

Multi-SFC: Orquestração de SFCs Distribuídas sobre Múltiplas Nuvens em Múltiplos Domínios com Múltiplas Plataformas NFV

Alexandre Huff^{1,2}, Giovanni Venâncio¹, Elias P. Duarte Jr.¹

¹Universidade Federal do Paraná (UFPR) – Programa de Pós-Graduação em Informática
Caixa Postal 19081 – CEP: 81531-980 – Curitiba – PR – Brasil

²Universidade Tecnológica Federal do Paraná – Câmpus Toledo (UTFPR)
CEP: 85902-490 – Toledo – PR – Brasil

alexandrehuff@utfpr.edu.br, gvsouza@inf.ufpr.br, elias@inf.ufpr.br

Abstract. *The NFV-MANO architecture, which is widely adopted by NFV (Network Function Virtualization) solutions, includes the NFVO (NFV Orchestrator) element to support the Service Function Chaining (SFC) orchestration. In this work we propose a strategy for the orchestration of Multi-SFCs: SFCs distributed over multiple clouds, possibly in multiple domains, orchestrated by NFVOs of multiple NFV platforms. Existing solutions are monolithic because they are restricted to SFCs composed of Virtual Network Functions (VNFs) from a single cloud/domain/orchestrator. The proposed strategy establishes tunnels by instantiating VNFs connecting segments of the Multi-SFC, allowing the secure communication across multiple domains. Several management tasks, including configuring NFV infrastructures, are transparent to the operator. Experimental results are presented for the evaluation of the proposed approach in terms of performance and resource consumption.*

Resumo. *A arquitetura NFV-MANO, amplamente adotada por soluções NFV (Network Function Virtualization) inclui o elemento NFVO (NFV Orchestrator) para apoiar a orquestração de SFC (Service Function Chaining). Este trabalho propõe uma estratégia para a orquestração de Multi-SFCs: SFCs distribuídas sobre múltiplas nuvens, possivelmente em múltiplos domínios, orquestradas por NFVOs de múltiplas plataformas NFV. Soluções existentes são monolíticas, pois se restringem a SFCs compostas por VNFs (Virtual Network Functions) de uma única nuvem/domínio/orquestrador. A estratégia proposta estabelece túneis através da instanciação de VNFs em pontos específicos da Multi-SFC, permitindo a instanciação e comunicação segura de recursos em múltiplos domínios. Diversas tarefas de gerenciamento, incluindo a configuração das infraestruturas NFV são transparentes para o operador. Resultados experimentais avaliam a solução, comparando a abordagem proposta em termos de desempenho e consumo de recursos.*

1. Introdução

A Virtualização de Funções de Rede (NFV - *Network Function Virtualization*) permite a implementação em software de diversas funções que tradicionalmente são executadas em *middleboxes*. Cada instância de uma função virtualizada de rede é denominada VNF (*Virtual Network Function*) e pode ser executada por sistemas de

virtualização em hardware de prateleira, aumentando a sua flexibilidade e diminuindo custos [Chiosi et al. 2013, Martins et al. 2014, Yousaf et al. 2017]. Neste contexto, a ETSI (*European Telecommunications Standards Institute*) propôs a arquitetura do *framework* NFV-MANO (NFV - *Management and Orchestration*) para o gerenciamento e orquestração de VNFs.

Uma das principais responsabilidades do NFV-MANO é coordenar a composição de VNFs para formar uma SFC (*Service Function Chaining*). Uma SFC consiste de uma composição de VNFs e do encaminhamento do tráfego entre as mesmas em uma ordem pré-definida para disponibilizar serviços de rede [Halpern and Pignataro 2015]. Na SFC, o tráfego da rede é encaminhado para a próxima função da cadeia de VNFs através de um identificador de fluxo, enquanto que no roteamento convencional as decisões de encaminhamento geralmente são baseadas apenas no endereço IP de destino.

Em geral, as VNFs da composição de uma SFC são instanciadas em uma mesma nuvem e dentro de um mesmo domínio [Moens and De Turck 2014, Bari et al. 2015, Kuo et al. 2018]. Entretanto, esta abordagem é limitada, mesmo por que existe um limite computacional de VNFs que podem ser executadas em uma máquina física [Ghaznavi et al. 2017]. Dependendo da complexidade do serviço, uma SFC pode ser composta por um grande número de VNFs, gerando alto consumo de recursos computacionais e degradando o desempenho da rede. Em alguns casos é necessário instanciar parte das SFCs fora da nuvem, por exemplo em ambientes de *Edge Computing*. Neste contexto podem ser fornecidos serviços de rede virtualizados em sistemas de computação com diferentes arquiteturas, tais como *Cloudlets*, *Mobile Edge Computing* e *Fog Computing* [Dolui and Datta 2017]. Dispositivos destes sistemas podem ser limitados em termos de recursos e não serem capazes de hospedar SFCs com um grande número de VNFs.

Neste trabalho é proposta a Multi-SFC: uma solução para permitir a composição de SFCs sobre múltiplas nuvens, possivelmente de múltiplos domínios administrativos orquestrados por múltiplas plataformas NFV. Estas plataformas – tais como o Tacker [Tacker 2018], Open Source MANO (OSM) [ETSI 2018], Open Baton [Fokus and Tu 2018], entre outras – implementam um NFVO (*NFV Orchestrator*) definido na arquitetura NFV-MANO. Na prática, a composição de SFCs em diferentes plataformas NFV é complexa e exige conhecimentos específicos das particularidades dos orquestradores NFV envolvidos. Além disso, como uma SFC pode executar entre domínios administrativos distintos, a configuração global da SFC (*e.g.*, encaminhamento de tráfego entre os segmentos das SFCs em cada nuvem/domínio/plataforma) deverá ser coordenado entre os operadores de rede de cada domínio. Ainda, embora a ETSI tenha definido interfaces para a comunicação entre orquestradores NFV de diferentes domínios administrativos [Haitao and Mann 2018], existe a dificuldade na composição de SFCs em múltiplas plataformas, uma vez que estas possuem particularidades específicas de operação e implementam diferentes modelos de dados.

A solução proposta utiliza uma abordagem holística e define um *framework* que permite abstrair a composição e o gerenciamento das Multi-SFCs distribuídas em diferentes nuvens de maneira transparente para o usuário operador de rede. A configuração da infraestrutura NFV para permitir encaminhamento do tráfego entre as VNFs de uma SFC distribuída sobre múltiplas nuvens/domínios/orquestradores também é abstraída pela abordagem proposta. Neste contexto, o *framework* realiza o estabelecimento automático

de túneis VPN (*Virtual Private Network*) através da instanciação de VNFs nas conexões de entrada e saída de segmentos específicos da SFC. A principal vantagem em utilizar a abordagem holística está em abstrair as configurações de baixo nível para a criação de SFCs entre múltiplas nuvens. Dessa forma, o gerenciamento da Multi-SFC é realizado de maneira simples, transparente e unificada, independente da plataforma NFV utilizada.

Foi implementado um protótipo da abordagem Multi-SFC que permite criar uma SFC sobre múltiplas nuvens, múltiplos domínios e múltiplas plataformas NFV. Resultados experimentais avaliam a Multi-SFC em termos de utilização de recursos computacionais e o desempenho da rede, em especial a vazão e latência. Os resultados mostram que a vazão está diretamente relacionada com o desempenho e latência dos *links* da VPN. Resultados também mostram que o protótipo implementado não introduz sobrecarga significativa de recursos computacionais durante a sua execução.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados no contexto de SFCs distribuídas. A proposta holística e a avaliação experimental são descritas nas Seções 3 e 4, respectivamente. Por fim, a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Diversas soluções NFV implementam o bloco funcional NFVO da arquitetura NFV-MANO. Entretanto, nenhuma delas oferece suporte para a execução distribuída de SFCs sobre múltiplas nuvens e domínios entre diferentes plataformas NFV. Os trabalhos a seguir propõem plataformas e *frameworks* para solucionar alguns destes desafios.

Em [Francescon et al. 2017] é proposto o *framework* X-MANO para a orquestração de serviços de rede em múltiplos domínios. De modo geral, o X-MANO somente permite encadear serviços de rede já existentes e que foram disponibilizados pelas interfaces dos orquestradores NFV de cada domínio. Isto pode gerar desperdício de recursos computacionais e energia, uma vez que o tráfego da rede pode atravessar VNFs desnecessárias. Embora a arquitetura proposta para o X-MANO envolva a orquestração de serviços de rede que executam em múltiplos domínios, em [Baggio et al. 2017] os autores apresentam um estudo de caso simples realizado com a plataforma NFV Open Baton [Fokus and Tu 2018]. Neste estudo, as VNFs que fazem a conexão inter-domínios devem ser instanciadas e configuradas manualmente pelos administradores de rede.

Salsano e outros [Salsano et al. 2017] propuseram o *framework* RDCL 3D (*Reusable Functional Blocks Description and Composition Language Design, Deploy and Direct*). Este *framework* permite editar e validar descritores de serviços, bem como implantar SFCs com VNFs executando sobre a plataforma ClickOS [Martins et al. 2014]. Embora o RDCL 3D também possibilite a composição de VNFs em diferentes plataformas NFV, existe a necessidade da modificação do *openvim* para executar o ClickOS. Ainda, o RDCL 3D utiliza descritores proprietários para descrever seus projetos de SFC e implementa a lógica para a composição e validação das SFCs no lado cliente.

O orquestrador NFV TeNOR [Riera et al. 2016] é um projeto da plataforma T-NOVA [Xilouris et al. 2014] para o gerenciamento de serviços de rede e o controle do ciclo de vida de VNFs sobre infraestruturas virtualizadas distribuídas. O orquestrador TeNOR automatiza a instanciação de SFCs, além de gerenciar e otimizar os recursos computacionais das VNFs que compõem a SFC. O TeNOR propõe uma solução baseada em

mapeamento para instanciar de maneira eficiente as VNFs de uma SFC sobre múltiplos PoPs (Pontos de Presença). No entanto, os autores não descrevem cenários reais que utilizam tecnologias de virtualização. Além disso, a plataforma TeNOR não considera a integração entre diferentes orquestradores NFV.

Open Platform for NFV (OPNFV) [Linux Foundation 2018] é uma solução NFV da Linux Foundation que desenvolve uma plataforma para simplificar o desenvolvimento e implantação de componentes NFV. O projeto OPNFV fornece uma plataforma NFV de referência que permite a interoperabilidade entre soluções NFV de diferentes desenvolvedores. No entanto, além da sua complexa instalação e configuração, o OPNFV não possui suporte para a execução de SFCs em múltiplas nuvens ou domínios administrativos.

O Blue Planet MDSO (*Multi-Domain Service Orchestration*) [Blue Planet 2018] propõe um *framework* para o gerenciamento e composição de serviços de rede entre múltiplos domínios e múltiplas infraestruturas NFV. Apesar de facilitar a configuração e instanciação de SFCs, o *framework* requer a utilização da sua própria solução de orquestração NFV. Além disso, o consumo de recursos computacionais para a orquestração de serviços de rede multi-domínio é elevado, necessitando ao menos 20 CPUs e 64 GiB de memória RAM.

Em [Sonkoly et al. 2015] os autores utilizam redes SDN (*Software-Defined Networks*) para distribuir SFCs em múltiplas nuvens e domínios administrativos. Entretanto, não é possível garantir que a interconexão das redes de nuvens distintas sempre sejam realizadas por redes SDN devido à obrigatoriedade destes equipamentos na rede, impossibilitando, por exemplo, que redes tradicionais com dispositivos legados sejam utilizadas neste cenário. Por fim, em [Guerzoni et al. 2016] é proposto um *framework* para a orquestração de SFCs entre múltiplos domínios. Entretanto, este *framework* não prevê mecanismos para facilitar a composição de SFCs nos múltiplos domínios.

Apesar dos vários avanços realizados para permitir a execução distribuída de SFCs, até então nenhuma solução é completa, de maneira a fornecer todo o *framework* necessário para a composição e orquestração de serviços de rede distribuídos sobre múltiplos domínios e diferentes plataformas NFV. Enquanto que algumas soluções abordam a execução dos serviços de rede em múltiplos domínios, estas são limitadas em termos de interoperabilidade entre diferentes plataformas NFV.

3. Orquestração de SFCs sobre Múltiplos Domínios e Plataformas NFV

Ainda não há no contexto de NFV soluções completas para a execução e gerenciamento de SFCs de forma distribuída em múltiplas plataformas NFV. As próprias plataformas possuem diversas lacunas relacionadas à orquestração de SFCs que executam sobre múltiplas nuvens e domínios. A solução Multi-SFC proposta neste trabalho tem por objetivo permitir a instanciação e gerência de SFCs sobre múltiplas nuvens, múltiplas plataformas NFV e múltiplos domínios administrativos. As próximas seções apresentam detalhes da arquitetura Multi-SFC, bem como a implementação do protótipo desenvolvido.

3.1. A Arquitetura Multi-SFC

A abordagem Multi-SFC permite que uma SFC seja distribuída em múltiplos segmentos. Um segmento é instanciado em um domínio e conectado a segmentos de outros domínios. A segmentação pode ser observada na Figura 1. Na medida em que há grande quantidade

de plataformas NFV disponíveis, diferentes domínios podem compor seus segmentos utilizando diferentes tecnologias e plataformas NFV.

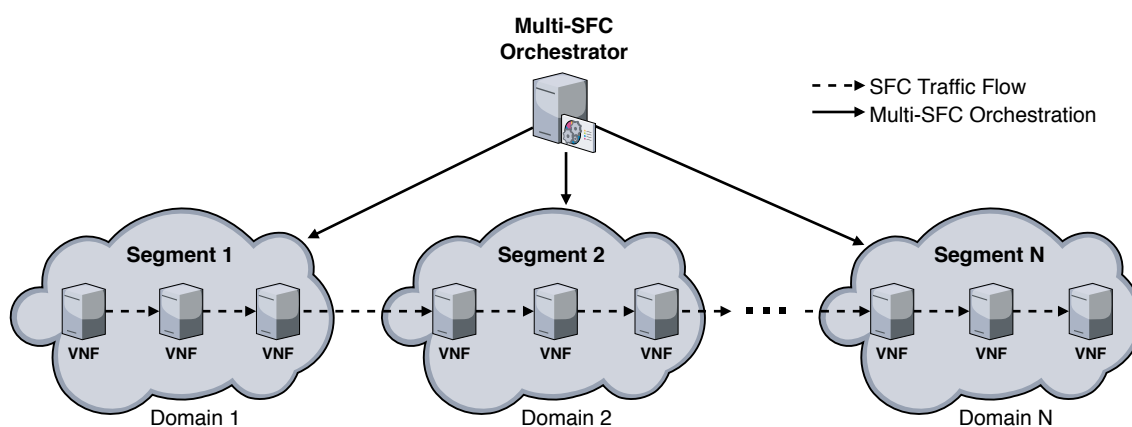


Figura 1. Multi-SFC: Segmentação.

Na Multi-SFC, o usuário compõe uma SFC que consiste de uma sequência pré-determinada de VNFs. Na abordagem tradicional centralizada, todas as VNFs da composição são instanciadas em um mesmo domínio, enquanto que a Multi-SFC divide a SFC entre múltiplos segmentos, um por domínio, conectados entre si. Cada segmento consiste de uma fração de VNFs da mesma composição. Vale destacar que a Multi-SFC é escalável, na medida em que permite distribuir a utilização de recursos computacionais e possibilita criar SFCs com maior número de VNFs.

A Figura 2 mostra a arquitetura de uma Multi-SFC composta por dois domínios. Neste exemplo, uma SFC composta originalmente por 6 VNFs é distribuída entre dois segmentos. O componente principal da solução é o *Multi-SFC Orchestrator*, utilizado para o gerenciamento das Multi-SFCs. O *Multi-SFC Orchestrator* fornece ao usuário uma interface de comunicação com funções em alto nível para a composição e gerenciamento do ciclo de vida de SFCs distribuídas sobre múltiplos domínios. Na arquitetura Multi-SFC, a comunicação inter-domínios é realizada através de VPNs, permitindo além de privacidade e segurança, o roteamento entre os múltiplos segmentos da SFC. Utilizando a nomenclatura da arquitetura da SFC da IETF, estas VPNs executam as funcionalidades de SFF (*Service Function Forwarder*) inter-domínios. Vale destacar que não conhecemos implementações de SFC com suporte nativo a SFF multi-domínio. As VPNs utilizadas na Multi-SFC são instanciadas e configuradas como VNFs que também fazem parte da SFC distribuída. De forma intuitiva, a conexão de uma sequência de dois segmentos de uma Multi-SFC é feita pela inclusão de uma VNF implementando VPN na saída do primeiro segmento e de outra VNF implementando VPN na entrada do segundo segmento.

O *Multi-SFC Orchestrator* também executa a configuração automática do encaminhamento de tráfego inter-domínios. Essa configuração pode ser baseada em roteamento ou *bridge*. Os detalhes das estratégias de encaminhamento são descritos na Seção 3.2. A Figura 3 ilustra a arquitetura do *Multi-SFC Orchestrator* que é proposta como uma solução genérica alinhada com as definições do *framework* NFV-MANO da ETSI. Isto permite que o *Multi-SFC Orchestrator* seja utilizado com diferentes plataformas NFV e em inúmeras aplicações finais. A arquitetura NFV-MANO também é apresentada para ilustrar os pontos de comunicação com os módulos do *Multi-SFC Orchestrator*.

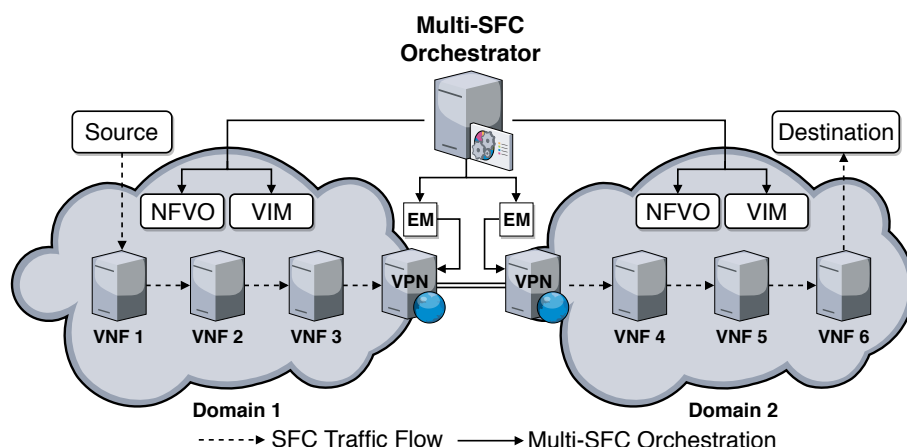


Figura 2. Arquitetura Multi-SFC entre pares de domínios.

O *Multi-SFC Core* é o módulo central da arquitetura do *Multi-SFC Orchestrator*. Este módulo tem responsabilidade de conduzir a composição das SFCs em múltiplos orquestradores e validar as requisições realizadas pelas aplicações cliente. O *Multi-SFC Core* provê uma interface de comunicação genérica e centralizada através de uma API (*Application Programming Interface*). Assim, as várias aplicações cliente com arquiteturas diversas implementadas em CLI (*Command Line Interface*), GUI (*Graphical User Interface*) ou sistemas inteligentes podem fazer o uso da arquitetura proposta para realizar a composição e gerenciamento do ciclo de vida de SFCs que atravessam múltiplos domínios orquestrados por múltiplas plataformas NFV.

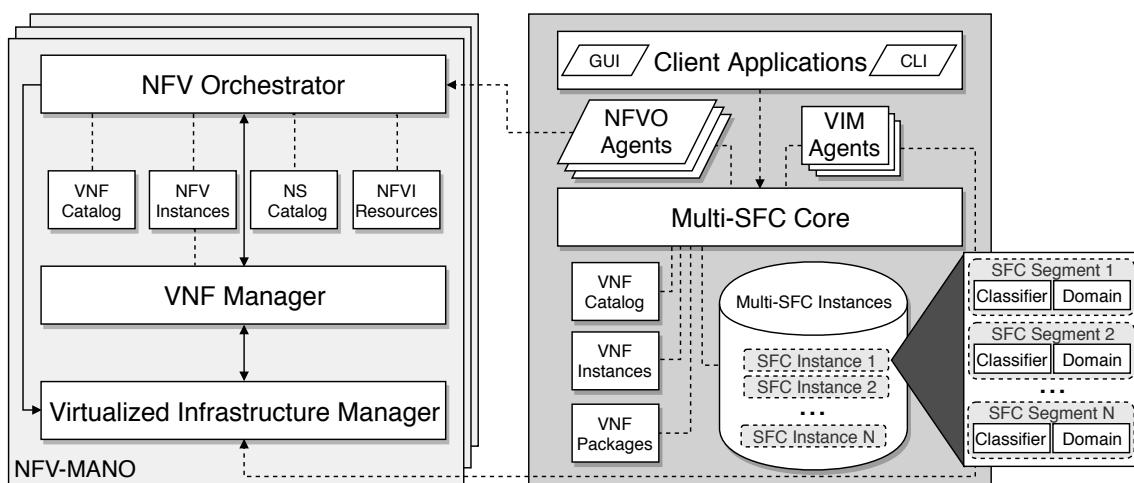


Figura 3. Arquitetura do Multi-SFC Orchestrator.

As Multi-SFCs são gerenciadas pelo *Multi-SFC Core* com o apoio dos módulos *NFVO Agents* e *VIM Agents*. O módulo *NFVO Agents* representa os agentes de comunicação com as respectivas implementações de cada orquestrador NFV, utilizados nos múltiplos domínios das Multi-SFCs. Estes agentes de comunicação implementam um conjunto de funcionalidades que permitem realizar a composição e orquestração de SFCs em cada domínio. As funcionalidades incluem desde o gerenciamento dos descritores de VNFs e SFCs, até a instanciação e destruição das mesmas. Os agentes do módulo *NFVO*

Agents consomem dados das interfaces de comunicação *northbound* dos seus respectivos orquestradores NFV que geralmente são providas através do modelo REST (*Representational State Transfer*). De modo geral, cada agente do módulo *NFVO Agents* tem a responsabilidade de traduzir as operações genéricas do *Multi-SFC Core* para as respectivas operações e particularidades de seu orquestrador NFV.

O módulo *VIM Agents* é responsável pela configuração automática e transparente do roteamento e dos túneis VPN nas extremidades de cada segmento da SFC. Esse tipo de configuração permite a interligação e encaminhamento do tráfego das SFCs entre os múltiplos domínios. Para cada plataforma que utiliza um tipo diferente de VIM, deve existir a respectiva implementação do seu agente de comunicação (*VIM Agent*) na arquitetura proposta. O módulo *VIM Agents* realiza a configuração das VPNs através de um *Element Management* (EM) implementado nas VNFs da VPN. Conforme previsto pela arquitetura NFV da ETSI [Quittek et al. 2014], o EM é responsável por realizar as operações FCAPS (*Fault, Configuration, Accounting, Performance, and Security*) das VNFs.

O elemento *VNF Catalog* é utilizado para gerenciar os meta-dados dos *VNF Packages* armazenados no repositório do *Multi-SFC Core*. O elemento *VNF Instances* tem a responsabilidade de mapear as diferentes instâncias de VNF executando nas diferentes plataformas NFV, que inclusive não necessitam estar associadas a uma SFC. Para gerenciar as Multi-SFCs, o *Multi-SFC Core* utiliza uma base de dados que armazena e mapeia as informações de cada instância de SFC. Cada instância possui informações relacionadas aos segmentos que compõem a SFC. Para cada segmento da SFC são armazenadas informações como, as VNFs que compõem aquele segmento, o classificador local do segmento da SFC e detalhes do domínio que irá hospedar o respectivo segmento.

As principais operações disponibilizadas pela API do *Multi-SFC Core* para as aplicações cliente para realizar a composição de SFCs distribuídas são destacadas a seguir. A operação “*compose_SFP*” permite o encadeamento das VNFs (*i.e.*, *Service Function Path* - SFP) em uma SFC a partir de informações dos identificadores dos *VNF Packages* armazenados no repositório da Multi-SFC. Para cada VNF que será incluída no SFP é necessário realizar uma requisição “*compose_SFP*”. A operação “*get_policies*” retorna a lista de políticas e restrições que podem ser utilizadas pelo classificador da SFC. A configuração dos classificadores é realizada pela operação “*post_policies*” na qual devem ser informadas as políticas de entrada de tráfego da SFC. A operação “*create_SFC*” recebe o descritor da SFC construído a partir das operações anteriores e tem a responsabilidade de instanciar as VNFs e os segmentos da SFC nos diferentes domínios e orquestradores NFV de maneira autônoma.

As funcionalidades abstraídas pela API do componente *NFVO Agents* referem-se basicamente às operações de instanciação, consulta e destruição de VNFs e SFCs nas diferentes plataformas NFV. A API do componente *VIM Agents* abstrai as configurações necessárias para realizar a interconexão dos segmentos da SFC que podem estar distribuídos em diferentes domínios. As principais operações abstraídas pela API do componente *VIM Agents* são descritas a seguir. A operação “*configure_networks*” é responsável por configurar as redes dos segmentos da SFC nas plataformas NFV dos diferentes domínios. A operação “*configure_routers*” é responsável por configurar os roteadores de cada rede, a fim de que o tráfego seja encaminhado para as VNFs que fazem a interconexão dos múltiplos segmentos da SFC. Por fim, a operação “*configure_endpoints*” abstrai

a configuração de protocolos de tunelamento para o encaminhamento do tráfego entre os múltiplos segmentos da SFC. Essencialmente a abordagem de tunelamento usando VNFs proposta na arquitetura Multi-SFC libera os operadores de rede de diversas configurações manuais necessárias para interconectar os diferentes domínios das Multi-SFCs.

3.2. Uma Implementação da Arquitetura Proposta

Um protótipo da arquitetura Multi-SFC foi implementado com o apoio de diversos facilitadores NFV, em particular: OpenStack, Tacker e OSM. Facilitadores NFV são ferramentas que viabilizam o desenvolvimento e implantação da NFV [Chiosi et al. 2012]. No contexto da SFC, a própria abordagem proposta pode ser considerada um facilitador SFC, uma vez que ela abstrai e viabiliza a composição e o gerenciamento do ciclo de vida de SFCs orquestradas em múltiplos domínios e múltiplas plataformas NFV.

A plataforma OpenStack [OpenStack 2018] é utilizada como VIM (*Virtualized Infrastructure Manager*) no contexto da arquitetura NFV-MANO da ETSI, pois permite virtualizar e gerenciar recursos computacionais, armazenamento e as redes utilizadas pelas VNFs. O Tacker [Tacker 2018] é um projeto oficial da plataforma OpenStack e é baseado na especificação da arquitetura NFV-MANO da ETSI. Este projeto tem a finalidade de implementar funcionalidades de *VNF Manager* e NFVO genéricos permitindo gerenciar o ciclo de vida de VNFs e orquestrar serviços de rede baseados em SFC na NFV. A plataforma OSM [ETSI 2018] também tem como objetivo a implementação da arquitetura NFV-MANO e é mantida pela ETSI. O OSM disponibiliza as funcionalidades de *VNF Manager* e NFVO genéricos e possibilita que diferentes implementações de VIMs sejam integrados em sua arquitetura. O componente *openvim* é a implementação padrão para realizar as funções de VIM no OSM. Entretanto, o *openvim* ainda não oferece funcionalidades para o gerenciamento e orquestração de SFCs. Diante disso, a abordagem proposta neste trabalho utiliza o OpenStack como a implementação do VIM para o OSM.

A implementação de um protótipo para a arquitetura do *Multi-SFC Orchestrator* ilustrada na Figura 3 foi realizada utilizando a linguagem Python e está disponível em <https://github.com/alexandre-huff/multi-sfc>. A interface do módulo *Multi-SFC Core* utiliza o modelo REST implementada a partir da biblioteca *Flask* da linguagem Python. Dois *NFVO Agents* foram implementados para trocar informações com os orquestradores NFV Tacker e OSM. Os *NFVO Agents* foram implementados com base na biblioteca *Requests* da linguagem Python. Também foi realizada a codificação de um *VIM Agent* para a plataforma OpenStack com base na biblioteca *Requests* da linguagem Python.

Foi implementado um EM para realizar a configuração automática das VPNs (que são VNFs) e permitir o encadeamento do tráfego entre os diferentes segmentos da SFC. O EM foi codificado com base na biblioteca *Flask* da linguagem Python e disponibiliza uma API REST para o gerenciamento do ciclo de vida da VPN. O túnel VPN entre os diferentes segmentos da SFC é estabelecido logo após a inicialização da máquina virtual e a obtenção dos respectivos *endpoints*. Assim, para cada par de segmentos da SFC distribuída é estabelecido um túnel VPN. Esta estratégia libera o operador de rede da necessidade de realizar a configuração manual do encaminhamento do tráfego da SFC entre as diferentes plataformas NFV e domínios administrativos.

A configuração e instanciação do túnel VPN são realizadas com base na solução [OpenVPN 2018] sobre o sistema operacional Linux Ubuntu. A implementação do

protótipo do *VIM Agent* realiza a configuração do OpenVPN utilizando roteamento entre as sub-redes de cada segmento da SFC. Além de configurar as VNFs da VPN, o *VIM Agent* também insere novas rotas nos roteadores virtuais das redes de cada segmento da SFC. Isto possibilita que o fluxo da rede seja encaminhado e retorne pelo mesmo túnel VPN, garantindo com isso a conectividade entre os nodos de origem e destino. A configuração destas rotas nos roteadores virtuais de cada segmento da SFC elimina a necessidade de modificações na configuração da rede dos nodos de origem e destino. Vale reiterar que a vantagem de se utilizar VPN proposta neste trabalho é viabilizar a comunicação entre os múltiplos segmentos da SFC distribuídos em múltiplos domínios administrativos, garantindo autenticidade, integridade e confidencialidade da comunicação inter-segmento.

Uma alternativa ao uso de roteamento em VPNs implementadas como VNFs para a conectividade de segmentos de uma Multi-SFC é o uso de *bridges* VPN. Este modelo de conectividade exige que o *VIM Agent* realize a instanciação e configuração de uma nova sub-rede para a comunicação entre os roteadores virtuais de cada segmento. Entretanto, o uso de *bridges* também requer que detalhes específicos de protocolos sejam configurados em cada roteador virtual para que os mesmos possam interoperar.

Embora a implementação do protótipo faça o uso de VPN para o encaminhamento do tráfego entre os múltiplos segmentos da SFC, a arquitetura genérica da Multi-SFC também permite que o encaminhamento do tráfego seja realizado usando outras tecnologias e protocolos de rede. Implementações de túneis utilizando protocolos tais como, *Virtual eXtensible Local Area Network* (VXLAN) e *Generic Routing Encapsulation* (GRE) poderiam ser utilizados neste contexto, substituindo as funções que implementam VPN.

A configuração dos classificadores dos segmentos da SFC é realizada automaticamente pelo módulo *Multi-SFC Core* em conjunto com os respectivos *NFVO Agents* de cada orquestrador NFV. Os classificadores são configurados com base nas políticas e restrições de tráfego gerais da SFC (ACLs - *Access Control Lists*) informada pelo usuário operador de rede. O fato de utilizar a VPN e roteamento para interligar os múltiplos segmentos permite que os classificadores sejam configurados com as mesmas regras. De modo geral, o processo de composição da SFC é transparente para o usuário que informa apenas a sequência de VNFs que compõem a SFC distribuída, e quando necessário, as conexões entre as mesmas caso existir mais de uma alternativa. Detalhes do processo de composição de uma SFC foram apresentados em um trabalho anterior [Huff et al. 2018].

4. Avaliação Experimental

Nesta seção são descritos os experimentos executados com o protótipo implementado com o objetivo de validar a proposta. São apresentados resultados que avaliam a vazão e latência do tráfego da SFC que atravessa diferentes domínios. Também foi avaliado o uso de recursos computacionais, em especial CPU e memória RAM para a composição de uma Multi-SFC. Os experimentos foram executados em duas máquinas físicas e outras duas máquinas virtuais executando sobre o sistema de virtualização Xen. As máquinas físicas foram utilizadas para executar as duas instâncias da plataforma OpenStack, simulando dois domínios distintos. Estas máquinas executaram as funcionalidades do bloco funcional VIM da arquitetura NFV-MANO da ETSI. A primeira máquina utilizada possui a seguinte configuração: CPU Intel(R) Core(TM) i7-6700HQ @ 3.5 GHz com 4 núcleos; 12 GiB de RAM; Ubuntu 16.04. A segunda máquina possui a configuração: CPU Intel(R)

Core(TM) i5 @ 3.2GHz com 2 núcleos; 6 GiB de RAM; Ubuntu 16.04.

O sistema de virtualização Xen possui as seguintes configurações: CPU AMD Opteron(tm) Processor 6136 @ 2.4GHz com 24 núcleos; 96 GiB de RAM; Ubuntu 14.04. Os experimentos foram executados para Multi-SFCs formadas por VNFs de duas plataformas NFV distintas, Tacker e OSM. Estas plataformas desempenharam as funções dos blocos funcionais VNFM e NFVO nos experimentos realizados. As máquinas virtuais usadas para executar as plataformas Tacker e OSM foram configuradas com 8 núcleos para processamento; 8 GiB de RAM; Ubuntu 16.04.

A Figura 4 ilustra o cenário criado para avaliar a vazão e latência do tráfego da SFC utilizando a abordagem Multi-SFC. Neste cenário os diferentes domínios são simulados por meio de duas instâncias da plataforma OpenStack instaladas nas duas máquinas físicas descritas anteriormente. Todas as VNFs usadas no experimento foram executadas utilizando máquinas virtuais com o sistema operacional Ubuntu Cloud 16.04, com 1 núcleo de processamento e 256 MiB de memória RAM. As VNFs não modificam o conteúdo do tráfego, apenas executam a função de encaminhamento. As máquinas físicas foram interconectadas por meio de uma rede *Gigabit*.

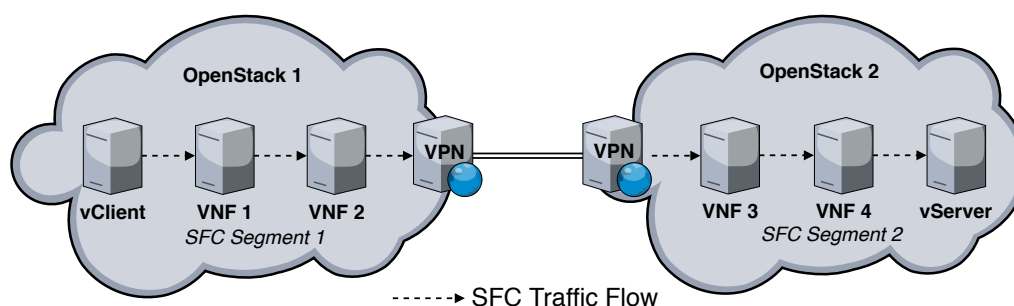


Figura 4. Cenário utilizado para avaliar a vazão e latência.

O primeiro experimento analisa o impacto na vazão da rede em diferentes cenários. A ferramenta *iPerf3* foi utilizada para mensurar o tráfego TCP entre a origem (*vClient*) e destino (*vServer*) nos diferentes domínios. O experimento foi executado 10 vezes e as médias são apresentadas no gráfico da Figura 5. Inicialmente foi medida a vazão TCP sem o encadeamento do tráfego na Multi-SFC e foi possível atingir em média a vazão de 931 Mbps. O mesmo experimento foi realizado nas máquinas que executam as duas instâncias do OpenStack e obteve-se vazão TCP idêntica ao experimento anterior.

Também foi analisada a vazão da interconexão dos múltiplos domínios por meio da VPN, representada na Figura 5 através da linha *VPN Link*. É possível observar que em média a vazão da VPN foi de 147,7 Mbps. A redução da vazão do tráfego TCP que passa através da VPN, neste experimento implementada pela solução OpenVPN, é resultado das chamadas às bibliotecas do sistema para realizar a criptografia, tarefas de encapsulamento de tráfego e pelo fato de utilizar apenas um núcleo de processamento. Também foi analisada a vazão do tráfego que atravessa todos os nodos da SFC representada pela linha Multi-SFC. Neste experimento foi possível atingir em média a vazão de 132,76 Mbps. Este resultado é esperado, uma vez que o tráfego é processado por cada VNF que compõe a SFC. Assim, é possível concluir que a maior degradação do desempenho da rede neste experimento foi devida à sobrecarga imposta pela OpenVPN.

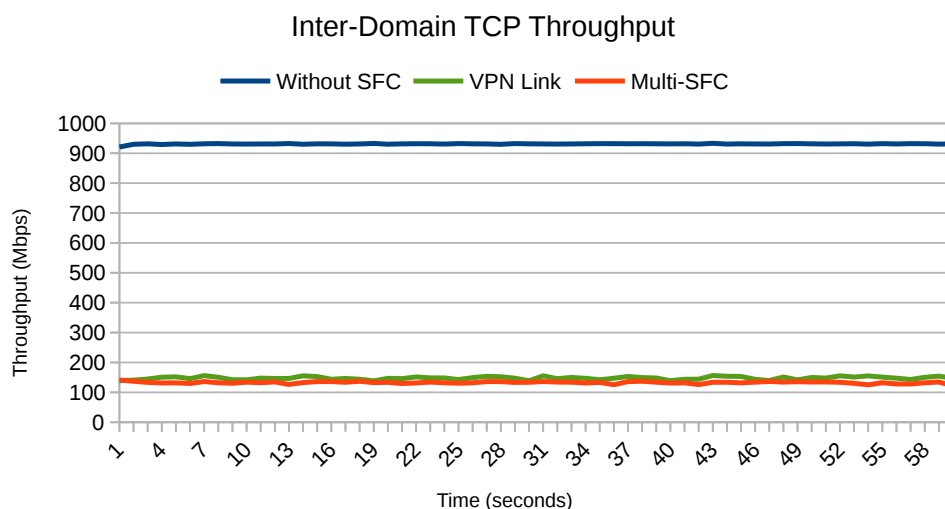


Figura 5. Vazão TCP em diferentes cenários de execução.

Além disso, foi realizado um experimento para medir a latência imposta pela abordagem Multi-SFC entre as máquinas virtuais *vClient* e *vServer*. A latência foi medida utilizando *ping* durante várias execuções de 60 segundos. O resultado de uma execução representativa é ilustrado na Figura 6 e comparada com a latência de mensagens ICMP entre as mesmas máquinas sem encadeamento do tráfego na SFC. É possível observar que a abordagem da Multi-SFC não aumenta significativamente a latência das mensagens ICMP. Em média, a latência na abordagem Multi-SFC foi de 3,872 ms, enquanto que a latência sem o encadeamento das mensagens na SFC foi em média de 1,063 ms. Este resultado também pode ser considerado o esperado, já que as mensagens são processadas por todas as VNFs da SFC.

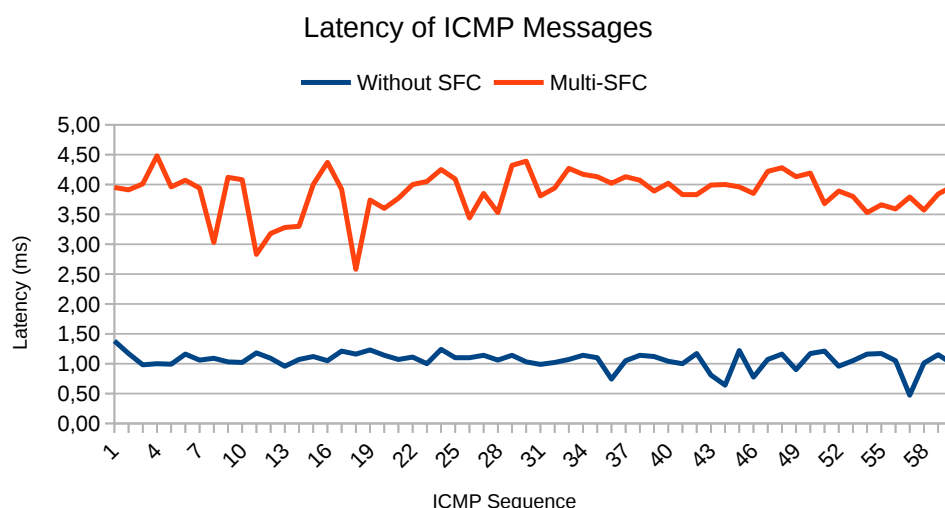


Figura 6. Latência de mensagens ICMP entre a abordagem Multi-SFC e sem SFC.

Também foi avaliado o uso dos recursos de processamento e memória requeridos pelo protótipo durante o processo de composição de uma SFC com 4 VNFs. O protótipo foi executado na máquina física com 4 núcleos e 12 GiB de RAM descrita anteriormente. Foram realizadas diversas execuções do experimento. A Figura 7 ilustra o resultado de

uma das execuções que foi selecionado aleatoriamente, uma vez que todas as execuções apresentaram padrão semelhante porém em instantes diferentes.

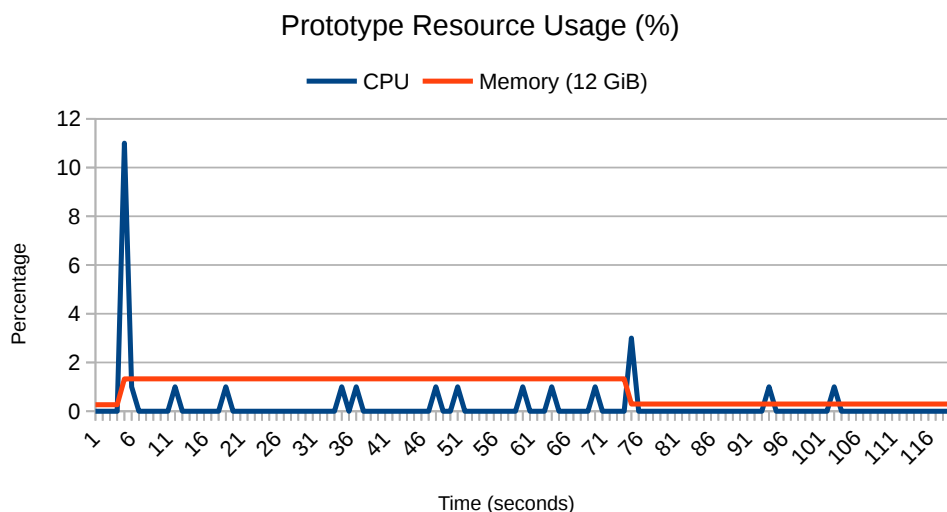


Figura 7. Porcentagem de uso de CPU e memória RAM do Protótipo.

Neste experimento foi utilizado um *pool* com 4 processos para paralelizar a instanciação de VNFs usando o *NFVO Agent* implementado para o OpenStack Tacker. Os dados de processamento e memória foram coletados utilizando a biblioteca *psutil* da linguagem Python, em intervalos de 1 segundo. Os recursos utilizados por todos os processos foram somados. Pode-se observar que o processamento atinge o pico de 11% na fase inicial da composição. Nesta fase são selecionadas as VNFs e as conexões para formar o descritor da SFC com diversas trocas de mensagens. Após a composição do descritor da SFC, é possível verificar o aumento no uso de memória RAM que passa de 0,26% (31,7 MiB) para 1,327% (176,1 MiB). Isto acontece devido à criação dos processos para a instanciação paralela das VNFs. É possível concluir que o uso de CPU permanece baixo com exceção do pico inicial, assim o protótipo não introduz sobrecarga significativa de processamento e memória. Vale ressaltar que a quantidade de processos usados para paralelizar as tarefas de instanciação das VNFs (*pool*) pode ser ajustada, permitindo balancear o tempo total de instanciação da SFC com os recursos utilizados.

5. Conclusão

Este trabalho apresentou a proposta da Multi-SFC para a composição de SFCs sobre múltiplas nuvens, possivelmente sobre múltiplos domínios administrativos orquestrados em múltiplas plataformas NFV. A arquitetura da Multi-SFC é proposta como uma solução extensível e alinhada à especificação da arquitetura NFV-MANO da ETSI. Isto possibilita o seu uso com diferentes plataformas NFV compatíveis com a arquitetura NFV-MANO e inúmeras aplicações finais. A solução proposta utiliza uma abordagem holística e permite abstrair a composição e o gerenciamento das SFCs em diferentes domínios administrativos de maneira transparente para o usuário operador de rede.

A proposta da Multi-SFC permite que SFCs complexas e que necessitam de um grande número de VNFs sejam distribuídas e executadas em diferentes infraestruturas NFV, inclusive em múltiplos domínios administrativos. A abordagem propõe a divisão

da SFC em múltiplos segmentos executados por diferentes plataformas NFV. Também é proposto o encadeamento do tráfego entre os diferentes segmentos da SFC através do estabelecimento automático de túneis VPN. Um protótipo da abordagem Multi-SFC foi implementado utilizando as plataformas NFV Tacker e OSM. Resultados experimentais mostram que a vazão da SFC está diretamente relacionada com a capacidade de vazão dos *links* da VPN. Além disso, a Multi-SFC não insere latência significativa entre os múltiplos segmentos da SFC e introduz pouca sobrecarga em termos de recursos computacionais.

Trabalhos futuros incluem a alocação eficiente de recursos no contexto de NFV (*e.g.*, elasticidade e migração de VNFs), a fim de otimizar a instanciação dos segmentos das Multi-SFCs. Fatores relevantes quando se consideram múltiplos domínios administrativos, por exemplo acordos de federação entre outros, também serão explorados. Ainda devem ser investigadas estratégias para melhorar o desempenho da VPN. Outra linha de trabalho futuro é a investigação de técnicas que utilizam o cabeçalho NSH (*Network Service Header*) no encaminhamento do tráfego entre os múltiplos domínios administrativos.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- Baggio, G., Francescon, A., and Fedrizzi, R. (2017). Multi-domain service orchestration with X-MANO. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–2, Bologna, Italy.
- Bari, M. F., Chowdhury, S. R., Ahmed, R., and Boutaba, R. (2015). On orchestrating virtual network functions. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 50–56. IEEE.
- Blue Planet (2018). <https://www.blueplanet.com/products/multi-domain-service-orchestration.html>. Acessado em dezembro de 2018.
- Chiosi, M., Clarke, D., Willis, P., Reid, A., Feger, J., Bugenhagen, M., Khan, W., Fargano, M., Cui, C., Deng, H., and Michel, U. (2012). Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action. White paper, ETSI.
- Chiosi, M., Wright, S., Clarke, D., Willis, P., Johnson, L., Bugenhagen, M., Feger, J., Khan, W., Chunfeng, C., and et al (2013). Network Functions Virtualisation (NFV): Network Operator Perspectives on Industry Progress. White paper, ETSI.
- Dolui, K. and Datta, S. K. (2017). Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6, Geneva, Switzerland.
- ETSI (2018). Open Source MANO. <https://osm.etsi.org/>. Acessado em dezembro de 2018.
- Fokus, F. and Tu, B. (2018). Open Baton: an open source reference implementation of the ETSI Network Function Virtualization MANO specification. <http://openbaton.github.io/>. Acessado em dezembro de 2018.
- Francescon, A., Baggio, G., Fedrizzi, R., Orsini, E., and Riggio, R. (2017). X-MANO: An open-source platform for cross-domain management and orchestration. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–6, Bologna, Italy.
- Ghaznavi, M., Shahriar, N., Kamali, S., Ahmed, R., and Boutaba, R. (2017). Distributed service function chaining. *IEEE Journal on Selected Areas in Communications*, 35(11):2479–2489.

- Guerzoni, R., Perez-Caparrós, D., Monti, P., Giuliani, G., Melian, J., and Biczók, G. (2016). Multi-domain orchestration and management of software defined infrastructures: A bottom-up approach. In *European Conference on Networks and Communications (EuCNC2016)*.
- Haitao, X. and Mann, A. (2018). Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on architecture options to support multiple administrative domains. GR NFV-IFA 028 V3.1.1. Group Report, ETSI.
- Halpern, J. and Pignataro, C. (2015). Service function chaining (sfc) architecture. RFC 7665, RFC Editor. <http://www.rfc-editor.org/rfc/rfc7665.txt>.
- Huff, A., Venâncio, G., da C. Marcuzzo, L., Garcia, V. F., dos Santos, C. R. P., and Duarte Jr., E. P. (2018). Uma abordagem holística para a definição de service chains utilizando Click-on-OSv sobre diferentes plataformas NFV. *Simpósio Brasileiro de Redes de Computadores (SBRC)*, v. 36.
- Kuo, T.-W., Liou, B.-H., Lin, K. C.-J., and Tsai, M.-J. (2018). Deploying chains of virtual network functions: On the relation between link and server usage. *IEEE/ACM Transactions on Networking (TON)*, 26(4):1562–1576.
- Linux Foundation (2018). <https://www.opnfv.org/>. Acessado em dezembro de 2018.
- Martins, J., Ahmed, M., Raiciu, C., Olteanu, V., Honda, M., Bifulco, R., and Huici, F. (2014). Clickos and the art of network function virtualization. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, NSDI'14*, pages 459–473, Berkeley, CA, USA. USENIX Association.
- Moens, H. and De Turck, F. (2014). Vnf-p: A model for efficient placement of virtualized network functions. In *10th International Conference on Network and Service Management (CNSM)*, pages 418–423.
- OpenStack (2018). OpenStack - open source software for creating private and public clouds. <https://www.openstack.org/>. Acessado em dezembro de 2018.
- OpenVPN (2018). <https://openvpn.net/>. Acessado em dezembro de 2018.
- Quittek, J., Bauskar, P., BenMeriem, T., Bennett, A., Besson, M., and et al (2014). Network Functions Virtualisation (NFV); Management and Orchestration. GS NFV-MAN 001 V1.1.1. Group specification, ETSI.
- Riera, J. F., Batallé, J., Bonnet, J., Días, M., McGrath, M., Petralia, G., Liberati, F., Giuseppi, A., Pietrabissa, A., Ceselli, A., et al. (2016). Tenor: Steps towards an orchestration platform for multi-pop nfv deployment. In *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, pages 243–250. IEEE.
- Salsano, S., Lombardo, F., Pisa, C., Greto, P., and Blefari-Melazzi, N. (2017). RDCL 3D, a model agnostic web framework for the design and composition of NFV services. In *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 216–222, Berlin, Germany.
- Sonkoly, B., Czentye, J., Szabo, R., Jocha, D., Elek, J., Sahhaf, S., Tavernier, W., and Risso, F. (2015). Multi-domain service orchestration over networks and clouds: a unified approach. *ACM SIGCOMM Computer Communication Review*, 45(4):377–378.
- Tacker (2018). Tacker - OpenStack NFV Orchestration. <https://wiki.openstack.org/wiki/Tacker>. Acessado em dezembro de 2018.
- Xilouris, G., Trouva, E., Lobillo, F., Soares, J., Carapinha, J., McGrath, M. J., Gardikis, G., Paglierani, P., Pallis, E., Zuccaro, L., et al. (2014). T-nova: A marketplace for virtualized network functions. In *2014 European Conference on Networks and Communications (EuCNC)*, pages 1–5. IEEE.
- Yousaf, F. Z., Bredel, M., Schaller, S., and Schneider, F. (2017). NFV and SDN - Key Technology Enablers for 5G Networks. *IEEE Journal on Selected Areas in Communications*, 35(11):2468–2478.