

Avaliação do uTP em Aplicações de Alto Desempenho

Tiago Heinrich¹, Rafael R. Obelheiro¹, Guilherme P. Koslovski¹

¹Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC

Resumo. *A latência na transferência de dados pela rede é um fator limitante para o desempenho de aplicações paralelas e distribuídas. Diversas melhorias foram propostas, incluindo novos algoritmos de controle de congestionamento para o TCP e implementação de controle de congestionamento na camada de aplicação, usando o UDP como protocolo de transporte. Este artigo avalia o uso do uTP, comparando o seu uso de banda com a de três variantes de TCP.*

1. Introdução

A latência de comunicação entre processos exerce grande influência sobre o desempenho de aplicações paralelas e distribuídas, especialmente quando são necessárias transferências de grandes volumes de dados [Evans et al. 2003, Kouzes et al. 2009]. Um fator importante para essa latência são os mecanismos de controle de congestionamento do TCP (*Transmission Control Protocol*), e por isso existe uma busca constante por mudanças no controle de congestionamento do TCP que permitam explorar melhor a largura de banda disponível e propiciem menor latência [Afanasyev et al. 2010, Cardwell et al. 2016].

Uma abordagem alternativa é usar o UDP como protocolo de transporte, acrescentando um mecanismo de controle de congestionamento na camada de aplicação para evitar o consumo excessivo de banda. Um exemplo popular é o uTP (*Micro Transport Protocol*) [Norberg 2018], um protocolo usado em transferência de arquivos no BitTorrent, cujo princípio é maximizar a utilização de banda na ausência de tráfego concorrente e reduzir a taxa de transmissão quando é detectado um aumento no atraso de rede.

Nesse contexto, uma questão que surge é se a adoção do uTP poderia beneficiar aplicações paralelas e distribuídas. Este artigo compara o desempenho de fluxos uTP e TCP concorrentes, analisando como a banda disponível é dividida entre os protocolos. A implementação do uTP disponível na *libtorrent* [Norberg 2018] foi comparada com três variantes de TCP, CUBIC [Ha et al. 2008], Westwood+ [Grieco and Mascolo 2004] e BBR [Cardwell et al. 2016], todas disponíveis no Linux. O CUBIC usa segmentos perdidos como indício de congestionamento, reduzindo sua janela de transmissão, e foi escolhido por ser a variante padrão do Linux. O Westwood+ foi escolhido por ser uma variante conhecida que tem uma proposta um pouco diferente, uma vez que define sua janela de transmissão em função de uma estimativa da largura de banda disponível. O BBR foi escolhido por ser uma variante emergente; ele é baseado em modelo, usando medições de largura de banda e RTT (*round-trip time*) para determinar o tamanho da janela. O uTP, por sua vez, baseia-se em medições de atraso unidirecional, reduzindo a janela quando observa aumento no atraso.

2. Experimentos e Resultados

Os experimentos foram realizados usando um topologia *dumbbell*, com um cliente e um servidor ligados por dois *switches* com a finalidade de ter mais de um *hop*, em uma rede

isolada. A topologia representa, de forma simplificada, a ocorrência de um gargalo de congestionamento. Os enlaces eram de 100 Mbps, e o MSS (*Maximum Segment Size*) dos fluxos foi de 1460 bytes. O tráfego TCP foi gerado usando o iperf3 [ESnet 2018], e o tráfego uTP foi gerado com o qBittorrent [Dumez 2018], um cliente de código aberto que usa a *libtorrent*. Foram realizados dois conjuntos de experimentos. O primeiro conjunto permite analisar de que forma a banda disponível é dividida entre um fluxo uTP e um fluxo TCP concorrente, e o segundo mostra como a banda é partilhada por um fluxo uTP e quatro fluxos TCP concorrentes. Os resultados estão mostrados na Figura 1.

A Figura 1(a) apresenta a divisão de banda entre um fluxo CUBIC e um fluxo uTP. O fluxo uTP recebe aproximadamente 80% da banda disponível ao longo do experimento, enquanto o CUBIC alcança uma média inferior a 20%, embora chegue a ser o fluxo dominante em vários períodos curtos. A soma dos fluxos não atinge 100%, o que mostra que nem toda a banda disponível consegue ser aproveitada.

A Figura 1(b) compara um fluxo BBR e um fluxo uTP. Observa-se uma divisão de banda semelhante à do CUBIC (cerca de 85% para o fluxo uTP e 15% para o fluxo TCP, em média). O BBR mostra-se menos agressivo que o CUBIC, cedendo ainda mais banda para o uTP. Ainda, a Figura 1(c) mostra a comparação entre Westwood+ e uTP. A divisão de banda fica equilibrada: o fluxo uTP recebe menos de 60%, enquanto o fluxo TCP obtém pouco mais de 40%. Esse equilíbrio é atingido cerca de 75 segundos após o início do experimento.

Nos experimentos com um fluxo uTP e quatro fluxos TCP concorrentes, uma divisão equitativa da largura de banda resultaria em 20% para o fluxo uTP e 80% para os fluxos TCP. Quando é usado o CUBIC (Figura 1(d)), observa-se uma situação bem diferente da expectativa: o fluxo uTP continua dominante, recebendo quase 60% da banda, enquanto os quatro fluxos TCP recebem menos de 40%. Com o BBR (Figura 1(e)) a discrepância é ainda maior: um único fluxo uTP consome mais de 70% da banda, enquanto os quatro fluxos TCP precisam dividir pouco mais de 20% entre si. Observa-se ainda neste cenário uma maior variabilidade na utilização da banda, com os fluxos TCP tornando-se dominantes por curtos períodos.

Uma situação mais próxima da esperada é observada novamente com o Westwood+, como pode ser visto na Figura 1(f). Os fluxos TCP obtêm cerca de 75% da banda, enquanto o fluxo uTP fica com pouco mais de 20%. Este experimento também obtém uma divisão mais estável da banda entre TCP e uTP.

No geral, o uTP mostra-se mais agressivo no uso de banda do que as variantes de TCP consideradas, notadamente CUBIC e BBR. Isso indica que o uTP pode reduzir o tempo de transferência de dados em aplicações paralelas e distribuídas, especialmente quando as transferências não são sincronizadas, em detrimento de outros fluxos concomitantes.

3. Conclusão

O processamento de grandes volumes de dados é hoje um dos principais cenários de uso de aplicações paralelas e distribuídas. Nesse cenário, a busca pela redução na latência da transferência de dados pela rede é uma constante. Os algoritmos de controle de congestionamento usados no TCP são frequentemente apontados como contribuidores para essa latência, o que suscita a investigação de novos métodos para reduzi-la. Este estudo

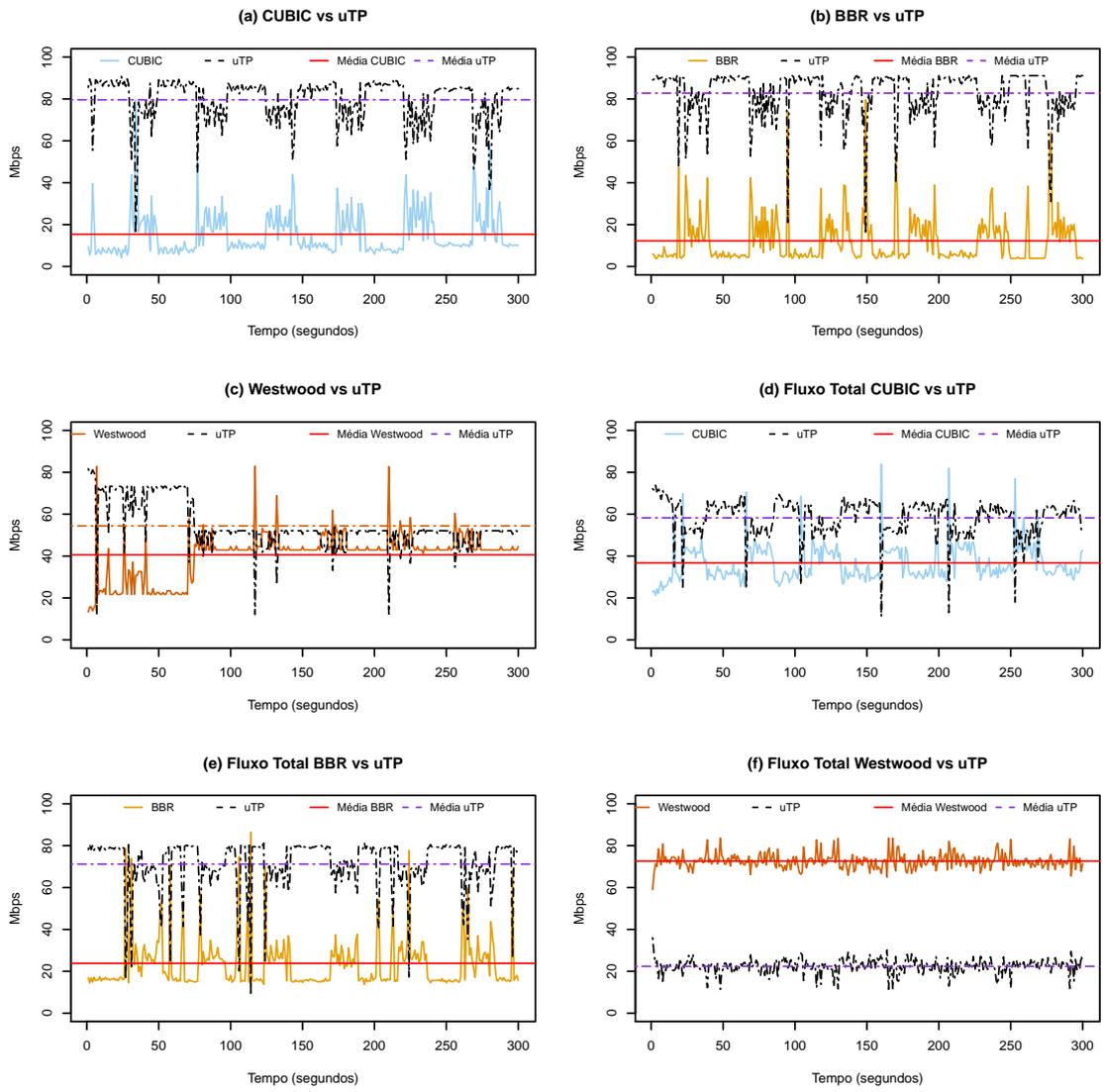


Figura 1. Divisão de largura de banda entre fluxos uTP e TCP

mostra que o uTP, um protocolo que implementa controle de congestionamento sobre o UDP, usa melhor a banda do que o tradicional TCP, propiciando assim transferências mais rápidas de dados pela rede. Como trabalho futuro, pretende-se aprofundar a investigação avaliando o desempenho de aplicações paralelas e distribuídas que utilizem o uTP para transferência de dados.

Referências

- [Afanasyev et al. 2010] Afanasyev, A., Tilley, N., Reiher, P., and Kleinrock, L. (2010). Host-to-host congestion control for TCP. *IEEE Communications Surveys & Tutorials*, 12(3):304–342.
- [Cardwell et al. 2016] Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., and Jacobson, V. (2016). BBR: Congestion-based congestion control. *Queue*, 14(5):50.
- [Dumez 2018] Dumez, C. (2018). qBittorrent – a BitTorrent client in Qt. <https://github.com/qbittorrent/qBittorrent>.
- [ESnet 2018] ESnet (2018). iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool. <https://github.com/esnet/iperf>.
- [Evans et al. 2003] Evans, J. J., Groop, W. D., and Hood, C. S. (2003). Exploring the relationship between parallel application run-time and network performance in clusters. In *IEEE International Conference on Local Computer Networks (LCN)*, pages 538–547.
- [Grieco and Mascolo 2004] Grieco, L. A. and Mascolo, S. (2004). Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control. *ACM SIGCOMM Computer Communication Review*, 34(2):25–38.
- [Ha et al. 2008] Ha, S., Rhee, I., and Xu, L. (2008). CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review*, 42(5):64–74.
- [Kouzes et al. 2009] Kouzes, R. T., Anderson, G. A., Elbert, S. T., Gorton, I., and Gracio, D. K. (2009). The changing paradigm of data-intensive computing. *IEEE Computer*, 42(1):26–34.
- [Norberg 2018] Norberg, A. (2018). Libtorrent manual. <https://www.libtorrent.org/utp.html>.