On-The-Fly NFV: Em Busca de uma Alternativa Simples para a Instanciação de Funções Virtualizadas de Rede

Giovanni Venâncio[1], Rogério C. Turchetti[2], Elias P. Duarte Jr.[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19.081 – 81.531-980 – Curitiba – PR – Brasil

[2]CTISM - Universidade Federal de Santa Maria (UFSM)
Avenida Roraima, 1000 – 97105-900 – Santa Maria – RS – Brasil

gvsouza@inf.ufpr.br, turchetti@redes.ufsm.br, elias@inf.ufpr.br

Abstract. *Network Function Virtualization (NFV) is a technology that allows the design, development and management of network functions that are usually implemented in hardware, using virtualization techniques. In this work we present On-The-Fly NFV, a tool that we believe to be as simple as possible to instantiate and delete Virtualized Network Functions (VNFs) in real time. The proposed tool provides a simple and intuitive way to create specific filters that are applied to packet flows, determining to which VNFs packets are forwarded. The implementation of a VNF called VNF-Filter is also described. This VNF is reponsible for the instantiation and removal of other VNFs and also for classifying and forwarding packets to the corresponding functions. Performance evaluation experiments are presented as well as a qualitative comparison with a traditional NFV platform based on OpenStack.*

Resumo. *Network Function Virtualization (NFV) é uma tecnologia que permite projetar, desenvolver e gerenciar funções de rede, normalmente implementados em hardware, usando virtualização. Neste trabalho é apresentada On-The-Fly NFV, uma ferramenta tão simples quanto possível que permite instanciar e remover funções virtualizadas de rede (Virtualized Network Functions ou VNFs) em tempo real. A ferramenta proposta fornece de maneira intuitiva a criação de filtros específicos que se aplicam ao fluxo de pacotes e que determinam para quais VNFs os pacotes são encaminhados. É apresentada também a implementação de uma VNF denominada de VNF-Filter que, além de participar na instanciação e remoção de VNFs, classifica e encaminha pacotes para funções adequadamente. São apresentados experimentos de avaliação de desempenho, bem como uma comparação qualitativa com uma plataforma NFV tradicional baseada em OpenStack.*

1. Introdução

As redes de computadores normalmente utilizam diversos serviços disponibilizados como *middleboxes*, como *firewalls*, *gateways* e sistemas de detecção de intrusão [Sekar et al. 2012]. Estes serviços representam uma importante fração das despesas de capital (*CAPital EXpenditures* - CAPEX) e despesas operacionais (*OPerational EXpenditures* - OPEX) [Cotroneo et al. 2014] de uma rede, já que cada um desses componentes requer hardware específico. Além disso, muitos destes serviços de rede implementados em hardware são complexos para implantar, gerenciar e solucionar problemas

[Sherry et al. 2012]. Por fim, existe um desperdício dos recursos disponíveis no sistema, já que os *middleboxes* não fornecem uma maneira fácil para ajustar os seus recursos de acordo com a demanda.

Network Function Virtualization (NFV) surge como uma tecnologia para resolver estes problemas. A tecnologia NFV utiliza técnicas de virtualização para disponibilizar serviços de rede através de dispositivos virtuais. Estes serviços são implementados e disponibilizados através de processos virtuais que executam em hardware genérico (e.g., arquitetura x86). Dessa forma, a inclusão, alteração e/ou atualização de serviços não necessitam da instalação de novos equipamentos específicos. As principais vantagens da NFV incluem melhorias na flexibilidade e custos da rede, já que dispositivos antes implementados em hardware são substituídos por dispositivos virtuais, denominados de *Virtualized Network Functions* (VNF). Em contraste com os serviços de rede presentes em uma rede tradicional, as VNFs podem ser implementadas e executadas utilizando uma camada de virtualização, sendo então mais fáceis de serem gerenciadas e operadas [Turchetti and Duarte Jr 2017]. Além disso, a flexibilidade da rede é melhorada na medida em que serviços de rede são facilmente instanciados, utilizados e removidos.

Na prática, a maioria das ferramentas existentes para a supervisão do ciclo de vida das VNFs são extremamente complexas, muitas vezes dependendo de múltiplas camadas, incluindo uma infraestrutura de nuvem (como o OpenStack [OpenStack 2017]), um serviço dedicado para as funções virtualizadas (como o Tacker [Tacker 2017]) e além disso há outros módulos para configurar e monitorar aspectos diversos das funções virtualizadas. A pergunta que nos fizemos para desenvolver o presente trabalho foi: qual o sistema mínimo necessário para a instanciação e remoção de VNFs, que não necessite de múltiplas plataformas complexas subjacentes para sua execução? Sendo assim, é proposta a *On-The-Fly NFV*, uma ferramenta que acreditamos ser tão simples quanto possível para a instanciação e remoção de VNFs *on-the-fly* em tempo real. *On-The-Fly NFV* permite a criação de regras – conjunto de filtros que são aplicadas aos fluxos de pacotes que passam pela rede. Através das regras criadas, os fluxos de pacotes são encaminhados para as diversas VNFs adequadamente. A ferramenta possui uma interface gráfica que permite a utilização de suas funcionalidades bem como a visualização de métricas provenientes das VNFs.

A fim de fornecer estas funcionalidades, foi implementada uma VNF denominada de *VNF-Filter*. Esta VNF é responsável pela instanciação e remoção de outras VNFs, bem como pela classificação e encaminhamento dos pacotes da rede aos serviços necessários. A partir desta VNF, resultados experimentais avaliam o desempenho da ferramenta *On-The-Fly NFV*. Além disso, é apresentada uma comparação qualitativa com uma plataforma NFV tradicional baseada em OpenStack.

O restante deste trabalho está organizado da seguinte maneira. A Seção 2 apresenta a ferramenta *On-The-Fly NFV* e a função virtualizada *VNF-Filter*. A Seção 3 mostra os resultados experimentais. A Seção 4 apresenta uma comparação qualitativa com uma plataforma NFV tradicional. Por fim, a Seção 5 apresenta as conclusões do trabalho.

2. A Ferramenta *On-The-Fly NFV*

Quando implementamos uma função virtualizada de rede e executamos suas funcionalidades, surge a necessidade de criar procedimentos que permitem realizar o gerenciamento

do ciclo de vida e preferencialmente, que as tarefas de gerenciamento não interrompam o funcionamento da rede. Atualmente, as plataformas que oferecem este tipo de gerenciamento são complexas e dependem de uma grande infraestrutura subjacente.

Neste contexto, o presente trabalho demonstra a experiência de criar uma ferramenta que não dependa de múltiplos elementos para sua execução. A ferramenta *On-The-Fly NFV* permite instanciar e remover VNFs *on-the-fly* em tempo real, sem a interrupção de outros serviços da rede. O objetivo é facilitar a instanciação e remoção de múltiplas VNFs.

A ferramenta permite também a criação de regras. Uma regra é um conjunto de filtros que determinam para qual VNF um fluxo de pacotes deve ser encaminhado. Os filtros disponíveis inicialmente são: IP de origem e destino, porta de origem e de destino (caso exista) e protocolo de comunicação. Além disso, uma VNF estará associada a cada regra, definindo para onde o pacote deve ser encaminhado. A Figura 1 mostra a interface gráfica da ferramenta *On-The-Fly NFV*. Nesta interface é possível utilizar todas as funcionalidades disponíveis, bem como visualizar métricas provenientes das VNFs por meio de texto simples ou gráficos.

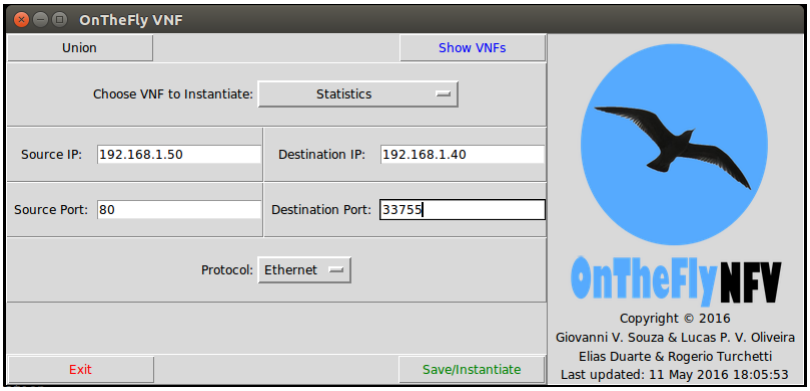


Figura 1. Interface gráfica da ferramenta *On-The-Fly NFV*.

A Figura 2 mostra a arquitetura da ferramenta. A estratégia utilizada é encaminhar o fluxo de dados dos *hosts* para a *VNF-Filter*. Através do protocolo *OpenFlow* [McKeown et al. 2008], para cada requisição *packet-in* feita ao controlador, uma regra *OpenFlow* é instalada no *switch* especificando que este fluxo de pacotes seja encaminhado a *VNF-Filter*. Desse modo, a *VNF-Filter* pode encaminhar todo o fluxo de pacotes as demais VNFs, baseando-se nas regras criadas.

A abordagem utilizada na ferramenta *On-The-Fly NFV* faz com que o controlador instale regras *OpenFlow* no *switch* especificando que cada fluxo de pacotes seja encaminhado para a *VNF-Filter*. A *VNF-Filter* por sua vez é responsável por diversas tarefas, como o recebimento de pacotes encaminhados pelos *switches* e a classificação dos pacotes de acordo com as regras criadas na ferramenta. Além disso, a *VNF-Filter* deve encaminhar os pacotes para as demais VNFs. É importante notar que apesar do protocolo *OpenFlow* também permitir a classificação dos pacotes, as regras utilizadas pela *On-The-Fly NFV* é genérica, sendo possível especificar outros campos sem da modificação do protocolo.

A *VNF-Filter* é composta por dois processos denominados de *process-sniffer* e

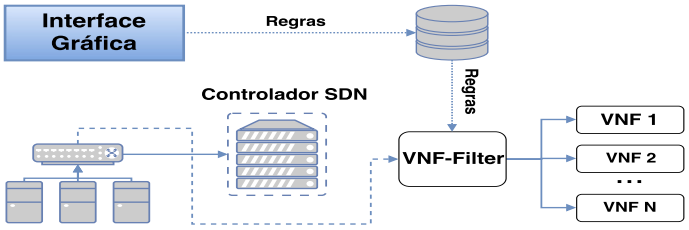


Figura 2. Arquitetura da ferramenta *On-The-Fly NFV*.

process-filter. Em primeiro lugar, todos os pacotes recebidos pelo *process-sniffer* são encaminhados para o *process-filter*. O *process-filter* recebe, processa e classifica todos os pacotes recebidos. Finalmente, uma vez que a etapa de classificação termina, o pacote pode ser encaminhado para a respectiva VNF. Como exemplo, considere uma regra que especifica que pacotes TCP provenientes do endereço IP `10.0.0.1` são encaminhados à VNF de *Deep Packet Inspection* (DPI). Qualquer pacote que se enquadre nesta regra é encaminhado para a VNF especificada. Caso contrário, o pacote é simplesmente descartado.

3. Resultados Experimentais

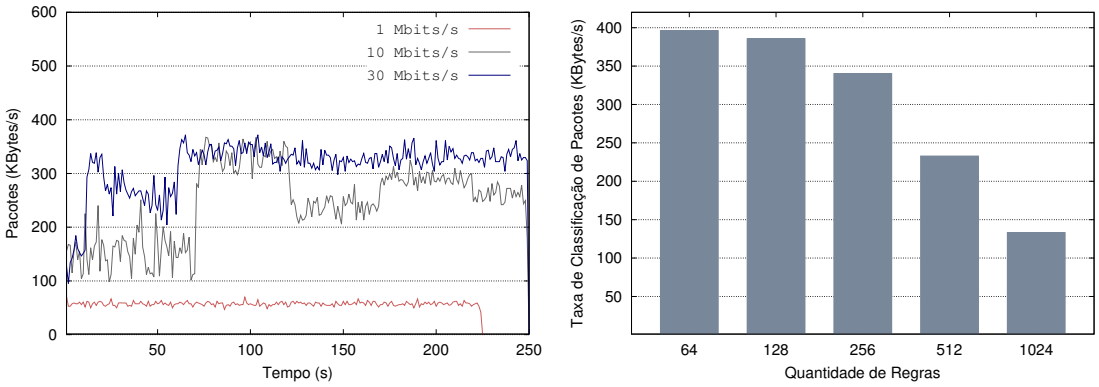
O objetivo desta seção é apresentar resultados experimentais obtidos com a ferramenta *On-The-Fly NFV*. A ferramenta foi desenvolvida em *Python* e seu código fonte pode ser acessado[1]. A rede SDN é criada a partir da ferramenta *Mininet*[2] e consiste de 3 *switches* com 3 *hosts* cada. Foi utilizado o controlador SDN *Ryu*[3].

A demonstração exemplificará as funcionalidades da ferramenta *On-The-Fly NFV*. Para isso, serão criadas diversas regras e uma VNF de estatística da rede será instanciada. Além disso, um fluxo de dados entre dois *hosts* será criado. Para os experimentos reportados foi utilizada uma máquina com a seguinte configuração: processador AMD FX-4300 com 4 núcleos, 8 GB de memória RAM e sistema operacional *Ubuntu* 16.04. O fluxo de pacotes foi gerado a partir da ferramenta *iperf*. Todos os testes enviam pacotes durante um intervalo de 100 segundos.

A Figura 3(a) mostra a taxa de classificação obtida para diferentes taxas de transferência utilizando uma única regra. Para uma taxa de transferência de 1 MBps, a taxa de classificação dos pacotes permanece constante. Isso mostra que a quantidade de pacotes recebidos é semelhante à quantidade de pacotes processados. Conforme a taxa de transferência aumenta, a taxa de classificação também aumenta, porém torna-se variável. Isso acontece devido a quantidade de pacotes recebidos ser muito maior do que a quantidade de pacotes que a *VNF-Filter* classifica, gerando perda de desempenho em alguns pontos.

Outro parâmetro que influencia na taxa de classificação é o número de regras processadas. A fim de determinar qual a perda de desempenho conforme o número de regras, a rede foi submetida a um fluxo constante de 30 MBps enquanto o número de regras varia. A Figura 3(b) mostra a média (considerando três execuções) da taxa classificação de pacotes obtida. É possível observar que conforme aumenta a quantidade de regras,

[1]https://github.com/giovannivenancio/otfnfv
[2]http://mininet.org/
[3]https://osrg.github.io/ryu/



(a) Variando a taxa de transferência.

(b) Variando o número de regras.

Figura 3. Taxa de classificação de pacotes da ferramenta *On-The-Fly NFV*.

a quantidade de pacotes classificados por segundo diminui. Essa perda de desempenho se agrava quando são criadas mais do que 256 regras, já que o tempo para a *VNF-Filter* executar as suas tarefas torna-se muito maior para processar o fluxo recebido.

4. Uma Comparação Qualitativa com uma Plataforma NFV Tradicional

O instituto ETSI (*European Telecommunications Standards Institute*) tem coordenado esforços para padronizar uma arquitetura NFV. Uma das principais funcionalidades desta arquitetura é o suporte ao controle do ciclo de vida de VNFs [ETSI 2015]. Essas VNFs são gerenciadas pelo NFV-MANO, módulo responsável por oferecer interfaces de comunicação e abstração dos recursos computacionais [ETSI 2014]. O NFV-MANO possui um módulo específico para realizar a gerência das VNFs em operação, denominado *VNF Manager* (VNFM). Em especial, o VNFM oferece as funcionalidades necessárias para gerenciar o ciclo de vida das VNFs [ETSI 2014].

Até então foram propostas várias soluções que realizam o gerenciamento de VNFs de acordo com as especificações ETSI [Tacker 2017, OpenBaton 2017, ONAP 2017]. Apesar de oferecerem diversas funcionalidades, estas plataformas são complexas em sua instalação, configuração e uso, além de exigir uma grande quantidade de recursos computacionais para o seu funcionamento. Ainda, é exigido do usuário a execução de diversos procedimentos para execução das operações de gerenciamento, onde é necessário informar parâmetros específicos do sistema.

Como referência, considere o VNFM Tacker [Tacker 2017] que executa suas funções, por padrão, em um ambiente OpenStack [OpenStack 2017]). Considere também a operação de instanciação de uma VNF. No Tacker, esta operação exige uma série de procedimentos. O primeiro procedimento consiste em criar um *VNF Descriptor* (VNFD) e o adicionar ao catálogo NFV. Após, uma VNF é criada baseando-se no VNFD criado anteriormente. Por fim, a configuração e execução de software desta VNF devem ser realizados manualmente. Por outro lado, como o objetivo da ferramenta *On-The-Fly NFV* é ser tão simples quanto possível, a criação da VNF é feita a partir da interface gráfica disponível, no momento da criação de um filtro. Esta operação abstrai todos os procedimentos descritos acima e exige apenas que o usuário informe qual VNF será instanciada, a partir de uma lista de VNFs disponíveis. Além disso, é notável a quantidade de recursos computacionais exigidos para execução da plataforma OpenStack e Tacker. Seguindo a

ideia de simplicidade no desenvolvimento da ferramenta, a quantidade de recursos computacionais consumido pela *On-The-Fly NFV* é desprezível.

Após comparar uma abordagem simplista com uma plataforma NFV complexa e compatível com a arquitetura definida pela ETSI, é importante considerar se tais plataformas são realmente necessárias em todos os cenários. Em casos onde existem poucas funções a serem executadas (e.g., ambiente de experimentação, redes de pequeno porte), a utilização de soluções mais simples como a *On-The-Fly NFV* pode ser suficiente.

5. Conclusão

Neste trabalho foi apresentada a *On-The-Fly NFV*, uma ferramenta simples para o gerenciamento de VNFs. A ferramenta proposta permite a criação de regras que são aplicadas aos fluxos de pacotes e determinam o encaminhamento para VNFs. O componente chave da ferramenta é a *VNF-Filter*, responsável por instanciar e remover outras VNFs, bem como classificar e encaminhar os pacotes da rede aos serviços necessários. Através de regras *OpenFlow*, pacotes são encaminhados para a *VNF-Filter*, que processa e encaminha fluxos adequadamente às demais VNFs. Todas as funcionalidades estão disponíveis a partir de uma interface gráfica, onde é possível também visualizar resultados provenientes das VNFs. Resultados experimentais mostram que o número de regras criadas na *VNF-Filter* possui influência desprezível na taxa de transferência dos pacotes. Além disso, é possível melhorar a taxa de classificação da ferramenta através da utilização de outro algoritmo para classificação de pacotes, bem como implementar a *VNF-Filter* de maneira replicada. Por fim, uma análise qualitativa compara a *On-The-Fly NFV* com uma plataforma tradicional de NFV e demonstra a possibilidade de utilizar ferramentas mais simples em determinados ambientes NFV.

Referências

[Cotroneo et al. 2014] Cotroneo, D., De Simone, L., Iannillo, A., Lanzaro, A., Natella, R., Fan, J., and Ping, W. (2014). Network function virtualization: Challenges and directions for reliability assurance. In *Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on*. IEEE.

[ETSI 2014] ETSI, N. (2014). Network functions virtualisation (nfv); management and orchestration. *NFV-MAN*, 1:v0.

[ETSI 2015] ETSI, N. (2015). Network functions virtualization (nfv) infrastructure overview. *NFV-INF*, 1:V1.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2).

[ONAP 2017] ONAP (2017). Onap. `https://www.onap.org/`. Accessed: 2017-11-21.

[OpenBaton 2017] OpenBaton (2017). Openbaton. `https://openbaton.github.io/`. Accessed: 2017-11-21.

[OpenStack 2017] OpenStack (2017). Openstack. `https://www.openstack.org/`. Accessed: 2017-11-24.

[Sekar et al. 2012] Sekar, V., Egi, N., Ratnasamy, S., Reiter, M. K., and Shi, G. (2012). Design and implementation of a consolidated middlebox architecture. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

[Sherry et al. 2012] Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., and Sekar, V. (2012). Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of the ACM SIGCOMM 2012*.

[Tacker 2017] Tacker (2017). Tacker. `https://wiki.openstack.org/wiki/Tacker`. Accessed: 2017-11-21.

[Turchetti and Duarte Jr 2017] Turchetti, R. C. and Duarte Jr, E. P. (2017). Nfv-fd: Implementation of a failure detector using network virtualization technology. *International Journal of Network Management*, 27(6).