

Análise do Controle de Congestionamento TCP em *Data Centers* de Nuvens IaaS

Guilherme Xavier¹, Arthur Schuelter¹, Guilherme Koslovski¹

¹Universidade do Estado de Santa Catarina – UDESC – LabP2D

{guilherme.carvalho, arthur.schuelter}@edu.udesc.br
guilherme.koslovski@udesc.br

Resumo. *O compartilhamento justo dos recursos de rede é foco comum no desenvolvimento de protocolos e data centers. Especificamente em nuvens de IaaS, nas quais os inquilinos podem configurar máquinas virtuais, a justiça de compartilhamento é afetada pelos diversos algoritmos para controle de congestionamento no TCP. Assim, o presente trabalho analisa o desempenho dos algoritmos quando executados em um cenário heterogêneo, comum nas nuvens IaaS.*

1. Introdução

As nuvens computacionais que ofertam infraestruturas como serviço são comumente compartilhadas por múltiplos inquilinos. A reserva de recursos ocorre através de máquinas virtuais (MVs) dinamicamente disponibilizadas aos clientes de acordo com a configuração previamente definida. Ou seja, o gerenciamento interno do sistema operacional, bibliotecas e aplicações é realizado pelo contratante. Como resultado, é comum a execução concorrente de versões distintas de sistemas, entre eles, os protocolos de controle de congestionamento do TCP.

Algoritmos de controle de congestionamento inferem informações da rede para aumentar ou diminuir o volume de dados trafegados. Bem como, tratam os possíveis gargalos nos equipamentos de encaminhamento de pacotes, que além da perda dos mesmos, acarretam na diminuição da vazão de tráfego útil, aumento da latência e degradação do serviço oferecido pelo provedor IaaS. No entanto, ao combinar diversos algoritmos, o princípio de justiça de compartilhamento é perdido e as MVs apresentam desempenhos diferentes na ocorrência de congestionamento em *data centers*.

Nesse contexto, o presente trabalho apresenta uma análise sobre a justiça de compartilhamento e desempenho obtido ao combinar diferentes versões dos protocolos para controle de congestionamento. O estudo é realizado sobre o *Mininet* [Karamjeet Kaur 2014], uma ferramenta para prototipação e análise de redes virtualizadas de alto desempenho. Destarte, o artigo está organizado da seguinte maneira: a Seção 2 revisa conceitos importantes da literatura para a contextualização da problemática tratada; enquanto a Seção 3 trata da análise experimental, descrevendo os testes realizados e os resultados obtidos. A Seção 4 é reservada às conclusões e perspectivas de continuidade.

2. Revisão de Literatura

2.1. Algoritmos para Controle de Congestionamento TCP

Dado um ambiente compartilhado, é desejável que todos os usuários sejam mantidos suficientemente satisfeitos pelo serviço recebido. Advindo deste princípio, a justiça é implementada pelo TCP para promover esse equilíbrio, mesmo que existam diferentes algoritmos para controle de congestionamento operantes na rede [Floyd 1994]. No entanto, tais

algoritmos utilizados pelas MVs não são capazes de manter a rede homogênea para todos os usuários, uma vez que, cada mecanismo implementa as características do protocolo da sua maneira. Os proeminentes neste quesito são: *Cubic*, *Reno*, *BBR* e *Vegas*.

O *Cubic* é um algoritmo de crescimento rápido da janela de transmissão, seguindo uma função cúbica. Para promover justiça, estabiliza o crescimento da janela quando este se aproxima do limite da rede. Por sua vez, o *Reno* possui ênfase na reação do algoritmo frente à perda de pacotes, aplicando métodos de recuperação de vazão quando pacotes são perdidos e limitação do crescimento da janela quando a rede está congestionada. O algoritmo *BBR*, desenvolvido pela *Google*, fornece alta vazão e justiça em ambientes congestionados e com alta latência. Por fim, o *Vegas*, diferente das outras implementações, tenta detectar o congestionamento antes de ocorrer a perda de pacotes. Ele estabelece valores de vazão máxima e mínima, executando de maneira eficiente dentro desse intervalo.

2.2. *Explicit Congestion Notification* - ECN

O ECN é uma das alternativas elaboradas para que se reduza a perda de pacotes em um *data center*. O mecanismo notifica os emissores e receptores sobre o congestionamento na rede por meio de marcações nos pacotes transmitidos. Tais marcas sinalizam o congestionamento da rede ao chegar aos pontos finais da conexão, fazendo com que o fluxo de transmissão seja reduzido e a perda de pacotes seja amenizada antes mesmo que ocorra [Bauer et al. 2011].

2.3. Trabalhos Relacionados

Atualmente alguns projetos apresentam propostas para viabilizar a equidade em *data centers*. O AC/DC (*Administrator Control over Data Center*) propõe que os administradores da rede sejam capazes de alterar os algoritmos de controle de congestionamento sem alterar as configurações dos inquilinos da rede [He et al. 2016]. Já o DCTCP (*Data Center TCP*) sugere a utilização de um único algoritmo otimizado no *data center* [Alizadeh et al. 2010]. Assim, cada nó pertencente a rede acredita estar utilizando seu próprio protocolo, por conta disso a rede permanece homogênea. Por fim, o VCC (*Virtualized Congestion Control*) aprimora o funcionamento dos algoritmos e facilita as atualizações por meio dos *hypervisors*. Os quais recebem um algoritmo determinado e distribuem sua tradução entre os *hosts*, fazendo com que componentes legados operem junto aos benefícios de uma nova implementação [Cronkite-Ratcliff et al. 2016].

3. Análise Experimental

3.1. Cenário de Testes

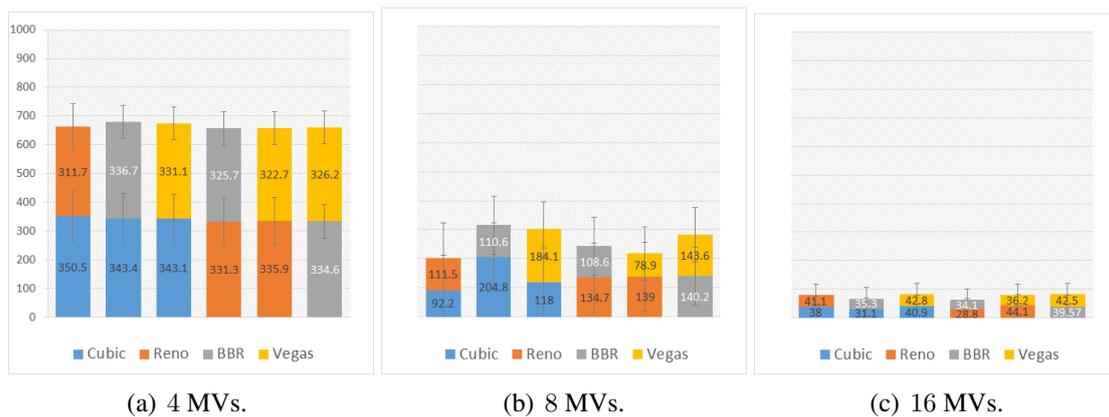
Para simular o cenário de concorrência por recursos em uma nuvem IaaS, um ambiente de testes foi criado utilizando a ferramenta *Mininet*. A topologia adotada foi a de Halteres, ou seja, MVs foram interconectadas por 2 *switches*, onde os enlaces foram configurados com 1 Gbps, sendo que a disputa por recursos ocorre na conexão dos *switches*. Para cada par cliente-servidor condicionado na execução, foi atribuído um algoritmo TCP diferente, dentre os elencados na Seção 2.1. Ainda, foram determinadas duas abordagens de controle de congestionamento. Na primeira, os algoritmos executavam sem o suporte do ECN, enquanto na segunda, ele estava presente junto ao RED (*Random Early Detection*). Um algoritmo que atua no controle de filas do roteador, selecionando pacotes aleatórios para descartar, prevenindo assim, o surgimento de filas. Este, foi configurado segundo a

literatura de [Alizadeh et al. 2010]. O tamanho do *buffer* foi estabelecido em 10^6 , sendo que para o mínimo da extensão da fila de marcação foi dado $9 * 10^4$ e para o máximo foi adicionando 1 a este último valor. Os pacotes permitidos por rajada de descarte foram 61, com probabilidade 1 e tamanho médio do pacote em 1500.

Dada a infraestrutura, 3 casos foram preparados com diferentes composições: a primeira composta por 4 MVs, a segunda por 8 e a última por 16. Optou-se por desenhar essas 3 organizações para analisar o comportamento dos algoritmos em ambientes com configurações distintas, por exemplo, clientes que reservaram MVs replicadas (com o mesmo algoritmo). Por conseguinte, amostras (10 execuções) foram colhidas e os resultados compõem uma média com desvio padrão de aproximadamente 70 Mbps. Vale ressaltar que, para a elaboração dos gráficos elaborados foi utilizado um intervalo de confiança de 95%.

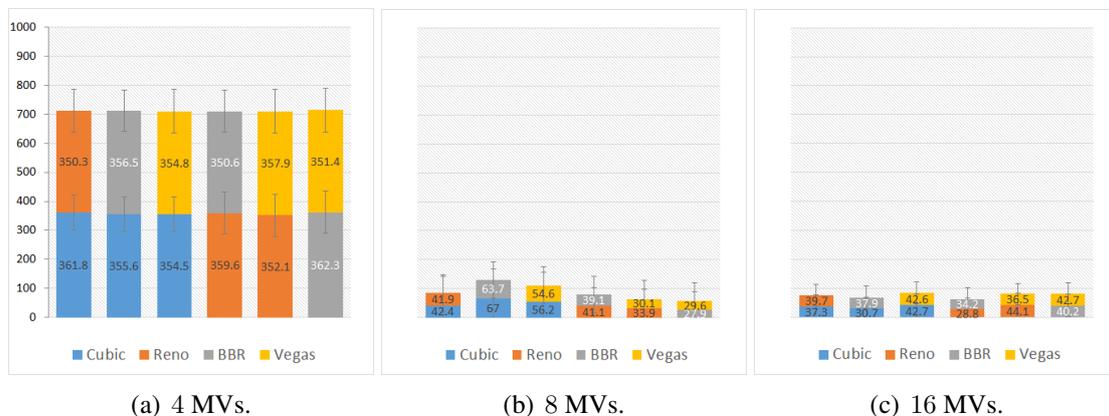
3.2. Discussão dos Resultados

Os resultados para a execução sem suporte ECN e com suporte são apresentados pelas Figuras 1 e 2, respectivamente. Vale ressaltar que, a análise realizada utilizou o emulador *Mininet*, cujos resultados produzidos podem se afastar, levemente, de situações reais.



(a) 4 MVs. (b) 8 MVs. (c) 16 MVs.

Figura 1. Compartilhamento do enlace de gargalo sem suporte ECN.



(a) 4 MVs. (b) 8 MVs. (c) 16 MVs.

Figura 2. Compartilhamento do enlace de gargalo com suporte ECN.

Ao avaliar os dados, nota-se que em uma distribuição com poucas MVs competindo pelo enlace (Figuras 1(a) e 2(a)), a vazão utilizada pelos pares é demasiada-

mente maior se comparada com os valores obtidos com conjuntos maiores (Figuras 1(c) e 2(c)). Uma vez que a quantidade de componentes na infraestrutura influencia a vazão de cada par. Pois, a vazão depende da capacidade dos enlaces da rede e não do número de máquinas virtuais.

Quanto aos algoritmos, as versões *Cubic* e *Reno* proporcionaram uma divisão próxima do ideal, enquanto o *BBR* e *Vegas*, apresentaram dispersões significativas no compartilhamento do canal. Ainda, é possível observar que para a maior parte dos casos com ECN ativo (Figura 2), houve uma aproximação nas métricas. Desse modo, o ideal de justiça foi devidamente visualizado através do comportamento similar dos algoritmos heterogêneos envolvidos. De modo geral, a aplicação de ECN contribuiu para que os recursos em rede fossem distribuídos com maior equilíbrio entre os inquilinos solicitantes.

No entanto, neste experimento, a quantidade de pacotes perdidos não foi devidamente otimizada, dada a agressividade no descarte dos pacotes oriundos da configuração dos *switches*. Em ambientes virtualizados de IaaS, a aplicabilidade de métodos que buscam a justiça são de suma importância quando se objetiva um comportamento homogêneo e eficiente do sistema.

4. Considerações Finais

A análise de justiça realizada sobre os algoritmos de controle de congestionamento indica que aplicação do ECN em uma infraestrutura distribuída, tal qual a de *data centers* de nuvens IaaS, aproximou as taxas de vazão entre as MVs, tornando o sistema mais justo e homogêneo no compartilhamento de recursos disponíveis em rede. Quanto aos trabalhos futuros, experimentos com outras versões de algoritmos para controle de congestionamento são previstos, bem como estudos sobre a virtualização deste gerenciamento.

Agradecimentos: Financiado pela UDESC e FAPESC, sendo desenvolvido no LabP2D.

Referências

- Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and Sridharan, M. (2010). Data center tcp (dctcp). In *Proc. of the SIGCOMM Conference*.
- Bauer, S., Beverly, R., and Berger, A. (2011). Measuring the state of ecn readiness in servers, clients, and routers. In *Proc. of the SIGCOMM Conf. on Internet Measurement Conference, IMC '11*.
- Cronkite-Ratcliff, B., Bergman, A., Vargaftik, S., Ravi, M., McKeown, N., Abraham, I., and Keslassy, I. (2016). Virtualized congestion control. In *Proc. of the SIGCOMM Conference*.
- Floyd, S. (1994). Tcp and explicit congestion notification. *SIGCOMM Comput. Commun. Rev.*, 24.
- He, K., Rozner, E., Agarwal, K., Gu, Y. J., Felter, W., Carter, J., and Akella, A. (2016). Ac/dc tcp: Virtual congestion control enforcement for datacenter networks. In *Proc. of the SIGCOMM Conference*.
- Karamjeet Kaur, Japinder Singh, G. S. (2014). Mininet as software defined networking testing platform.