

Um estudo preliminar sobre o impacto de protocolos para controle de congestionamento em *Edge-Cloud Continuum*

Nicolas Keiji Cattani Sakashita¹, Guilherme Piêgas Koslovski²

¹Universidade do Estado de Santa Catarina (UDESC), Joinville, SC, Brasil

Resumo. *Edge-Cloud Continuum (ECC) é uma arquitetura que combina o conceito de Cloud e Edge Computing, melhorando o desempenho dos serviços distribuídos. Essa combinação reduz latência, tempo de processamento e consumo de energia. A arquitetura ECC é dependente de interconexões de redes, e herda esses desafios relacionados a gerência e aos recursos compartilhados. Ou seja, assim como em nuvens computacionais, as aplicações ainda compartilham diversos enlaces de comunicação. O presente trabalho investiga o impacto dos algoritmos de congestionamento Cubic e Bottleneck Bandwidth and Round-trip propagation time (BBR) no desempenho de aplicação ECC.*

1. Introdução

A Computação em Nuvem é um modelo que traz acesso a rede de forma onipresente, conveniente e sob demanda a uma variedade de recursos computacionais que podem ser rapidamente instanciados com o mínimo de esforço de gerenciamento e serviço [Mell et al. 2011]. Por suas vantagens, grande parte das aplicações, dados, serviços e sistemas computacionais foram movidos para a nuvem, centralizando a oferta de serviços. A centralização negligencia a oportunidade de utilizar todo o poder computacional dos dispositivos modernos distribuídos nas bordas. Nesse contexto, *Edge Computing* é um paradigma que utiliza os recursos disseminados nessas bordas, mantendo as principais vantagens da nuvem, como suporte de infraestrutura, mas deixa o controle e as decisões nas bordas, trazendo a computação mais próxima do ser humano [Garcia Lopez et al. 2015]. A combinação da computação em nuvem e nas bordas levou a definição de *Edge-Cloud Continuum* (ECC) [Bittencourt et al. 2018]. Embora revolucionário, a literatura especializada carece de informações sobre o impacto dos algoritmos de controle de congestionamento em aplicações na arquitetura ECC, visto que a infraestrutura permanece compartilhada, assim como ocorre em nuvens computacionais. O presente trabalho é a continuação do estudo [Sakashita et al. 2023], acrescentando novos cenários e protocolos de controle. Dessa vez, os algoritmos Cubic [Ha et al. 2008] e BBR [Cardwell et al. 2017], comumente utilizados nos servidores, são analisados em cenários de ECC. O conteúdo está organizado da seguinte forma: A Seção 2 apresenta o cenário experimental, ambiente e ferramentas. A Seção 3 mostra os resultados, enquanto a Seção 4 conclui o trabalho e apresenta perspectivas de trabalhos futuros.

2. Cenário experimental

Os testes foram realizados em 4 configurações distintas para analisar a aplicabilidade do BBR e do Cubic em cenários de ECC, conforme resumido na Figura 1: *Client-Cloud*, *Client-Edge*, *Client-Edge-Cloud w/o* (sem output de dados) e *Client-Edge-Cloud w* (com output de dados). Tais cenários representam as diferentes estruturas de redes utilizadas

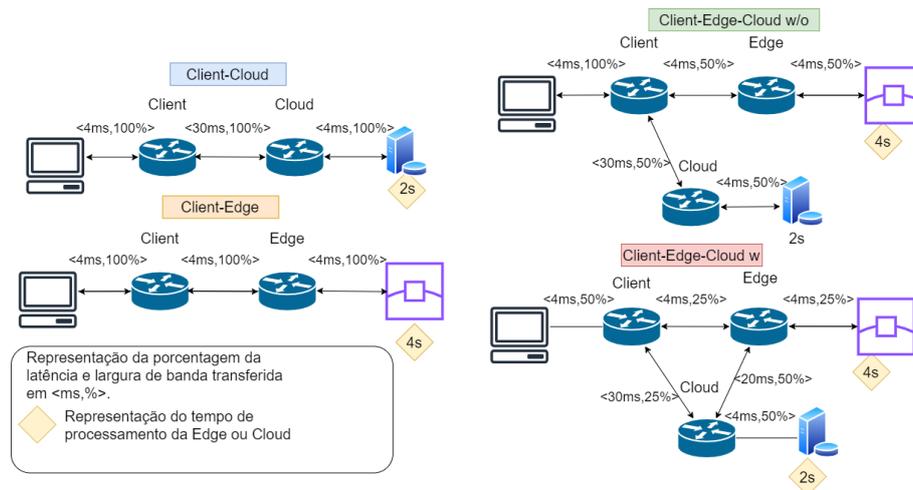


Figure 1. Cenário de testes do experimento

atualmente no desenvolvimento de aplicações, ou seja, emulam os cenários reais. *Client-Edge-Cloud w* e *w/o* nas cores vermelha e verde, representam a arquitetura ECC, que aloca e transmite simultaneamente uma quantidade de dados do cliente para a nuvem e a *edge*. Em cada enlace é transferido de 25% a 50% dos dados. Nos *Client-Cloud* e *Client-Edge* representados pelas cores azul e laranja, 100% do volume de dados é transferido em apenas um enlace. Essas porcentagens de dados transmitidos em cada enlace trará uma diferença considerável na velocidade de transferência, o que impacta diretamente na qualidade de serviço Quality-of-Service (QoS) da rede. Em cada teste foram utilizados 16 clientes simultâneos em cada um dos cenários, sendo que cada cliente transfere um volume de dados entre 1 e 100MB. A aplicação recebe os dados, processa e devolve para o cliente o mesmo volume de dados originalmente recebido. Como protocolos experimentais foram utilizados (a) o Cubic, (b) o BBR e (c) uma mistura entre Cubic e BBR, usando o BBR apenas em volumes de dados inferiores a 10MB, entre a *edge* e o cliente. A escolha dos protocolos se deve ao fato de que o Cubic é o algoritmo de congestionamento padrão atualmente, e o BBR vem se tornando cada vez mais utilizado por sua alta taxa de convergência e vazão para baixas filas.

2.1. Ferramentas e implementação

Os cenários foram implementados utilizando o emulador Mininet [Lantz et al. 2010] em conjunto com o controlador Ryu¹, sendo implementados cada um, em uma máquina virtual (VM) com 4GB de RAM, 4CPUs e no GNU/Linux Ubuntu 20.04. Para representar cada enlace, sub-rede e rotas estáticas, as regras foram configuradas pelo controlador Ryu. Para gerar o fluxo de dados, foi utilizado a ferramenta iPerf3, que permite a escolha do algoritmo de controle de congestionamento. Foi utilizada a métrica de Flow-Completion Time (FCT) para cada enlace, que representa o tempo em que uma conexão demora para ser completamente finalizada. O FCT representa o desempenho em um nível de aplicação, contabilizando todo o tempo de transferência e de retransmissão. O FCT também é uma das mais importantes métricas para o usuário final, pois ela é percebida diretamente pela velocidade que as informações chegam [Dukkipati and McKeown 2006]. Os dados foram coletados no iPerf3 e representados em Cumulative Distribution Functions (CDFs).

¹Controlador Ryu. Disponível em: <https://ryu-sdn.org>

3. Resultados experimentais

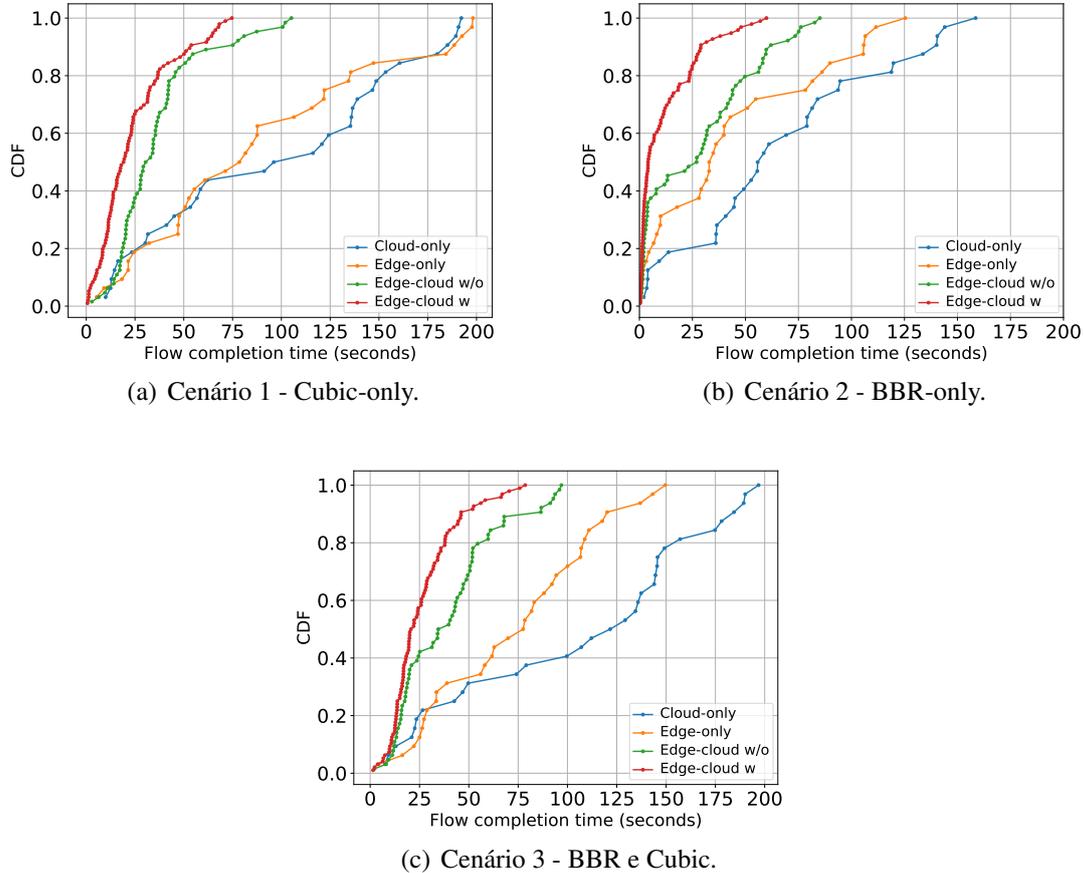


Figure 2. Resultados dos cenários experimentais.

As Figuras 2(a)-(c) apresentam os resultados dos experimentos em categorias: (a) Cubic-only, (b) BBR-only e (c) uma mistura de Cubic e BBR. Em todas as categorias o *Edge-Cloud w* conseguiu o menor tempo de completude do FCT, seguido pelo *Edge-Cloud w/o*. Para o *Cubic-only* o *Edge-Cloud w* contempla 70% dos casos para 25 segundos, enquanto o *Edge-Cloud w/o* contempla 40% dos casos para esse mesmo tempo. Já para a *edge* e o *cubic only*, esse tempo se da a apenas 20% dos casos. Os resultados com BBR obtiveram desempenho superiores em todas as categorias em relação ao Cubic: 20 segundos resulta em 80% dos casos para *Edge-Cloud w* e perto de 50% para *Edge-Cloud w/o*, sendo que de 20 a quase 40% para o Cubic e edge only. É interessante mencionar que o BBR também demonstrou-se mais eficiente com grandes volumes de dados, mesmo não sendo uma das características do BBR. O BBR nas conexões *Edge-Cloud* conseguiu em perto de 1 segundo por volta de 20% das conexões, e acima de 60% das conexões terminaram dentro de 10 segundos. Por último, a proposta de combinar Cubic e BBR resultou em resultados semelhantes ao Cubic-only. Isso se deve pelo fato de ter sido utilizado Cubic na maioria das conexões, e apenas quando o volume de dados era 10MB ou inferior, no enlace entre o cliente e a edge, o BBR era utilizado. Como esperado, as categorias do *edge* e *cloud only* demoraram mais tempo para completar a transferência de dados. Isso se deve pois, nas categorias de *Edge-Cloud* existem mais nós sendo processados ao mesmo

tempo, dividindo o volume de dados entre esses enlaces, trazendo à tona as vantagens do BBR. Esses resultados destacam a superioridade do BBR em comparação com o Cubic, principalmente com pequenos volumes de dados para o FCT em arquiteturas ECC.

4. Conclusão

O aumento de novas tecnologias de comunicação, dispositivos móveis, Internet of Things (IoT) e novas arquiteturas, fazem com que surjam novas demandas computacionais para processamento, armazenamento de dados, latência e largura de banda. Uma das formas de solucionar é utilizando a arquitetura ECC, que traz o processamento dos dispositivos mais próximo dos usuários. Embora revolucionário, as aplicações baseadas em ECC ainda herdam os desafios que gerenciamento das redes distribuídas. Portanto a arquitetura da rede e os protocolos de comunicação impactam no desempenho final da aplicação já que não são componentes isolados mas sim, partes que compõem toda a aplicação. Esse trabalho investigou o impacto dos algoritmos de controle de congestionamento Cubic e BBR, em termos de FCT para aplicações simuladas em cenários de ECC. BBR se mostrou superior em FCT em relação ao Cubic. Os resultados mostraram que a seleção de um algoritmo de congestionamento é crucial para conseguir mais vantagens dos benefícios oferecidos pela ECC. Como trabalhos futuros, aplicações em larga escala e o impacto da mobilidade do usuário vão ser investigados.

Agradecimento: Este trabalho recebeu apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa e Inovação (FAPESC), desenvolvido no Laboratório de Processamento Paralelo e Distribuído (LabP2D).

References

- Bittencourt, L., Immich, R., Sakellariou, R., Fonseca, N., Madeira, E., Curado, M., Villas, L., DaSilva, L., Lee, C., and Rana, O. (2018). The internet of things, fog and cloud continuum: Integration and challenges. *Internet of Things*, 3:134–155.
- Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., and Jacobson, V. (2017). Bbr: Congestion-based congestion control. *Communications of the ACM*, 60(2):58–66.
- Dukkipati, N. and McKeown, N. (2006). Why flow-completion time is the right metric for congestion control. *ACM SIGCOMM Computer Communication Review*, 36(1):59–62.
- Garcia Lopez, P., Montessor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P., and Riviere, E. (2015). Edge-centric computing: Vision and challenges.
- 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.
- Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, pages 1–6.
- Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.
- Sakashita, N. K., Albuquerque, P. R., and Koslovski, G. P. (2023). O impacto dos algoritmos de controle de congestionamento em aplicações edge-cloud continuum. In *Anais da XX Escola Regional de Redes de Computadores*, pages 19–24. SBC.