Posicionamento e Encadeamento Otimizado de Funções Virtuais de Rede usando Regras *SDN/OpenFlow*

Eduardo Stein Brito, Weverton Cordeiro, Luciano Paschoal Gaspary

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{eduardo.brito,weverton.cordeiro,paschoal}@inf.ufrgs.br

Abstract. *Network Function Virtualization (NFV) enables shifting network functions (such as firewall, proxy, and load balancing), from specialized appliances to software centric solutions running on commodity hardware, using virtualization. In conjunction with Software Defined Networking (SDN), it becomes feasible to place these functions within the network with great flexibility. In spite of the convenience, optimized network function placement and chaining is non-trivial, thus being the subject of various investigation efforts. In this paper, we present a first approach for materializing optimized network function placement and chaining in a network infrastructure. From preliminary results we assessed that, in each network switch, configured rules ensured that packets would follow an optimal path, thus leading to rational use of network resources.*

Resumo. *A Virtualização de Funções de Rede (NFV) permite transferir funções de rede (como* firewall*,* proxy *e balanceamento de carga), de equipamentos especializados para soluções centradas em* software *executando em servidores de prateleira, usando virtualização. Aliado a Redes Definidas por Software (SDN), torna-se possível posicionar funções virtuais de rede de forma flexível na infraestrutura. Apesar da conveniência, o posicionamento e encadeamento otimizado de funções de rede não é trivial, tendo sido foco de diversas pesquisas. Nesse artigo, apresenta-se uma primeira proposta para materializar esse encadeamento. A partir de resultados preliminares, observou-se que as regras instaladas em cada* switch *garantiam o trânsito otimizado dos pacotes de dados, promovendo assim o uso mais racional dos recursos da rede.*

1. Introdução

A Virtualização de Funções de Rede (*Network Function Virtualization*, NFV) é um paradigma que permite transferir funções de rede (como *firewall*, *proxy* e balanceamento de carga), de equipamentos especializados para soluções centradas em *software* executando em servidores de prateleira, usando virtualização [NF ISG 2012]. Aliado ao paradigma de Redes Definidas por Software (*Software-Defined Networking*, SDN) [McKeown 2009], torna-se possível que funções virtuais de rede (*Virtual Network Functions*, VNFs) sejam posicionadas de forma flexível na rede [Luizelli et al. 2015]. Mais importante, SDN torna possível encadear fluxos de dados entre as VNFs necessárias e na ordem desejada.

Um dos principais problemas em NFV refere-se ao posicionamento, encadeamento e orquestração de VNFs, algo não trivial do ponto de vista operacional. Isso porque fluxos de rede podem requerer várias VNFs para processá-los, além de larguras de banda específicas nos enlaces entre VNFs. Além disso, o posicionamento das mesmas precisa levar em consideração a capacidade disponível nos pontos de presença (*Network Points of Presence*, N-PoPs) em que podem ser instanciadas.

Várias investigações abordaram o problema acima. Por exemplo, Moens *et al.* [Moens and De Turck 2014] o formalizaram como um problema de otimização, escalável para um pequeno conjunto de VNFs. Luizelli *et al.* [Luizelli et al. 2015] também abordaram o problema por uma perspectiva de otimização, introduzindo um conjunto de heurísticas para reduzir o espaço de buscas, reduzindo portanto a complexidade para encontrar soluções ótimas. No âmbito da orquestração, um dos esforços mais notórios é o *Open Platform for NFV* (OPNFV) [Price et al. 2015], uma plataforma que visa fomentar a adoção e inter-operabilidade de soluções NFV, entre outros.

Apesar dos avanços na área, observa-se uma lacuna existente entre as soluções voltadas ao posicionamento e encadeamento, e aquelas voltadas à orquestração das mesmas. Mais especificamente, as soluções para orquestração existentes não agregam os requisitos de posicionamento e encadeamento ótimo de funções de rede, restringindo-se a implantá-las e gerenciá-las. Por outro lado, as pesquisas voltadas ao posicionamento e encadeamento assumem, como premissa simplificatória, que os resultados gerados podem ser trivialmente acomodados na infraestrutura.

2. Posicionamento e Encadeamento Otimizado de VNFs usando OpenFlow

Dado o problema exposto, o objetivo do presente trabalho é a investigação e concepção de uma solução para operacionalizar o encadeamento otimizado de VNFs, com foco na programação de regras de fluxos de dados em redes SDN usando o padrão *OpenFlow*. Visa-se assegurar o encaminhamento dos fluxos por instâncias de VNFs hospedadas em N-PoPs, na ordem necessária, tal como definido em um documento de encadeamento de funções de rede (*Service Function Chaining*, SFC). Portanto, o foco da solução esteve no desenvolvimento de um algoritmo que permita o encadeamento de fluxos de rede, passando por VNFs, a partir de parâmetros fornecidos.

Lançando mão do sistema NFBRIDGE [Luizelli et al. 2015], e emulando *switches OpenFlow* utilizando o *mininet*, foi possível construir um ambiente para experimentação do processo de encadeamento otimizado de VNFs. O NFBRIDGE gera como saída um arquivo contendo SFCs que devem ser implementadas. Apesar de indicar a melhor forma de alocar e encadear as VNFs, ele não operacionaliza esses procedimentos na infraestrutura.

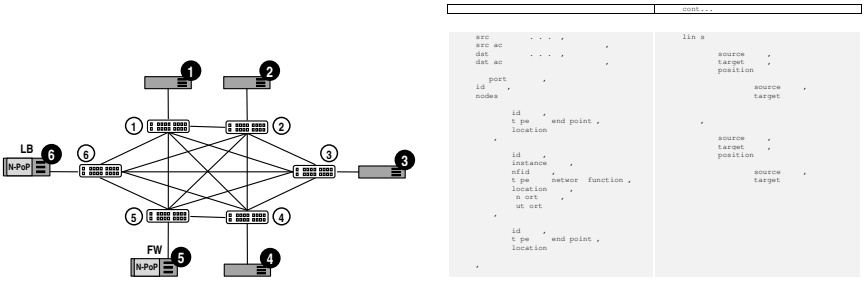
Para suprir essa lacuna, foi utilizado o controlador *Floodlight*, o qual mostrou-se ser um dos mais promissores para o projeto. Este controlador tem por objetivo gerar as regras nos *switches* às quais seriam baseadas em arquivo de saída gerado pelo NFBRIDGE. Desenvolveu-se, então, um algoritmo para processar esta saída, e materializar as alocações de VNFs e encadeamentos na rede. A implementação das VNFs foi feita aproveitando-se do *software open source Click Router*. *Click* permite programar diversas funções de rede, executado-as em um *host*. Para fins de experimentação, criou-se duas VNFs: um *firewall* e um balanceador de carga (*load balancer*); ambas foram programadas utilizando a linguagem *click*. Na criação de regras (a partir do arquivo de saída do NFBRIDGE), cada vez que há a conexão do *switch* com uma VNF, regras especiais devem ser geradas para que se garanta a troca de pacotes entre os mesmos (do *switch* para a VNF e vice-versa). A seguir é dada uma breve descrição do algoritmo proposto.

Identificação de SFCs. Na leitura do arquivo, para a correta identificação da SFCs, estas devem conter IDs próprios. Além disso, deve-se informar as portas TCP/UDP de comunicação, e qual o IP de origem e IP de destino das conexões. Assim, torna-se possível gerar regras não ambíguas, às quais levam como *matching* essas informações. Já para o ID da SFC, é utilizada uma *Vlan Tag* que será o ID informado mais um (ID + 1).

Interpretação de Nodos. Os "nodos" (que podem ser do tipo funções de rede, *end-points* de fluxos, etc.) permitem identificar em qual(is) local(is) da topolo-

gia encontra(m)-se o(s) *host(s)* e também a(s) VNF(s). Os *hosts*, nesse caso, podem estar conectados a *switches* da topologia, sendo tanto um *end-point* de origem/destino, como também uma VNF. No exemplo ilustrado, para o nodo VNF do tipo *Firewall*, é indicado por quais portas ele está conectado ao *switch*. No caso do *Load Balancer*, são indicados os *hosts* entre os quais a carga será balanceada, com seus respectivos IPs e MACs. Assim, consegue-se saber o que fazer com os pacotes nos *switches* conectados a esses nodos.

Interpretação de Links. Os "links" indicam entre quais pares de *switches* os pacotes devem transitar. A partir da leitura de cada conexão, e verificando em cada *switch* se existe algum nodo conectado, é possível criar as regras *OpenFlow* via controlador. Assim, torna-se possível encaminhar o pacote para o próximo *switch*, sempre fazendo o *matching* dos pacotes conforme a identificação de cada SFC.



(a) Topologia (b) SFC

Figura 1. Topologia totalmente conexa com um balanceador de carga e um *firewall*, e SFC descrita pelo arquivo gerado pelo NFbridge.

Para ilustrar a solução desenvolvida, considere a topologia da Figura 1. Usando como entrada um arquivo gerado pelo NFBRIDGE, pode-se verificar as regras a serem instaladas nos *switches*. No exemplo, o ID da SFC é "1". Conforme mencionado, esse ID será utilizado para colocar uma *Vlan Tag*, visando identificar todas as regras referentes àquela SFC. Dessa forma, cada *switch* da topologia possui regras que tenham *matching* de acordo com a *Vlan Tag*, de forma a discriminar as regras de diferentes SFCs. Um detalhe importante é que a *Vlan Tag* é removida no *switch* que está conectado ao *host* de destino, para que o pacote seja reconhecido ao enviá-lo ao *host*. A partir da verificação dos nodos, é possível saber que a SFC descreve o encaminhamento de pacotes do *host* conectado no *switch* 1 para o *host* conectado em 2. Note que o ID do *switch* é o valor de "location" + 1. Os pacotes serão encaminhados então pela VNF (neste caso um *firewall*, identificado por um "nfid" igual a zero) que se encontra no N-PoP conectado ao *switch* 5. Por fim, os enlaces são verificados para determinar o caminho que os pacotes devem percorrer.

Observe ainda na Figura 1 que é especificado que os pacotes devem percorrer os *switches* 1, 5, 2, nesta ordem. Como também foi especificado que esses pacotes devem passar por um *firewall*, os pacotes em 5 serão enviados para essa VNF, localizada no N-PoP 5, antes de serem encaminhados para o *switch* 2. A interpretação de todos esses dados é feita pelo controlador *Floodlight*. Depois da criação das regras nos *switches*, é possível encaminhar os pacotes pelo caminho especificado. Para ilustrar, temos um exemplo (Tabela 1) de regras criadas no *switch* 1, para a SFC mostrada na Figura 1.

3. Resultados Preliminares

Para aferir a eficácia e eficiência do algoritmo desenvolvido, foram avaliadas diversas topologias com várias SFCs submetidas simultaneamente. Além disso, foi verificado, nos

```
*** s1 ----------------------------------------------------
cookie=0x20000000000000, duration=48.978s, table=0,
n_packets=30, n_bytes=2575,
idle_age=6, priority=0,tcp,in_port=1,nw_dst=10.0.0.2,
tp_dst=8000 actions=mod_vlan_vid:2,output:5
cookie=0x20000000000000, duration=48.991s, table=0,
n_packets=30, n_bytes=10210, idle_age=6, priority=0,
ip,in_port=5,dl_vlan=2 actions=strip_vlan,output:1
```

Tabela 1. Regra criada no *switch 1* da topologia da Figura 1

switches Open Flow virtualizados no *mininet*, se as regras eram criadas corretamente. Por fim, foi aferida a efetividade das VNFs no processamento de fluxos de tráfego.

Para avaliação foram instanciados, nos *hosts end-points* das SFCs, servidores web e clientes *wget*. O objetivo foi verificar se as regras habilitavam o encaminhamento otimizado dos fluxos entre ambos. Os resultados obtidos foram promissores: verificou-se a efetividade das regras instaladas, sendo possível obter as páginas web, e minimizando o tráfego na rede. Como parte dos experimentos, analisou-se também o número de regras criadas por SFC. Em um dos experimentos, constatou-se por exemplo que para qualquer *switch* não esteja conectado a uma VNF parte daquela SFC, são criadas apenas duas regras: uma para ida do pacote em direção ao destino, e outra para tráfego de resposta. Quando o *switch* está conectado a um VNF, são criadas regras conforme o tipo de VNF: no caso do *Firewall*, quatro regras são criadas; para o *Load Balancer*, seis regras.

4. Considerações Finais e Trabalhos Futuros

Os resultados alcançados, apesar de demonstrar a efetividade da materialização do encadeamento ótimo de fluxos entre funções de rede, sugeriram direções de pesquisa importantes para permitir, via NFBRIDGE [Luizelli et al. 2015], uma implantação mais eficaz das SFCs submetidas. Além disso, experimentos adicionais deverão ser conduzidos, considerando SFCs mais complexas e/ou específicas, para verificar a robustez do algoritmo de geração de regras SDN implementado. Mais ainda, faz-se necessário a implementação de outras VNFs, bem como avaliação contemplando topologias de rede mais complexas (por exemplo, topologias de *datacenter* e de *backbones*).

Referências

Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P., and Gaspary, L. P. (2015). Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *Integrated Network Management (IM 2015), 2015 IFIP/IEEE International Symposium on*, IM '15, pages 98–106, New York, NY, USA. IEEE.

McKeown, N. (2009). Software-defined networking. *INFOCOM Keynote*, 17(2):30–32.

Moens, H. and De Turck, F. (2014). Vnf-p: A model for efficient placement of virtualized network functions. In *Network and Service Management (CNSM), 2014 10th International Conference on*, pages 418–423.

NF ISG (2012). Network function virtualisation (nfv): An introduction, benefits, enablers, challenges and call for action. In *SDN and OpenFlow World Congress*, pages 1–16.

Price, C., Rivera, S., Peled, A., Wolpin, M., Brockners, F., Chinnakannan, P., Sardella, A., Hou, P., Young, M., Mehta, P., Nguyenphu, T., and Neary, D. (2015). OPNFV: An Open Platform to Accelerate NFV. Available: <https://www.opnfv.org/sites/opnfv/files/pages/files/opnfv_whitepaper_ 103014.pdf>.