

# NFV-PEAR: Posicionamento e Encadeamento Adaptativo de Funções Virtuais de Rede

Gustavo Miotto<sup>1</sup>, Marcelo Caggiani Luizelli<sup>1</sup>  
Weverton Luis da Costa Cordeiro<sup>1</sup>, Luciano Paschoal Gaspar<sup>1</sup>

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

{gmiotto, mcluizelli, wlccordeiro, paschoal}@inf.ufrgs.br

**Resumo.** *O projeto de mecanismos flexíveis e eficientes para o posicionamento e encadeamento de funções virtualizadas de rede (VNFs) é central para o sucesso de Virtualização de Funções de Rede (Network Function Virtualization, NFV). As soluções existentes, no entanto, consideram custos fixos (e imutáveis) de processamento de fluxos e de largura de banda ao posicionar as VNFs em Pontos de Presença da Rede (N-PoPs). Essa limitação torna-se crítica em redes NFV com fluxos cujos comportamentos são altamente dinâmicos e nas quais os requisitos de processamento e os recursos disponíveis nos N-PoPs mudam constantemente. Para preencher essa lacuna é apresentado o NFV-PEAR, um framework para o posicionamento e encadeamento adaptativo de VNFs. O NFV-PEAR visa a (re)organizar periodicamente os posicionamentos e encadeamentos de VNFs previamente determinados, objetivando-se manter um desempenho fim-a-fim aceitável mesmo durante flutuações nos custos de processamento e nos requisitos dos fluxos. Paralelamente, busca-se minimizar as mudanças na rede (por exemplo, a realocação de VNFs ou de fluxos) realizadas para cumprir esse objetivo. Os resultados obtidos, a partir de uma avaliação experimental, mostram que o NFV-PEAR tem potencial para reduzir significativamente o número de mudanças na rede necessárias para assegurar o desempenho fim-a-fim esperado para os fluxos, garantindo assim o funcionamento estável dos serviços.*

**Abstract.** *The design of flexible and efficient mechanisms for proper placement and chaining of virtual network functions (VNFs) is key for the success of Network Function Virtualization (NFV). State-of-the-art solutions, however, consider fixed (and immutable) flow processing and bandwidth requirements when placing VNFs in the Network Points of Presence (N-PoPs). This limitation becomes critical in NFV-enabled networks having highly dynamic flow behavior, and in which flow processing requirements and available N-PoP resources change constantly. To bridge this gap, we present NFV-PEAR, a framework for adaptive VNF placement and chaining. In NFV-PEAR, one may periodically (re)arrange previously determined placement and chaining of VNFs, with the goal of maintaining acceptable end-to-end flow performance despite fluctuations of flow processing costs and requirements. In parallel, NFV-PEAR seeks to minimize network changes (e.g., reallocation of VNFs or network flows) done to fulfill this goal. The results obtained from an experimental evaluation provide evidence that NFV-PEAR has potential to significantly reduce the number of network changes required to ensure end-to-end flow performance, thus ensuring stable operation of network services.*

## 1. Introdução

Virtualização de Funções de Rede (*Network Function Virtualization*, NFV) é um paradigma relativamente novo que visa migrar funções tradicionalmente executadas por *hardware* especializado (*middleboxes*) para implementações centradas em *software*, rodando em instâncias de máquinas virtuais. Exemplos de funções incluem *firewall*, balanceamento de carga, *proxy*, detecção de intrusão, entre outras. Essa migração leva a vários benefícios, entre os quais a redução no custo de aquisição e operação de *hardware* para executar funções de rede, além de tornar mais flexível o processo de posicionamento e encadeamento dessas funções na infraestrutura [Han et al. 2015].

Embora recente, NFV experimentou diversos avanços em várias frentes, desde o projeto e implantação de funções virtuais de rede (*Virtual Network Functions*, VNFs) [Cohen et al. 2015, Bari et al. 2015] até a operação e gerência das mesmas [Open Networking Lab 2015, Zhang et al. 2016]. Apesar do progresso alcançado, muitas oportunidades de pesquisa permanecem. Um dos tópicos que merece destaque é o posicionamento e encadeamento de VNFs. Em resumo, trata-se de como melhor posicionar VNFs em pontos de presença na rede (*Network Points of Presence*, N-PoPs), e encadeá-las de modo que fluxos de dados sejam processados pelas VNFs na ordem especificada em documentos chamados de SFCs (*Service Function Chains*).

Além de especificar quais funções processarão determinados fluxos, e em qual ordem, as SFCs especificam, ainda, requisitos de poder computacional (nos N-PoPs em que cada função será executada), de largura de banda (entre N-PoPs) e de atraso fim a fim, para processar/encaminhar os fluxos em questão. Para materializar o encadeamento, o paradigma de Redes Definidas por Software (*Software Defined Networking*, SDN) [McKeown et al. 2008] pode ser considerado um aliado conveniente, por permitir que VNFs possam ser posicionadas e encadeadas de forma altamente flexível.

Uma limitação importante das soluções que lidam com o posicionamento e encadeamento de VNFs é que as mesmas consideram os custos de operação das VNFs, e os recursos disponíveis nos N-PoPs, como sendo fixos e imutáveis, ao decidir qual a melhor forma de instanciar um conjunto de SFCs na infraestrutura. Em ambientes reais, no entanto, tanto os custos quanto os recursos disponíveis podem mudar dinamicamente, conforme a carga à qual a rede é submetida. Dessa forma, os requisitos de processamento dos fluxos, tal como especificado nas SFCs, podem ser violados em momentos de alta demanda. Algumas soluções visam a superar essa lacuna analisando o comportamento de uma VNF individual (*firewall*, por exemplo) e ativando novas VNFs como resposta ao aumento da carga de fluxos. A busca individualizada por ótimos locais, como no exemplo do *firewall*, pode não levar a um ótimo global no que se refere ao equilíbrio entre oferta e demanda no posicionamento de funções e encadeamento de fluxos. Mais importante, pode levar a desperdício de recursos, pelo não aproveitamento da capacidade de processamento de fluxos ociosa em VNFs/N-PoPs.

Para suprir essa lacuna, neste artigo propõe-se NFV-PEAR, um *framework* para a orquestração adaptativa de VNFs. O objetivo é permitir o (re)arranjo (periódico) de funções/encadeamentos previamente alocados, em paralelo à instanciação de novas SFCs, visando a lidar com o comportamento dinâmico dos fluxos e flutuações na disponibilidade de recursos nos N-PoPs. Para isso, busca-se tanto (re)encadear fluxos através de VNFs com capacidade de processamento e de larguras de banda ociosas, bem como (re)organizar VNFs entre N-PoPs com maior quantidade de recursos disponíveis. As contribuições do artigo se desdobram em três: (i) um modelo formal para assegurar o melhor provisionamento de SFCs frente a alterações dinâmicas de demanda e/ou custos

associados aos equipamentos de rede (virtuais ou não); (ii) uma arquitetura de referência para o replanejamento e implantação de SFCs, agnóstica de tecnologias de virtualização e infraestrutura; e (iii) um mecanismo para aferir dados sobre a operação de VNFs. A partir de resultados obtidos com o NFV-PEAR via avaliação experimental, observou-se um melhor aproveitamento no uso de recursos da rede, ao mesmo tempo que tornou-se possível lidar mais adequadamente com fluxos se comportando de forma dinâmica, sem violar seus requisitos de processamento (tais como determinados nas SFCs).

O restante do artigo está organizado como segue. Na Seção 2 discute-se, a título de motivação, como o desempenho de funções virtualizadas de rede é fortemente influenciado por demandas distintas de tráfego de rede. Na Seção 3 apresenta-se um modelo de programação linear inteira para provisionamento adaptativo de VNFs. Na Seção 4 descreve-se o NFV-PEAR, a solução proposta para a orquestração adaptativa de VNFs. Na Seção 5 discorre-se sobre o ambiente utilizado para avaliação e os principais resultados alcançados. Na Seção 6 discute-se os principais trabalhos relacionados. Por fim, na Seção 7 são apresentadas considerações finais e perspectivas de trabalhos futuros.

## **2. O Impacto de Tráfego de Rede no Desempenho de Funções de Rede Virtualizadas**

No contexto de um arcabouço adaptativo como NFV-PEAR, é fundamental entender como funções virtualizadas de rede estão desempenhando. Como forma de motivar e ilustrar como indicadores de desempenho são afetados à medida que funções de rede são submetidas a diferentes padrões de tráfego, conduziu-se uma série de experimentos em um ambiente NFV típico. A seguir é apresentado um resumo das principais constatações, considerando as métricas de CPU, vazão de dados e perda de pacotes.

Os experimentos foram realizados em ambiente controlado formado por dois servidores A e B, equipados com 1 processador Intel Xeon E5-2420 (1.9GHz, 12 Threads e 15MB cache), 32GB de memória RAM (1333MHz), 1 HD SAS (1TB de capacidade), 1 interface de rede Gigabit e Fedora 21 (kernel 3.17). Os servidores foram conectados diretamente entre si utilizando um cabo de rede Gigabit. No servidor A, foi instalado um *hypervisor* KVM e um *switch* virtual Open vSwitch. No KVM foi instanciada uma máquina virtual com duas interfaces *ethernet* lógicas, 1 vCPU e 1GB de memória RAM. O *switch* virtual foi conectado à interface *ethernet* física e às interfaces da máquina virtual. No servidor B, foram instalados dois *containers* do tipo Docker, cada qual com uma interface *ethernet* lógica, e um *switch* Open vSwitch conectado aos *containers* e à interface *ethernet* física.

O cenário de experimentação foi configurado como segue. No servidor B, os *containers* funcionaram como cliente e servidor Iperf, configurados no modo UDP; no servidor A, a máquina virtual KVM encaminhava os pacotes entre suas interfaces. Durante o experimento, o tráfego iniciava no cliente Iperf, em B, passava pela máquina virtual, em A, e retornava ao servidor Iperf, em B. Optou-se por essa organização para que o custo de geração de tráfego não interferisse no desempenho da máquina virtual, alvo da medição. Ademais, executou-se dois experimentos: (i) máquina virtual com roteamento por tabelas de rotas e (ii) máquina virtual com roteamento por função de rede executando em Click Router [Kohler et al. 2000]. Os resultados apresentados representam a média de 30 repetições dos experimentos.

A partir dos resultados obtidos, apresentados na Figura 1, percebe-se que conforme a utilização de CPU se aproxima de 100%, perdas de pacote começam a ocorrer, tornando possível prever quando o desempenho da função de rede tende a degradar.

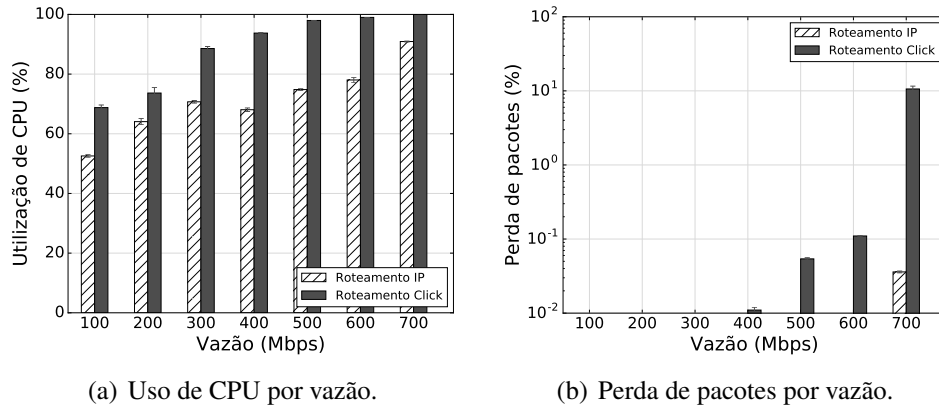


Figura 1. Métricas com impacto no desempenho das SFCs.

Nesses mesmos experimentos não foram observadas variações estatisticamente significativas para medições de ocupação de memória. Os resultados observados reforçam a importância do mecanismo adaptativo sendo proposto neste trabalho. NFV-PEAR permitirá o ajuste fino no provisionamento de encadeamentos de funções virtualizadas frente a degradações do desempenho de VNFs ou mudanças no perfil de tráfego.

### 3. Modelo Formal para Provisionamento Adaptativo de VNFs

Para lidar com o comportamento dinâmico dos fluxos de rede e reorganizar a alocação de VNFs sem desperdício de recursos físicos ou degradação de desempenho, faz-se necessário revisar os modelos e heurísticas de alocação existentes na literatura e usados para esse fim. Na primeira iteração sobre esse problema, adotou-se uma versão adaptada do modelo proposto por Luizelli et al. [Luizelli et al. 2015, Luizelli et al. 2017], o qual formula o posicionamento e encadeamento estático de funções virtualizadas usando um conjunto de restrições em um sistema linear.

A seguir é descrita a abordagem proposta para provisionamento adaptativo de VNFs, começando pela notação formal (Subseção 3.1) e seguida pelo modelo baseado em Programação Linear Inteira (Subseção 3.2). No modelo, letras superescritas  $P$  e  $S$  indicam símbolos relacionados aos recursos físicos e às SFCs, respectivamente. De forma análoga, as letras superescritas  $N$  e  $L$  referem-se aos N-PoPs/*endpoints*<sup>1</sup> e enlaces que os conectam. Por fim, utiliza-se a letra superescrita  $H$  para denotar subgrafos de uma SFC.

#### 3.1. Notação e Descrição do Modelo

**Informações de Entrada.** O modelo proposto por Luizelli et al. [Luizelli et al. 2015] considera como entrada um conjunto de SFCs  $Q$  e uma infraestrutura física  $p$ , esta última sendo uma tripla  $p = (N^P, L^P, S^P)$ .  $N^P$  representa o conjunto de nós da infraestrutura física (N-PoPs ou dispositivos de encaminhamento), enquanto que os pares  $(i, j) \in L^P$  representam enlaces físicos unidirecionais. Representa-se enlaces bidirecionais por meio de dois enlaces em direções opostas (isto é,  $(i, j)$  e  $(j, i)$ ). O conjunto de tuplas  $S^P = \{\langle i, r \rangle \mid i \in N^P \wedge r \in \mathbb{N}^*\}$  contém a localização (representada como um identificador único) de cada N-PoP. O modelo proposto captura as seguintes restrições relacionadas aos recursos físicos: poder computacional dos N-PoPs (representado por  $c_i^P$ ), bem como largura de banda e *delay* dos enlaces físicos representados, respectivamente, por  $b_{i,j}^P$  e  $d_{i,j}^P$ .

<sup>1</sup>Um *endpoint* é uma entidade que representa, em uma SFC, a origem/destino dos fluxos de dados.

SFCs  $q \in Q$  representam qualquer topologia de encaminhamento. Uma SFC é representada por uma tripla  $q = (N_q^S, L_q^S, S_q^S)$ . O conjunto  $N_q^S$  representa os nós virtuais (isto é, *endpoints* e VNFs), enquanto que o conjunto  $L_q^S$  representa os enlaces virtuais que os conectam. Note que cada SFC  $q$  apresenta, no mínimo, dois *endpoints*, os quais representam regiões específicas da infraestrutura. Os *endpoints* são conhecidos previamente e dados por  $S_q^S = \{\langle i, r \rangle \mid i \in N_q^S \wedge r \in \mathbb{N}^*\}$ . Além disso, cada SFC captura os seguintes requisitos relacionado aos recursos virtuais: processamento requerido por uma VNF  $i$  (representado por  $c_{q,i}^S$ ), largura de banda mínima requerida para o tráfego entre VNFs (ou *endpoints*)  $i$  e  $j$  (representada por  $b_{q,i,j}^S$ ), e latência máxima tolerável entre qualquer par de *endpoints* (representada por  $d_q^S$ ).

Por simplicidade, assume-se que cada SFC  $q$  apresenta um conjunto de caminhos virtuais representado por  $H_q$ . Cada elemento  $H_{q,i} \in H_q$  representa um caminho possível no subgrafo  $q$ , contendo uma origem e um destino. O subconjuntos  $N_{q,i}^H \subseteq N_q^S$  and  $L_{q,i}^H \subseteq L_q^S$  contém, respectivamente, as VNFs e os enlaces virtuais pertencentes ao caminho  $H_{q,i}$ .

O conjunto  $F$  denota os tipos de VNFs disponíveis (*firewall*, *proxy*, etc). VNFs podem ser instanciadas no máximo  $U_m$  vezes. Define-se a função  $f_{type} : N^P \cup N^S \rightarrow F$  para o tipo de uma dada VNF, a qual pode estar instanciada em um N-PoP ou ser parte de uma requisição de SFC. Adicionalmente, as funções  $f_{cpu} : (F \times U_m) \rightarrow \mathbb{R}_+$  e  $f_{delay} : F \rightarrow \mathbb{R}_+$  denotam poder computacional e atrasos relacionadas a uma função virtualizada de rede. Assume-se que as VNFs provisionadas podem atender uma demanda maior que a capacidade pré-dimensionada (*overcommitment*). O parâmetro  $\lambda \{\lambda \in \mathbb{R}_+, \lambda \geq 0\}$  define o percentual da capacidade das VNFs que pode ser violado.

**Informações de Saída.** A solução do modelo é expressa por conjuntos de variáveis binárias, descritos a seguir. Variáveis  $Y = \{y_{i,m,j}, \forall i \in N^P, m \in F, j \in U_m\}$  indicam o posicionamento de uma VNF. Isto é, se a instância  $j$  da função de rede  $m$  está mapeada no N-PoP  $i$ . De maneira similar, as variáveis  $\bar{Y} = \{\bar{y}_{i,m,j}, \forall i \in N^P, m \in F, j \in U_m\}$  indicam que o posicionamento atual de uma VNF  $j$  não foi alterado em relação a um dado posicionamento anterior representado por  $P_{i,m,j}^y$ .

As variáveis  $A^N = \{a_{i,q,j}^N, \forall i \in N^P, q \in Q, j \in N_q^S\}$  representam a atribuição de uma VNF requisitada (ou um fluxo) a uma VNF provisionada. Isto é, a variável indica se o nó  $j$  (sendo uma VNF ou um *endpoint*), requerido pela SFC  $q$ , é atribuído ao nó  $i$  (N-PoP). Similarmente, as variáveis  $\bar{A}^N = \{\bar{a}_{i,q,j}^N, \forall i \in N^P, q \in Q, j \in N_q^S\}$  indicam que a VNF  $j$  (ou o fluxo) da SFC  $q$  permanece alocada para uma mesma instância em relação a uma atribuição anterior denotada por  $P_{i,q,j}^{a^N}$ .

Por último, as variáveis  $A^L = \{a_{i,j,q,k,l}^L, \forall (i,j) \in L^P, q \in Q, (k,l) \in L_q^S\}$  indicam o provisionamento do encadeamento na infraestrutura física, *i.e.* se o enlace virtual  $(k,l)$  da SFC  $q$  está alocado no enlace físico  $(i,j)$ . Respectivamente, as variáveis  $\bar{A}^L = \{\bar{a}_{i,j,q,k,l}^L, \forall (i,j) \in L^P, q \in Q, (k,l) \in L_q^S\}$  indicam que os encadeamentos do enlace virtual  $(k,l)$  da SFC  $q$  permanece utilizando o enlace físico  $(i,j)$ .

### 3.2. Formulação do Modelo

O modelo proposto considera uma função multi-objetivo, a qual minimiza simultaneamente (i) os recursos consumidos na infraestrutura (*i.e.*, capacidade de processamento nos N-PoPs, nas VNFs e nos enlaces físicos), e (ii) as (possíveis) alterações nos mapeamentos decorrentes da flutuação da demanda alocada (*e.g.*, provisionamento de novas VNFs, readequação nos encadeamentos de SFCs e redistribuição de fluxos às VNFs existentes).

A primeira parte da função objetivo minimiza o consumo de recursos na rede. Essa minimização se materializa pela redução do número de VNFs alocadas (descritas pelas variáveis  $y$ ) e do comprimento dos encadeamentos realizados (descritos pelas variáveis  $a^L$ ). Por sua vez, a segunda parte da equação refere-se às alterações realizadas na infraestrutura e está definida por três componentes. O primeiro refere-se à minimização de alterações no posicionamento de VNFs já alocadas (descritas pelas variáveis  $\bar{y}$ ); o segundo refere-se à minimização nas modificações dos encadeamentos existentes (descritas pelas variáveis  $\bar{a}^L$ ); e o terceiro captura as modificações relacionadas com as (re)atribuições de fluxos (ou SFCs) às VNFs (descritas pelas variáveis  $\bar{a}^N$ ). Cada componente é ponderado, respectivamente, por  $\alpha$ ,  $\beta$  e  $\gamma$  de acordo com as prioridades estabelecidas.

**Objetivo:**

$$\begin{aligned} \text{Min.} \quad & \left( \sum_{i \in N^P} \sum_{m \in F} \sum_{j \in U_m} y_{i,m,j} + \sum_{(i,j) \in L^P} \sum_{q \in Q} \sum_{(k,l) \in L_q^S} a_{i,j,q,k,l}^L \right) + \\ & \left( -\alpha \cdot \sum_{i \in N^P} \sum_{m \in F} \sum_{j \in U_m} \bar{y}_{i,m,j} - \beta \cdot \sum_{(i,j) \in L^P} \sum_{q \in Q} \sum_{(k,l) \in L_q^S} \bar{a}_{i,j,q,k,l}^L - \gamma \cdot \sum_{i \in N^P} \sum_{q \in Q} \sum_{k \in N_q^S} \bar{a}_{i,q,k}^N \right) \end{aligned}$$

**Sujeito a:**

$$\sum_{m \in F} \sum_{j \in U_m} y_{i,m,j} \cdot F_{m,j}^{cpu} \leq C_i^P \quad (\forall i \in N^P) \quad (1)$$

$$\sum_{q \in Q} \sum_{j \in N_q^S: f(j)=f(m)} C_{q,j}^S \cdot a_{i,q,j}^N \leq \lambda \cdot \sum_{j \in U_m} y_{i,m,j} \cdot F_{m,j}^{cpu} \quad (\forall i \in N^P)(\forall m \in F) \quad (2)$$

$$\sum_{q \in Q} \sum_{(k,l) \in L_q^S} B_{q,k,l}^S \cdot a_{i,j,q,k,l}^L \leq B_{i,j}^P \quad (\forall (i,j) \in L^P) \quad (3)$$

$$\sum_{i \in N^P} a_{i,q,j}^N = 1 \quad (\forall q \in Q)(\forall k \in N_q^S) \quad (4)$$

$$a_{i,q,k}^N \cdot l = a_{i,q,k}^N \cdot j \quad (\forall \langle i,j \rangle \in S^P)(\forall q \in Q)(\forall \langle k,l \rangle \in S_q^S) \quad (5)$$

$$a_{i,q,k}^N \leq \sum_{m \in F} \sum_{j \in U_m: m=f(k)} y_{i,m,j} \quad (\forall i \in N^P)(\forall q \in Q)(\forall k \in N_q^S) \quad (6)$$

$$\sum_{j \in N^P} a_{i,j,q,k,l}^L - \sum_{j \in N^P} a_{j,i,q,k,l}^L = a_{i,q,k}^N - a_{i,q,l}^N \quad (\forall q \in Q)(\forall i \in N^P)(\forall (k,l) \in L_q^S) \quad (7)$$

$$\begin{aligned} \sum_{(i,j) \in L^P} \sum_{(k,l) \in L_{q,t}^H} a_{i,j,q,k,l}^L \cdot D_{i,j}^P \\ + \sum_{i \in N^P} \sum_{k \in N_{q,t}^H} a_{i,q,j}^N \cdot F_k^{delay} \leq D_q^S \end{aligned} \quad (\forall q \in Q)(\forall (N_{q,t}^H, L_{q,t}^H) \in H_q) \quad (8)$$

$$\overline{y_{i,m,j}} = P_{i,m,j}^y \cdot y_{i,m,j} \quad (\forall i \in N^P)(\forall m \in F)(\forall j \in U_m) \quad (9)$$

$$\overline{a_{i,q,j}^N} = P_{i,q,j}^{a^N} \cdot a_{i,q,j}^N \quad (\forall i \in N^P)(\forall q \in Q)(\forall k \in N_q^S) \quad (10)$$

$$\overline{a_{i,j,q,k,l}^L} = P_{i,j,q,k,l}^{a^L} \cdot a_{i,j,q,k,l}^L \quad (\forall (i,j) \in L^P)(\forall q \in Q)(\forall (k,l) \in L_q^S) \quad (11)$$

A seguir descreve-se os conjuntos de restrições que compõem o modelo. Os três primeiros referem-se às limitações de recursos da infraestrutura física. O conjunto de restrições (1) garante que o somatório de todas as instâncias de VNFs aprovionadas em um dado N-PoP não exceda a capacidade computacional disponível. O conjunto (2) garante que a demanda requerida pelos fluxos das SFCs não exceda a capacidade de processamento aprovionada para as VNFs. Note que a capacidade aprovionada das VNFs por ser ultrapassada (em momentos de alta demanda, por exemplo) por um fator  $\lambda$ . Por último, o conjunto (3) garante que as demandas dos encadeamentos aprovionados em um dado enlace físico não exceda a largura de banda disponível no enlace.

O conjunto de restrições (4)-(6) garante o posicionamento dos recursos virtuais. O conjunto de restrições (4) garante que cada elemento de uma SFC seja mapeado na infraestrutura. Por sua vez, o conjunto (5) garante que os *endpoints* das SFCs sejam mapeados nos dispositivos de rede localizados em regiões específicas da infraestrutura física. O conjunto (6) garante a disponibilidade de instâncias de VNFs nos N-PoPs em que as requisições das SFCs são mapeadas. Isto é, caso uma VNF requisitada por uma SFC seja mapeada em um dado N-PoP  $i$ , então (no mínimo) uma instância da VNF estará posicionada e executando em  $i$ .

As restrições referentes ao encadeamento das SFCs são descritas pelos conjuntos de restrições (7) e (8). O conjunto (7) garante que haja uma caminho válido na infraestrutura física entre todos os *endpoints* e VNFs da SFC. Por sua vez, o conjunto (8) garante

que os caminhos adotados para encaminhar o tráfego respeite os limites de atraso máximo entre os *endpoints*. A primeira parte da equação refere-se ao atraso proveniente dos enlaces físicos, enquanto que a segunda parte refere-se ao atraso oriundo do processamento de pacotes nas próprias VNFs.

Por fim, os conjuntos de restrições (9)-(11) determinam a similaridade do posicionamento e do encadeamento de SFCs em relação a um dado mapeamento anterior conhecido (denotado pelo conjunto  $P$ ). Os conjuntos (9), (10) e (11) definem, respectivamente, a similaridade das variáveis relacionadas ao posicionamento de VNFs (variáveis  $y$ ), relacionadas à atribuição de requisições às VNFs posicionadas (variáveis  $a^N$ ) e em relação aos encadeamentos adotados (variáveis  $a^L$ ). Note que o objetivo de tais equações é identificar os casos em que as variáveis da alocação invertem os valores assumidos de 1 para 0. Esses casos, em particular, identificam quando as alocações são modificadas.

#### 4. Posicionamento e Encadeamento Adaptativo de Funções Virtuais de Rede com NFV-PEAR

Tendo sido apresentado o modelo ILP para posicionamento e encadeamento adaptativo de VNFs, nesta seção introduz-se NFV-PEAR: uma proposta de arquitetura para implantação e orquestração de funções de rede. NFV-PEAR apoia-se no modelo ILP proposto para permitir a realocação dinâmica de funções de rede, em resposta a oscilações nas demandas de processamento de fluxos. É importante destacar que a arquitetura foi projetada em sintonia com os principais blocos construtores preconizados pela interface MANO (*Management and Orchestration*) do padrão ETSI [ETSI 2016].

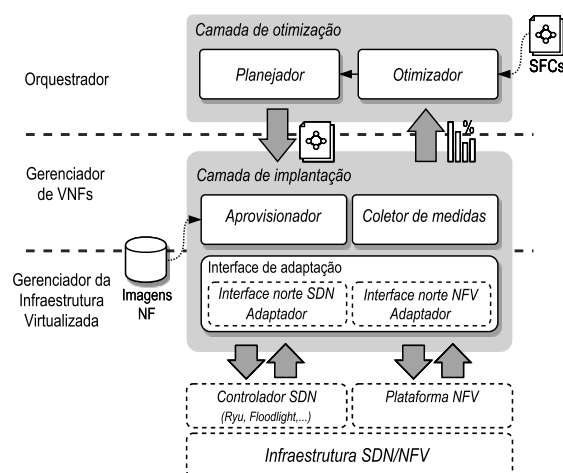


Figura 2. Visão geral da arquitetura proposta.

Uma visão geral da arquitetura proposta é ilustrada na Figura 2, destacando as camadas de *otimização* e de *implantação*. Elas são descritas em detalhes nas Subseções 4.1 e 4.2, respectivamente. A Figura 2 destaca ainda as interações com os controladores SDN e plataforma NFV em uso na infraestrutura, e a relação entre os elementos da arquitetura e os blocos construtores da interface MANO. A camada de otimização da arquitetura proposta, por exemplo, provê os serviços esperados para o bloco construtor “orquestrador” da interface MANO.

Outra peça-chave de NFV-PEAR são as medidas de desempenho, que permitem aferir o estado das VNFs em operação na infraestrutura e identificar realocações necessárias para responder a flutuações no volume de fluxos. As métricas propostas no



escopo deste artigo, e a metodologia empregada para aferir a importância das mesmas, são descritas na Seção 2.

#### 4.1. Camada de Otimização

A camada de otimização reúne os módulos responsáveis pela otimização e pelo planejamento da instanciação e do encadeamento de SFCs na infraestrutura. Observe nesse caso que tanto SFCs já implantadas como as em implantação são processadas por esses módulos, ao se (re)planejar a alocação de VNFs contidas nas mesmas.

O módulo *otimizador* é responsável por computar a melhor alocação possível de VNFs na rede, considerando o conjunto de SFCs mencionado anteriormente, bem como informações sobre o estado atual da infraestrutura (*endpoints*, N-PoPs/VNFs e recursos disponíveis nos mesmos, enlaces, etc.). Para esse fim, o otimizador implementa o modelo ILP discutido na seção anterior. A saída desse módulo – a solução para o modelo ILP no cenário dado – é repassada ao planejador.

O módulo *planejador* é responsável por determinar, algoritmicamente, a melhor forma de conduzir, na prática, as alterações necessárias na alocação de VNFs na rede e nos encadeamentos correspondentes. Seu objetivo é manter a infraestrutura em um estado próximo ao de operação ótima, efetuando, para isso, o mínimo de mudanças necessárias. Diversas estratégias podem ser adotadas para assegurar transições suaves entre estados que a infraestrutura deve assumir e, com isso, evitar interrupções. Tais não fazem parte do escopo deste artigo e serão detalhadas em um trabalho futuro.

#### 4.2. Camada de Implantação

A camada de implantação reúne os módulos responsáveis por provisionar as SFCs na rede física. O *aprovisionador* é responsável por implementar na rede as alocações de VNFs e os encadeamentos, conforme os mapeamentos de SFCs recebidos da camada de otimização. O módulo *coletor de medidas* implementa as funcionalidades de monitoração das VNFs implantadas na rede, consolidando estatísticas de operação das mesmas, as quais são repassadas para a camada de otimização.

Ambos os módulos comunicam-se com a *interface de adaptação* para realizar as atividades de orquestração/monitoração de VNFs na infraestrutura física. Essa interface é composta por dois sub-módulos, (i) *interface norte SDN*, responsável por traduzir requisições de instalação de encadeamentos e consultas de estado (por exemplo, nos *switches*) para o protocolo utilizado pelo controlador SDN, e (ii) *interface norte NFV*, responsável por adaptar requisições pertinentes às VNFs ao protocolo utilizado pelo controlador NFV da infraestrutura.

Para facilitar o processo de integração com outras soluções compatíveis com a interface MANO, os módulos da camada de implantação expõem uma interface de programação (API) para orquestração e implantação simplificada de VNFs. A API é projetada de modo a reduzir a complexidade de programação da rede SDN e facilitar o gerenciamento das funções virtuais. Em resumo, a API expõe métodos para instanciação de VNFs, posicionamento de VNFs nos N-PoPs e instanciação de encadeamentos.

### 5. Avaliação

Para aferir a eficácia e a efetividade de NFV-PEAR, definiu-se um processo sistemático de avaliação. Para tal, o modelo formalizado na Seção 3 foi implementado e executado no *CPLEX Optimization Studio* versão 12.4<sup>2</sup>. Os experimentos foram conduzidos em uma

<sup>2</sup><http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>

máquina com quatro processadores Intel i5 2.6 GHz, 8 GB de memória RAM, executando sistema operacional Ubuntu/Linux Server 11.10 x86\_64.

## 5.1. Carga de Trabalho

Para realizar os experimentos, adotou-se uma estratégia similar à empregada em trabalhos anteriores na área [Luizelli et al. 2015]. A infraestrutura física foi gerada com a ferramenta Brite<sup>3</sup> utilizando o modelo Barabasi-Albert (BA-2) [Albert and Barabási 2000]. O modelo adotado apresenta características topológicas semelhantes às infraestruturas típicas de ISPs (*Internet Service Providers*). As infraestruturas físicas consideradas contêm um total de 50 N-PoPs, cada um com capacidade computacional de 100%. Em média, cada rede apresenta 300 enlaces com capacidade uniforme de 10 Gbps e atrasos médios de 10ms. Os N-PoPs são posicionados em locais diversos na rede.

Considera-se dois tipos de imagens de VNFs disponíveis para instanciação. Para cada tipo de VNF, assume-se a disponibilidade de instâncias de pequena e grande capacidade computacional (demandando, respectivamente, 25% e 100% da capacidade computacional de um N-PoP). Para a avaliação conduzida, considera-se um conjunto de 20 SFCs sendo submetidas à infraestrutura. Os tipos das VNFs requisitadas pelas SFCs são aleatoriamente selecionadas dentre as possíveis. Cada VNF requisita entre 25% e 50% da capacidade de uma imagem instanciada em um N-PoP. As SFCs consideradas seguem uma topologia em linha com os seus *endpoints* selecionados aleatoriamente na infraestrutura física.

Para avaliar a capacidade do modelo em replanejar a infraestrutura com o mínimo de interrupções, considera-se que um percentual das SFCs aprovionadas alternam entre um modo de consumo considerado normal (ou pré-aprovionado) e um modo de consumo acima do planejado (por exemplo, demanda de pico). Nesse último caso, replanejamentos pontuais são necessários para manter o desempenho e a estabilidade do sistema. Compara-se o modelo proposto com o modelo de Luizelli et al. [Luizelli et al. 2015]. Naquele, quando há a necessidade de replanejamento, todas as SFCs são resubmetidas e aprovionadas na infraestrutura.

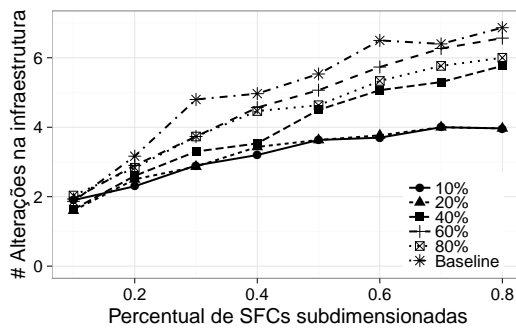
## 5.2. Resultados

A avaliação concentra-se principalmente na qualidade das soluções geradas pelo otimizador. Primeiramente, avalia-se o número de modificações necessárias na infraestrutura em função da variação na demanda. A Figura 3 apresenta a quantidade média de modificações relacionadas com: (i) o reposicionamento de VNFs (Figura 3(a)), (ii) a reatribuição de SFCs às VNFs (Figura 3(b)) e (iii) o reencadeamento de SFCs (Figura 3(c)). Varia-se a proporção de SFCs subdimensionadas (*i.e.*, aquelas que apresentam demandas maiores que o aprovionado) de 10% a 80% do total de SFCs alocadas na infraestrutura. Além disso, varia-se o percentual da demanda excedida de 10% a 80% (curvas distintas). Para esses experimentos, considera-se os valores de  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\lambda$  (do modelo ILP) iguais a 1.

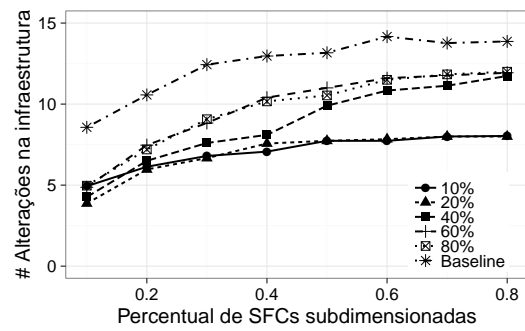
Observa-se que o número de alterações necessárias (eixo y) para readequar a rede à nova demanda é proporcional 1) ao percentual de SFCs com aumento de demanda e 2) aos valores de demanda excedidos (eixo x). Além disso, observa-se que o número de alterações relacionadas ao reposicionamento de VNFs é substancialmente menor em comparação ao observado para reatribuições de fluxos e reencadeamentos de SFCs. Isso indica a factibilidade do modelo proposto em ambientes reais, uma vez que o tempo requerido para instanciar (ou migrar) uma VNF é substancialmente maior (na ordem de

---

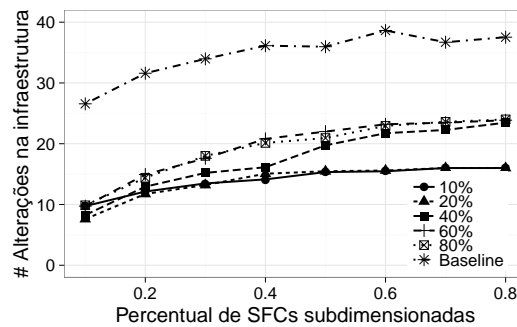
<sup>3</sup><http://www.cs.bu.edu/brite/>



(a) Alterações no posicionamento de VNFs.



(b) Alterações nas atribuições de SFCs às VNFs.



(c) Alterações nos encadeamentos das SFCs.

**Figura 3. Análise do replanejamento da alocação de SFCs.**

milissegundos a segundos) que o de reprogramar um dispositivo de encaminhamento (na ordem de milissegundos), por exemplo. Ainda, em comparação com o *baseline*, observa-se que o modelo proposto reduz em 25% o número de alterações relacionadas ao posicionamento de VNFs e em até duas vezes a quantidade de alterações relacionadas ao encadeamento e reatribuições de SFCs nas VNFs.

A seguir, avalia-se o impacto do fator de *overcommitment* (parâmetro  $\lambda$ ) de VNFs no replanejamento da infraestrutura. Note que quanto maior é o fator de *overcommitment*, maior é a chance de uma determinada VNF apresentar degradação de desempenho. Para esta avaliação, fixou-se o percentual de SFCs subdimensionadas em 80% e o percentual de aumento da demanda em 40%. O parâmetro  $\lambda$  é variado entre 0 e 40%. A Figura 4 ilustra o número de alterações necessárias para readequar a demanda de tráfego para esse cenário. Observa-se que quanto maior é o fator de *overcommitment*, menor é o número de alterações necessárias na infraestrutura. Por exemplo, com 10% de *overcommitment*, há uma redução de 30% no número total de alterações.

Por fim, discute-se o tempo médio necessário para encontrar uma solução ótima para o problema de replanejamento do encadeamento de SFCs. Em todos os experimentos, o tempo médio necessário para a resolução do modelo proposto permanece abaixo de 2 segundos. Apesar do problema apresentado e modelado ser NP-Difícil, tais resultados indicam que o modelo exato pode ser aplicado para instâncias do problema de pequena e média escala. Para ser aplicada em infraestruturas com escalas maiores, avaliações adicionais são necessárias para identificar os limites de tempo de computação.

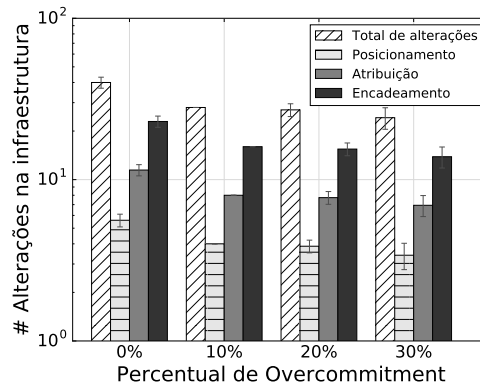


Figura 4. Impacto de *overcommitment* no replanejamento da infraestrutura.

## 6. Trabalhos Relacionados

Os trabalhos de pesquisa no âmbito de NFV podem ser organizados considerando várias perspectivas. Na área de orquestração, um dos esforços mais notórios é o *Open Platform for NFV* (OPNFV) [Open Networking Lab 2015]. Essa plataforma *open source* visa a fomentar a interoperabilidade entre as tecnologias habilitadoras de NFV (por exemplo, Open vSwitch, KVM e Xen) com as demais camadas da arquitetura proposta pelo ETSI [ETSI 2016] (por exemplo, de orquestração e de monitoramento).

Em paralelo, várias outras plataformas tem sido propostas para superar lacunas específicas na orquestração de VNFs, por exemplo o ClickOS [Martins et al. 2014], o OpenNetVM [Zhang et al. 2016] e o VirthPhy [Domicini et al. 2016]. O ClickOS, baseado no hipervisor Xen e usando funções virtuais escritas em Click [Kohler et al. 2000], visa a reduzir a sobrecarga de cópia de pacotes entre interfaces, permitindo alcançar vazão próxima da velocidade do enlace. O OpenNetVM, por sua vez, introduz uma camada de encaminhamento virtual para integrar o mecanismo de virtualização leve Docker à biblioteca de aceleração de pacotes Intel DPDK. Por fim, VirthPhy apresenta uma plataforma programável para *data centers* de pequeno porte, nos quais tanto as funções quanto os elementos de rede que as interconectam são virtualizados.

Na área de monitoração, uma das principais iniciativas é o NFV-VITAL [Cao et al. 2015]. Nesse trabalho, os autores propõem um *framework* para caracterizar o desempenho de VNFs executando em ambientes de nuvem. A partir dessa caracterização, torna-se possível (i) estimar a melhor alocação de recursos computacionais para executar as VNFs, e (ii) determinar o impacto de diferentes configurações de virtualização e de *hardware* no desempenho das VNFs. Outra iniciativa é o NFVPerf [Naik et al. 2016], uma ferramenta para detecção de gargalos em ambientes NFV. Por meio da análise dos fluxos de dados que transitam entre as VNFs, torna possível calcular vazão e atrasos médios, e assim possíveis degradações de desempenho em tempo real.

No âmbito de posicionamento e encadeamento de VNFs, vários trabalhos merecem destaque, entre os quais os de Bari et al. [Bari et al. 2015] e Luizelli et al. [Luizelli et al. 2015, Luizelli et al. 2017]. Bari et al. [Bari et al. 2015] descrevem o problema de orquestração de VNFs, que consiste em determinar o número de VNFs e suas localizações na rede de modo que os custos operacionais sejam ótimos. Os autores formulam o problema via sistema linear (*Integer Linear Programming, ILP*), e utilizam CPLEX e programação dinâmica para otimizar as alocações em ambientes NFV de menor escala. Mais recentemente, Luizelli et al. [Luizelli et al. 2017] abordaram o problema

para ambientes de larga escala, via proposta de um algoritmo de otimização que combine programação matemática e meta-heurísticas de busca.

Apesar dos avanços observados, as soluções existentes não abordam cenários de flutuações e gargalos localizados que ocorrem devido a variações no volume de fluxos em trânsito na rede. Uma estratégia *ad hoc* para lidar com essas flutuações é reexecutar os algoritmos de alocação de VNFs, e reorganizá-las conforme o resultado obtido. Apesar de efetiva, essa estratégia é computacionalmente mais cara (por reexecutar globalmente os algoritmos de otimização), e não permite reagir eficientemente à dinamicidade no comportamento dos fluxos. O trabalho de Rankothge et al. [Rankothge et al. 2015] é o que mais se aproxima de uma solução efetiva para esse problema. Nele, os autores utilizam algoritmos genéticos para introduzir funções de rede com capacidade de processamento escalável. No entanto, as funções de rede são consideradas de forma isolada, portanto sem levar em conta possíveis otimizações globais como, por exemplo, reencadear fluxos com requisitos similares para VNFs de maior capacidade.

Considerando as limitações discutidas acima, NFV-PEAR apresenta-se como uma solução para readequar a rede frente as variações de demanda, via identificação de gargalos no processamento de fluxos, reorganização do posicionamento e encadeamento de funções de rede localmente/globalmente, e visando à minimização da interrupção no processamento dos fluxos em trânsito.

## 7. Considerações Finais

Neste trabalho propôs-se NFV-PEAR, um *framework* para orquestração adaptativa de funções de rede em ambientes NFV. As contribuições deste trabalho se desdobram em (i) um modelo formal para assegurar o melhor provisionamento de SFCs frente a alterações dinâmicas de demanda e/ou custos associados aos equipamentos de rede, (ii) uma arquitetura de referência para o replanejamento e implantação de SFCs, agnóstica de tecnologias de virtualização e infraestrutura e (iii) um conjunto de métricas para representar dados sobre a operação de VNFs. Após a formalização do modelo ótimo para o problema de replanejamento adaptativo do encadeamento de funções virtualizadas de rede, realizou-se uma avaliação analítica. Os resultados obtidos evidenciaram que o modelo proposto contribui significativamente para uma redução no número de modificações na infraestrutura física (de até 25% no reposicionamento de VNFs e de mais de 200% no reencadeamento de encadeamentos de funções de rede).

Como perspectivas de trabalhos futuros, pretende-se materializar o modelo e a arquitetura propostos em um ambiente de orquestração real. O objetivo é avaliar métricas de desempenho como tempo de resposta e de reação do sistema. Outra proposta de trabalho futuro é estender a avaliação para identificar correlações na valoração dos parâmetros do modelo (em especial,  $\alpha$ ,  $\beta$  e  $\gamma$ ) na qualidade das soluções obtidas. Por fim, visa-se a desenvolver e integrar métodos de predição de demanda de tráfego ao modelo proposto.

## Referências

- Albert, R. and Barabási, A.-L. (2000). Topology of evolving networks: Local events and universality. *Physical Review Letters*, 85:5234 – 5237.
- Bari, M. F., Chowdhury, S. R., Ahmed, R., and Boutaba, R. (2015). On orchestrating virtual network functions. In *Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM)*, CNSM '15, pages 50–56.
- Cao, L., Sharma, P., Fahmy, S., and Saxena, V. (2015). NFV-VITAL: A framework for characterizing the performance of virtual network functions. In *2015 IEEE Conference*

- on *Network Function Virtualization and Software Defined Network (NFV-SDN)*, pages 93–99.
- Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015). Near optimal placement of virtual network functions. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1346–1354.
- Dominicini, C. K., Vassoler, G. L., Ribeiro, M. R., and Martinello, M. (2016). Virtphy: A fully programmable infrastructure for efficient nfv in small data centers. In *2016 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*.
- ETSI (2016). Network Functions Virtualisation (NFV). Available: <<http://www.etsi.org/technologies-clusters/technologies/nfv>>.
- Han, B., Gopalakrishnan, V., Ji, L., and Lee, S. (2015). Network function virtualization: Challenges and opportunities for innovations. *Communications Magazine, IEEE*, 53(2):90–97.
- Kohler, E., Morris, R., Chen, B., Jannotti, J., and Kaashoek, M. F. (2000). The click modular router. *ACM Transactions on Computer Systems (TOCS)*, 18(3):263–297.
- Luizelli, M., Bays, L., Buriol, L., Barcellos, M., and Gaspary, L. (2015). Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 98–106.
- Luizelli, M. C., da Costa Cordeiro, W. L., Buriol, L. S., and Gaspary, L. P. (2017). A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining. *Computer Communications*, 102:67 – 77.
- 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 *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 459–473, Seattle, WA. USENIX Association.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- Naik, P., Shaw, D. K., and Vutukuru, M. (2016). NFVPerf: Online Performance Monitoring and Bottleneck Detection for NFV. In *2016 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*.
- Open Networking Lab (2015). Open Platform for NFV (OPNFV). Disponível em <<https://www.opnfv.org/>>. Acesso em: 28 jul. 2016.
- Rankothge, W., Le, F., Russo, A., and Lobo, J. (2015). Experimental results on the use of genetic algorithms for scaling virtualized network functions. In *IEEE SFV-SDN*.
- Zhang, W., Liu, G., Zhang, W., Shah, N., Lopreiato, P., Todeschi, G., Ramakrishnan, K., and Wood, T. (2016). OpenNetVM: A Platform for High Performance Network Service Chains. In *Proceedings of the 2016 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*. ACM.