



VIRTUALIZAÇÃO DE CONTROLE DE CONGESTIONAMENTO DO TCP EM NUVENTS COMPUTACIONAIS

Lucas Litter Mentz¹, Arthur Felipe Herdt Schuelter², Guilherme Piegas Koslovski³

¹ Acadêmico do Curso de Bacharelado em Ciência da Computação – CCT – bolsista PIVIC/UDESC.

² Acadêmico do Curso de Bacharelado em Ciência da Computação – CCT.

³ Orientador, Departamento de Ciência da Computação – CCT – guilherme.koslovski@udesc.br.

Palavras-chave: Controle de congestionamento. Justiça de tráfego em redes. Nuvem computacional.

Virtualização de controle de congestionamento – ou VCC – em TCP é a base de estudo para o projeto que estamos estudando e baseia-se no uso do ECN (*Explicit Congestion Notification*) – *flag* no cabeçalho TCP que sinaliza que a *stream* (fluxo de dados TCP) deve desacelerar seu tráfego de dados – para prevenir o fenômeno denominado *bufferbloat*. *Bufferbloat* ocorre quando um *buffer* (acumulador) de um switch/roteador está cheio e por isso rejeita (*drop*) os pacotes seguintes até ter espaço para acomodar mais pacotes, ocasionando perda de pacotes e retransmissão nas *streams* envolvidas, podendo agravar a situação de *bufferbloat*.

Roteadores e switches capazes de ECN são os responsáveis por sinalizar às *streams* que seus *buffers* estão próximos da capacidade máxima por meio da sinalização da *flag* ECN, causando com que o receptor da *stream* replique a *flag* para o remetente que por sua vez reduz sua janela de envio para reduzir a taxa de transmissão. O protocolo que rege a seleção das *streams* a serem avisadas pelo ECN é o RED (*Random Early Detection*) que é parametrizado e configurável para refletir a melhor opção em um determinado cenário. O RED é um algoritmo de seleção probabilístico que faz a seleção aleatória de *streams* para sinalizar com ECN baseado em parâmetros como ocupação mínima e máxima dos *buffers* entre outros.

Buscamos com esse projeto analisar as implicações de, ao invés de habilitar o ECN e selecionar um determinado algoritmo TCP nos sistemas operacionais das máquinas virtuais envolvidas, habilitar a virtualização de controle de congestionamento – VCC – nos hospedeiros de virtualização, mais especificamente no controlador da implementação de rede virtualizada utilizada na nuvem computacional – neste projeto, o Open vSwitch. Testes preliminares foram feitos analisando um cenário específico:

- uma rede com topologia Dumbbell (fig. 1) com um determinado número de dispositivos se comunicando pelo gargalo de 1Gbps da rede;
- os dispositivos identificados com “s#” recebem dados do dispositivo “c#” correspondente;
- os dispositivos envolvidos são simulados com uso de Docker e a rede é gerenciada com uso do Mininet – a solução que envolve Mininet e Docker que usamos é o projeto de código aberto Containernet;
- ao invés de uma implementação de VCC no Open vSwitch, usamos um *patch* no kernel Linux 3.19.0 que implementa ECN a nível de sistema operacional compatível com TCP New Reno;

- o tráfego é gerado com uso do software utilitário iperf.

Obtivemos resultados satisfatórios com o uso da virtualização de controle de congestionamento no cenário apresentado, tendo uma distribuição quase ideal entre as *streams* que usaram ECN nativo e as que usaram ECN simulado com VCC (fig. 2).

Para prosseguir com o projeto, vamos substituir o iperf pelo benchmark MPI NAS para ter uma medida mais comparável ao usar benchmarks bastante conhecidos. O benchmark MPI NAS cria vários cenários de computação distribuída utilizando um número definido de *hosts* e mede o tempo de execução. Planejamos comparar resultados de execuções desse benchmark com 16 *hosts* primeiramente em três casos: sem ECN, com ECN nativo e com VCC na rede. Em seguida iremos repetir os testes, mas incluir um *stream* iperf para competir com a execução do MPI NAS nos mesmos três casos.

Como objetivo final do projeto, planejamos desenvolver um módulo para o Open vSwitch que implementa a virtualização do controle de congestionamento, permitindo assim usar o VCC em qualquer solução que usa Open vSwitch ao invés de estar preso ao kernel Linux 3.19.0 com o *patch* usado.

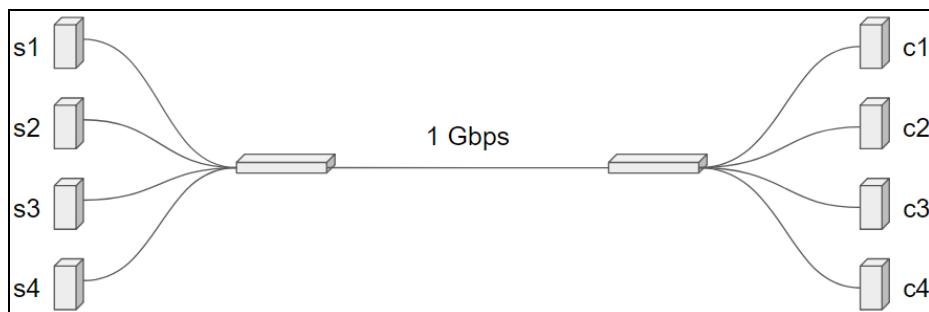


Fig. 1 Topologia Dumbbell. Próprio autor (2018).

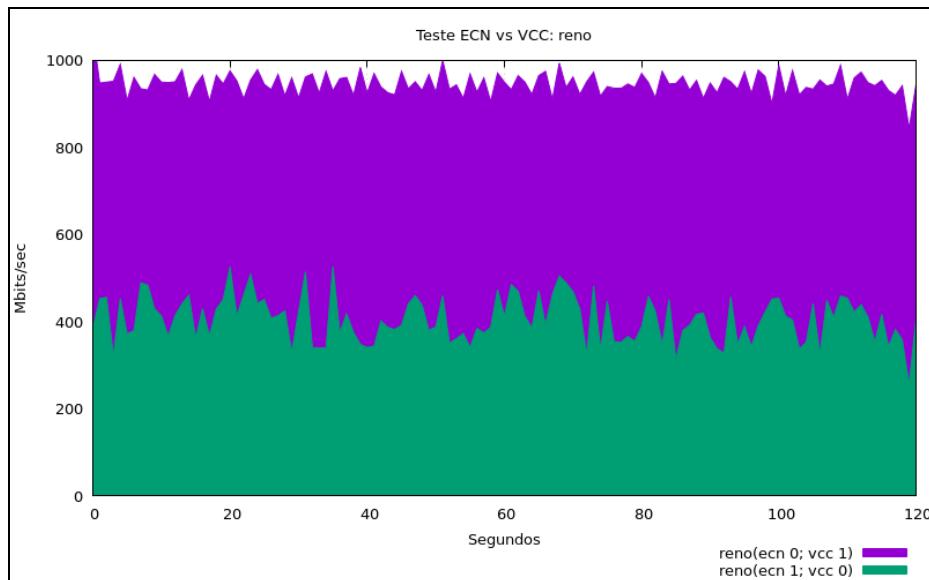


Fig. 2 Resultados de VCC e ECN com TCP New Reno com contêineres. Próprio autor (2018).

Referência:

CRONKITE-RATCLIFF, Bryce et al. Virtualized congestion control. In: **Proceedings of the 2016 ACM SIGCOMM Conference**. ACM, 2016. p. 230-243.