

# Uma Ferramenta para Evitar Enlaces Congestionados em Redes Definidas por *Software*

Gustavo Diel<sup>1</sup>, Anderson Marcondes<sup>2</sup>, Guilherme Koslovski<sup>1,2</sup>

<sup>1</sup>Ciência da Computação – UDESC

<sup>2</sup>Programa de Pós-graduação em Computação Aplicada – UDESC

{gustavodiel, anderson.marcondes}@gmail.com, guilherme.koslovski@udesc.br

**Resumo.** *Por ser um recurso ubíquo e compartilhado, o congestionamento em redes de computadores é comum. Utilizando recursos de redes definidas por software, o presente trabalho apresenta uma solução para diminuir o impacto de enlaces congestionados através do redirecionamento de fluxos de dados. A ferramenta proposta monitora a carga nos enlaces da rede selecionando caminhos com menor congestionamento.*

## 1. Introdução

Com o aumento dos dispositivos que fazem uso de redes, um problema é o surgimento de gargalos entre os equipamentos de encaminhamento de tráfego. Tais gargalos podem ocasionar aumento de latência e diminuição da vazão de dados. Aplicações que geram grandes quantidades de dados têm seu desempenho comprometido. Muitas vezes, o congestionamento acontece pelo roteamento incorreto dos pacotes ou pela falta da possibilidade de priorizar certas conexões.

As Redes Definidas por *Software* (SDNs) introduziram a separação dos planos de controle e de dados, aumentando as formas de gerenciamento do tráfego da rede [McKeown et al. 2008]. Esta separação retira do *software* embutido no dispositivo de encaminhamento a exclusividade de gerenciar os caminhos dos dados dentro da rede, criando, assim, a possibilidade de um controlador externo à rede exercer esta tarefa.

O controlador possui diversas informações centralizadas sobre a rede. Assim, é possível identificar todos os caminhos entre qualquer dispositivo, bem como detectar quais caminhos estão congestionados. Com essa informação, pode-se definir políticas especiais para cada tráfego na rede, melhorando o desempenho sem alterar o *hardware*. Para se realizar a conexão lógica entre os dispositivos e o controlador, é utilizado o protocolo OpenFlow, que define as métricas, o conjunto de dados e as instruções a serem suportadas por *switches*. Em SDN, a caracterização de fluxos de dados combina informações das camadas, 2, 3 e 4, como por exemplo, endereços *Medium Access Control* (MAC), *Internet Protocol* (IP) e portas.

Desta forma, o presente trabalho propõe uma ferramenta para reconfiguração dinâmica de rotas de encaminhamento de fluxos SDN. Baseado no estado atual da rede, a ferramenta periodicamente seleciona os caminhos com menor congestionamento para direcionar o tráfego. O restante deste trabalho está organizado na seguinte forma: a Seção 2 descreve a ferramenta proposta. A análise experimental é discutida na Seção 3. Os trabalhos relacionados são discutidos na Seção 4, enquanto as considerações finais são apresentadas na Seção 5.

## 2. Ferramenta Proposta

Utilizando um controlador SDN, um administrador possui maior controle sobre os dispositivos de comunicação. Em específico, o controlador escolhido, OpenDaylight [Medved et al. 2014] possui uma REST API que possibilita a qualquer aplicativo a manipulação dos equipamentos da rede. Sobretudo, OpenDaylight possui diversas funcionalidades implementadas para monitoração da carga dos *switches*.

Baseado em OpenDaylight, foi desenvolvido uma ferramenta que monitora a carga da rede, tentando diminuir os enlaces congestionados. O seu funcionamento consiste em analisar a configuração da topologia e a quantidade de bytes que atravessaram determinado *switch*, para que regras possam ser geradas. Tais regras podem garantir funcionalidades como uma certa largura de banda (dentro dos limites do próprio *hardware*) ou até mesmo a seleção do menor caminho entre os dispositivos.

O usuário da ferramenta decide quais fluxos de dados devem ser priorizados, e qual deve ser a prioridade (latência ou largura de banda). Quando selecionada a configuração de largura de banda, o usuário pode informar o valor mínimo desejado, ou solicitar por enlaces não congestionados. A primeira abordagem realiza uma reserva estática, enquanto a segunda monitora a rede buscando a melhor configuração possível para aquele instante. A ferramenta foi desenvolvida em *Python* devido à sua rápida manipulação e vasta gama de recursos para conexão com a interface REST do controlador.

## 3. Análise Experimental

Para compor o cenário experimental, o controlador OpenDaylight versão Beryllium foi utilizado, com o padrão OpenFlow versão 1.4. Para gerar a topologia e toda a infraestrutura da rede virtual, foi utilizada a ferramenta Mininet [de Oliveira et al. 2014], que cria *switches* e dispositivos virtuais, conectados em uma rede isolada. A topologia utilizada é apresentada na Figura 1. É possível observar a existência de um equipamento atuando como controlador, além dos clientes que geram e recebem o tráfego de dados. A rede foi hospedada em um equipamento enquanto o controlador foi posicionado externamente. Os 2 hospedeiros eram *quad-core* AMD Phenom II X4 2.8 GHz, 4 GB de RAM, 500 GB de armazenamento (SATA 7200 rpm) e SO Ubuntu 14.04-64bit.

Para gerar a carga de trabalho foi utilizada a ferramenta *iperf* [Tirumala et al. 2005]. Dois pares de comunicação foram compostos: entre os clientes 1 e 4 (H1 e H4 respectivamente), e entre os clientes 2 e 3 (H2 e H3 respectivamente). O *iperf* quantifica a máxima largura de banda disponível em uma rede. No experimento, utilizou-se comunicação com o protocolo TCP. Os clientes 3 e 4 serviram como servidores para os clientes 2 e 1, respectivamente. Para garantir a confiabilidade dos resultados, dez rodadas de testes foram feitas. Cada teste foi executado por aproximadamente 20 segundos, e a cada segundo o *iperf* retorna a largura de banda máxima alcançada entre o cliente e o servidor.

Dois cenários foram executados, sendo o primeiro sem qualquer mudança na rede (identificado como configuração padrão), e o segundo com a ferramenta proposta (identificado como técnica aplicada). Para cada cenário os pares cliente-servidor foram executados isoladamente e depois de forma concorrente, introduzindo o congestionamento na rede. Os resultados podem ser visualizados na Figura 2 (média e o desvio padrão).

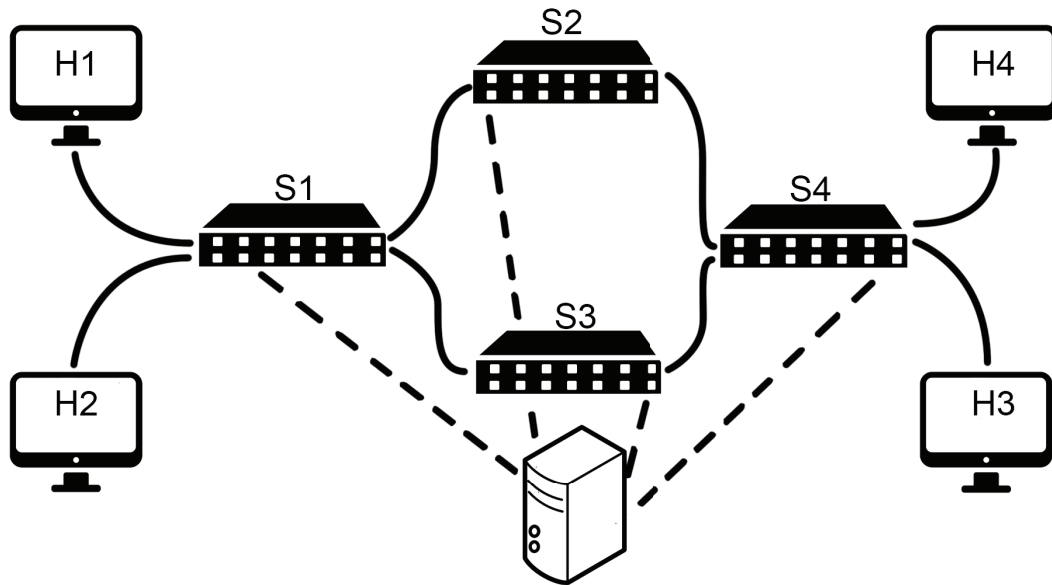


Figura 1. Topologia personalizada gerada pelo Mininet. Traços contínuos são conexões físicas, tracejados são conexões lógicas.

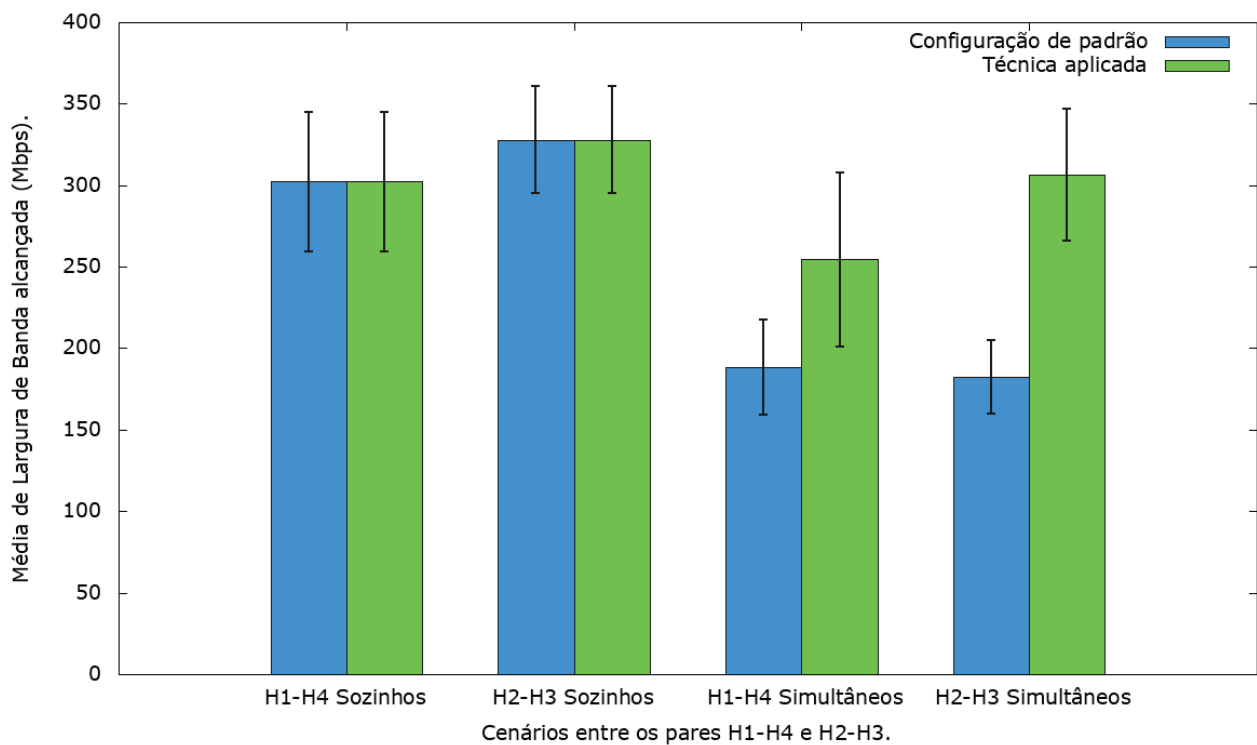


Figura 2. Resultados do *iperf* para os cenários executados.

Quando não existe ocorrência de congestionamento na rede, os pares cliente-servidor observam uma vazão (em Mbps) equivalente. Entretanto, com a utilização da ferramenta proposta, é possível observar que na execução simultânea, a ferramenta permitiu a obtenção de maiores vazões quando comparada a configuração padrão. A diferença ocorre pela utilização do segundo caminho (Figura 1) para encaminhamento de fluxos.

## 4. Trabalhos Relacionados

O uso do paradigma das redes definidas por *software* para melhorar o desempenho de aplicações já é uma realidade. Aplicações multimídia, como o YouTube [Jarschel et al. 2013], já conseguem ter o seu desempenho aprimorado com o uso de SDN e entregam uma experiência para usuário mais satisfatória do que somente quando utilizadas técnicas tradicionais. Wang e outros [Wang et al. 2016] propuseram um novo paradigma de rede, o *Application Drive Network* (ADN), que provê serviços diferenciados sob demanda. A ferramenta proposta neste trabalho pode ser conectada com ADN para configurar o encaminhamento de fluxos utilizando os dados monitorados.

Fontes e outros [Fontes et al. 2015] desenvolveram uma ferramenta baseada no Mininet, chamado Mininet Wifi, que permite utilizar recursos de rede sem fio para qualquer cenário possível de ser feito com o Mininet. Foi proposto o uso da ferramenta para permitir a realização de experimentos que cheguem o mais próximo da realidade em um ambiente virtualizado. Por fim, os trabalhos relacionados reforçam que aplicações podem ter suas métricas de desempenho melhoradas com o uso do paradigma das SDN.

## 5. Considerações Finais

Enlaces congestionados afetam o desempenho das aplicações hospedadas. Utilizando uma rede SDN, é possível contornar esse problema, diminuindo o congestionamento gerado pelas diversas aplicações na rede. Como a ferramenta é desenvolvida à parte, executando em um computador que não precisa estar diretamente conectado na rede em si, é possível a integração deste serviço com qualquer configuração SDN. Como trabalhos futuros, a ferramenta será integrada a propostas específicas para otimização de aplicações.

**Agradecimentos:** o trabalho foi financiado pelos editais da UDESC e desenvolvido no LabP2D.

## Referências

- de Oliveira, R. L. S., Schweitzer, C. M., Shinoda, A. A., and Prete, L. R. (2014). Using mininet for emulation and prototyping software-defined networks. In *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, pages 1–6. IEEE.
- Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M. A. S., and Rothenberg, C. E. (2015). Mininet-wifi: Emulating software-defined wireless networks. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 384–389.
- Jarschel, M., Wamser, F., Hohn, T., Zinner, T., and Tran-Gia, P. (2013). Sdn-based application-aware networking on the example of youtube video streaming. In *2013 Second European Workshop on Software Defined Networks*, pages 87–92.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- Medved, J., Varga, R., Tkacik, A., and Gray, K. (2014). Opendaylight: Towards a model-driven sdn controller architecture. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*.
- Tirumala, A., Qin, F., Dugan, J., Ferguson, J., and Gibbs, K. (2005). Iperf: The tcp/udp bandwidth measurement tool.
- Wang, Y., Lin, D., Li, C., Zhang, J., Liu, P., Hu, C., and Zhang, G. (2016). Application driven network: Providing on-demand services for applications. In *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference, SIGCOMM '16*, pages 617–618, New York, NY, USA. ACM.