janela de congestionamento na fase de Congestion Avoidance que causa subutilização de banda por longos períodos. Diversas variantes do protocolo TCP Reno têm sido propostas com o objetivo de superar seu desempenho na obtenção de banda em enlaces de grande capacidade. Neste artigo investigamos a propriedade de escalabilidade de sete protocolos variantes do TCP para redes de alta velocidade: BIC, CUBIC, FAST, HSTCP, HTCP, Libra e STCP seguindo as recomendações atuais para avaliação de protocolos.*

## 1. Introdução

Em resposta à série de colapsos de congestionamentos ocorridos na Internet durante os anos 80, foi adicionado o mecanismo de controle de congestionamento ao *Transmission Control Protocol* (TCP) permitindo controlar a quantidade de dados injetados na rede por conexão. Este mecanismo é conhecido como Incremento Aditivo Decremento Multiplicativo (AIMD). Entretanto, o padrão AIMD de crescimento da janela do TCP Reno 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* 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 ( $N = W^2/1,5$ ). Isto é aproximadamente um pacote a cada  $S = 6000$  segundos, o que é irrealista [Floyd 2002].

Diversas variantes do TCP para operar em redes com produto banda-atraso elevado foram desenvolvidos nos últimos anos com o objetivo de melhorar o desempenho do TCP em caminhos de alta velocidade e são chamados TCPs para Redes de Alta Velocidade. As variantes do TCP investigadas neste artigo são: BIC [Xu et al. 2004], CUBIC [Rhee and Xu 2005], FAST [Jin et al. 2004a], HTCP [Leith and Shorten 2004], Libra [Marfia et al. 2005], STCP [Kelly 2003] e HSTCP [Floyd et al. 2002].

Um protocolo ideal para operar em redes de alta velocidade deve apresentar diversas características desejáveis como ser compatível com TCP Reno, alcançar estabilidade na taxa de transmissão, apresentar justiça inter e intra protocolos, e principalmente ter escalabilidade. A propriedade de escalabilidade caracteriza a adaptação do sistema a alterações na intensidade de utilização, volume ou demanda, sem impactos significativos no desempenho. Esta característica é de grande importância para protocolos que se propõem a operar em ambientes com grande disponibilidade de recursos, pois precisam controlar grandes janelas de congestionamento. Neste artigo investigamos a propriedade de escalabilidade dos protocolos variantes do TCP para redes de alta velocidade.

A maioria dos estudos de simulação disponíveis na literatura não leva em consideração diversos fatores importantes na construção dos cenários de simulação descritos em recomendações [Floyd 2007] e propostas de *bechmark* [David X. Wei 2005], o que pode influenciar nos resultados a serem obtidos.

A contribuição do presente artigo é a investigação da escalabilidade das variantes do protocolo TCP para redes de alta velocidade realizada através de simulações com cenários que obedecem as atuais recomendações [Floyd 2007] [David X. Wei 2005] para análise de protocolos.

## 2. Propriedades dos protocolos para redes de alta velocidade

Diversos aspectos podem caracterizar e qualificar um algoritmo de controle de congestionamento. No contexto de redes de alta velocidade estamos interessados nos algoritmos que conseguem lidar com grandes janelas de congestionamento e taxas de perdas realistas. Para estes protocolos algumas propriedades são fundamentais como justiça entre fluxos, estabilidade, convergência para um estado de equilíbrio e escalabilidade.

A justiça caracteriza o comportamento a longo prazo. Captura informações sobre a capacidade de coexistência entre fluxos de quaisquer protocolos. Um protocolo justo não impede que fluxos coexistentes adquiram recursos suficientes para execução. A propriedade de convergência avalia quanto rápido o protocolo reage a mudanças na configuração da rede, por exemplo, quando um enlace estiver altamente utilizado e um novo fluxo entrar, este não seja impedido de obter uma porção justa da banda. E finalmente, a estabilidade caracteriza a magnitude em que as variáveis do sistema oscilam. É desejável que os fluxos alcancem um estado de equilíbrio, onde oscilações na taxa de transmissão sejam mínimas. A propriedade de escalabilidade caracteriza a adaptação do sistema a alterações na intensidade de utilização, volume ou demanda. Indica a habilidade de lidar com alterações de trabalho sem impactar no desempenho.

## 3. Protocolos TCP para redes de alta velocidade

Os protocolos variantes do TCP para redes de alta velocidade diferem em relação ao sinal de congestionamento utilizado para definir a dinâmica de ajuste do tamanho da janela. Algumas propostas são baseadas em perdas de pacotes, outras baseadas em atraso, ou ainda híbridas. As principais variantes são descritas nas subseções a seguir.

### 3.1. BIC TCP

Em 2004 Rhee 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 e escalabilidade de banda. Seu algoritmo de controle de congestionamento utiliza duas políticas para atualização do tamanho da janela: incremento aditivo e incremento de busca binária. A escolha entre estas políticas é dependente da utilização do enlace.

A injustiça de RTT entre dois fluxos concorrentes é definida como a vazão em termos do RTT. Isto implica em garantir que fluxos com baixo RTT não provoquem *starvation* de fluxos com longos RTTs. Para assegurar a justiça entre conexões com RTTs diferentes o protocolo BIC adapta a taxa de transmissão de acordo com o tamanho da janela. Na fase de incremento de busca binária (implementada como uma função logarítmica) a taxa de incremento é reduzida à medida em que o tamanho da janela aproxima-se de um valor máximo  $cwnd_{max}$  estimado (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 alto.

Quando o tamanho da janela passa do tamanho máximo previamente estimado, o algoritmo de busca binária é substituído pelo *slow start* para sondar o novo tamanho máximo da janela. Nesta fase os incrementos ocorrem a cada RTT em passos

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

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 de pacotes, 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 ( $S_{max}$ ) ajudam a reduzir o tempo de subutilização de recursos. Tornar o incremento dependente apenas da diferença entre o tamanho das janelas atual e máxima permite aos fluxos convergirem para um compartilhamento justo.

Fluxos com RTT curto podem alcançar o tamanho máximo estimado para a janela em menor espaço de tempo, mas não obtém maiores porções de banda, mesmo em ambientes com perdas sincronizadas. Portanto, a combinação das técnicas de incremento linear agressivo quando a janela é pequena e incremento logarítmico quando a janela é grande, favorece o compartilhamento justo da banda entre fluxos com diferentes RTTs.

A 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 evitando perda de pacotes e consequentemente 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.

### 3.2. CUBIC TCP

Cubic é uma variante do TCP para redes de alta velocidade proposto por Rhee [Rhee and Xu 2005]. Surgiu da busca por uma função de crescimento da janela de conges-

tionamento que preservasse as características de estabilidade e escalabilidade alcançadas pelo protocolo BIC com um mecanismo de controle da janela simplificado.

A janela de congestionamento do protocolo Cubic é determinada pela seguinte equação:

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

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 após uma redução provocada por um evento de perda de pacotes é rápido, e à medida em que a janela se aproxima de  $cwnd_{max}$ , o crescimento torna-se mais lento, chegando a quase zero. 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  $cwnd_{max}$ . Este crescimento lento próximo de  $cwnd_{max}$  contribui para a estabilidade do protocolo e aumenta a utilização do enlace, enquanto que o rápido crescimento da janela quando a diferença entre as janelas atual e máxima é grande, assegura a escalabilidade do protocolo.

A função cúbica contribui para a 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 o fluxo com menor  $cwnd_{max}$  crescer mais lentamente ( $K$  aumenta conforme  $cwnd_{max}$  aumenta). Assim, ambos os fluxos eventualmente convergem para o mesmo tamanho de janela [Rhee and Xu 2005].

A taxa de crescimento da janela é determinada pelo tempo decorrido  $t$ . Isto assegura RTT linear uma vez que quaisquer fluxos concorrentes com diferentes RTTs terão o mesmo  $t$  após perdas de pacotes sincronizadas. Para melhorar a estabilidade e justiça o incremento da janela é limitado para não ser superior a  $S_{max}$  por segundo. Esta característica mantém a janela em crescimento linear quando distante de  $cwnd_{max}$ . O incremento linear do Cubic é realizado fazendo a janela ser dependente do tempo real - um fluxo com RTT pequeno terá menos incrementos em um RTT, mas serão constantes no tempo [Rhee and Xu 2005]. Esta forma de incremento melhora a compatibilidade do protocolo, pois não privilegia fluxos com RTT maior ou menor. 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 massivas de pacotes enquanto mantém alta utilização dos enlaces.

### 3.3. FAST TCP

FAST TCP é uma abreviação para *Fast Active Management Scalable Transmission Control Protocol* proposto por [Jin et al. 2004a]. Seu algoritmo de controle de congestionamento ajusta o tamanho da janela baseado em estimativas de atraso e eventos de perda de pacotes. É considerado a versão *high-speed* do TCP Vegas.

O mecanismo de controle de congestionamento do TCP FAST é 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, pacotes perdidos (reconhecimento negativo) e pacotes 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 massivas perdas de pacotes [Jin et al. 2004b].

FAST TCP usa o atraso na fila como o principal fator no algoritmo de ajuste da janela de congestionamento. Sob condições normais a atualização periódica baseada no RTT médio e no atraso médio nas filas são calculados no componente de estimativa de acordo com a equação:

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

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, 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 na tentativa de estabilizar em torno do ponto de equilíbrio, alcançado quando um certo número de pacotes permanecem armazenados na rede para mantê-la totalmente utilizada. Isto elimina oscilações, mas tem duas dificuldades: o controle baseado em equações requer uma explícita estimativa da probabilidade de perda fim-a-fim, o que é difícil quando a probabilidade de perda é muito baixa; mesmo se a probabilidade de perda seja estimada a dinâmica de fluxo do TCP Reno leva a um sistema de *feedback* instável quando o atraso e a capacidade da rede aumentam, o que pode levar a oscilações. Permanece estável independente da quantidade de fluxos. Converge exponencialmente rápido para o ponto de equilíbrio, podendo divergir após um tempo, dependendo das condições e da topologia da rede.

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.

FAST alcança justiça proporcional, utilizando somente a banda excedente não sendo 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.

### 3.4. HighSpeed TCP

O protocolo HighSpeed TCP (HSTCP) [Floyd et al. 2002] [Souza and Agarwal 2003] 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.

HSTCP usa o tamanho da janela como indicação do produto banda-atraso de um caminho. Em ambientes com taxa de perda de pacotes maior que  $10^{-3}$  opera de forma idêntica ao TCP Reno, não aumentando o risco de colapso de congestionamento. E quando os eventos de congestionamento tornam-se raros (menores que  $10^{-3}$ ), adota uma função de crescimento da janela mais agressiva para ser escalável com a banda disponível.

Neste protocolo foi introduzida uma nova relação entre a janela de congestionamento média ( $cwnd$ ) e a taxa de perda de pacotes ( $p$ ) no estado estacionário. A 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  $cwnd < W_{low}$  usa a função TCP padrão, caso contrário adota a função HSTCP.  $W_{high}$  (janela de congestionamento média) e  $P_{high}$  (taxa de exclusão de pacotes) são utilizados para especificar os limites da função HSTCP. Os valores padrão para  $W_{low}$ ,  $W_{high}$  e  $P_{high}$  são respectivamente 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 seguintes equações:

$cwnd \leftarrow cwnd + \frac{a(cwnd)}{cwnd}$ , onde  $a(cwnd) = \frac{2+cwnd^2*b(cwnd)*p(cwnd)}{2-b(cwnd)}$ . Quando perdas ocorrem a taxa de transmissão deve ser reduzida  $cwnd \leftarrow cwnd - b(cwnd) * cwnd$ , onde

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

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

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 TCP padrão.

O HSTCP possui alguns objetivos mais específicos: sustentar alta velocidade com taxas de perdas realistas, recuperar-se de múltiplos *timeouts* e períodos com pequenas janelas de congestionamento sem incorrer em atrasos excessivos, e manter eqüidade de banda entre conexões.

O primeiro objetivo é alcançado com o ajuste dos parâmetros de incremento e decremento realizados em função do tamanho da janela atual. Isto permite alcançar janelas de dezenas de milhares de segmentos com taxas de perdas de pacotes realistas. Taxas de transmissão mais agressivas permitem que fluxos de melhor esforço iniciem com janelas maiores permitindo que grandes janelas sejam alcançadas ainda na fase *Slow Start*.

HSTCP não responde prontamente à 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 somente por alguns RTTs, abreviando o período de subutilização de recursos.

Pelo princípio da eqüidade, fluxos TCP deveriam obter a mesma quantidade de banda que fluxos HSTCP, porém considerando-se que quando a taxa de perda de pacotes é muito pequena o TCP não consegue adquirir maiores porções da banda, é aceitável que os fluxos HSTCP façam uso da banda excedente no ambiente. Entretanto, quando a banda está altamente utilizada, os fluxos HSTCP induzem um grande número de perdas que afetam fluxos TCPs forçando-os a reduzir sua janela de congestionamento e, por natureza, levarão muitos ciclos de RTT para alcançar a mesma largura de banda.

O ajuste da taxa de transmissão baseada no tamanho da janela pode provocar *starvation* de fluxos com longo RTT, pois fluxos com curto RTT incrementam suas janelas mais rapidamente obtendo grande parte da banda do canal.

À medida em 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 banda é frequentemente super-estimada, o que dificulta o processo de convergência do protocolo, que pode nunca ocorrer, principalmente quando existem muitos fluxos HSTCP concorrentes.

### 3.5. H-TCP

O protocolo H-TCP [Leith and Shorten 2004] [Leith et al. 2005] possui um algoritmo de controle de congestionamento projetado para executar em redes convencionais, de alta velocidade e longas distâncias. Generaliza o paradigma 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  $\bar{\alpha}(\Delta)$ . Este parâmetro é ajustado para fornecer uma taxa de incremento que considere tanto a duração como o número de eventos de congestionamento.

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 obtém em média a mesma quantidade de banda, considerando a mesma probabilidade de perda.

O número de épocas de congestionamento até a convergência é determinada pelo

fator de decremento  $\beta$ . No TCP Reno,  $\beta = 0.5$  e leva aproximadamente 4 épocas para convergir. Para melhorar o tempo de resposta é preciso escolher uma função  $\bar{\alpha}(\Delta)$  tal que a duração das épocas de congestionamento permaneçam razoavelmente pequenas 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 uma época de congestionamento e com o ajuste da polinomial ajusta-se a duração da época de congestionamento.

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. O uso do barrado sobre  $\bar{\alpha}$  indica que a taxa de crescimento é 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 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 na duração e no número de épocas de congestionamento antes da convergência e na 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 [Leith and Shorten 2004].

Uma vez que os fluxos ajustam seu fator de *backoff* para o 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 onde as filas são pequenas em relação ao produto banda-atraso.

### 3.6. TCP Libra

TCP Libra [Marfia et al. 2005] foi desenvolvido com o objetivo de obter o melhor equilíbrio entre utilização, justiça e compatibilidade com TCP New Reno. O elemento chave do TCP Libra é o algoritmo de ajuste de janela de transmissão que é compensado pela diferença no RTT, o que leva a uma solução que fornece justiça entre fluxos TCP que compartilham o mesmo enlace gargalo independente do RTT e portanto não é sensível ao comportamento "auto-clock". Denomina-se comportamento auto-clock a redução na taxa de envio de segmentos devido a detecção automática de congestionamento incipiente realizada pelos mecanismos de controle de congestionamento baseados em perda de segmentos.

Este protocolo utiliza estimativas de RTT para avaliar a capacidade do enlace disponível e controlar a taxa de incremento da janela. Possui um projeto modular cujos componentes 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*.

Libra pertence à classe dos algoritmos AIMD. Reage a formação de filas diminuindo o incremento das janelas 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 governado 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.

O controle de escalabilidade pode ser identificado por  $k_1 c$ , sendo  $k_1$  um parâmetro fixo que deve levar em conta o requisito de tempo de resposta do algoritmo e a execução na região estável. 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 é que um fluxos com RTT acima de um limiar podem ser interpretados como fluxos que estejam enfrentando problemas no caminho.

A prevenção de perdas sincronizadas e múltiplas reduções nas janelas são resolvidas com a introdução de *pacing* aleatório, no qual os pacotes são transmitidos aleatoriamente dentro de um RTT. Com isso é 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 próxima 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, aumento de  $T_0$  deve ser seguido por um aumento em  $k_1$ .

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 TCP Libra não penaliza fortemente a dinâmica da janela quando o buffer começa a encher, assim quando o RTT começa a aumentar, é o incremento que 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 competindo com TCP New Reno.

### 3.7. Scalable TCP

O *Scalable TCP* (STCP), proposto por Tom Kelly [Kelly 2003], consiste em modificar a forma de atualização da janela de congestionamento dos transmissores TCP para tornar o tempo de recuperação após um evento de congestionamento rápido e independente do tamanho da janela.

Este protocolo adota o paradigma de incremento aditivo e decremento multiplicativo (MIMD), sendo agressivo na obtenção de banda e drástico na redução da janela. O algoritmo obedece às seguintes regras básicas:

- 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  $cwnd \leftarrow cwnd - \lceil b * cwnd \rceil$ ,  $b$  constante  $0 < b < 1$ .

Os valores de  $a$  e  $b$  são ajustados respectivamente para 0.01 e 0.125 [Kelly 2003], escolhidos 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, utilizando rapidamente os recursos subutilizados. Por outro lado, quando a banda está totalmente utilizada, o incremento multiplicativo provoca uma grande quantidade de perdas de pacotes devido ao constante erro ao estimar a banda passante disponível.

Grandes janelas de congestionamento aliadas a elevadas taxas de exclusão de pacotes, tornam o STCP compatível com TCP Reno. Isto ocorre devido às constantes reduções nas janelas de congestionamento e o aumento do atraso fim-a-fim provocado pelas filas nos buffers dos roteadores. Entretanto, quando a taxa de exclusão é pequena,

torna-se injusto por utilizar um incremento extremamente agressivo comparado ao incremento aditivo dos fluxos TCP, alocando grande parte dos recursos e induzindo perdas que forçam os fluxos TCP a reduzirem suas taxas de transmissão.

Da mesma forma que protocolo HSTCP, descrito na seção 3.4, é injusto com fluxos com longo RTT, pois fluxos com RTT curtos incrementam sua janela muito mais rapidamente obtendo 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.

A concepção de protocolos variantes do TCP tem recebido grande atenção enquanto que uma avaliação sistemática destes protocolos têm sido negligenciada. Neste artigo, descrevemos comparações entre os protocolos BIC, CUBIC, FAST, HSTCP, HTCP, STCP e LIBRA em relação às propriedades de escalabilidade, justiça, estabilidade e convergência.

#### 4. Trabalhos relacionados

Alguns estudos comparativos entre protocolos de transmissão em alta velocidade foram realizados, cada qual com sua modelagem de rede e tráfego.

Sally Floyd [Floyd 2007] propôs um *framework* para avaliação de algoritmos de controle de congestionamento. Este *framework* inclui várias métricas como vazão, taxa de perda de pacotes, atraso e justiça além de diferentes ambientes de rede. Wei et al. [David X. Wei 2005] estabeleceu um *benchmark* incluindo vários cenários realísticos para avaliações de desempenho, tais como distribuições *heavy-tailed* e valores de atraso de propagação. Estes trabalhos mostram a necessidade de avaliações realísticas de desempenho para novos algoritmos de controle de congestionamento.

Weigle [Weigle et al. 2006] investigou a propriedade de justiça de RTT entre fluxos de diferentes protocolos de alta velocidade utilizando topologia *dumbbell*, com gargalo de  $622Mbps$ , RTT de  $100ms$ , mas desconsiderou qualquer outro tipo de tráfego coexistente.

Bullet [Bullet et al. 2003] realizou comparações experimentais entre o protocolo TCP Newreno e as variantes para alta velocidade para redes de curtas ( $10ms$ ), médias ( $70ms$ ) e longas ( $170ms$ ) distâncias. Segundo seus estudos, o TCP Newreno apresenta baixo desempenho e instabilidade enquanto que a maioria dos protocolos variantes do TCP alcançam melhorias significativas. Embora seus experimentos tenham sido realizados em redes reais, nenhum controle sobre tráfego *background* foi realizado.

Li et al [Yee-Ting Li and Shorten 2006] realizou experimentos para os protocolos BIC, FAST, HSTCP, HTCP e STCP em uma rede de laboratório. Relatou que todos os protocolos, exceto HTCP, apresentaram substancial injustiça, e enfatizou o bom desempenho do HTCP. Nenhum tipo de tráfego *background* foi empregado nestes experimentos, o que pode levar a resultados tendenciosos.

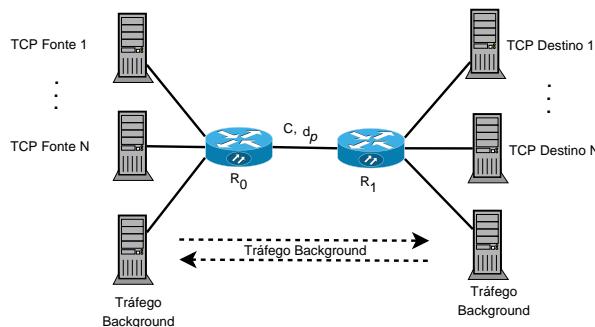
A escalabilidade é uma propriedade de grande importância para protocolos que pretendem operar em redes de alto desempenho, entretanto não existem estudos específicos que determinem a escalabilidade dos protocolos para redes de alta velocidade.

## 5. Descrição dos cenários e experimentos

Embora muitos protocolos de transporte para redes de alta velocidade tenham sido desenvolvidos nos últimos anos, pouca atenção têm sido dada à avaliação do desempenho destes protocolos. 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 adotadas levando a resultados tendenciosos, com conclusões divergentes e até mesmo contraditórias [Ha et al. 2006].

Atualmente existe uma força-tarefa no sentido de estabelecer um conjunto de testes padrão que permita um melhor entendimento das diferenças entre os vários algoritmos de controle de congestionamento, estabelecendo métricas de desempenho para diferentes implementações e configurações de hardware [Floyd 2007] [Sally Floyd 2006]. Baseados nestas recomendações derivamos nosso conjunto de experimentos para analisar a propriedade de escalabilidade dos protocolos de transporte para redes de alta velocidade.

Utilizamos o software *Network Simulator 2* como ferramenta de simulação e adotamos uma topologia *dumbbell*, caracterizada por apresentar um único enlace gargalo, por ser bastante simples e amplamente empregada e aceita no contexto de simulações. A Figura 1 ilustra a topologia utilizada composta por  $N$  fontes de tráfego TCP e  $N$  destinos conectados por um par de roteadores, além de fontes de tráfego *background* em ambos os sentidos de transmissão.



**Figura 1. Topologia *dumbbell* utilizada nas simulações**

Para evitar os efeitos de sincronização, o tempo de início das fontes TCP e o atraso de propagação nos enlaces sofreram pequenos ajustes aleatórios. O tempo total de simulação utilizado em cada experimento foi fixado em  $T = 600$  segundos, tempo suficiente para observar grande parte dos fenômenos que ocorrem na rede. O tamanho do *buffer* de cada enlace foi fixado em 2500 pacotes, estando em conformidade com tamanho de *buffers* reais [David X. Wei 2005].

Para tornar os cenários mais próximos de uma representação real de rede, incluímos tráfego *background*, responsável por ocupar uma porcentagem da capacidade do enlace nos sentidos direto e reverso de transmissão e composto por 20% de tráfego UDP

CBR, 24% FTP e 56% WEB. Além disso, variamos a intensidade deste tráfego entre 20%, 50% e 80% da capacidade do enlace gargalo.

A propriedade de escalabilidade é de vital importância para um protocolo designado a operar em ambientes com grande disponibilidade de recursos, como são as redes de alta velocidade. Segundo Wei, [David X. Wei 2005] estes protocolos devem ser escaláveis com a capacidade, atraso e nível de multiplexação. Investigamos a propriedade de escalabilidade dos protocolos BIC, CUBIC, FAST, HSTCP, HTCP, Libra e HTCP a partir destes três aspectos.

- Escalabilidade em função do número de conexões: para cada protocolo, variamos o número de conexões concorrentes no intervalo  $2^1, 2^2, \dots, 2^8$  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 da capacidade do enlace: para cada protocolo, variamos a capacidade do enlace gargalo entre  $155Mbps$  (OC-03),  $622Mbps$  (OC-12),  $1Gbps$  (Gigabit Ethernet),  $5Gbps$  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.
- Escalabilidade em função do atraso de propagação: para cada protocolo, variamos 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.

## 6. Análise dos resultados

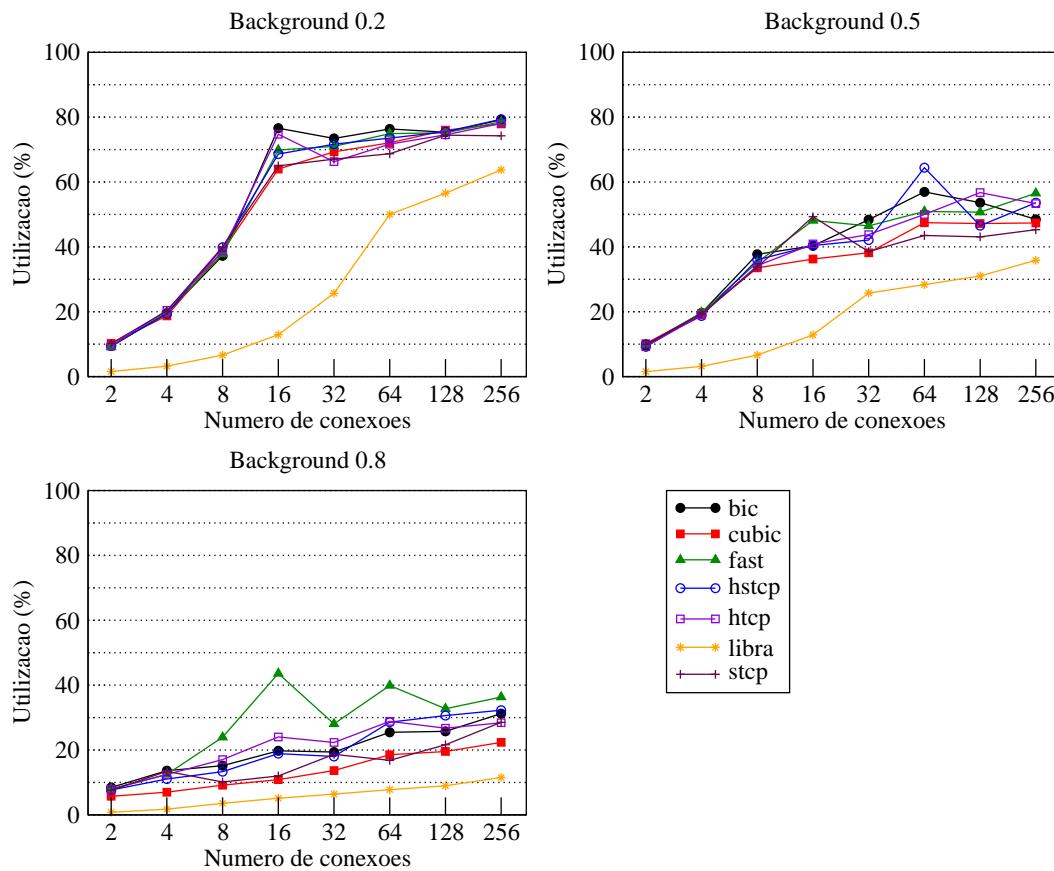
A Figura 2 é composta por três gráficos que apresentam os resultados obtidos para a escalabilidade em função do número de conexões para as diferentes intensidades de tráfego *background*.

Nestes experimentos utilizamos um enlace gargalo de  $1Gbps$  com atraso de propagação de  $100ms$ . Podemos observar que em um cenário com 20% de tráfego *background* são 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.

Observamos também que mesmo com 50% enlace ocupado por tráfego *background* os protocolos FAST, BIC, HTCP e HSTCP alcançam utilização acima de  $500Mbps$  quando o número de conexões é superior a 64. O mesmo comportamento pode ser observado quando 80% do enlace é ocupado pelo tráfego *background*, porém neste caso todos os protocolos, exceto Libra, alcançam utilização acima de  $200Mbps$ .

Neste conjunto de experimentos o protocolo Libra mostrou-se pouco escalável em relação ao número de conexões. Por outro lado, os protocolos FAST e HTCP apresentaram melhores resultados em relação aos demais protocolos.

A escalabilidade com a capacidade do enlace é apresentada na Figura 3. Nestes experimentos fixamos em 20 o número de conexões para cada protocolo e o atraso de

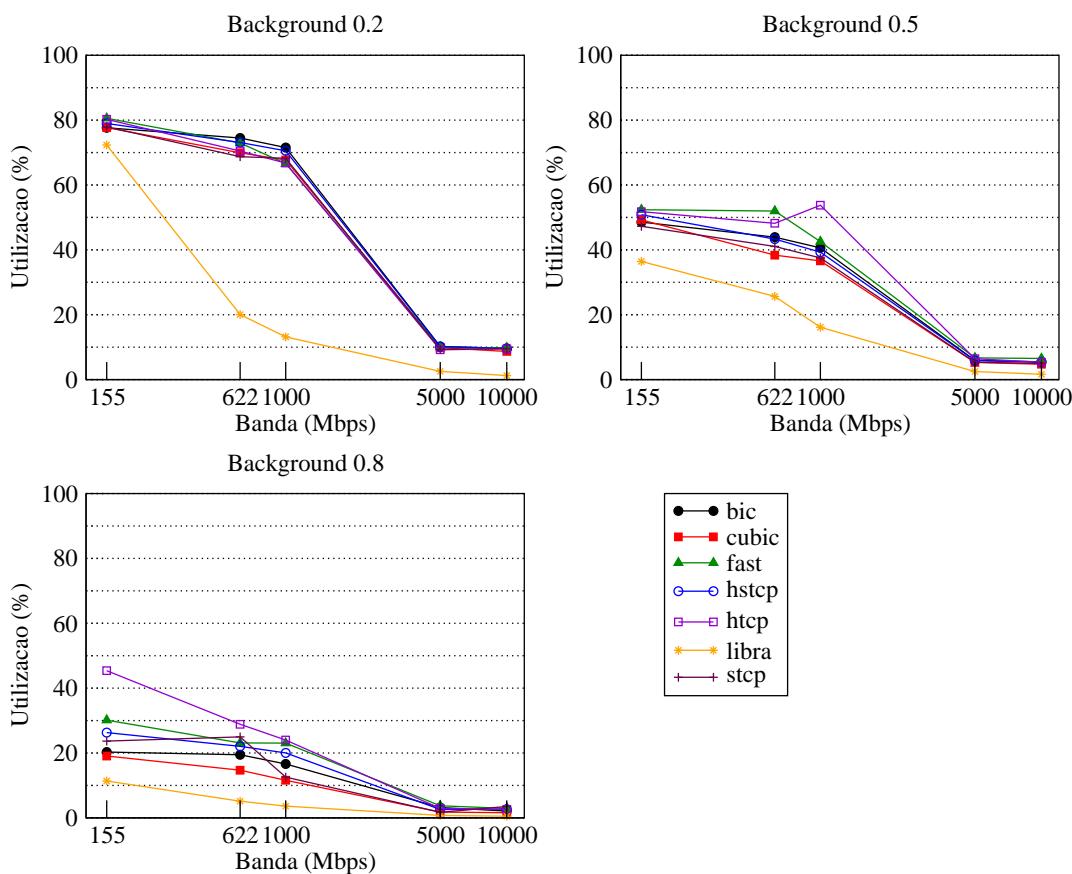


**Figura 2. Utilização em função do número de conexões para diferentes intensidades de tráfego *background*. Capacidade do enlace gargalo: 1Gbps. Atraso de propagação no enlace gargalo: 100ms.**

propagação no enlace gargalo em 100ms. Podemos observar que 20 conexões não são capazes de obter alta utilização em enlaces de capacidade superior a 1Gbps, independentemente da intensidade do tráfego *background*. A queda na utilização de todos os protocolos à medida em que a capacidade do enlace aumenta sugere que nenhum protocolo é capaz de lidar eficientemente com enlaces de grande capacidade.

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

A escalabilidade em função do atraso de propagação é apresentada na Figura 4. Nestas simulações fixamos o número de conexões para cada protocolo em 20 e a capacidade do enlace gargalo em 1Gbps. Verificamos que à medida em que o atraso de propagação aumenta, a utilização do enlace diminui independente da intensidade do tráfego *background*. Entretanto a presença do tráfego *background* influencia na variação da utilização do enlace. A Tabela 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 pode ser interpretado da seguinte forma: com 20% da capacidade do enlace ocupada por tráfego *back-*



**Figura 3. Utilização em função da capacidade do enlace para diferentes intensidades de tráfego *background*. Número de conexões: 20. Atraso de propagação no enlace gargalo: 100ms.**

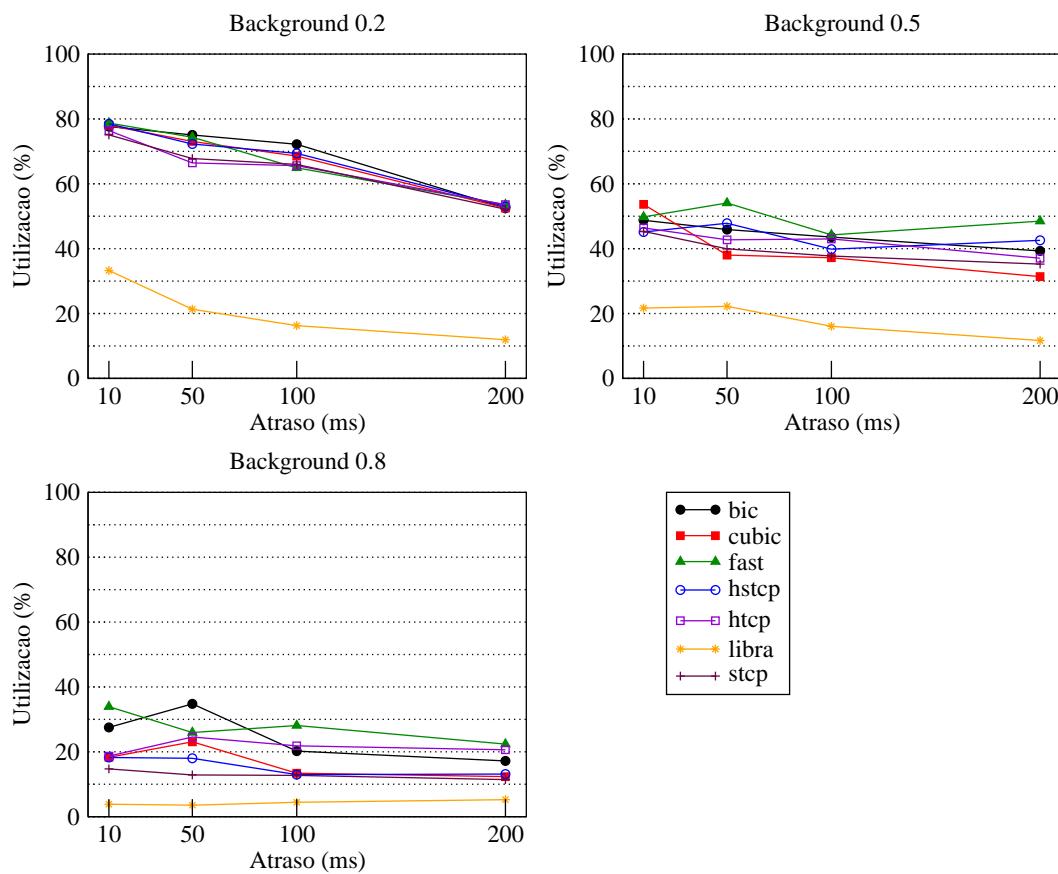
*ground* basicamente é o atraso de propagação que influencia no mecanismo de atualização da janela; enquanto que quando o enlace é ocupado por uma grande quantidade de tráfego, a formação das filas equilibra a diferença no atraso de propagação.

Neste aspecto de escalabilidade, 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. Mesmo assim, podemos destacar que os protocolos HSTCP e BIC conseguem utilizar o enlace com melhor eficiência que os fluxos dos protocolos STCP e HTCP. Com 50% de tráfego *background* o protocolo FAST obteve o melhor desempenho, seguido pelos protocolos HSTCP e BIC. E novamente, no cenário com 80% de tráfego *background*, 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 de 20%, mostrando-se eficiente na alocação de banda para enlaces altamente utilizados.

Em resumo, a investigação dos aspectos da propriedade de escalabilidade nos leva a concluir que o protocolo Libra não apresenta um comportamento desejável de escalabilidade, entretanto vale ressaltar que o objetivo principal do protocolo Libra é alcançar justiça de RTT [Marfia et al. 2005]. O protocolo FAST apresentou bom desempenho na escalabilidade em função do atraso de propagação para intensidades de tráfego *back-*

**Tabela 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**

	Tráfego <i>background</i>		
	20%	50%	80%
BIC	25.1444%	9.4507%	10.3093%
CUBIC	25.4979%	22.265%	5.9711%
FAST	25.1676%	1.2910%	11.5455%
HSTCP	25.4681%	2.5813%	5.1196%
HTCP	22.8241%	9.3567%	1.9551%
LIBRA	21.3569%	10.003%	1.4051%
STCP	22.9155%	10.092%	3.3147%



**Figura 4. Utilização em função do atraso para diferentes intensidades de tráfego *background*. Capacidade do enlace gargalo: 1Gbps. Número de conexões: 20.**

*ground* de 50% e 80%, além disso obteve o melhor resultado de utilização de enlace no cenário de escalabilidade em função do número de conexões para 80% de tráfego *background*. O protocolo HTCP obteve melhor utilização que os demais protocolos nos experimentos de escalabilidade em relação a capacidade do enlace. Em relação à escalabilidade em função do número de conexões podemos destacar o desempenho dos protocolos FAST, BIC, HTCP e HSTCP que conseguem atingir um bons níveis de utilização em relação ao

número de conexões para intensidades de tráfego *background* acima de 50%.

## 7. Conclusões e trabalhos futuros

Neste artigo investigamos a escalabilidade de sete variantes do protocolo TCP propostas para melhorar o desempenho do TCP Reno em redes de alta velocidade. Os protocolos investigados foram BIC, CUBIC, FAST, HSTCP, HTCP, Libra e STCP. A propriedade de escalabilidade foi analisada sob três aspectos principais: capacidade, atraso e multiplexação para diferentes intensidades de tráfego *background*.

Nossa investigação sobre escalabilidade nos leva a diversas conclusões acerca do desempenho dos protocolos. Observamos que apenas 16 conexões são necessárias para ocupar um enlace de  $1Gbps$  e que aumentar a quantidade de conexões não garante aumento significativo na utilização do enlace. Outra conclusão interessante é que a escalabilidade dos protocolos é limitada quando enlaces de grande capacidade ( $5Gbps$  -  $10Gbps$ ) estão disponíveis.

Ao nível dos protocolos cabe ressaltar a falta de escalabilidade do protocolo Libra e o desempenho não tão bom dos protocolos STCP e CUBIC. Tendo em vista que o protocolo STCP foi projetado para ser escalável, não destacou-se em nenhum dos três aspectos de escalabilidade avaliados. Por outro lado, o protocolo FAST apresentou bom desempenho na escalabilidade em função do atraso de propagação para intensidades de tráfego *background* de 50% e 80%, e alcançou a melhor utilização de enlace no cenário de escalabilidade em função do número de conexões quando 80% do enlace estava ocupado por tráfego *background*. O protocolo HTCP destacou-se nos experimentos de escalabilidade em relação a capacidade do enlace, alcançando nível de utilização maior que os demais protocolos. Os fluxos dos protocolos FAST, HTCP e HSTCP obtiveram os melhores níveis de utilização em função do número de conexões.

A escalabilidade é apenas uma de muitas propriedades que devem ser levadas em consideração na análise de um protocolo, tais como justiça inter/intra protocolos, estabilidade e convergência. Além disso, estudos experimentais devem ser realizados para confirmar os resultados obtidos por simulação, entretanto tais estudos estão fora do escopo deste trabalho.

Futuramente investigaremos outras propriedades dos protocolos para redes de alta velocidade como estabilidade, justiça, convergência e compatibilidade com o Protocolo TCP Reno. Além disso, vamos investigar estas propriedades em outras topologias mais realistas, como a *parking lot*<sup>1</sup>, para então alcançar conclusões definitivas sobre as diversas propriedade dos protocolos para redes de alta velocidade.

## Referências

- Bullet, H., Cottrell, R. L., and Hughes-Jones, R. (2003). Evaluation of advanced TCP stacks on fast long-distance production networks. *J. Grid Comput.*, 1(4):345–359.
- David X. Wei, Pei Cao, S. H. L. (2005). Time for a TCP benchmark suite? Disponível em: <http://www.cs.caltech.edu/weixl/research/technical/benchmark/sumary.ps>.

<sup>1</sup>*Parking lot*: um modelo de topologia que apresenta mútiplos enlaces gargalos. Permite representar caminhos com diferentes *round-trip times* e número de saltos.

- Floyd, S. (2002). Highspeed TCP for large congestion windows. RFC-3742.
- Floyd, S. (2007). Metrics for the evaluation of congestion control mechanisms. Internet-Draft: draft-irtf-tmrg-metrics-07.
- Floyd, S., Ratnasamy, S., and Shenker, S. (2002). Modifying TCP's congestion control for high speeds. Internet-Draft.
- Ha, S., Kim, Y., Le, L., Rhee, I., and Xu, L. (2006). A step toward realistic performance evaluation of high-speed TCP variants. In *In Proc of PFLDnet2006*, Nara, Japan.
- Jin, C., Wei, D. X., and Low, S. H. (2004a). FAST TCP: Motivation, architecture, algorithms, performance. In *IEEE Infocom*.
- Jin, C., Wei, D. X., Low, S. H., Buhrmaster, G., Bunn, J., Choe, D. H., Cottrel, R. L. A., Doyle, J. C., Feng, W., Martin, O., Newman, H., Paganini, F., Ravot, S., and Singh, S. (2004b). FAST TCP: From theory to experiments. *IEEE Network*, 19(1):4–11.
- Kelly, T. (2003). Scalable TCP: Improving performance in highspeed wide area networks. Technical Report 2, CERN-IT Division.
- Leith, D. and Shorten, R. (2004). H-TCP: TCP for high-speed and long-distance networks. In *In Proc of PFLDnet2004*, Argonne.
- Leith, D., Shorten, R., and Lee, Y. (2005). H-TCP: A framework for congestion control in high-speed and long-distance networks. Technical report, Hamilton Institute.
- Marfia, G., Palazzi, C., Pau, G., Gerla, M., and Roccati, M. (2005). TCP-libra: Exploring RTT fairness for TCP. Technical Report TR-050037, UCLA-CSD.
- Rhee, I. and Xu, L. (2005). CUBIC: A new TCP-friendly high-speed TCP variant. In *In Proc of PFLDnet2005*, Lyon, France.
- Sally Floyd, E. K. (2006). Tools for the evaluation of simulation and testbed scenarios. Internet-Draft: draft-irtf-tmrg-tools-03.
- Souza, E. and Agarwal, D. (2003). A highspeed TCP study: Characteristics and deployment issues. Technical Report LBNL-53215, Lawrence Berkeley National Lab.
- Weigle, M. C., Sharma, P., and IV, J. R. F. (2006). Performance of competing high-speed TCP flows. In *Networking*, pages 476–487, Coimbra, Portugal.
- Xu, L., Harfoush, K., and Rhee, I. (2004). Binary increase congestion control (BIC) for fast long-distance networks. In *IEEE Infocom*, volume 4, pages 2514–2524.
- Yee-Ting Li, D. L. and Shorten, R. N. (2006). Experimental evaluation of TCP protocols for high-speed networks. Technical report, Hamilton Institute, NUI Maynooth.