

Avaliação Experimental e Simulação do Protocolo TCP em Redes de Alta Velocidade

José Ferreira de Rezende, Luís Henrique M. K. Costa e Marcelo G. Rubinstein

Resumo—O protocolo TCP não consegue utilizar toda a capacidade de transmissão de redes de alta velocidade. Novos protocolos, como o HighSpeed TCP (HSTCP), tornam-se necessários. Neste artigo, é analisado o desempenho dos protocolos TCP e HSTCP. O trabalho compara resultados de simulação aos obtidos através de experimentos práticos e identifica os parâmetros de configuração que influenciam a obtenção de altas taxas de transmissão. Os dois parâmetros de rede que mais influenciam a vazão máxima, o atraso de transmissão e taxa de perda de pacotes, foram variados. O protocolo HSTCP é mais eficiente, porém para taxas de perda mais altas, o TCP obtém desempenho semelhante.

Palavras-Chave—Redes de alta velocidade, TCP, Avaliação.

Abstract—The TCP protocol is unable to efficiently use the capacity of high speed networks. New protocols, such as HighSpeed TCP (HSTCP), are needed. This paper analyzes the performance of TCP and HSTCP. The work compares simulation with experimental results and identifies the configuration parameters that limit the transmission rate obtained. The two most important network parameters, round-trip delay and packet loss probability, are varied. HSTCP is more efficient than TCP, but for high loss rates, TCP performs similarly.

Keywords—High speed networks, TCP, Evaluation.

I. INTRODUÇÃO

Com o advento das redes gigabit, diversas aplicações tais como as aplicações multimídia distribuídas de tempo real e as aplicações de grade computacional de larga escala estão sendo viabilizadas. De modo a utilizar essas aplicações em redes de longa distância como a Internet, a escolha de um protocolo de transporte adequado é de vital importância.

O TCP (*Transmission Control Protocol*) tem sido utilizado há mais de vinte anos na Internet. O controle de congestionamento do TCP é constituído por diferentes mecanismos. Em estado estacionário, o principal algoritmo de controle de congestionamento utilizado é o AIMD (*Additive-Increase, Multiplicative-Decrease*). Assim, a janela de congestionamento do TCP aumenta linearmente a cada reconhecimento positivo (ACK) recebido e decresce exponencialmente a cada segmento perdido. Desta forma, o AIMD adapta a taxa de transmissão ao congestionamento da rede. No entanto, o AIMD apresenta alguns problemas para redes de alta velocidade, principalmente quando o produto *banda passante* \times *atraso* é grande, pois a lentidão do aumento da janela de congestionamento provoca uma baixa utilização da banda passante disponível.

José Ferreira de Rezende e Luís Henrique M. K. Costa, Grupo de Telemática e Automação, PEE-COPPE/DEL-POLI, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, E-mails: rezende@gta.ufrj.br e luish@gta.ufrj.br. Marcelo G. Rubinstein, Departamento de Eletrônica e Telecomunicações, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brasil, E-mail: rubi@uerj.br. Este trabalho foi realizado com recursos da RNP/FINEP/FUNTEL, FUJB, CNPq, CAPES e FAPERJ.

Vários protocolos de transporte voltados para redes gigabit estão sendo propostos, visando o desenvolvimento de novas aplicações que envolvam a transmissão confiável de grande quantidade de dados [1]. Dentre os principais protocolos, destacam-se o XCP (*Explicit Congestion Control Protocol*) [2], o HSTCP (*HighSpeed TCP*) [3], o STCP (*Scalable TCP*) [4] e o FAST TCP (*Fast Active-queue-management Scalable TCP*) [5]. Alguns protocolos, como o XCP, exigem modificações nos roteadores, enquanto outros, como o HSTCP, o STCP e o FAST TCP são implementados nos sistemas finais. O STCP é baseado no HSTCP. O FAST TCP se baseia no TCP Vegas, variante pouco utilizada na Internet. Por esses motivos, foi escolhido o HSTCP para estudo neste trabalho.

A análise de protocolos de transporte para redes de alta velocidade através de experimentos práticos comparados a resultados de simulações ainda não foi muito explorada. Dunigan [6] e Coccetti e Cottrell [7] realizaram testes entre máquinas distantes geograficamente. O desempenho do TCP e do HSTCP foi comparado para uma configuração fixa, com RTT (*Round-Trip Time*) e taxa de perda fixos. Souza e Agarwal [8] compararam o desempenho do TCP e do HSTCP através de simulações, nas quais foram variados o congestionamento, a taxa de perda de pacotes, o tráfego de fundo e a política de gerenciamento de fila dos roteadores.

A principal contribuição deste trabalho é comparar resultados de simulação aos obtidos através de experimentos práticos e identificar os parâmetros de configuração que influenciam a obtenção de altas taxas de transmissão. Este trabalho avalia o desempenho dos protocolos TCP e HSTCP, levando em consideração o mecanismo de controle de congestionamento. Parâmetros como o RTT e a taxa de perda de pacotes foram variados, de modo a verificar seus efeitos na vazão dos protocolos. As medidas realizadas experimentalmente foram comparadas a resultados de simulação obtidos com o simulador *ns-2* [9]. Os resultados experimentais e de simulação mostraram a eficiência do HSTCP em manter enlaces de alta velocidade totalmente ocupados, diferentemente do TCP original. A vantagem do HSTCP em relação ao TCP é maior para RTTs altos, porém, para taxas de perdas mais altas, os dois possuem desempenho semelhante.

Este trabalho está organizado da seguinte forma. A Seção II descreve as características principais do protocolo TCP. A Seção III mostra como o protocolo HSTCP foi projetado, a partir das observações realizadas sobre o TCP. A Seção IV descreve os parâmetros utilizados nos cenários de experimentação prática e de simulação, enquanto a Seção V faz a análise comparativa dos resultados. A Seção VI conclui este trabalho.

II. TCP

O protocolo TCP é o responsável pela transmissão confiável e pelo controle de congestionamento na Internet. A transmissão confiável é obtida através da utilização de mensagens de reconhecimento positivo, ou *acknowledgements* (ACKs). O TCP permite o envio de w unidades de transmissão, ou segmentos, de tamanho máximo chamado MSS (*Maximum Segment Size*), antes da recepção de uma mensagem de reconhecimento. Esse número de mensagens não-reconhecidas permitido é chamado de tamanho de janela. Desta forma, a taxa de transmissão máxima do TCP pode ser controlada através deste tamanho de janela. Para uma conexão TCP que transmite w segmentos de tamanho MSS bytes a cada RTT segundos, a vazão é dada por

$$\frac{w \times \text{MSS}}{\text{RTT}}. \quad (1)$$

O protocolo TCP também possui um mecanismo de controle de fluxo, de modo a não sobrecarregar o receptor dos dados. Cada segmento TCP contém um campo de 16 bits chamado tamanho da janela de recepção, que indica o número de bytes que o receptor pode aceitar sem estourar o seu *buffer* de recepção. Consequentemente, o número máximo de segmentos que o TCP pode enviar sem receber um ACK é limitado pelo tamanho dessa janela de recepção. Assim, os 16 bits do campo tamanho da janela de recepção seriam um empecilho para que o número de segmentos em trânsito fosse maior do que 64 kbytes. De modo a contornar esse problema, foi criado um mecanismo de multiplicação do valor da janela de recepção por um fator negociado na fase de estabelecimento de conexão. Esse mecanismo denominado *Window Scale* está descrito na RFC 1323 [10].

Já o controle de congestionamento do TCP visa adaptar a taxa de envio de dados da fonte à carga da rede. O controle de congestionamento do TCP é composto dos algoritmos partida lenta (*slow-start*) e prevenção de congestionamento (*congestion avoidance*).

A partida lenta é usada no início de uma conexão TCP. O objetivo desta fase é lentamente fazer com que a taxa de transmissão da fonte converja para a capacidade de transmissão disponível na rede, de forma a aproveitar toda esta capacidade. O tamanho da janela de transmissão, w , começa em 1 e aumenta de 1 segmento a cada ACK recebido. O crescimento da janela é exponencial e acaba quando um patamar chamado *SSThresh* (*Slow-Start Threshold*) é alcançado, ou quando ocorre a perda de um segmento.

Depois da fase da partida lenta, o TCP passa à fase de prevenção de congestionamento, onde é utilizado o algoritmo AIMD (*Additive-Increase, Multiplicative-Decrease*) para controlar o tamanho da janela de transmissão de forma dinâmica. O AIMD funciona da seguinte forma. Primeiramente, a janela w cresce como na partida lenta, porém utilizando incrementos menores. Quando ocorre uma perda detectada por ACKs duplicados, a janela w é diminuída e a fase de prevenção de congestionamento recomeça desse novo patamar. Quando a perda é detectada por temporização, o *SSThresh* é ajustado para a metade do valor atual da janela de congestionamento e a janela de congestionamento é colocada em 1. Então, a

partida lenta é utilizada novamente até que a janela atinja o *SSThresh*.

Resumidamente, na fase de partida lenta, a cada ACK recebido a janela cresce de acordo com a seguinte equação:

$$w = w + c, \quad (2)$$

onde $c = 1$. Já para a fase de prevenção de congestionamento, para cada ACK recebido:

$$w = w + \frac{a}{w}, \quad (3)$$

onde $a = 1$, e para cada segmento perdido:

$$w = w - b \times w, \quad (4)$$

onde $b = 0,5$.

Esse mecanismo de prevenção de congestionamento do TCP limita o tamanho máximo da janela que pode ser alcançado nas redes gigabit atuais. Por exemplo, no caso de uma conexão TCP de segmentos de 1500 bytes e de RTT igual a 100 ms, para que seja atingida uma vazão em estado estacionário de 10 Gbps, obtém-se a partir da Equação 1 que o tamanho médio de w deve ser 83333 segmentos [3]. Como existe para o TCP uma relação em estado estacionário entre o tamanho médio da janela e a taxa de perda de pacotes dada por:

$$w = \frac{1,2}{\sqrt{p}}, \quad (5)$$

a taxa de perda não deve ultrapassar 2×10^{-10} , o que não é possível nas redes atuais.

III. HIGH SPEED TCP

Para evitar a dependência de taxas de perda irrealmente baixas, Sally Floyd propôs uma versão modificada do protocolo TCP, chamada HighSpeed TCP (HSTCP). O HSTCP altera o crescimento e a diminuição da janela de congestionamento para conexões com grandes janelas, na fase de prevenção de congestionamento [3]. Por outro lado, o HSTCP não altera a fase de partida lenta. A idéia, conservadora, é não aumentar a janela em mais do que o dobro, durante 1 RTT, sem realimentação explícita dos roteadores.

A função de resposta do HSTCP é mantida linear na escala log-log, como no TCP. No entanto, o HSTCP muda a inclinação dessa reta, fazendo com que para um determinado valor p de taxa de perda de segmentos, a janela w possua um valor maior no HSTCP do que no TCP. Foram escolhidos dois pontos que determinam a função de resposta do HSTCP. Esses pontos são especificados pelos parâmetros *Low_Window* e *High_Window*, que determinam a menor e a maior janelas entre as quais a modificação atuará, sendo que para cada uma dessas janelas, as taxas de perda de segmentos são respectivamente *Low_P* e *High_P*. Para taxas de perda maiores do que *Low_P*, o HSTCP se comporta exatamente como o TCP.

A função de resposta do HSTCP [11] é então dada por:

$$w = 10^{S(\log p - \log Low_P) + \log Low_Window}, \quad (6)$$

onde a inclinação da reta é:

$$S = \frac{\log High_Window - \log Low_Window}{\log High_P - \log Low_P}. \quad (7)$$

Logo,

$$w = \left(\frac{p}{Low_P} \right)^S Low_Window. \quad (8)$$

Para os valores padrões $Low_Window = 38$, $High_Window = 83000$, $Low_P = 10^{-3}$ e $High_P = 10^{-7}$, a função de resposta do HSTCP é aproximadamente igual a:

$$w = \frac{0,12}{p^{0,83}}. \quad (9)$$

Essa função de resposta do HSTCP é traduzida para os parâmetros a e b utilizados na fase de prevenção de congestionamento (Seção II). A idéia é aumentar mais o valor de a para grandes valores de w , de modo a permitir que uma maior vazão seja atingida mesmo para uma taxa de perda de segmentos significativa, e utilizar menores valores de b para grandes valores de w , para que a diminuição da janela com a perda de um segmento não seja muito grande. Para tanto, no HSTCP, os valores de a e b das Equações 3 e 4 passam a ser função do tamanho da janela, w :

$$w = w + \frac{a(w)}{w}, \quad (10)$$

e

$$w = w - b(w) \times w. \quad (11)$$

Além dos quatro parâmetros descritos anteriormente, um último parâmetro, chamado *High_Decrease*, determina o valor de $b(w)$ para $w = High_Window$.

A partir desses parâmetros, $b(w)$ e $a(w)$ podem ser calculados [11]:

$$b(w) = \frac{(High_Decrease - 0,5)(\log w - \log Low_Window)}{\log High_Window - \log Low_Window} + 0,5 \quad (12)$$

e

$$a(w) = \frac{w^2 \times 2 \times b(w)}{(2 - b(w)) \times w^{1,2} \times 12,8}. \quad (13)$$

Os valores de $a(w)$ e $b(w)$ são os mesmos para o TCP para $w \leq Low_Window$, ou seja, nesta faixa, a função de resposta do HSTCP é igual à do TCP.

A Figura 1 apresenta $a(w)$ e $b(w)$, para os valores padrões $Low_Window = 38$, $High_Window = 83000$, $Low_P = 10^{-3}$, $High_P = 10^{-7}$ e $High_Decrease = 0,1$.

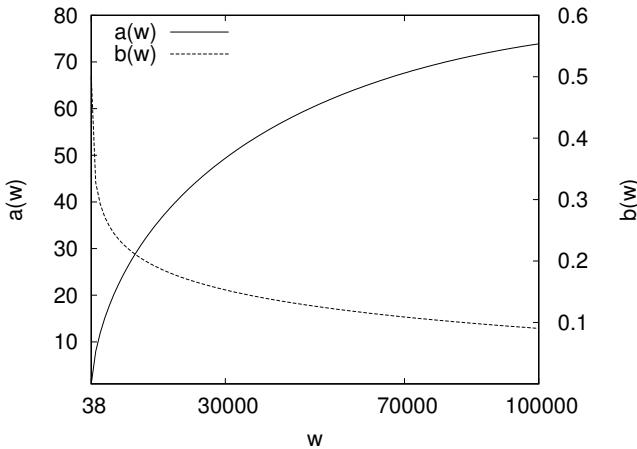


Fig. 1. Exemplo de valores dos parâmetros a e b no HSTCP.

Pode-se observar pelos valores de $a(w)$ que a janela w cresce mais rápido conforme w aumenta. Além disso, de acordo com a Figura 1, ao ocorrer uma perda, a janela decresce menos rápido para valores maiores de w . Ou seja, quanto maior a janela, maior sua taxa de crescimento e menor sua taxa de decrescimento. Com isso, uma janela média suficiente para transmissões a alta velocidade pode ser alcançada.

IV. AVALIAÇÃO DE DESEMPENHO

Foram realizados testes experimentais e simulações de modo a avaliar a vazão obtida pelos protocolos TCP e HSTCP sob diferentes valores de RTT e de taxa de perda de pacotes. Além disso, novos resultados baseados em uma topologia mais complexa foram obtidos através de simulações. Esta topologia mais complexa representa a rede experimental em fibra óptica utilizada no Projeto GIGA, da Rede Nacional de Ensino e Pesquisa (RNP) [12].

A. Experimentos Práticos

A topologia utilizada nos testes consiste de duas máquinas conectadas através de uma rede Ethernet de 1 Gbps. A ligação entre as máquinas foi feita diretamente através de um cabo *cross* de pares trançados. O computador *A* é um Pentium IV 2,8 GHz com 1 Gbyte de memória rodando Linux. A máquina *B* possui a mesma configuração de *A*, exceto pela memória de 512 Mbytes. Os dois computadores possuem placas Gigabit Ethernet 3C2000-T com o *driver* mais atual obtido de [13].

A taxa de transmissão das máquinas é limitada pelo barramento PCI de 32 bits a 33 Mhz, o que significa uma capacidade do barramento de 1,056 Gbps. A taxa de transmissão não pode atingir este valor pois deve-se descontar a sobrecarga devida à arbitragem do barramento pelo protocolo PCI. Testes preliminares indicaram uma vazão de 700 Mbps para o protocolo UDP, o que fornece um limite superior para a vazão do TCP.

A configuração das placas de rede e de diferentes parâmetros do sistema operacional é fundamental para a obtenção de altas vazões. Todas as modificações descritas a seguir foram feitas para ambos os protocolos, TCP e HSTCP. As placas de rede foram configuradas no modo *full-duplex* a 1000 Mbps. O controle de fluxo realizado na sub-camada MAC pelo protocolo Ethernet foi desabilitado para que não influísse nas medidas. Além disso, o modo de moderação dinâmica de interrupções foi desabilitado. A moderação dinâmica limita o número máximo de interrupções que o *driver* deve servir quando a carga do sistema está muito alta, o que prejudica a vazão máxima.

A versão do núcleo do sistema operacional utilizada foi a 2.4.26. Para implementar o HSTCP, o *patch* escrito por Fairey [14] para a versão 2.4.20 do núcleo foi portado para a versão 2.4.26. Além disso, diversos parâmetros de configuração do núcleo foram alterados [15], [16]. As filas de transmissão (*txqueuelen*) e de recepção (*netdev_backlog*) entre os subsistemas de rede do núcleo e o *driver* da interface de rede foram aumentadas de modo a limitar as perdas nessas filas. Com isso, a influência dessas filas nas medições foi diminuída. Outro ajuste realizado se refere ao

cache do TCP. O TCP guarda em um cache estatísticas de transferências anteriores, o que pode fazer com que as medições realizadas previamente influenciem novas medições. Por isso, os parâmetros de cache do TCP foram descartados a cada conexão. O mecanismo de reconhecimento seletivo de segmentos conhecido como SACK (Selective Acknowledgment) foi habilitado e os buffers de transmissão e recepção dos sockets foram aumentados para permitir que a janela de congestionamento do TCP alcançasse valores altos. O flag que permite o uso de janelas grandes, através de window scaling, também foi habilitado, de modo a permitir janelas superiores a 64 kbps.

De modo a avaliar o desempenho dos dois protocolos para diferentes condições de rede, foi utilizada uma ferramenta de emulação denominada NIST Net [17]. Esta ferramenta permite variar o RTT e a taxa de perda de pacotes, que são os parâmetros de maior influência na obtenção de altas taxas de transmissão. Vale observar que o NIST Net possui uma deficiência de implementação que impossibilita o uso de taxas de perda muito baixas, devido à precisão utilizada. Desta forma, o código da ferramenta foi modificado para aumentar sua precisão.

Além do NIST Net, a ferramenta Iperf versão 1.7.0 foi utilizada para medição de vazão. Foram realizadas transmissões de segmentos de 1460 bytes, com tamanho do buffer do socket do transmissor igual a 8 Mbytes e tamanho dos buffers da aplicação, de leitura e escrita, de 64 kbytes de tamanho.

B. Experimentos por Simulação

A avaliação do desempenho dos protocolos TCP e HSTCP também foi realizada através de simulações. Para tanto, o simulador ns-2 [9] versão 2.27 foi utilizado. O HSTCP é um dos protocolos disponíveis nessa versão do ns-2.

Foram utilizadas duas topologias nas simulações. A primeira consiste em dois nós ligados através de dois enlaces (um em cada sentido) de 1 Gbps. Nesse caso, o principal objetivo foi validar os resultados obtidos através dos experimentos práticos. O modelo de perda utilizado gera perdas de pacotes independentes com probabilidade p . Além disso, a latência desse mesmo enlace foi variada para simular as mudanças no RTT.

A segunda topologia usada consiste em uma rede de longa distância composta por vários roteadores interligados por enlaces de 1 Gbps (Figura 2). Esta topologia é semelhante à da rede experimental do Projeto GIGA, da RNP. No sub-projeto TAQUARA (Tecnologia, Aplicações e Qualidade de Serviço em Redes Avançadas) [12], uma das atividades consiste no teste em escala real de protocolos de transporte de alta velocidade. Os valores de RTT utilizados nas simulações, e representados na figura, são estimativas do RTT esperado na rede real.

Diversos parâmetros das simulações foram alterados em relação aos seus valores padrões no ns-2. O tamanho das filas dos enlaces é dado pelo produto banda passante \times atraso dividido pelo número de enlaces. Tanto o HSTCP quanto o TCP utilizam o mecanismo SACK. O tamanho dos segmentos TCP é de 1460 bytes. A janela de controle de fluxo

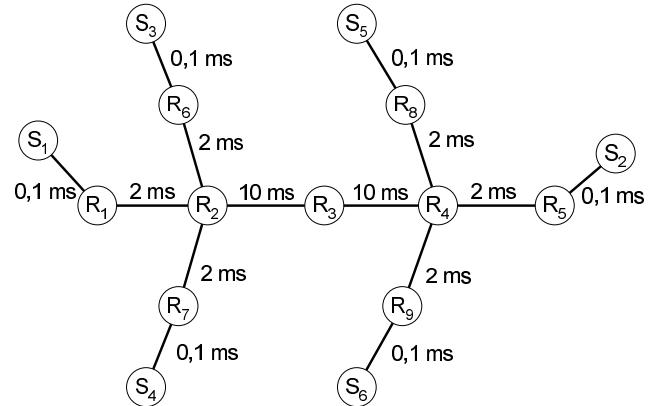


Fig. 2. Topologia simulada da rede GIGA.

é de 6000 pacotes, equivalente aos 8 Mbytes utilizados nos experimentos práticos. Os parâmetros específicos do HSTCP são os apresentados na Seção III.

V. ANÁLISE DOS RESULTADOS

Em ambas as topologias, o parâmetro de desempenho considerado foi a vazão. Na primeira topologia, a latência do enlace da origem ao destino foi variada de 0 a 40 ms e a taxa de perda de pacotes foi variada entre 10^{-6} e 10^{-4} . Em cada experimento realizado, um fluxo TCP com duração de 60 s foi gerado. Foram obtidas dez amostras de cada experimento e intervalos de confiança de 95% para a média foram representados nas figuras através de barras verticais.

Os resultados obtidos, tanto nos experimentos quanto nas simulações, são mostrados nas Figuras 3 a 5. Os resultados mostram que o HSTCP atinge um melhor desempenho do que o TCP, sendo maior o ganho quanto maior é o RTT e para taxas moderadas de perda. Isto é, quando a taxa de perda é alta, a partir de 10^{-4} , o desempenho dos dois protocolos tende a se igualar, em um valor baixo de vazão. Quando o RTT é próximo de zero, ambos os protocolos atingem a capacidade máxima do enlace, independente da taxa de perda. Isto pode ser explicado pelo fato dos mecanismos de controle de congestionamento empregados pelos dois protocolos conseguirem subir rapidamente suas taxas depois da ocorrência de uma perda. Isto se deve ao fato de os pacotes de reconhecimento chegarem rapidamente ao transmissor, o que permite que suas janelas de congestionamento cresçam rapidamente.

Pode-se notar pelas curvas experimentais e de simulação que, apesar de terem uma mesma tendência, existe uma diferença nos níveis de desempenho, principalmente quando se trata de vazões mais elevadas, superiores a 600 Mbps. Nesse caso, o desempenho obtido na simulação é maior do que aquele obtido através dos experimentos. Isto se explica pelo fato de os experimentos terem sido realizados em máquinas cujo barramento PCI não é capaz de garantir uma taxa superior a 700 Mbps, como mencionado na Seção IV-A.

Na segunda topologia (Figura 2), as simulações realizadas consistem no aumento do número de fluxos TCP e HSTCP utilizados num mesmo experimento. O objetivo desta simulação é verificar a possibilidade de se utilizar uma técnica conhecida como striping [1], empregada para acelerar o download de arquivos na Internet. Através desta técnica, a informação a

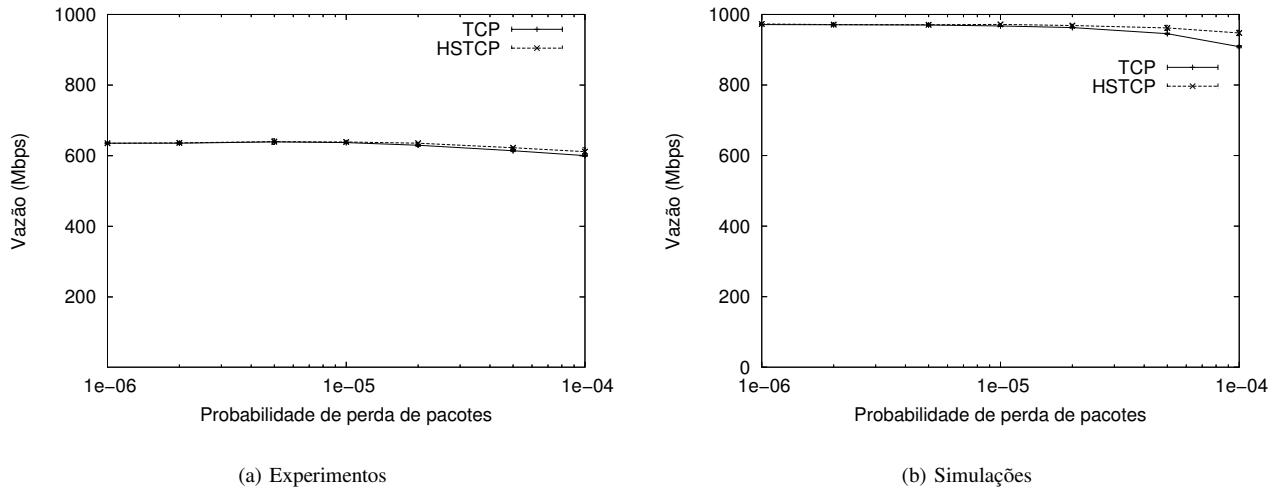


Fig. 3. Vazão com latência igual a 0 ms.

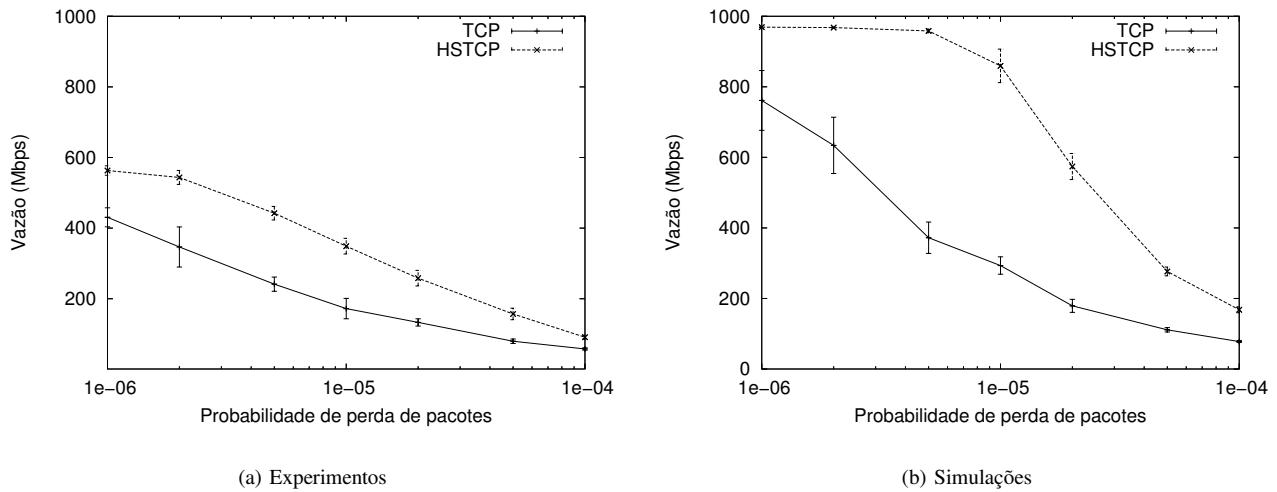


Fig. 4. Vazão com latência igual a 20 ms.

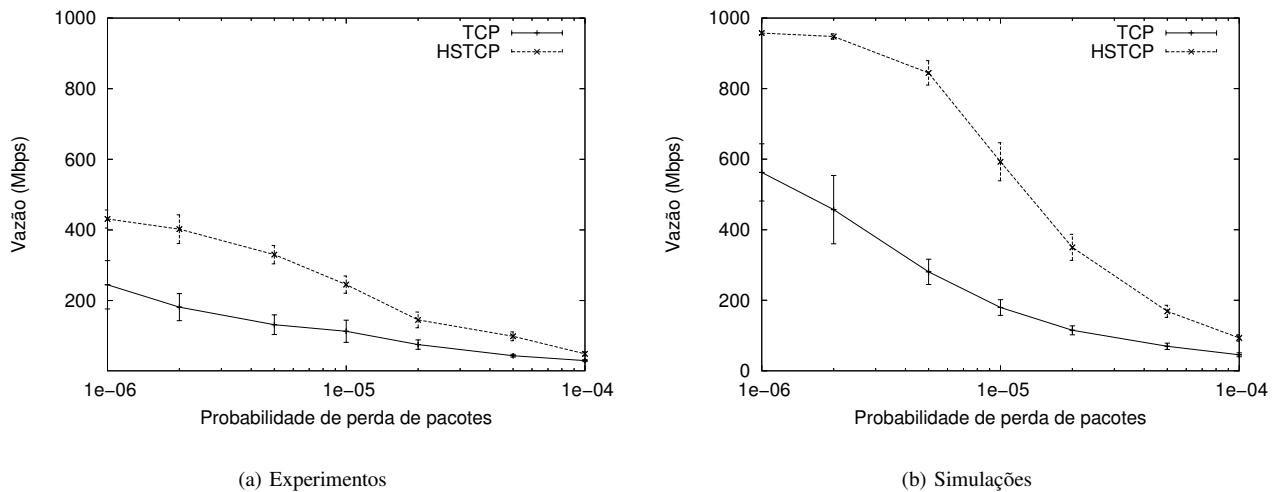


Fig. 5. Vazão com latência igual a 40 ms.

ser transferida é particionada, sendo cada parte transferida em conexões TCP independentes.

Os resultados obtidos estão mostrados na Figura 6. A vazão

plotada corresponde à média da vazão agregada de todos os fluxos obtida em dez amostras de simulação, com barras de erro correspondentes a um intervalo de confiança de 95%.

Pode-se concluir que ao empregarmos a técnica de *striping* é possível se atingir um desempenho bem próximo do HSTCP e, consequentemente, da capacidade do enlace a partir da utilização de vários fluxos TCP simultâneos, da ordem de uma dezena.

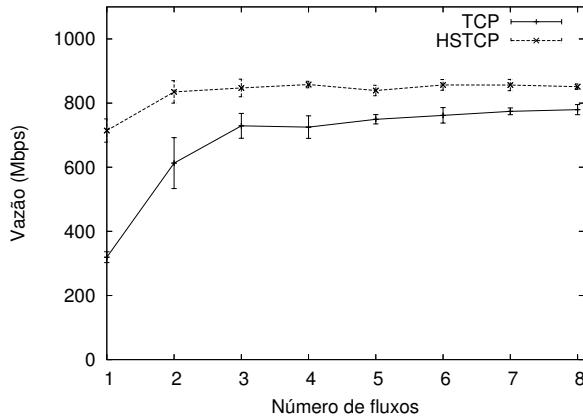


Fig. 6. Vazão agregada variando-se o número de fluxos.

VI. CONCLUSÕES

Neste artigo, foi avaliado o desempenho do protocolo TCP em redes de alta velocidade. Os resultados obtidos comprovam o problema de desempenho desse protocolo em cenários onde o produto *banda passante* \times *atraso* é alto. O problema de eficiência do TCP está relacionado ao mecanismo de controle de congestionamento empregado que, por sua natureza conservadora, não é capaz de manter enlaces de alta velocidade totalmente ocupados. O algoritmo AIMD, que controla o tamanho da janela de transmissão do TCP, depende de taxas de perda de pacotes irrealmente baixas para manter alto o valor da janela. Uma modificação deste mecanismo, chamada HSTCP, permite resolver essa deficiência através do uso de um novo algoritmo para o controle de congestionamento. Em altas taxas de perda, esse algoritmo tem um funcionamento similar ao original, mas quando em baixas taxas de perda, é mais agressivo no crescimento da janela e, por outro lado, mais conservador na sua redução, quando ocorrem perdas de segmentos.

Para avaliar o desempenho desta modificação do TCP em redes de alta velocidade, foram realizados experimentos práticos e por simulação. Nos cenários de testes, foram variados dois parâmetros que possuem influência direta na vazão do TCP, o RTT e a taxa de perda de pacotes, análise não encontrada na literatura. Os testes experimentais envolveram duas máquinas PC Linux conectadas diretamente através de placas Gigabit Ethernet. A ferramenta NIST Net foi utilizado para aumentar o RTT e a taxa de perda de pacotes desta conexão. O cenário real foi repetido no simulador de redes ns-2.

Os resultados mostraram a eficiência do HSTCP em manter enlaces de 1 Gbps totalmente ocupados, diferentemente do TCP original. Os resultados obtidos, tanto experimentais quanto por simulação, demonstraram esta tendência. Os valores medidos experimentalmente são mais baixos que os obtidos por simulação, devido às limitações do barramento das máquinas utilizadas. Pôde-se também notar que a vantagem

do HSTCP em relação ao TCP é maior para RTTs altos. No entanto, para taxas de perda mais altas (maiores que 10^{-4}), os dois protocolos tendem à mesma vazão, independente do RTT.

Finalmente, foi analisado, através de simulação, o desempenho dos protocolos numa topologia semelhante a da rede experimental do Projeto GIGA. Nesta topologia, foi mais uma vez demonstrado que o TCP é menos eficiente que o HSTCP na utilização de toda a banda passante de enlaces de alta velocidade. No entanto, várias conexões TCP em conjunto tendem a ocupar uma maior parte da banda disponível, inclusive aproximando-se do desempenho do HSTCP.

REFERÊNCIAS

- [1] A. Falk, T. Faber, J. Bannister, A. Chien, R. Grossman e J. Leigh, “Transport protocols for high performance”, *Communications of the ACM*, vol. 46, no. 11, pp. 43–49, novembro de 2003.
- [2] D. Katabi, M. Handley e C. Rohrs, “Congestion control for high bandwidth-delay product networks”, em *ACM SIGCOMM*, pp. 89–102, outubro de 2002.
- [3] S. Floyd, *HighSpeed TCP for Large Congestion Windows*. RFC 3649, dezembro de 2003.
- [4] T. Kelly, “Scalable tcp: Improving performance in highspeed wide area networks”, em *ACM Computer Communication Review*, vol. 33, pp. 83–91, abril de 2003.
- [5] C. Jin, D. X. Wei e S. H. Low, “FAST TCP: Motivation, architecture, algorithms, performance”, em *IEEE INFOCOM*, março de 2004.
- [6] T. Dunigan, “Floyd’s TCP slow-start and AIMD mods”. <http://www.csm.ornl.gov/~dunigan/net100/floyd.html>, 2004.
- [7] F. Coccetti e L. Cottrell, “TCP stacks comparison with a single stream”. <http://www-icpm.slac.stanford.edu/monitoring/bulk/fast/tcp-comparison.html>, 2004.
- [8] E. de Souza e D. Agarwal, “A HighSpeed TCP study: Characteristics and deployment issues”, relatório técnico, 2003. LBNL Technical Report Number LBNL-53215.
- [9] K. Fall e K. Varadhan, “The ns manual”, relatório técnico, março de 2005.
- [10] V. Jacobson, R. Braden e D. Borman, *TCP Extensions for High Performance*. RFC 1323, maio de 1992.
- [11] S. Floyd, S. Ratnasamy e S. Shenker, “Modifying TCP’s congestion control for high speeds”, relatório técnico, maio de 2002. <http://www.icir.org/floyd/hstcp.html>.
- [12] “Projeto GIGA”. <http://www.rnp.br/pd/giga/>, 2004. Rede Nacional de Ensino e Pesquisa.
- [13] <http://www.syskonnected.com/>, 2004.
- [14] G. Fairey, “High-Speed TCP”. <http://www.hep.man.ac.uk/u/garethf/hstcp>, 2004.
- [15] M. Rio, “Datatag”. <http://datatag.web.cern.ch/datatag/howto/tcp.html>, 2004.
- [16] “Enabling high performance data transfers”. <http://www.psc.edu/networking/projects/tcptune>, 2004. Pittsburgh Supercomputing Center.
- [17] M. Carson e D. Santay, “NIST Net - a Linux-based network emulation tool”, *ACM Computer Communication Review*, vol. 33, no. 3, pp. 111–126, julho de 2003.