

Avaliando recursos de hardware e software em redes definidas por software: Um estudo de caso

Nadyan Suriel Pscheidt¹, Adriano Fiorese¹

¹Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – Joinville – SC – Brasil

nadyan.suriel@gmail.com, adriano.fiorese@gmail.com

Resumo. *Nas redes definidas por software, os planos de dados e de controle não estão mais estritamente acoplados no mesmo hardware. Assim, neste trabalho, são apresentados alguns testes desenvolvidos com o intuito de verificar o desempenho tanto do plano de dados, testando a capacidade de transferência do switch, quanto o plano de controle, testando a capacidade do controlador de manipular fluxos nas tabelas de fluxo do switch.*

1. Introdução

O avanço e crescimento das redes de computadores as tornaram peça chave em diversos cenários, sejam eles empresariais, universitários e de pesquisa. Esse crescimento afetou também a forma como as redes são gerenciadas, tornando essa tarefa mais complexa do que costumava ser. Este fato fica evidente com a quantidade de dispositivos conectados crescendo significativamente, o que aumenta também a quantidade de dados trafegados na rede, comprometendo seu desempenho.

Para auxiliar na gerência e desempenho da rede, surgiram as chamadas redes definidas por *software*, que dissociam o plano de dados do plano de controle. O plano de dados tem a tarefa de encaminhar os pacotes de dados de acordo com uma tabela de fluxos, que ditará as regras de envio. Assim, o plano de dados é composto basicamente pelo *switch* SDN.

Já o plano de controle é responsável por controlar e administrar as tabelas de fluxo, criando novas regras para o caso de pacotes desconhecidos que cheguem ao controlador, assim como enviar comandos para o plano de dados. O papel de plano de controle é realizado por um *software* externo chamado controlador, que terá sua comunicação com o plano de dados efetuada através de um protocolo. O *OpenFlow* [Lara et al. 2014], por exemplo, é um dos protocolos mais conhecidos na área e possui licença aberta. Com este controlador é possível modificar, da forma desejada, as tabelas de fluxo do *switch* que está sendo configurado, explicitando como cada pacote deve ser roteado através da topologia da rede.

Existem alguns trabalhos demonstrando o desempenho de redes definidas por *software* em diversos cenários. Em [Bianco et al. 2010] é estudada uma rede definida por *software* em um ambiente virtual analisando seu *throughput* modificando a quantidade de fluxos nas tabelas de fluxos, assim como os tamanhos dos pacotes. Já os experimentos conduzidos em [Tootoonchian et al. 2012] comparam o desempenho entre quatro controladores SDN utilizando um *framework* denominado *cbench*, desenvolvido pelos autores.

Neste trabalho, será analisado o desempenho, em termos da largura de banda disponibilizada, do *switch* modelo N3048P da marca Dell™, que possui suporte ao protocolo *OpenFlow* versão 1.3 atuando em uma rede definida por *software*, e também o desempenho do controlador dessa rede SDN. O controlador compatível com o *hardware* em teste é o **OpenDayLight** versão Helium [OpenDaylight Foundation 2016], desenvolvido em linguagem Java®, indicado pelo próprio fabricante e que foi utilizado para o desenvolvimento dos testes de largura de banda. Juntamente com os testes no *hardware*, foram executados testes de estresse no controlador.

Na Seção 2 serão descritos os testes realizados e os resultados obtidos, assim como o ambiente no qual os testes foram desenvolvidos e a topologia na qual os equipamentos estavam dispostos. Por fim, a Seção 3 concluirá o trabalho, indicando alguns trabalhos e experimentos que podem dar continuidade aos aqui apresentados.

2. Desenvolvimento dos experimentos

Nesta Seção são apresentados os testes de desempenho executados no equipamento mencionado operando em uma rede definida por *software*, com seu controlador atuando como plano de controle externamente. Também serão apresentados testes desenvolvidos para verificar a capacidade do controlador no quesito capacidade de manipulação da tabela de fluxos. O objetivo é estressar o controlador adicionando fluxos às tabelas de fluxos do *switch* para verificar alterações de comportamento ou eventuais falhas.

2.1. Ambiente de teste

O ambiente no qual os testes foram realizados é controlado, situado no laboratório LabP2D (Laboratório de Processamento Paralelo e Distribuído) da Universidade do Estado de Santa Catarina. O *switch* foi conectado a três máquinas, sendo que uma das máquinas foi utilizada como controlador, formando a topologia ilustrada na Figura 1.

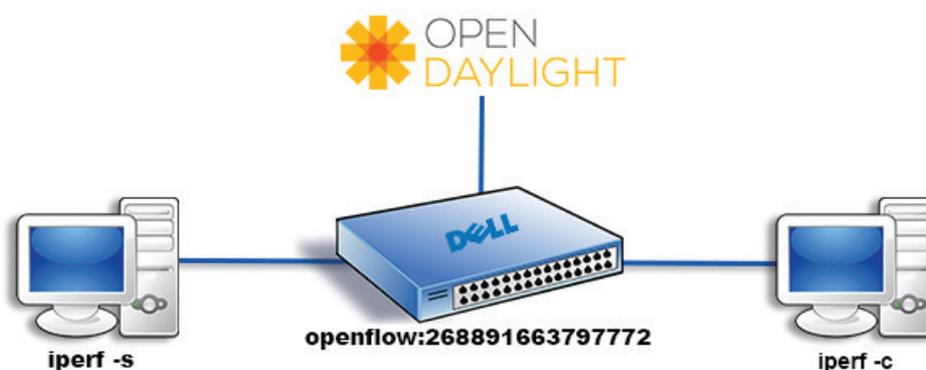


Figura 1. Topologia da rede

Os testes de largura de banda para a mensuração do *throughput* da rede foram executados aplicando a ferramenta **iperf** versão 3.0.11, que gera a maior quantidade possível de tráfego entre duas máquinas em determinado período de tempo. Uma das máquinas conectadas ao *switch* foi configurada pelo **iperf** para ser o servidor através do comando **\$ iperf -s**. O outro *host* também conectado ao *switch* atuou como o gerador de tráfego, utilizando o comando **\$ iperf -c <ip do servidor> -l <tamanho do pacote> -t <tempo de teste>**. As máquinas utilizadas possuem processador AMD Phenom II operando a

2.8Ghz com 4GB de memória RAM, com sistema operacional Ubuntu 14.04 LTS. A conexão entre *hosts* e *switch* é feita através de cabeamento Gigabit Ethernet.

2.2. Testes de largura de banda

Para testar a largura de banda do *switch*, foram executados os comandos *iperf*. Cada mensuração durou 60 segundos, configurada com o parâmetro *-t*, enquanto o tamanho do pacote foi modificado a cada interação com o parâmetro *-l*. A sequência de testes foi a seguinte: pacote de 64, 96, 128, 256, 512, 1024, 1500 e 2048 bytes. Cada teste foi repetido dez vezes, para obtenção da média. Os resultados estão ilustrados na Figura 2.

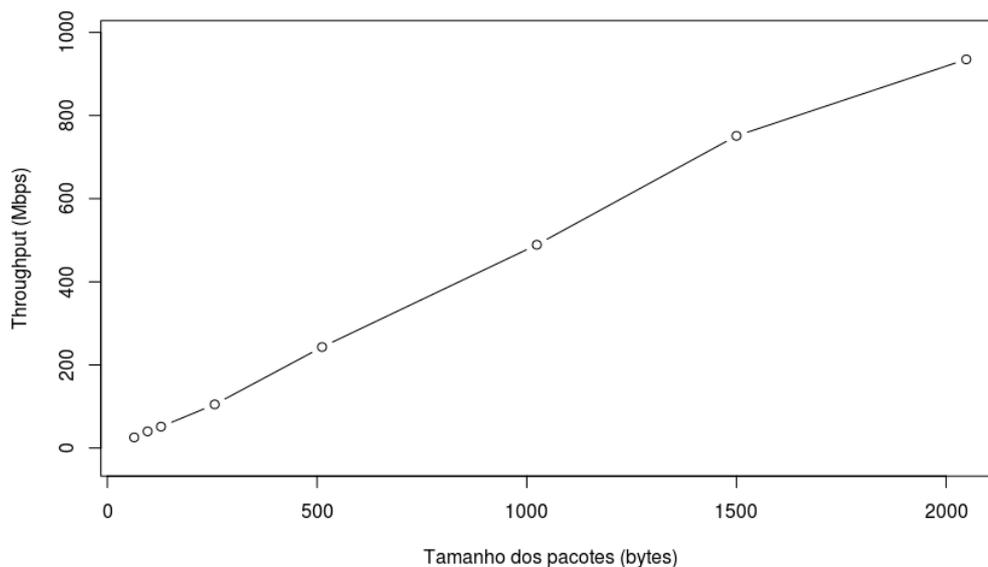


Figura 2. *Throughput* para cada tamanho de pacote em megabits por segundo

Como pode ser observado, o consumo de largura de banda do *switch* teve um crescimento linear. O valor obtido com pacotes de 2048 bytes é próximo ao valor máximo fornecido pelos equipamentos utilizados, que é de cerca de 1Gb/s. Como esperado, os pacotes de maior tamanho demonstraram um maior *throughput*. Isso se deve pelo fato de que pacotes menores são mais custosos por precisarem de mais operações envolvendo cabeçalhos para encapsulamento [Bianco et al. 2010].

2.3. Testes de estresse no controlador

Muito importante para as redes definidas por *software* é a capacidade do controlador gerenciar os fluxos através dele inseridos. Esses testes foram executados pois uma quantidade excessiva de fluxos sendo tratados pelo controlador e armazenados nas tabelas de fluxo do *switch* pode causar quedas de desempenho na rede devido a sobrecarga.

As tabelas de fluxos possuem campos, utilizados para classificar e reconhecer um determinado tipo de fluxo de dados. Esses campos possuem informações de cada fluxo, como IP de origem, IP de destino, portas de origem e destino, tipo de conexão *Ethernet*, assim como qual ação será tomada quando certos pacotes chegarem ao controlador. Ou seja, esses campos irão determinar as ações que serão tomadas e como o fluxo de dados será roteado pela topologia.

A adição de fluxos nas tabelas do *switch* para verificar a capacidade de manipulação e inserção dos fluxos foi executada através de um *script* que inseria alguns

dos campos de forma aleatória, como **id**, **nome**, **prioridade**, **endereço de origem e destino**, dentre outros. Nos experimentos realizados, esses campos tem o objetivo de ocupar a memória das tabelas de fluxos do *switch*. Assim, primeiramente, foram adicionados 1000 fluxos na tabela de índice 0 (existem 256 tabelas no total). Com essa quantidade foi verificado que o desempenho do controlador e *switch* não foi alterado, inclusive os valores obtidos de *throughput* foram os mesmos apresentados nos testes de largura banda. Para prosseguir, foram adicionados 64 mil fluxos na tabela de índice 0, ainda sem alteração de desempenho e sem falhas de funcionamento evidentes. Nos testes seguintes, todas as tabelas foram preenchidas com 1000 fluxos cada, porém não houve alterações.

No intuito de forçar ao máximo o controlador, todas as tabelas foram preenchidas com 8000 fluxos cada. Nesse cenário, o controlador teve uma queda de desempenho drástica, apresentando lentidão e travamentos, até o momento em que parou de responder.

3. Conclusão

Este trabalho apresentou um estudo de caso para o *switch* N3048P da DellTM relativo a avaliação de *throughput* e carga do controlador OpenDaylight em uma topologia de rede SDN. No cenário avaliado, o controlador bem como o *hardware* demonstraram um bom desempenho alcançando a velocidade máxima de transmissão permitida pelo equipamento com pacotes de tamanho de 2048 bytes, como evidenciado nos testes de largura de banda realizados. Também foi possível concluir que o controlador foi capaz de manipular uma quantidade relativamente grande de fluxos em suas tabelas sem sofrer algum tipo de falha, apenas demonstrando problemas quando estressado ao máximo, com 8 mil fluxos em cada uma de suas 256 tabelas.

Trabalhos futuros visando a continuidade desse estudo, incluem a realização de pesquisas relacionadas à mensuração da latência ou *jitter*, e também pesquisas envolvendo algum tipo de interferência no equipamento para verificar mudanças no desempenho nesses cenários, como queda de *throughput* e aumento no tempo de *lookup* dos fluxos corretos a serem utilizados.

Agradecimentos

Os autores gostariam de agradecer ao PROBIC, programa de apoio financeiro da UDESC, bem como ao Laboratório de Processamento Paralelo e Distribuído (LabP2D) pela disponibilização dos recursos computacionais.

Referências

- Bianco, A., Birke, R., Giraud, L., and Palacin, M. (2010). Openflow switching: Data plane performance. In *Proceedings of IEEE International Conference on Communications*, pages 1–5.
- Lara, A., Kolasani, A., and Ramamurthy, B. (2014). Network innovation using openflow: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):493–512.
- OpenDaylight Foundation (2016). Opendaylight: Open source sdn platform. <https://www.opendaylight.org/>. Accessed: 2016-10-30.
- Tootoonchian, A., Gorbunov, S., Ganjali, Y., Casado, M., and Sherwood, R. (2012). On controller performance in software-defined networks. In *Hot-ICE*, volume 12, pages 1–6.