

# Adaptação da Camada de Transporte Através da Correta Seleção de Algoritmo de Controle de Congestionamento

Ronaldo M. Salles<sup>1</sup>, Marcelo R. Silva<sup>1</sup>

<sup>1</sup>Departamento de Engenharia da Computação(SE/8) – Instituto Militar de Engenharia (IME)  
Praça General Tibúrcio, 80 Praia Vermelha - Rio de Janeiro - RJ CEP: 22290-270

salles@ime.eb.br, imereis@ime.eb.br

**Abstract.** *The increasing importance of wireless network together with the rapid growth of high speed networks pose new challenges to the transport protocols. Among these challenges, the most prominent are long round trip times(RTT), high packet error rates in wireless networks and the very large capacity of some links. To overcome them, a wide variety of TCP enhancements has been presented in the literature, some of them distributed in current versions of operational systems. However, the choice of the best transport procedure seems to be an open problem. This work proposes a selection methodology among different transport protocol versions, characterizing an adaptive transport layer. Such proposal is simple, low cost and built from real experiments.*

**Resumo.** *O rápido crescimento das redes sem fio somado à expansão das redes de alta velocidade, impõem novos desafios aos protocolos de transporte. Dentre eles, destacam-se as conexões de longo round trip time (RTT), elevadas taxas de erros em redes sem fio e a alta capacidade de alguns enlaces. Para superá-los, um grande número de variantes TCP é apresentado na literatura, algumas delas distribuídas em versões atuais dos sistemas operacionais. Porém, a escolha da melhor abordagem de transporte parece ser um problema em aberto. Este trabalho propõe uma metodologia para a seleção de diferentes versões de protocolos de transporte caracterizando uma camada de transporte adaptativa. A proposta é simples, de baixo custo e baseada em experimentos reais.*

## 1. Introdução

Ao considerar-se as redes de comunicação é fácil observar que elas tem permeado praticamente todas as áreas da atividade humana. Tão diversas quantas as realidades permeadas pelas redes de comunicação são as características dos componentes da sua malha. Essa heterogeneidade exige a adaptação das camadas superiores para que haja comunicação entre quaisquer dois pontos desta rede. As versões tradicionais do TCP (Reno, New-Reno), se mostram inadequadas para aproveitarem de forma eficiente a capacidade de enlaces com elevado produto banda retardo (BDP), por isso diversos autores propõem novas funções para atualização da janela de congestionamento do TCP.

Entretanto, a maioria dos trabalhos realizados para confrontar o desempenho das novas versões TCP são baseadas em simulações, onde há maior chance de a influência de fatores importantes relacionados à transmissão de dados em ambientes reais ser tratada de maneira inadequada. O presente trabalho adota uma abordagem prática, através de experimentos em laboratório, para avaliação da escalabilidade de algumas destas versões.

Este artigo traz uma investigação da viabilidade de uma camada de transporte adaptativa e levanta, de forma prática, alguns critérios para definição da escolha entre abordagens de transporte diferentes. Para isso, uma tabela relacionará o protocolo de transporte (versão TCP), mais indicado para os valores do RTT e velocidade apresentados por uma ligação fim-a-fim. Esta tabela servirá de base para as camadas de transporte adaptativas adequarem suas atividades às condições impostas pela rede em um dado momento.

O presente artigo na seção 2 apresenta um panorama da evolução do transporte TCP; a seção 3 descreve como foram feitos os experimentos e os resultados obtidos nos mesmos; na seção 4 há uma análise dos resultados obtidos; a seção 5 apresenta a sistemática de uma camada de transporte adaptativa entre as variantes TCP e por fim são apresentadas algumas conclusões e propostas para trabalhos futuros.

## **2. Evolução do TCP**

O TCP é um dos principais protocolos de transporte devido a sua predominância no conjunto de protocolos da Internet. A evolução do TCP perseguiu a maior escalabilidade deste protocolo. Para isso, as modificações propostas ao longo dos anos focaram o controle de congestionamento. Nas implementações mais antigas do TCP [Postel 1981], o emissor repassava para o canal de comunicação múltiplos segmentos até o limite da janela de fluxo anunciada pelo receptor.

Posteriormente, surgiram três novas versões do TCP: TCP Tahoe [JACOBSON 1988], dotado do “Congestion avoidance” e “slow start”, o TCP Reno [JACOBSON 1990] que trouxe o “fast retransmit” e introduziu o “fast recovery” no “congestion avoidance” e o TCP new Reno [S. Floyd 2004], propondo uma otimização no “fast recovery”.

### **2.1. Novas variantes TCP**

Apesar de toda contribuição e utilidade das versões tradicionais do TCP, estudos comprovam [Rezende et al. 2005] que o mesmo apresenta um desempenho muito restrito em redes de alta velocidade, nas quais o produto banda retardo (PBD), seja elevado, bem como em redes sem fio. Com o objetivo de aproveitar melhor os recursos destas redes, diversas variantes do TCP foram propostas, como veremos a seguir.

O **Binary Increase Congestion Control (BIC) -TCP**[Xu et al. 2004] utiliza o valor médio entre a janela de congestionamento imediatamente antes e depois de uma perda( $w$ ). Quando a “distância” da janela de congestionamento ( $cwnd$ ) de  $w$  é maior do que um determinado valor o incremento de  $cwnd$  é mais agressivo; quando  $cwnd$  está “perto” de  $w$  este incremento passa a ser mais suave. Além disso, para redes subutilizadas, onde o crescimento da janela de congestionamento passa a ser sistemático, esta variante implementa um algoritmo que aumenta esta janela mais agressivamente (exponencialmente).

No **CUBIC-TCP**[Xu and Rhee 2005] a atualização da janela de congestionamento leva em consideração o tempo decorrido desde o último evento de perda, não privilegiando, assim, fluxos com RTT's maior ou menor. Sua função de crescimento da janela de congestionamento é semelhante a do BIC, possuindo, porém, um formato mais simples e maior cautela antes de tornar o incremento agressivo.

Baseado em quatro componentes, o **Fast-TCP**[Jin et al. 2004] utiliza o *controle de estimativa*, para fornecer as informações necessárias às demais componentes (por exemplo, a estimativa do RTT mínimo e médio, integrantes da função de atualização da janela de congestionamento, calculada pela *componente controle de janela*); o *controle de dados* seleciona quais segmentos, dentre os ainda não reconhecidamente transmitidos, devem ser enviados e o *controle de rajada* regula a quantidades de dados a serem transmitidos a fim de evitar grandes rajadas de dados, as quais podem criar longas filas e aumentar a probabilidade de erros em massa.

O **High Speed TCP(HSTCP)**[Floyd 2003] segue a mesma filosofia do Reno, propondo, porém, uma melhoria no congestion avoidance, ou seja, o algoritmo AIMD é modificado de tal forma que o incremento e o decremento da janela de congestionamento variem de acordo com o tamanho da mesma.

O **Scalable TCP(STCP)**[Kelly 2003] introduz um mecanismo de atualização da janela de congestionamento em função apenas do RTT da conexão, sem levar em consideração o tamanho desta janela no momento da perda.

A versão **TCP Vegas**[Brakmo et al. 1994] utiliza o RTT como indicador da iminência ou não de um evento de perda de pacotes. Quanto maior o RTT, maior é, em princípio, a carga nos roteadores, o que requer adaptações na atualização da janela de congestionamento. O HSTCP é considerado uma evolução deste algoritmo.

No caso do **H-TCP**[R.S ] a janela de congestionamento sofre influência tanto do número de perdas de pacotes ocorridas quanto do tempo decorrido desde o último evento de congestionamento. Sendo assim, quanto melhor for a resposta da rede ao fluxo de dados (longo período sem perdas), mais agressivo será o aumento da janela de congestionamento.

A proposta dos protocolos **Westwood**[Casetti et al. 2001] e **Westwood+**[Dell'Aera et al. 2004] visa monitorar continuamente os pacotes ACK's para estimar o ERE (Eligible Rate Estimation), fundamental na atualização da janela de congestionamento e cuja técnica de estimação está em aberto. Como eles estimam a taxa de transmissão para atualizar a janela de congestionamento estes protocolos se mostram extremamente eficiente em redes wireless onde as perdas de pacotes, geralmente causadas por ruídos, muitas vezes requerem um aumento na quantidade de retransmissões e não uma redução do número de quadros inseridos no meio.

A tabela 1 faz a correspondência entre as variantes TCP e as situações mais indicadas para aplicação de cada uma delas, além disso, é fornecida uma breve explicação de por que se espera uma boa resposta destas variantes nas situações mencionadas.

Infelizmente a maioria dos testes para avaliação da escalabilidade das atuais variantes TCP são realizados a partir de simulações. Naturalmente, a comparação destes protocolos utilizando-se ambientes reais (um dos objetivos deste trabalho), representa um diferencial, conduzindo a conclusões mais concretas.

**Tabela 1. Variantes TCP e suas indicações**

Protocolo	Indicação	Explicação
BIC TCP	Fluxos com RTT's diferentes e não TCP.	Atualização da Janela de congestionamento baseada no tamanho da mesma. Agressivo com fluxos TCP.
CUBIC	Fluxos com longos RTT's e diferentes entre si.	Janela atualizada conforme o tempo desde o último evento de perda.
Fast TCP	Redes com RTT's estáveis.	Fluxos com RTT's muito variável dificultam as estimativas próprias deste protocolo.
H-TCP	Redes não congestionadas	Se há muitas perdas, este protocolo se comporta conforme o TCP.
High Speed TCP	Fluxos com RTT's próximos	Os fluxos com RTT's mais curtos são privilegiados, deixando pouca banda para os de RTT's mais longos
Scalable TCP	Conexões com RTT homogêneo.	O tamanho da janela depende da frequência com que chegam os ACK's.
TCP Vegas	Redes de pequeno jitter.	Grandes oscilações do RTT dificultam a detecção da iminência de um congestionamento.
TCPWestwood TCPWestwood+	Redes wireless	Pouco agressivo na redução da janela de congestionamento.

### 3. Experimentos

Para uma avaliação prática das versões TCP de alta velocidade foram realizados downloads de arquivos variando-se o protocolo de transporte<sup>1</sup> e a capacidade do enlace. Para cada “clock rate” ajustado, acrescentou-se, através do Netem[Hemminger 2005], atrasos (200, 400, 600, 800 e 1000ms), ao RTT. Fixados o RTT e a versão Suse 11, kernel 2.6.25.11-0.1-pae do protocolo de transporte (RENO,BIC,CUBIC,STCP,HSTCP e HTCP), mediu-se o tempo de download. O download foi realizado de um servidor apache2, utilizando-se como cliente o FireFox 3.0.1. A partir destes testes podemos escolher qual dos protocolos se comportou melhor, isto é, proporcionou menor tempo de download. Os testes foram realizados no laboratório de redes do IME. As máquinas utilizadas seguem a seguinte configuração: Intel(R) Pentium(R) 4 HT 3.00; memória de 512 MB; placa de rede ASUSTeK 82540EM Gigabit Ethernet; HD Samsung SP 0802N 7200 rpm IDE ATA 133 e swap de 1Gb

#### 3.1. De 1200 bps a 1Mbps

A flexibilidade na velocidade do enlace foi garantida pela infra-estrutura da figura 1: Duas máquinas, grecia01, na rede 4 e grecia11, na rede 6, se comuni-

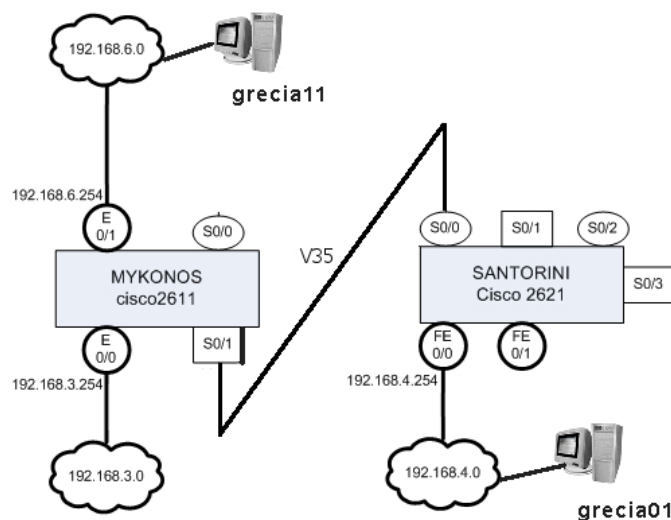


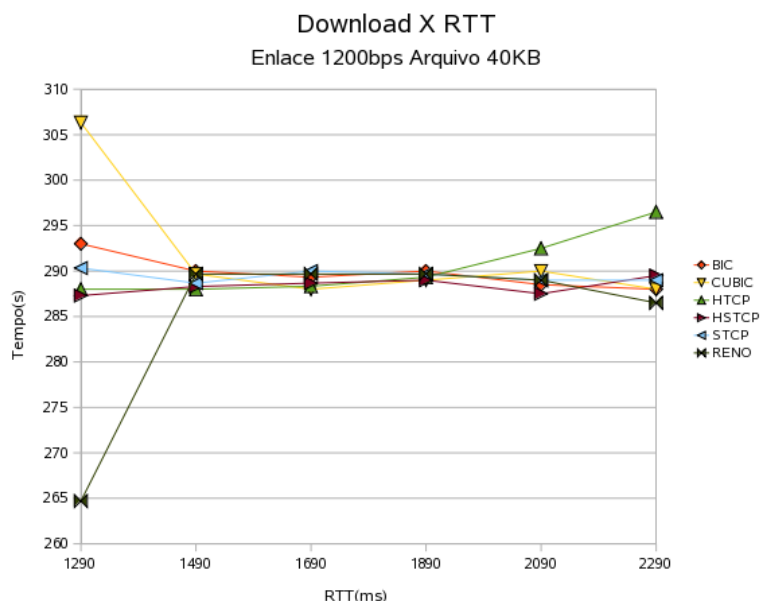
Figura 1. Estrutura para Enlaces de 1200bps a 1Mbps.

cavam através de um enlace V.35, que fazia a ponte entre os roteadores Mykonos (Cisco(R)2611), e Santorini (Cisco(R)2621), e cuja velocidade foi configurada para os valores 1.2, 56, 512 e 1000 Kbps.

##### a) Enlace de 1200bps

Com a velocidade do enlace V35 ajustada para 1200bps, foi levantado o tempo médio de download de um arquivo de 40K para cada um dos acréscimos no RTT considerados, chegando-se ao gráfico 2. Como se pode observar, nestas condições o TCP Reno

<sup>1</sup>a partir do kernel 2.6.7 algumas distribuições linux já trazem a opção de carregamento de outros algoritmos de controle de congestionamento em tempo de execução através do comando sysctl.



**Figura 2. Desempenho dos Protocolos: 1200bps**

se comportou de forma extremamente satisfatória, conseguindo em alguns caso um comportamento melhor do que as outras abordagens. Para o menor RTT (1290ms), o Reno alcançou um tempo de download menor do que aquele obtido com os outros protocolos, principalmente em relação ao CUBIC. Conforme aumentamos o RTT, os protocolos, com exceção do HTCP, praticamente se igualaram durante os testes, com ligeira vantagem para o Reno. O HTCP começa a se destacar negativamente de um RTT de 2090ms em diante.

#### **b) Enlace de 56Kbps**

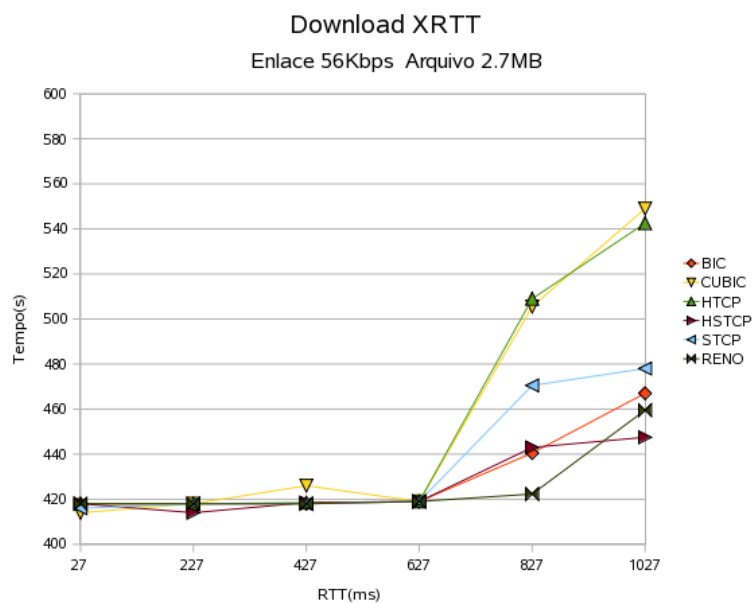
Agora os downloads realizados transferiram um arquivo de 2.7M. Os tempos de download para cada RTT estão sumarizados no gráfico 3. Mais uma vez o TCP Reno se mostra uma excelente escolha para este tipo de configuração. Até o RTT de 627ms os protocolos se equivalem; a partir daí, observa-se uma grande vantagem dos protocolos BIC, HSTCP e RENO em relação aos protocolos STCP, HTCP e CUBIC estes dois últimos com desempenho semelhante, fornecendo os maiores tempos de download. O desempenho global mostra ligeira vantagem para o RENO nesta fase dos testes.

#### **c) Enlace de 512Kbps**

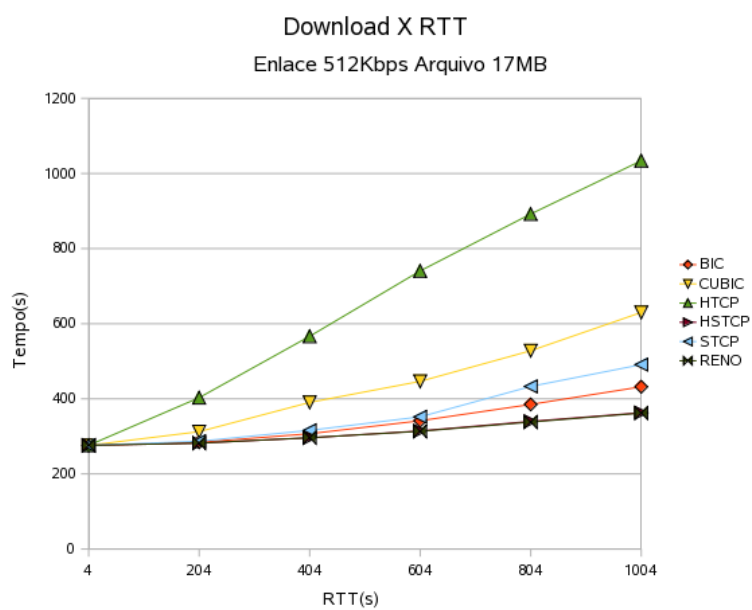
O tamanho de arquivo considerado nesta rodada de teste foi de 17M, o que resultou no gráfico 4. Para esta fase dos testes, o HTCP impôs um tempo de download muito acima dos outros. Os protocolos de melhor comportamento (praticamente idênticos), são o RENO e o HSTCP. Os demais protocolos podem ser ordenados de forma crescente em relação ao tempo de download para cada um dos valores de RTT considerados da seguinte forma: BIC, STCP e CUBIC.

#### **d) Enlace de 1Mbps**

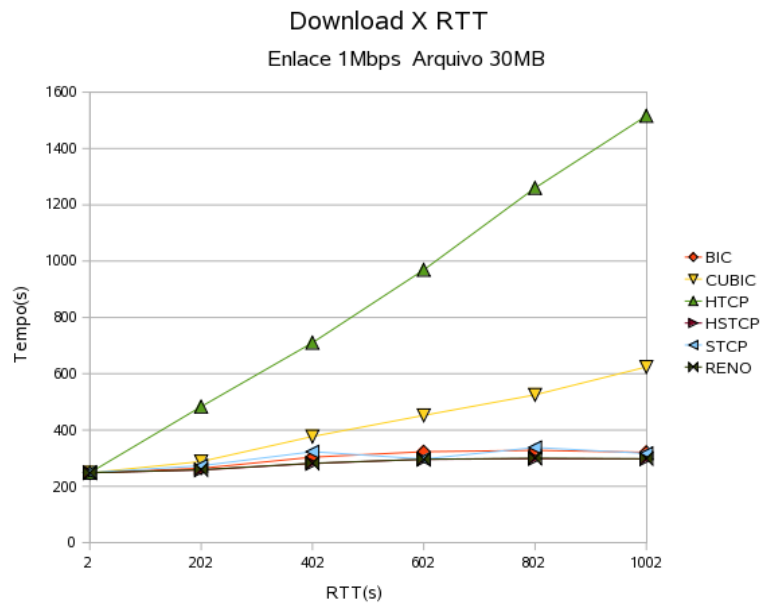
Para um arquivo de 30M, chega-se ao gráfico 5. Como no caso anterior, o HTCP atinge os maiores tempos de download; em seguida vem o CUBIC, mas com tempos de



**Figura 3. Desempenho dos Protocolos: 56Kbps**



**Figura 4. Desempenho dos Protocolos: 512Kbps**

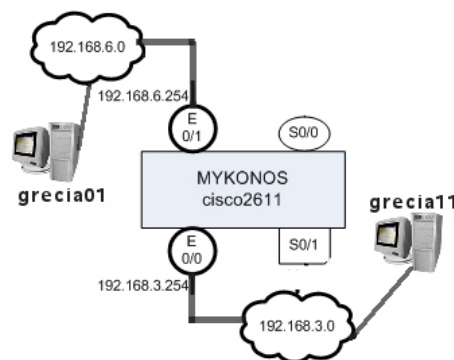


**Figura 5. Desempenho dos Protocolos: 1Mbps**

transferência bem menores. Os protocolos RENO, STCP, HSTCP e BIC tiveram desempenhos muito semelhantes, proporcionando os menores tempos de download.

### 3.2. Enlace de 10Mbps

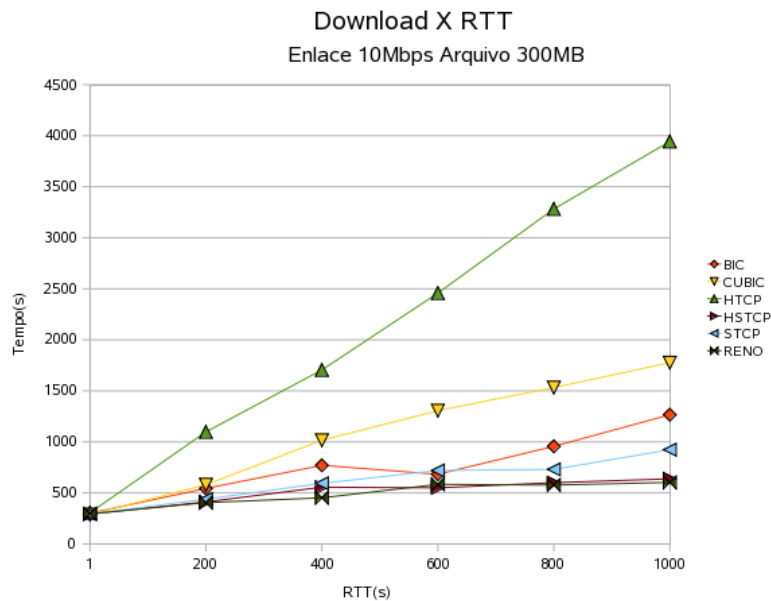
Para ligarmos duas máquinas a 10Mbps, utilizou-se a configuração ilustrada na figura 6. Como pode-se observar na figura, as redes 3 e 6 se interligam por interfaces ethernet (ve-



**Figura 6. Estrutura para Enlace de 10Mbps.**

locidade de 10Mbps), do roteador Mykonos . Durante esta fase um arquivo de 300M foi transferido da máquina greCIA01, na rede 6, à máquina greCIA11, na rede 3, para o levantamento dos pontos do gráfico 7. Como vemos, o TCP Reno continua entre os protocolos de melhor desempenho. Mais uma vez o HTCP é o destaque negativo nesta configuração, seguido pelo CUBIC que, apesar disto, mostra-se bem mais eficiente do que o HTCP. Os protocolos STCP e BIC demonstram um comportamento intermediário entre os protocolos considerados. Os menores tempos de download estão associados aos protocolos RENO e HSTCP, os quais mantêm suas respostas praticamente idênticas durante todas as situações consideradas nesta fase.





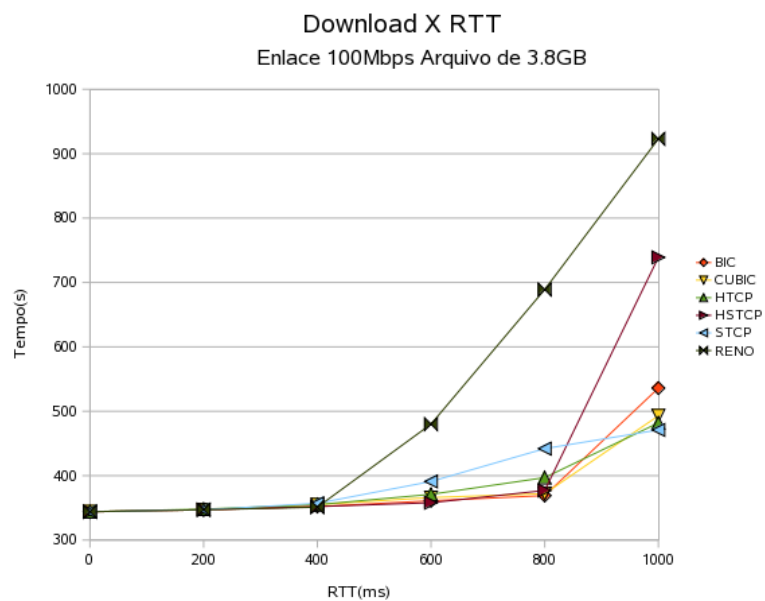
**Figura 7. Desempenho dos Protocolos: 10Mbps**

### 3.3. Enlace de 100Mbps

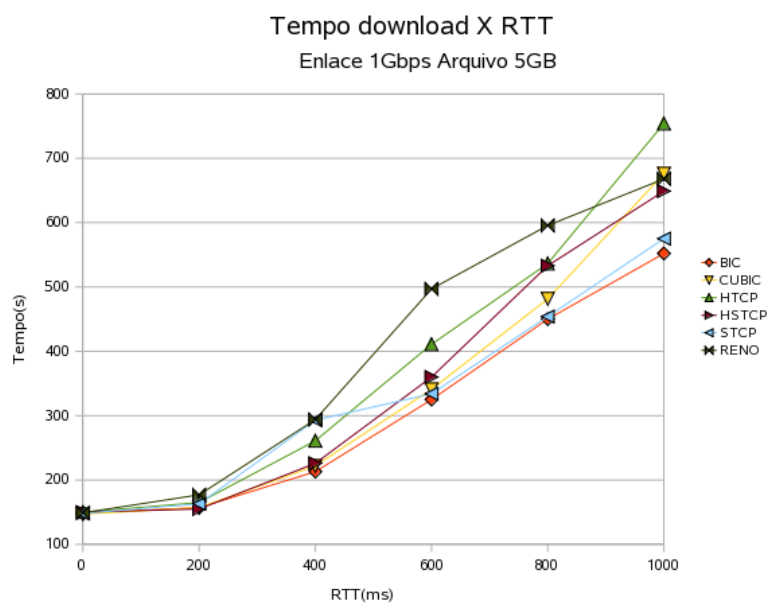
Como estamos trabalhando com placas FastEthernet concentradas em um switch catalyst(R) 2950, através de cabos par trançado da categoria 5e, colocamos duas máquinas em uma mesma sub-rede para alcançarmos a velocidade de 100Mbps, já que a velocidade máxima do V35 nos roteadores disponíveis chega no máximo a 80Mbps. Um arquivo de 1G, trocado entre as máquinas grecia01 e grecia 11, ambas na rede 6, serviu de base para a construção do gráfico 8. Não se pode deixar de destacar que, nesse enlace, a partir de um acréscimo de 600ms no RTT o TCP Reno passa a ter um desempenho abaixo de todas as outras variantes TCP consideradas. Destaque positivo para o CUBIC-TCP. Até o RTT de 400ms o desempenho dos protocolos é idêntico, mas para RTT's acima deste valor o RENO é apontado pelos testes como a pior opção, seguido pelo HSTCP. Agora o HTCP junto com o CUBIC e o BIC constituem as melhores opções seguidos de perto pelo STCP. Entretanto, o CUBIC obtém uma ligeira vantagem média quando considerados todos os pontos.

### 3.4. Enlace de 1Gbps

As máquinas grecia01 e grecia11 foram diretamente ligadas por um cabo par trançado da categoria 5e, pois duas placas FastEthernet conectadas por um cabo desse tipo trabalham a uma velocidade de 1Gbps. Um arquivo de 5GB foi transportado por cada um dos protocolos a cada acréscimo no RTT, de acordo com o gráfico 9. A partir de um acréscimo de 400ms no RTT, o desempenho do TCP Reno começa a se degradar. Destaque positivo para o BIC TCP nesta configuração. Os menores tempos de downloads nesta configuração estão associados ao BIC e ao STCP, com o BIC conseguindo alcançar tempos um pouco menores. Entre 400 e 800 ms os protocolos em ordem decrescente em relação ao tempo de download são RENO, HTCP, HSTCP e CUBIC. Em 1000ms, porém, a ordem passa a ser HTCP, CUBIC, RENO e HSTCP.



**Figura 8. Desempenho dos Protocolos: 100Mbps**



**Figura 9. Desempenho dos Protocolos: 1Gbps**

## 4. Análise geral

### 4.1. Enlaces de até 10Mbps

Os dados extraídos dos experimentos deixam claro que para enlaces de velocidade até 10Mbps a versão Reno constitui uma excelente opção, independente do RTT. Esta conclusão era esperada, pois os enlaces em questão constituem o “nicho” do Reno, ou seja, estamos nos enlaces em função dos quais o Reno baseou toda sua evolução; a utilização de uma outra abordagem nestes casos pode inclusive comprometer o desempenho da transmissão, basta observarmos o desempenho do HTCP nos gráficos correspondentes. Como segunda opção em termos de desempenho, temos o BIC TCP que em geral se comportou melhor do que as outras abordagens estando, porém, abaixo do TCP-Reno.

### 4.2. Enlace de 100Mbps

Para enlaces de velocidade de 100Mbps o Reno manteve-se entre as abordagens de melhor desempenho até um acréscimo de 600 ms no RTT, a partir do qual o CUBIC-TCP e o STCP passaram a ser os protocolos de transporte mais indicados, com ligeira vantagem para o CUBIC, quando considerados todos os demais pontos até 1000ms. Os gráficos mostram que o tempo de download pode ser reduzido em até 46,7%, escolhendo-se o protocolo de transporte adequado (CUBIC em vez do RENO). Os diversos ganhos percentuais proporcionados pelo protocolo CUBIC em relação ao RENO  $\left( \frac{(T_{RENO} - T_{CUBIC}) * 100}{T_{RENO}} \right)$ , onde  $T_X$  representa o tempo de download utilizando-se o protocolo X durante os experimentos para o enlace em questão), são sumarizados na tabela 2:

**Tabela 2. Ganhos Percentuais para Enlaces de 100Mbps**

RTT(ms)	Ganho Percentual(CUBIC sobre RENO)
600	27,7
800	46,0
1000	46,7

### 4.3. Enlace de 1Gbps

Agora, a partir de um acréscimo de 400ms o Reno é superado pelos protocolos BIC e STCP. Um fato interessante neste tipo de enlace é que o Reno conseguiu um desempenho melhor ou igual ao de algumas variantes(CUBIC, HTCP e HSTCP) que teoricamente deveriam superá-lo neste tipo de cenário. Como o BIC se destacou vamos sumarizar na tabela 3 os ganhos percentuais  $\left( \frac{(T_{RENO} - T_{BIC}) * 100}{T_{RENO}} \right)$ , que ele proporcionou em relação ao Reno

### 4.4. Paralelo com a literatura

Verdadeiramente, a adoção de um protocolo de transporte preparado para redes de alto produto banda retardo pode aproveitar melhor a capacidade fornecida pelo enlace. Entretanto, não há um consenso entre os pesquisadores sobre qual seria a abordagem de

**Tabela 3. Ganhos Percentuais para Enlaces de 1Gbps**

RTT(ms)	Ganho Percentual(BIC sobre RENO)
400	27,5
600	34,6
800	24,4
1000	17,3

transporte mais escalável. Em [Michel and Fonseca 2007], por exemplo, o HTCP é tido como protocolo altamente escalável; o CUBIC e o STCP são destacados negativamente sob este aspecto no referido trabalho. Já o “Advanced Computing for Science Department” [Tierney 2008] sugere o HTCP e o CUBIC como as melhores opções em redes com BDP elevado. Na prática, porém, para um enlace de 1G o HTCP demonstrou um desempenho muito aquém de algumas outras variantes, inclusive do próprio CUBIC, cuja performance está abaixo do BIC e do STCP, por exemplo, apontados como protocolos de excelente escalabilidade por [Xu et al. 2004], fato comprovado pelos experimentos.

## 5. Adaptação entre as variantes TCP

Como já foi mencionado anteriormente, a própria evolução da camada de transporte TCP evoluiu (infelizmente de maneira estática), conforme às novas demandas da rede. Antes de propor como seria uma camada de transporte adaptativa dentro das variantes TCP, analisemos a função de atualização da janela de congestionamento do TCP Reno: o gráfico desta função é linear acima da threshold e exponencial abaixo dela.

Este comportamento forma a base para o estudo da viabilidade de uma camada de transporte adaptativa entre as variantes TCP, pois a atualização da janela de congestionamento do TCP Reno já pressupõe uma adaptação da mesma de acordo com o “feed-back” que a rede lhe proporciona.

Do acima exposto, podemos concluir que o chaveamento entre funções de atualização da janela de congestionamento é algo inerente às versões tradicionais do TCP. Um outro fato muito interessante é que os algoritmos que procuraram dar ao TCP uma maior escalabilidade concentram seus esforços na forma de atualização da janela de congestionamento; e não poderia ser diferente, pois é esta a função que vai determinar a quantidade de datagramas inseridos a cada rodada de transmissão na rede.

Fica claro, portanto, que podemos disponibilizar na camada de transporte um conjunto de funções que possam ser, de acordo com as condições pontuais da rede, escolhidas durante a troca de dados. Sendo assim, para uma camada de transporte adaptativa precisaríamos apenas trabalhar no controle de congestionamento da camada em questão.

Agora, baseados nos nossos testes, apresentamos a tabela 4, que sintetiza de forma prática e objetiva a análise dos dados levantados nos testes. Esta tabela serve de parâmetro para o chaveamento da melhor função de atualização da janela de congestionamento.

Para aplicação dos resultados levantados durante o trabalho, foi desenvolvido um programa que realiza a adaptação da camada de transporte da seguinte forma: durante a linha de execução do referido programa faz-se uma chamada ao *pathrate* que gera um arquivo contendo o RTT e a velocidade estimados para a ligação entre dois endereços de

**Tabela 4. Condições de emprego dos protocolos**

Velocidade do Enlace	RTT	Protocolo Indicado
Até 10Mbps	Qualquer	RENO
100Mbps	$\leq 400\text{ms}$	RENO
	$> 400\text{ms}$	CUBIC
1Gbps	$\leq 200\text{ms}$	RENO
	$> 200\text{ms}$	BIC

rede. Estes valores são então utilizados pelo programa para, de acordo com a tabela 4, selecionar-se (através do sysctl), a abordagem de transporte mais adequada.

Tal programa poderia ser executado em background e de forma totalmente transparente para os usuários finais ajustar de tempos em tempos (ex.: crontab), o protocolo de transporte mais adequado às condições impostas pela rede.

## 6. Conclusões e trabalhos futuros

Os resultados comprovaram que uma grande economia, tanto de tempo quanto de recursos, pode ser atingida quando o transporte se adequa ao momento vivido pela infraestrutura de comunicação. Como os limiares apresentados na tabela 4 são bem espaçados, uma boa escolha no que diz respeito à função de atualização da janela de congestionamento dispensa medidas extremamente precisas, o que torna as implementações de camadas de transporte adaptativas independentes de sondagens muito eficientes para levantamento dos parâmetros da rede.

Outro fato bem interessante é o bom desempenho do TCP Reno, como opção de transporte de dados para um grande número de situações, principalmente naquelas mais tradicionais (velocidades até 10Mbps), e até em alguns enlaces mais robustos, como observado nos enlaces de 100MB, para RTT's abaixo de 400ms e ligações de 1GB com RTT inferior a 200ms.

Durante todo o trabalho, baseamos nossas conclusões, como geralmente é feito em trabalhos científicos, em um modelo simplificado, isto é, assumimos que a camada de transporte adaptativa é capaz de chavear entre as abordagens disponíveis de forma instantânea, sem fazermos qualquer consideração sobre a latência que pode existir durante este chaveamento, principalmente no que diz respeito ao ajuste das variáveis de contexto para realização das atividades da camada de transporte e como aproveitarmos as informações presentes nestas variáveis para alimentarmos aquelas instanciadas pela nova abordagem de transporte ora chaveada.

## Referências

- Brakmo, L., O'Malley, S., and Peterson, L. (1994). Tcp vegas: New techniques for congestion detection and avoidance. SIGCOMM '94 Symposium.
- Casetti, C., Gerla, M., Mascolo, S., Sanadidi, M. Y., and Wang, R. (2001). Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. ACM Mobicom 2001.

- Dell'Aera, A., Grieco, L. A., and Mascolo, S. (2004). Implementation of westwood+ tcp with rate-halving: A performance evaluation over the internet. IEEE International conference on Communication (ICC 2004).
- Floyd, S. (2003). Fast tcp: Motivation, architecture, algorithms, performance.
- Hemminger, S. (2005). Network emulation with netem. LCA.
- JACOBSON, V. (1988). Congestion avoidance and control. Proc. SIGCOMM '88 Conf.ACM, pp. 314-329.
- JACOBSON, V. (1990). Berkeley tcp evolution from 4.3-tahoe to 4.3-reno. In *Proceedings of the Eighteenth Internet Engineering Task Force*, p.365 (Sept.). University of British Columbia, Vancouver, B.C.
- Jin, C., Wei, D. X., and Low, S. H. (2004). Fast tcp: Motivation, architecture, algorithms, performance. IEEE INFOCOM.
- Kelly, T. (2003). Scalable tcp: Improving performance in high-speed widearea networks. In *ACM SIGCOMM Computer Communication Review*, Volume 33, Issue 2, pp. 83-91.
- Michel, N. F. and Fonseca, N. L. S. (2007). Uma investigação sobre a escalabilidade de variantes do protocolo tcp para redes de alta velocidade. XXVII Congresso da SBC.
- Postel, J. (1981). Internet protocol rfc 791. USC/Information Sciences Institute.
- Rezende, J. F., Costa, L. H., and Rubinstein, M. G. (2005). Avaliação experimental e simulação do protocolo tcp em redes de alta velocidade. XXII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES - SBrT'05, 04-08 DE SETEMBRO DE 2005, CAMPINAS, SP.
- S. Floyd, T. Henderson, A. G. (2004). Rfc 3782, "the newreno modification to tcp's fast recovery algorithm". IETF, RFC 3782, April 2004.
- Tierney, B. L. (2008). Tcp tuning guide. Advanced Computing for Science.
- Xu, L., Harfoush, K., and Rhee, I. (2004). Binary increase congestion control (bic) for fast long-distance networks". IEEE INFOCOM 2004.
- Xu, L. and Rhee, I. (2005). Cubic: A new tcp-friendly high-speed tcp variant. Workshop on Protocols for Fast Long Distance Networks.