

## Busca de caminhos como serviço em vSDNs

André Bahia<sup>1</sup>, Pedro Mourão<sup>1</sup>, Billy Pinheiro<sup>1</sup>, Antônio Abelém<sup>1</sup>

<sup>1</sup>Grupo de Estudos em Redes de Computadores e Sistemas Distribuídos - GERCOM  
UFPA – Belém – PA  
Caixa Postal 470 – 66075-110 – Belém – PA – Brasil

{andrebahia, billy, abelem}@ufpa.br, pedro.araujo@itec.ufpa.br

**Abstract.** *Virtual Software Defined Networks (vSDNs) were the combination of SDN and Virtualization. In this context, the hypervisor is responsible for managing the physical network, while the traditional SDN network maintains this function in the controller. This article introduces Search Path, a graph-based path finder that avoids unnecessary translations between the hipervisor de rede and the controllers in the vSDN context. The slice forwarded by the hipervisor de rede is received by Search Path and redesigned in the form of graphs, Facilitating the development of specific routing logics for each network, through the manipulation of graphs. The tests performed demonstrate that Search Path has better results than traditional hypervisors and controller.*

**Resumo.** *As Redes virtuais Definidas por Software (vSDNs) são a junção de SDN e Virtualização. Neste contexto, o hipervisor de rede é responsável pela gestão da rede física, enquanto a rede SDN não virtualizada mantém esta função no controlador. Este artigo apresenta o Search Path, um buscador de caminhos baseado em grafos que evita traduções desnecessárias entre o hipervisor de rede e os controladores no contexto de vSDNs. A fatia encaminhada pelo hipervisor de rede é recebida pelo Search Path em forma de grafos, facilitando o desenvolvimento de lógicas de encaminhamento específicas para cada rede, através da manipulação de grafos. Os testes realizados demonstram que o Search Path possui resultados melhores quando comparado com hipervisor de rede e controlador tradicionais.*

### 1. Introdução

A Virtualização de Rede (*Network Virtualization - NV*) habilita o compartilhamento do substrato físico entre diferentes instâncias virtuais, sendo este substrato formado por redes de computadores [Blenk et al. 2016] que podem estar distribuídas geograficamente.

No contexto de Redes Definidas por Software (*Software Defined Networking - SDN*), o hipervisor de rede organiza a arquitetura da rede em fatias e entrega cada fatia a um controlador, assim, permite que diferentes controladores usem a mesma rede física enquanto mantém o isolamento lógico entre eles.

O Search Path (SP) tem como objetivo simplificar as funções exercidas por um controlador em vSDNs, retirando do controlador as tarefas que são executadas pelo hipervisor de rede e criando um buscador de caminhos.

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta as funções dos controladores e virtualizadores. A Seção 3 descreve o *Search Path*. Por fim, a conclusão e os trabalhos futuros são apresentados na Seção 4.

## 2. Trabalhos Relacionados

Esta seção apresenta os trabalhos relacionados as funções exercidas por controladores e virtualizados em vSDNs. As funções exercidas pelos controladores SDN [Stancu et al. 2015] são:

- Verificar o estado da rede: O controlador verifica o comportamento da rede, identificando as cargas dos enlaces e criando a topologia da rede;
- Definir o encaminhamento de dados: O controlador é responsável pela criação e manutenção das regras *OpenFlow* que são encaminhadas para os *switches*.

A Tabela 1 faz a comparação entre os diferentes controladores NOX<sup>1</sup>, Onos [Berde et al. 2014], Beacon [Erickson 2013], Opendaylight [Medved et al. 2014] e Ryu<sup>2</sup> com desempenho de acordo com Andrade [Andrade et al. 2016].

**Tabela 1. Tabela Comparativa dos controladores**

	Utiliza grafo	Desempenho	Independente de Versão Openflow
<b>NOX</b>	Não	Moderado	Não
<b>Ryu</b>	Não	Ruim	Não
<b>Beacon</b>	Não	Bom	Não
<b>Floodlight</b>	Não	Bom	Não
<b>Opendaylight</b>	Não	Moderado	Não
<b>Onos</b>	Parcialmente	Moderado	Não

As funções atualmente exercidas pelos hipervisores de rede [Blenk et al. 2016] são:

- Dividir a rede e dar a cada controlador uma fatia (*slice*). O controlador possui conhecimento apenas da fatia da rede alocada para ele, enquanto o hipervisor de rede tem a visão global da rede;
- Mapear as topologias físicas e de todas as fatias, uma tarefa maior que a do controlador, que gerencia apenas a própria fatia;

São exemplos de hipervisores de rede o *OpenVirteX* [Al-Shabibi et al. 2014], e *Graph Virtualization Layer (GVL)*<sup>3</sup> [Pinheiro 2016], o último utilizado neste trabalho por reduzir a troca de mensagens entre a camada que controla o *slice* e a camada de virtualização. Através da análise das funções executadas pelos controladores e hipervisores de rede é possível identificar que a presença do hipervisor de rede, um requisito das vSDNs para acabar com a duplicação nas funções executadas na rede.

## 3. Search Path (SP)

O *Search Path* é um buscador de caminhos para vSDNs, que usa abstração de grafos e permite uma visão mais abrangente das redes através do dissociação com a linguagem

<sup>1</sup><https://github.com/noxrepo/nox>

<sup>2</sup>Disponível em <https://osrg.github.io/ryu/>

<sup>3</sup>Disponível em <https://gitlab.com/gercom/gvl/>

de rede utilizada (OpenFlow). Usando como base as funções dos hipervisores de rede e controladores mapeadas na Seção 2, é possível definir os requisitos desejáveis para um buscador vSDN, são eles:

- Receber informações do hipervisor de rede em vez de coletá-las. Para cada alteração na estrutura da rede, o hipervisor deve mapear a alteração e reenca-minhá-la para o buscador, evitando fluxos de dados desnecessários. Como o SP é usado em vSDNs, ele suporta esse recurso.
- Receber dados do hipervisor sobre a entrada de novos membros na rede: Não é função de um controlador para vSDNs mapear a entrada de novos elementos da rede, mas sim receber essas informações do hipervisor. O *Search Path* usa os dados do hipervisor de rede e não faz varreduras no *slice*, uma vantagem do SP sobre os controladores.
- O buscador deve ser responsável apenas pela criação do caminho onde os dados devem prosseguir, deixando de ser responsável pela criação de regras nos equipa-mentos, o SP usa os conceitos de grafos e o mapeamento já feito pelo hipervisor de rede para a busca dos caminhos e posterior inserção deste no hipervisor de rede.

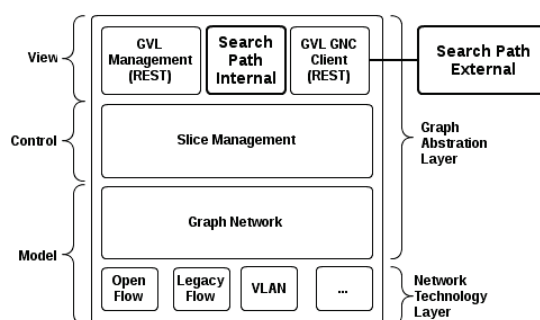


Figura 1. Definição do SPi e SPe em vSDN

A Figura 1 mostra em destaque a arquitetura do *Search Path internal*, para testes de menor caminho, e a *Search Path external*, que deverá ser utilizada em aplicações mais complexas, como *firewall*, por exemplo. As partes não destacadas compõem a arquitetura do hipervisor de rede GVL, utilizado neste trabalho para validação da proposta.

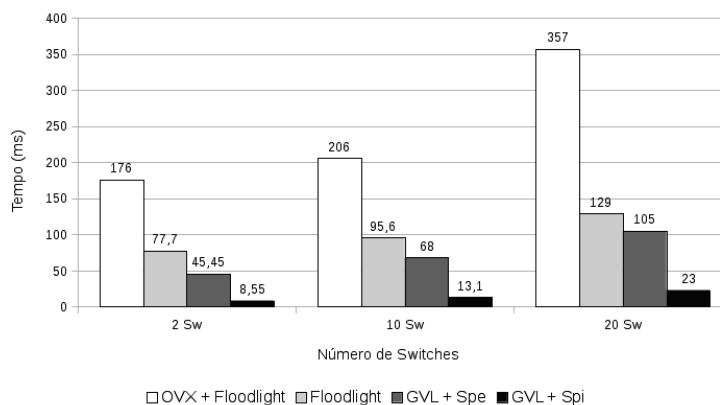


Figura 2. Gráfico entre as diferentes arquiteturas

Os testes consistiram em verificar o tempo de resposta do primeiro pedido ICMP utilizado em uma rede de topologia linear, para isso utilizamos a ferramenta Ping. Apenas o primeiro pacote atinge a camada de controle, os restantes seguem o fluxo das regras nos *switches*. A Figura 2 mostra os resultados dos testes do SP interno e SP externo utilizando o hipervisor de rede GVL, hipervisor de rede capaz de exportar a topologia dos *slices* sem usar Openflow. Nos testes tanto o *SPe* quando o *SPi* apresentaram melhores resultados que o *OpenVirtex* com o *Floodlight* e até mesmo no experimento usando apenas o *Floodlight*, ou seja, sem a camada de virtualização.

#### 4. Conclusão e Trabalhos Futuros

Este artigo apresentou o *Search Path*, um buscador de caminhos para vSDNs que utiliza os conceitos de grafos para diminuir a troca de mensagens *OpenFlow* e apresenta ganhos de desempenho quando comparado ao *OpenVirtex* com *Floodlight*. Novas topologias devem ser utilizadas em novos testes de desempenho nos trabalhos futuros para comprovar a escalabilidade da solução.

#### Referências

- Al-Shabibi, A., De Leenheer, M., Gerola, M., Koshibe, A., Parulkar, G., Salvadori, E., and Snow, B. (2014). Openvirtex: Make your virtual sdn programmable. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 25–30, New York, NY, USA. ACM.
- Andrade, L., Borba, M., Ishimori, A., Farias, F., Cerqueira, E., and Abelém, A. (2016). On the benchmarking mainstream open software-defined networking controllers. In *Proceedings of the 9th Latin America Networking Conference*, LANC '16, pages 9–12, New York, NY, USA. ACM.
- Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., and Parulkar, G. (2014). Onos: Towards an open, distributed sdn os. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 1–6, New York, NY, USA. ACM.
- Blenk, A., Basta, A., Reisslein, M., and Kellerer, W. (2016). Survey on network virtualization hypervisors for software defined networking. *IEEE Communications Surveys Tutorials*, 18(1):655–685.
- Erickson, D. (2013). The beacon openflow controller. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, pages 13–18, New York, NY, USA. ACM.
- 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*.
- Pinheiro, B. (2016). *Uma Abordagem SDN para Virtualização de Redes*. PhD thesis, Universidade Federal do Pará.
- Stancu, A. L., Halunga, S., Vulpe, A., Suci, G., Fratu, O., and Popovici, E. C. (2015). A comparison between several software defined networking controllers. In *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, pages 223–226.