

vSDNLight: Uma Proposta de Arquitetura Leve para Provisionamento de Redes Virtuais Definidas por Softwares

Fernando N. N. Farias¹, Antônio J. G. Abelém¹

¹Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Pará (UFPA)
Belém – Pará – Brasil

{fernndf, abelem}@ufpa.br

Abstract. *For building virtual software defined networks, which also are well-known as slices, is indispensable to use a hypervisor SDN. However, these solutions have introduced serious limitations of performance and scalability, because of your architecture based on a proxy. Thus, this paper presents a proposal of orchestrator that provides virtual software defined network based on a new model through the allocation of virtual switch instance on demand available in a physical infrastructure of whiteboxes.*

Resumo. *Para construção de redes virtuais definidas por softwares (vSDN), também conhecidas como slices da infraestrutura física, é necessário a utilização de soluções de hipervisores SDN. No entanto, essas soluções vêm apresentando grandes limitações de escalabilidade e desempenho, por causa de sua arquitetura baseada em proxy de serviços. Este artigo apresenta uma proposta de orquestração diferente do modelo atual provendo redes virtuais definidas por softwares através da alocação de instancias de switches virtuais sob demanda diretamente em dispositivos de comutação de baixo custo.*

1. Introdução

A redes definidas por software (SDN) [Kreutz et al. 2015] vem sendo a principal tecnologia utilizada para habilitar a virtualização de redes através de múltiplas redes lógicas isoladas em redes virtuais definidas por softwares (vSDN) [Blenk et al. 2016]. A partir do SDN foi possível otimizar a utilização de recursos físicos através da abstração de diferentes instâncias lógicas de maneira concorrente e isoladas, de forma que os recursos compartilhados são resumidamente: nós e enlaces. As vSDNs têm a habilidade de redesenhar a topologia da infraestrutura física em topologias virtuais, que por sua vez pode servir de suporte a uma aplicação ou serviço específico [Bozakov and Papadimitriou 2014].

Para habilitar o compartilhamento de uma infraestrutura física SDN por diferentes *tenants* (inquilinos) foi desenvolvida uma camada de abstração chamada de hipervisor SDN. Esta camada é responsável por monitorar, isolar e abstrair os recursos disponíveis na infraestrutura física e prover uma rede virtual para as camadas superiores através de um *slice* [Sherwood et al. 2009]. Desta forma, cada *tenant* que utilizar um *slice* pode instanciar seu próprio controlador, com a impressão que está utilizando a rede em sua plenitude quando, na verdade, ele apenas controla parte do recurso que o hipervisor lhe ofereceu.

No entanto, soluções de hipervisores, sejam elas centralizadas ou distribuídas, vêm apresentando problemas de desempenho que impactam diretamente na execução, tanto das redes virtuais como do próprio hipervisor [Blenk et al. 2016]. Isto acontece, pois, as traduções aplicadas pelo hipervisor elevam o índice de processamento que, conseqüentemente, degrada o desempenho do mesmo para execuções de outras funções. Além disso, esses hipervisores acumulam vários níveis de funções de controle e orquestração, causando um verdadeiro inchaço na estrutura do mesmo, ocasionando falhas que impossibilitam o aumento da escala de redes virtuais. Como exemplo, uma mensagem ou pacote pode ser descartado devido à alta utilização de recursos da rede (ex. sobrecarga de regras na tabela de encaminhamento) ou a exaustão da execução do hipervisor, como consequência, todas as redes virtuais são comprometidas e com isso o isolamento entre elas é afetado.

Devido a sua característica de abertura, o paradigma SDN vem sofrendo constantes evoluções e oferecendo oportunidades para o desenvolvimento de novas soluções em quaisquer de suas camadas. Nesse contexto, a introdução de *whiteboxes* (baseados em hardware ou software) [Farias et al. 2018b] na infraestrutura SDN permitiu a exploração de novas perspectivas de virtualização mais próximas do plano de dado. Atualmente, esses *whiteboxes* estão oferecendo dispositivos capazes de estabelecer instâncias virtuais de switches na infraestrutura física. No entanto, não existem arquiteturas definidas que explorem essas características de alocação de switches virtuais em *whiteboxes* para a aplicação de virtualização de redes e, com isso, apresentar um modelo de redes virtuais em SDN baseados na alocação de switches virtuais sob demanda.

Este trabalho apresenta uma proposta de arquitetura chamada de *vSDNLight*, que tem como objetivo propor uma arquitetura leve que define uma nova abordagem de criação de redes virtuais definidas por softwares (*vSDN*) através de switches *whiteboxes* que permitam a alocação de instâncias virtuais de switches. Neste caso o hipervisor deixa de existir entre o controlador da rede virtual e infraestrutura tal como a necessidade de traduções (i.e., *proxy*). Na proposta a camada de virtualização é parametrizada entre os switches da infraestrutura física, e o seu conjunto dessas instâncias de switches virtuais caracteriza a rede virtual.

Além desta seção introdutória, este artigo está dividido em mais 3 seções. Na Seção 2, tem-se os conceitos e descrição da proposta virtualização de redes baseados na instanciação de switches virtuais. Na Seção 3, descreve-se a arquitetura do *vSDNLight*. Por fim, na Seção 4, tem-se as conclusões e trabalhos futuros.

2. Slice baseado em VSIs

O *vSDNlight* é um orquestrador leve para prover redes virtuais definidas por softwares, que através de agentes espalhados pela infraestrutura reduz os overheads que ocasionam problemas, tanto na escalabilidade quanto na latência das soluções atuais, afetando o desempenho desses hipervisores.

O principal desafio científico da proposta é aplicar uma nova arquitetura para a criação de slices em redes SDN. Na arquitetura cria-se um modelo de slice baseado em vínculos entre pequenas abstrações de switches virtuais nomeadas de VSI (*Virtual Switch Instance*), de forma que o conjunto dessas instâncias resultam na rede virtual ou slice. Para isso, a gerência dessas VSIs é definida por agentes que negociam com a tecnologia

de virtualização disponível no plano de dados (ex. ¹*OpenvSwitch*) a criação, atualização ou remoção das mesmas. Na Figura 1, ilustra-se a diferença entre modelo tradicional e o proposto.

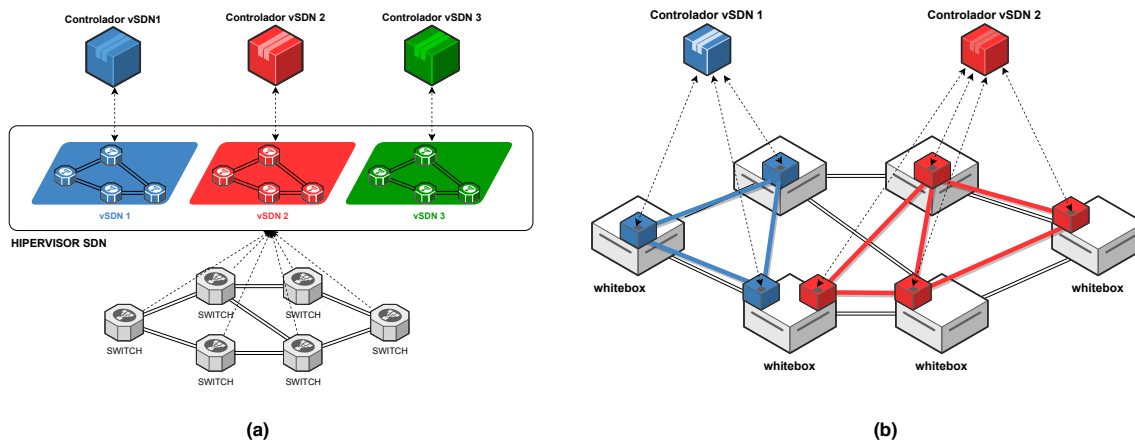


Figura 1. Diferença entre o modelo de *slice* tradicional (a) e o proposto (b).

No modelo tradicional, ilustrado na Figura 1(a), o hipervisor tem que tratar e identificar todas as mensagens recebidas de cada switch participante no *flowspace*. Logo depois, essas mensagens são reescritas e entregues ao controlador responsável por este *flowspace*. A tabela de encaminhamento é compartilhada entre os *flowspaces* nessa proposta. Além de ser um ponto de falha, isso compromete a escalabilidade, pois dependendo do número de redes virtuais essa tabela pode ser rapidamente congestionada de regras, o que inviabilizaria a sua utilização.

No modelo proposto, ilustrado pela Figura 1(b), a proposta é leve, pois não há tratamento de mensagens de controle pelo orquestrador e sim apenas o gerenciamento do ciclo de vida e monitoramento das redes virtuais, eliminando assim, a camada de virtualização no plano de controle e levando-a para mais próximo do plano de dados. Além disso, o orquestrador também coordena agentes espalhados na infraestrutura para tratar apenas das abstrações de switches virtuais, que é uma parte pequena da topologia criada, diminuindo a sobrecarga de controle e gerenciamento no sistema como um todo, se comparado a um hipervisor SDN. Desta forma, o controlador do usuário atua diretamente com o switch virtual.

Esta proposta também contribui, apresentando uma nova visualização do slice, sendo composto da junção das instâncias de virtuais de switches que compõe a topologia do slice, conforme ilustrado na Figura 2, e administrado pelo orquestrador (*vSDNLight*) que é executado no plano de gerenciamento. Na mesma Figura 2, a elipse vermelha ilustra o slice e os recursos alocados pelos agentes na infraestrutura física.

Com a remoção da camada de virtualização do plano de controle, a camada de tradução (i.e., camada de proxy) não é mais necessária, de forma que os dados pertencentes a um determinado slice são entregues diretamente a instância virtual do switch. Além disso, o controle dos slices também não são mais necessários, de forma que todo o tipo de orquestração foi migrado para plano de gerenciamento da arquitetura SDN, através do

¹Projeto OpenvSwitch: <https://www.openvswitch.org>

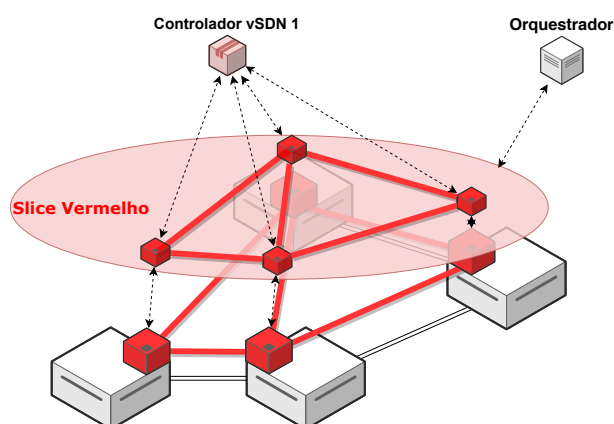


Figura 2. Visão de slice no modelo proposto.

componente *vSDNOrches*.

Por fim, a proposta ainda define dois modelos de operação do *slice*: no primeiro modo o controle e de responsabilidade do tenant, ou seja, o *slice* é controlado pelo sistema operacional de rede (NOS – *Networking Operating System*) atribuído pelo tenant com suas próprias regras de negócios. O outro modo, o controle do *slice* feito por uma interface de gerência através de um WIM (*Wide-area Network Infrastructure Manager*) [Necos 2019] aonde existe um controle da rede virtual pré-definido e apenas algumas funções especiais de controle do *slice* são disponibilizadas externamente.

3. Proposta do *vSDNLight*

Na arquitetura do *vSDNLight*, ilustrada na Figura 3, é possível observar os componentes responsáveis pela gerência dos recursos de virtualização disponíveis pela infraestrutura. Esta arquitetura contém dois elementos essenciais para funcionamento da orquestração: *vSDNOrches* e *vSDNAgent*. Na Figura 3, observa-se a visão geral da arquitetura destacando essas estruturas e seus componentes internos.

O *vSDNOrches* é responsável por fazer toda a orquestração dos slices e mapeá-los em forma de VSIs, além disso, ele também mapeia a interconexão entre os VSIs através de enlaces virtuais ou físicos, disponíveis na infraestrutura. A estrutura interna dele é composta pelos seguintes componentes: *Slice Manager*, *Slice Monitor*, *Topology Manager*, *CLI Manager* e *Slice Builder*.

O *Slice Manager* é responsável administrar a criação, remoção e atualização dos *slices* solicitados pelos tenants. O *slice* é o efeito do mapeamento dos nós em forma de VSIs na infraestrutura física e os enlaces virtuais podem ser um compartilhamento do enlace físico através da implementação de protocolos como *802.1q VLAN* e *802.1ad Stack VLAN* ou túneis *VXLAN* ou *GENEVE*. O *Slice Manager* também permite que o *slice* possa ser alocado de duas maneiras, no primeiro chamado de *on-demand*, o *tenant* repassa a topologia e o *Slice Manager* decide em que pontos da infraestrutura, ou seja, quais switches os elementos virtuais serão alocados. Já o segundo é chamado de *self-assigned*, o próprio *tenant* determina em que lugar da infraestrutura os VSIs que compõem o *slice* serão mapeados. Por fim, ele também faz a administração dos tenants e como será o controle do *slice* se através de um controlador externo ou se por um WIM.

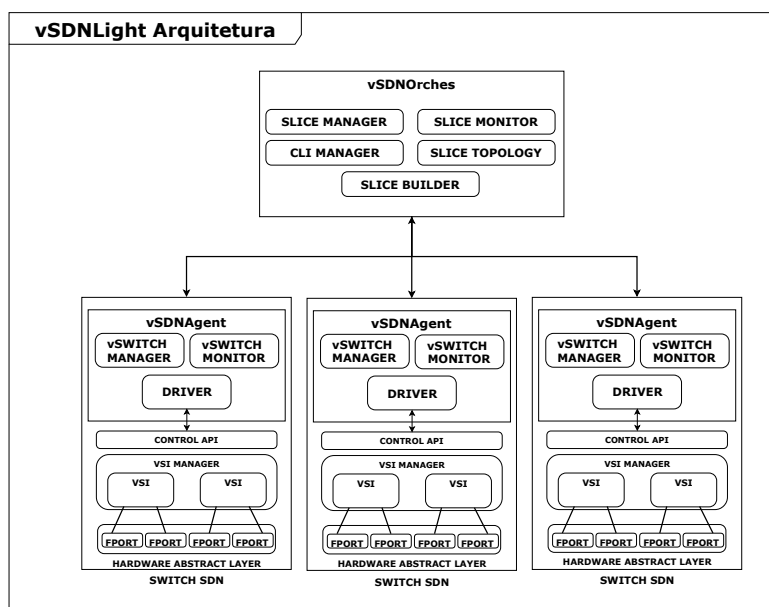


Figura 3. Arquitetura do vSDNLight.

O *Slice Monitor* permite ao vSDNOrches a capacidade de monitorar os slices e seus dispositivos físicos. Além disso, também há trocas de alertas vindos do vSDNAgent (ex. congestionamento de enlace e falhas tanto no switch virtual quanto o físico) ou consultas diretas aos agentes conectados aos dispositivos físicos. O *Topology Manager* é responsável por fazer à persistência dos dados da infraestrutura física e das topologias virtuais criadas. Ele é baseado em banco de dados não relacional baseado em grafos. Como as informações a serem tratadas são praticamente topologias, então esse tipo de banco de dados facilita a modelagem e otimiza as buscas de recursos. No *CLI Manager*, tem-se a interface primária com usuário onde é possível desenhar a topologia da rede virtual que será requisita ao *Slice Manager* para ser aplicada na infraestrutura física. Por fim, tem-se o *Service Bus* que o componente que permite troca de mensagens entre os componentes do vSDNLight com qualquer outro componente da arquitetura. No *Slice Builder*, tem-se o componente responsável por coordenar e registrar os vSDNAgents que entram e saem na infraestrutura. Ele é o componente responsável por negociar os recursos a serem alocados na infraestrutura. Essa negociação é definida pelo *Slice Manager* e repassada para o *Slice Builder* que aplicará a solicitação aos vSDNAgents.

O vSDNOrches é composto de 3 componentes: *vSwitch Manager*, *vSwitch Monitor* e *Driver*. No *vSwitch Manager*, tem-se o componente responsável por solicitar a criação, remoção e atualização das VSIs junto a tecnologia de virtualização disponível (ex. *OpenvSwitch* ou ²*OF-DPA*), ele também faz integração das portas virtuais com as portas físicas, configuração dos protocolos a serem utilizados na VSI e o mapeamento do VSI ao slice solicitante. Já o *vSwitch Monitor* monitora o comportamento do VSI e encaminha para o orquestrador, além disso, ele também é capaz de enviar alertas a respeito do comportamento dos VSIs, como sobrecargas ou falhas. Já o componente *Driver* é responsável por fazer a comunicação do vSDNAgent com hipervisor local, o objetivo deste componente é fazer o vSDNAgent tolerar qualquer tecnologia de virtualização de switch

²OF-DPA Library: <https://github.com/Broadcom-Switch/of-dpa>

e com isso permitir que a proposta alcance a maior quantidade de switches possíveis ou fazer com que qualquer fabricante possa desenvolver a sua proposta de vSDNAgent.

4. Conclusão e Trabalhos Futuros

Esta proposta apresentou o *vSDNLight*, uma arquitetura que explora a característica de *whiteboxes* de permite à criação de instâncias de switches virtuais sobre demanda. A proposta oferece uma arquitetura mais enxuta ou leve, a partir da remoção da camada de virtualização do plano de controle e a sua distribuição no plano de dados através de agentes que negociam a alocação dos switches virtuais com o switch físico. Neste caso, os elementos da rede virtual tratam diretamente com o seu controlador não precisando passar por qualquer hipervisor para fazer traduções, diminuindo o overhead do tratamento dos dados de controle quando comparado as soluções atuais.

Como trabalhos futuros, pretende-se fazer uma avaliação de desempenho do *vSDNLight* e compara-la com outras soluções mais conhecidas como *OpenVirteX* e *Flowvisor*. Também deseja-se avançar no desenvolvimento da arquitetura e disponibiliza-la junto ao emulador *vSDNEmul* [Farias et al. 2018a]. Este trabalho pode progredir ainda em questões como: migração de redes virtuais, tolerância a falhas e slice como serviço.

Agradecimentos

Esta pesquisa foi patrocinada pelo H2020 na 4ª chamada de colaboratividade EU-BR, no acordo no 777067 (*NECOS - Novel Enablers for Cloud Slicing*), financiada pela Comissão Europeia de Tecnologia e Ministério da Ciência e Tecnologia.

Referências

- Blenk, A., Basta, A., Reisslein, M., and Kellerer, W. (2016). Survey on network virtualization hypervisors for software defined networking. *IEEE Communications Surveys and Tutorials*, 18(1):655–685.
- Bozakov, Z. and Papadimitriou, P. (2014). Towards a scalable software-defined network virtualization platform. In *IEEE/IFIP NOMS 2014 - IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World*.
- Farias, F. N. N., Salvador, P. M., and Abelém, A. J. G. (2018a). vSDNEmul: Emulador de Redes Definidas Por Software Usando Contêineres. In *Anais do IX Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF - SBRC 2018)*, volume 9. SBC.
- Farias, F. N. N., Schwarz, M. F., de Rezende, J. F., and Abelém, A. J. G. (2018b). vSDN-Box: Um Hardware Especializado de Baixo Custo Gerenciado via SDN. In *Anais do XXIII Workshop de Gerência e Operação de Redes e Serviços (WGRS - SBRC 2018)*, volume 23. SBC.
- Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-Defined Networking : A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14 – 76.
- Necos (2019). NECOS System Architecture and Platform Specification. Acessado em: 22/03/2019. Disponível em: <http://www.maps.upc.edu/public/NECOS D3.1.final.pdf>.
- Sherwood, R., Gibb, G., Yap, K.-k., Appenzeller, G., Casado, M., Mckeown, N., and Parulkar, G. (2009). FlowVisor: A Network Virtualization Layer. In *Network*, page 15.