

Investigando Algoritmos de Balanceamento de Carga para Engenharia de Tráfego em Virtualização de Funções de Rede

Gabriel do Amaral Almeida¹, Weverton Luis da Costa Cordeiro¹,
Luciano Paschoal Gaspar¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 – Porto Alegre, RS – Brasil

{gaalmeida, weverton.cordeiro, paschoal}@inf.ufrgs.br

Abstract. *Network Function Virtualization (NFV) is a networking paradigm that reshapes the concept of network functions (such as firewalls, proxies and load balancers), shifting them from specialized appliances to software-centric solutions running on commodity hardware, using virtualization. In conjunction with Software Defined Networking (SDN), NFV becomes very convenient in enabling flexible placement of network functions in the infrastructure, and steering data flows between them. Beyond the huge potential of NFV, it brings along various novel challenges, for example the need to steer data flows between network functions in a rational manner (i.e., saving network resources and avoiding possible jams). In this work we investigate existing load balancing strategies, and propose the implementation of traffic engineering strategies for flow steering, designed with the goal of linking the flows over the network functions.*

Resumo. *Virtualização de Funções de Rede (Network Function Virtualization, NFV) é um paradigma que reformula o conceito de funções de rede, (como firewalls, proxies e balanceadores de carga), transferindo-as de equipamentos especializados para soluções centradas em software executando em servidores de prateleira, usando virtualização. Aliado ao conceito de Redes Definidas por Software (Software Defined Networking, SDN), NFV torna-se um paradigma bastante conveniente, por flexibilizar o posicionamento de funções de rede na infraestrutura, e o encadeamento de fluxos de dados entre as mesmas. Além das potencialidades, NFV traz consigo uma série de novos desafios, por exemplo a necessidade de encadear fluxos de dados entre as funções de rede de forma racional (isto é, economizando recursos de rede e evitando possíveis congestionamentos). Desta forma, este trabalho realiza um levantamento de estratégias de balanceamento de carga existentes e sugere a aplicação de estratégias projetadas em engenharia de tráfego com o objetivo de encadear os fluxos ao longo das funções de rede.*

1. Introdução

Virtualização de Funções de Rede (*Network Function Virtualization*, NFV) [Network Functions Industry Specification Group 2012] é um paradigma que visa conferir maior flexibilidade à gerência e à operação de funções de rede (por exemplo, *firewall*, *cache*, e *proxy*). O principal objetivo é migrar essas funções, que antes eram realizadas por equipamentos especializados e de alto custo, para soluções centradas em *software* e que possam executar em *hardware* de prateleira, usando virtualização.

Embora recente, NFV tem observado uma intensa atividade de pesquisa nos últimos três anos. Os esforços de investigação tem abordado vários aspectos independentes, por exemplo criar condições técnicas para implantar e executar funções

de rede em um ambiente virtualizado [Hwang et al. 2014], encadear fluxos de tráfego entre elas [Xia et al. 2015], permitir a migração consistente de fluxos de rede entre funções [Gember-Jacobson et al. 2014], otimizar o posicionamento e o encadeamento de funções de redes virtuais [Lewin-Eytan et al. 2015, Luizelli et al. 2015], controle de acesso de rede baseado em fluxos [Matias et al. 2015], funções de redes tolerantes a falhas [Sherry et al. 2015], etc. Em vários desses aspectos, NFV tem-se beneficiado de forma significativa de Software Defined Networking (SDN) [McKeown 2009], o qual tem apoiado o encadeamento automatizado de fluxos entre funções implantadas na rede e, consequentemente, aspectos-chave de gerenciamento tais como balanceamento de carga e observância a políticas de rede.

SDN permite separar o plano de controle do plano de dados da rede, respectivamente representados pelo controlador e pelos dispositivos de encaminhamento (roteadores, comutadores, etc.). Essa separação funciona devido à programabilidade do controlador, o qual é responsável por analisar os fluxos de dados recebidos pelos comutadores e informá-los (via mensagens de controle) por onde devem encaminhar cada fluxo. A consolidação das instruções que o controlador passa aos dispositivos de encaminhamento sucede-se por meio de regras instaladas nas tabelas de encaminhamento dos comutadores, usando protocolos abertos como o *OpenFlow* [McKeown et al. 2008]. O controlador é capaz de tomar tais decisões fundamentando-se na topologia, no estado atual de congestionamento ou até mesmo em políticas empregadas pelo administrador da rede.

Ao combinar SDN e NFV, torna-se possível encadear o tráfego (via regras *OpenFlow* instaladas nos comutadores via controlador) ao longo das funções virtualizadas de rede (*Virtualized Network Functions*, VNFs) implantadas nos pontos de presença da rede (*Network Points of Presence*, N-PoPs). Para processar os fluxos de dados via VNFs, define-se um documento de encadeamento de funções (ou *Service Function Chaining*, SFC) [Halpern and Pignataro 2015]. As SFCs definem como os fluxos deverão ser encaminhados entre as funções, ou seja, definem por quais funções de rede um fluxo com determinadas características deve ser encaminhado previamente ao encaminhamento até seu destino final.

Ao definir o encadeamento de fluxos pelas VNFs que devem processá-los, é preciso levar em conta aspectos como uso racional dos recursos da rede, além de garantir requisitos de latência e de largura de banda para o encaminhamento dos fluxos, evitando ainda o congestionamento dos enlaces por onde os fluxos trafegam. Considerando esses aspectos, é crucial utilizar estratégias de engenharia de tráfego [Awduche et al. 2002], empregadas a fim de otimizar o desempenho e o aproveitamento de recursos de rede. Desta forma, é possível que a rede opere de acordo com uma ou mais estratégias definidas pelo administrador, com o objetivo de não onerar demasiadamente os elementos da rede (comutadores, roteadores e os enlaces que os conectam).

Apesar dos diversos avanços na área de engenharia de tráfego (que reúne soluções clássicas como *Equal-cost Multi-path Routing* (ECMP) [Hopps 2000] e *Multiprotocol Label Switching* (MPLS) [Rosen et al. 2001]), as abordagens existentes desconsideram o estado global da rede na etapa de encaminhamento de dados. Apesar desse problema ser facilmente abordado com o uso de SDN, a virtualização de funções de rede torna mais complexa a engenharia de tráfego na rede. Isso porque surge a necessidade de encontrar mais de uma rota dentro de um caminho entre origem e destino. Essa necessidade manifesta-se pelo dever de localizar rotas que tratem as VNFs como nós de origem e/ou destino. As sub-rotas dentro de uma rota entre *host* origem e *host* destino intercorrem devido à obrigação de cumprimento das SFCs, que ditam sobre quais VNFs um determinado

fluxo de dados deve trafegar antes de chegar ao destino.

Para preencher essa lacuna, este trabalho tem por objetivo investigar e comparar procedimentos que permitam a utilização equilibrada dos recursos de rede disponíveis em uma infraestrutura que abriga as tecnologias referidas previamente. O artigo apresenta um levantamento das abordagens tradicionais de balanceamento de carga e de engenharia de tráfego visando à compreensão do problema e analisando possíveis formas de solução. Vislumbra-se que a proposição dessas estratégias possa resultar em projetos e implementações com o objetivo de apresentar soluções para o problema apresentado.

O restante do artigo está organizado como segue. Na Seção 2 apresenta-se uma análise dos trabalhos relacionados ao problema estudado. Na Seção 3 são citados aspectos de uma possível solução proposta para o problema de engenharia de tráfego no âmbito de NFV. A Seção 4 discute aspectos relacionados à avaliação preliminar dos algoritmos propostos para engenharia de tráfego. A Seção 5 fecha o artigo com apresentação de considerações finais e perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Como mencionado anteriormente, um dos elementos-chave deste trabalho são as estratégias de balanceamento de carga em redes de computadores. Em função disso, a seguir são discutidos pormenores das investigações que representam o estado da arte em estratégias de balanceamento de carga e de engenharia de tráfego. Vale ressaltar, entretanto, que pouco tem sido feito no sentido de desenvolver soluções que englobem redes com suporte à NFV.

Os artigos sobre engenharia de tráfego em redes SDN possuem algumas características únicas. Dentre elas, destacam-se a aplicação na qual a infraestrutura é focada (por exemplo *streaming*, aplicações de baixa latência, *web servers*, etc.), a técnica específica de balanceamento de carga (por exemplo número de saltos, banda disponível, número de conexões, etc.), a forma como tratam a entrada de novos fluxos, etc. As investigações levantadas na revisão do estado da arte são sumarizadas a seguir, com foco em suas propriedades.

Su *et al.* [Su et al. 2014] e Handigol *et al.* [Handigol et al. 2009] buscam minimizar a latência na comunicação entre quaisquer dois *hosts* de uma rede. O algoritmo proposto por Su *et al.* separa os fluxos em *mice flows* (fluxos de curta duração e que trafegam poucos dados) e *elephant flows* (fluxos contínuos e que trafegam um grande volume de dados), com o objetivo de isolar fluxos de maior ocupação dos enlaces e agregar fluxos de baixa ocupação. Já o algoritmo proposto por Handigol *et al.* realiza balanceamento de carga em servidores com um controlador configurado para tomar decisões fundamentadas no congestionamento da rede e na carga dos servidores. Nenhuma das duas propostas considera a existência de *middleboxes* na rede.

Lee *et al.* [Lee et al. 2015] propõem um mecanismo baseado na escolha de rotas com o menor número de saltos. Para desempate, Lee *et al.* fundamenta-se em um cálculo de auto-vetores de matrizes, formadas a partir de informações relativas às rotas candidatas, como quantidade de regras instaladas nos comutadores e banda disponível. Aquela que possuir menor auto-vetor associado é escolhida como rota entre os *hosts*. Todavia, confirmou-se o não suporte de VNFs dispostas em topologias sobre as quais o algoritmo pode atuar.

Adami *et al.* [Adami et al. 2015] desenvolveram um controlador que opera sobre uma aplicação de engenharia de tráfego para aplicações diferenciadas em SDN. A es-

tratégia consiste em separar a topologia em duas representações diferentes. A primeira trata de tráfego diferenciado (*VoIP*, *streaming*, jogos, etc.). A segunda trata de tráfego não diferenciado. O tráfego é classificado através de um módulo de inspeção de pacotes (*Deep Packet Inspection*, DPI) instalado na rede. Cada novo fluxo é classificado e, em seguida, encaminhado de acordo com a sua classificação. Diferentemente das soluções já apresentadas, existe um suporte parcial a apenas uma função de rede não virtualizada pré-posicionada na topologia.

Li e Pan [Li and Pan 2013] propõem uma forma de realizar balanceamento de carga para *web servers* por meio de SDN em uma topologia *fat-tree*. O cálculo da rota ideal é realizado a partir da consideração da banda disponível em cada enlace da árvore. Mais especificamente, o algoritmo é baseado em divisão e conquista, pois calcula as melhores sub-rotas entre dois níveis da árvore, formando uma rota completa. Tal como nas investigações anteriores, não se oferece suporte a N-PoPs dispersos pela topologia.

A partir do exposto acerca da literatura relacionada é importante ressaltar que, no contexto do presente trabalho, alguns desafios se apresentam. Mais especificamente, a tarefa de engenharia de tráfego torna-se mais complexa, pois é necessário considerar a existência de VNFs e as SFCs responsáveis por encadeá-las no ato de encontrar a forma ideal de encaminhar tráfego ao longo da rede.

3. Estratégias para Engenharia de Tráfego em NFV

Nesta seção propõem-se e discutem-se três estratégias de balanceamento de carga aplicadas no contexto de engenharia de tráfego, visando infraestruturas SDN com suporte a NFV. Como premissa simplificadoria, assume-se que a infraestrutura alvo possui SFCs pré-definidas e funções de rede pré-posicionadas, utilizando alguma arquitetura habilitadora de NFV presente na literatura. Como consequência dessa premissa, as estratégias propostas são agnósticas dos métodos de posicionamento e/ou implantação das funções de rede que são utilizadas pelo administrador da rede.

Considerando que o controlador de rede pode visualizar e coletar informações relativas aos enlaces e comutadores da rede, oportuniza-se a atribuição de pesos lógicos aos enlaces presentes na infraestrutura de uma forma mais descomplicada do que em redes de computadores tradicionais. Tendo em vista que o programa de controle tem acesso à visão da rede materializada na forma de um grafo, os pesos serão consultados sempre que houver a necessidade do controlador tomar uma decisão relacionada à instauração de uma nova rota. Nas estratégias propostas, os pesos atribuídos aos enlaces são baseados exclusivamente na utilização dos mesmos.

Para ilustrar os algoritmos propostos, lança-se mão de três exemplos, que serão descritos nas seções subsequentes, embasados na SFC representada na Figura 1. Nessa SFC, há dois *endpoints* A (origem) e B (destino), e duas funções de rede que processam os fluxos originados em A, durante o trânsito ao *endpoint* B: um *Firewall* e um servidor *Proxy*.



Figura 1. Exemplo de SFC

As estratégias propostas fundamentam-se na noção de que uma rota em uma SFC é composta por $N + 1$ sub-rotas, onde N indica o número de funções de rede presentes na

SFC. Portanto, considerando o exemplo de SFC ilustrado na Figura 1, é preciso calcular 3 rotas. A primeira é responsável por guiar o tráfego de um *host* até o N-PoP que executa o *firewall*. Analogamente, a segunda orienta o tráfego do *firewall* em direção ao *proxy*. Por fim, a terceira rota conduz o tráfego do *proxy* até o *host* destino.

A primeira estratégia trata da *evasão de enlace mais congestionados*. Em resumo, ela consiste na remoção iterativa dos enlaces mais congestionados. Objetiva-se que a rota vencedora seja a última rota restante após a remoção dos enlaces. No caso da existência de um peso máximo em comum, mantém-se o enlace de maior peso e é feita a pesquisa (e eventual remoção) do enlace de segundo maior peso de cada rota.

A segunda estratégia, denominada *contagem de saltos, opção por enlace de maior banda disponível*, visa utilizar a rede de forma mais equilibrada. Em resumo, ela fundamenta-se no número de saltos entre dois nós da rede. Opta-se pela rota que possuir o menor número de saltos, se esse número for compartilhado por mais de uma rota, busca-se realizar a utilização do enlace que possui a maior banda disponível, isto é, menor peso atribuído. No caso de um peso mínimo compartilhado mais de uma rota, verifica-se o segundo menor peso dentre as rotas.

Por fim, a terceira estratégia, *contagem de saltos, opção por média de utilização mais baixa*, baseia-se no número mais baixo de saltos entre origem e destino, de forma semelhante à estratégia anterior. Considerando que existe a possibilidade de haver mais de uma rota com o menor número de saltos, o desempate é realizado através do cálculo média dos pesos. A média dos pesos de uma rota arbitrária entre os nós (A, B) é calculado através da Equação 1, onde n representa o número de enlaces na rota e $w(e_i)$ representa o peso do i -ésimo enlace que compõe rota.

$$\bar{w}(A, B) = \frac{\sum_{i=1}^n w(e_i)}{n} \quad (1)$$

4. Avaliação

Tendo sido apresentadas as estratégias propostas no escopo desse trabalho, a seguir apresenta-se uma discussão sobre as potencialidades das mesmas, considerando a SFC apresentada na Figura 1, e a topologia ilustrada na Figura 2. Nessa topologia são apresentados elementos típicos de uma topologia de NFV: *hosts* (identificados com números em círculos preenchidos), roteadores e *comutadores* (identificados por números em círculos não preenchidos) e N-PoPs. Observa-se ainda a existência de enlaces conectando os elementos da topologia, e seus respectivos pesos.

A seguir são discutidos os critérios de seleção de caminhos para materializar as sub-rotas da SFC apresentada no exemplo, considerando cada uma das estratégias propostas. Para a primeira estratégia, observa-se que a primeira sub-rote (*Host 1*→*firewall*), tem-se apenas um caminho a seguir, através dos comutadores (2, 3, 1). Em relação à segunda sub-rote (*firewall*→*proxy*), existem três caminhos possíveis. São elas: (1, 3, 4), (1, 3, 5, 4) e (1, 3, 6, 5, 4). Como o objetivo desta estratégia é remover os enlaces mais sobrecarregados, o primeiro enlace a ser removido do cálculo é o enlace (5, 6), que possui peso 25. O grafo restante possui apenas duas rotas caminháveis: (1, 3, 4) e (1, 3, 5, 4). Aplicando a estratégia recursivamente no grafo remanescente, o enlace (5, 4) é removido, pois é o de maior peso. Desta forma, a rota restante é composta pelos comutadores (1, 3, 4). Por fim, a última sub-rote (*proxy*→*Host 2*) possui quatro alternativas: (4, 5, 6),

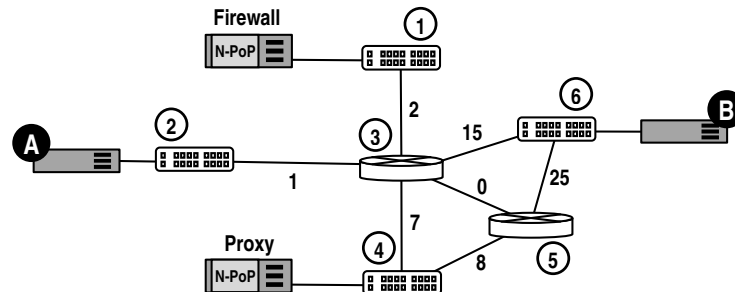


Figura 2. Exemplo de topologia com dois *hosts* e dois N-PoPs

(4, 5, 3, 6), (4, 3, 5, 6) e (4, 3, 6). Dentre essas, o maior peso presente é 25, portanto o enlace (5, 6) é removido. No próximo passo verificamos que o enlace (3, 6), de peso 15, é comum às duas rotas restantes, neste caso, busca-se o enlace com o segundo maior peso de cada rota, portanto o enlace (4, 5) é removido, restando a rota (4, 3, 6).

Focando agora na segunda estratégia, a primeira sub-rotas (*Host 1* → *firewall*) não precisa ser calculada para o exemplo dado, pois existe apenas uma possível: (2, 3, 1). Com o objetivo de encontrar a segunda sub-rotas (*firewall* → *proxy*), verificamos os dois caminhos candidatos possíveis: (1, 3, 4) e (1, 3, 5, 4). Procura-se pela rota com o menor número de saltos, desta forma, opta-se pela rota (1, 3, 4). Por fim, as quatro sub-rotas (*proxy* → *Host 2*) aspirantes são (4, 5, 6), (4, 5, 3, 6), (4, 3, 5, 6) e (4, 3, 6). Um algoritmo que implemente essa estratégia pode verificar que as rotas (4, 5, 6) e (4, 3, 6) possuem o menor número de saltos. Para realizar o desempate, a escolha realizada visa à utilização da rota que possuir o enlace de menor utilização dentre as duas candidatas. Neste caso, o enlace (4, 3) é retornado, resultando na escolha da rota (4, 3, 6).

A terceira estratégia por sua vez, e de forma análoga às estratégias anteriores, considera apenas uma rota possível para a primeira sub-rotas (*Host 1* → *firewall*): (2, 3, 1). Devido ao fato desta estratégia utilizar como primeiro critério o menor número de saltos, tem-se apenas um caminho candidato para a segunda sub-rotas (*firewall* → *proxy*): (1, 3, 4). Finalmente, existem quatro caminhos entre os comutadores 4 e 6 (*proxy* → *Host 2*), são eles: (4, 5, 3, 6), (4, 3, 5, 6), (4, 5, 6) e (4, 3, 6). Primeiramente, verifica-se que as rotas (4, 5, 6) e (4, 3, 6) possuem o menor número de saltos. Consequentemente, é necessário calcular a média de cada caminho. Para o caminho (4, 5, 6), tem-se que: $w(4, 5, 6) = \frac{8+25}{2} = 16.5$. Já para o caminho (4, 3, 6), tem-se que: $w(4, 3, 6) = \frac{7+15}{2} = 11$. Observa-se que a sub-rotas (4, 3, 6) possui o menor peso médio, sendo escolhida como a rota percorrida pelos dados destinados ao *host 2*.

A partir da avaliação descritiva das estratégias apresentadas acima, deve-se considerar eventuais limitações no processo de escolha de caminhos. Primeiramente, para encontrar a melhor rota a partir da primeira estratégia (de evasão do enlace mais congestionado), verifica-se um alto custo computacional associado em caso de grandes infraestruturas. Isso porque se faz necessário remover de forma iterativa enlaces lógicos pelo controlador até que uma única rota persista. Desta forma, esta solução pode oferecer uma baixa escalabilidade. As estratégias a seguir tem como primeira opção rotas com o menor número de saltos, mas diferem na política de desempate. A segunda estratégia visa rotas que possuam o enlace de maior banda disponível. Portanto, deve-se ter em mente que uma mesma rota pode conter o enlace menos e o mais congestionado de

todas as candidatas. Ainda assim, tal rota será escolhida, por possuir enlaces de menor utilização, mas não estará evitando o congestionamento, pois os dados trafegarão pelo enlace mais congestionado. Por fim, a terceira estratégia, responsável pelo cálculo da média das utilizações de cada rota, sugere a utilização mais equilibrada entre as três estratégias sugeridas, considerando que se avalia o estado geral da rota e não enlaces isolados.

5. Considerações Finais

A engenharia de tráfego consiste em um componente fundamental para o efetivo sucesso de Virtualização de Funções de Rede, dada a necessidade de encaminhamento eficiente de fluxos entre funções de rede, visando principalmente o uso racional dos recursos e a satisfação de requisitos de qualidade de serviço. Apesar da importância da engenharia de tráfego para a área de NFV, as abordagens existentes tornam-se inadequadas para esse contexto, devido à necessidade de encontrar mais de uma rota dentro de um caminho entre origem e destino, tal como determinado em uma SFC.

Para preencher essa lacuna, foi apresentada nesse trabalho uma análise sucinta sobre desafios relacionados a técnicas engenharia de tráfego, aplicadas ao contexto de Virtualização de Funções de Rede (NFV). Foram destacadas ainda as principais estratégias desenvolvidas com o objetivo de realizar engenharia de tráfego em SDN.

Tendo em vista o desenvolvimento recente das tecnologias expostas, desafios vão surgindo de forma gradual ao longo da disseminação e do emprego das mesmas. O questionamento da ausência de estratégias de balanceamento de carga nesse contexto é necessário, pois se pode dar um passo adiante em rumo à utilização de SDN e NFV em maior escala. Desta forma, objetiva-se que este trabalho atue como uma primeira abordagem ao problema evidenciado e fomentando a proposição e o desenvolvimento de novas técnicas a fim de enfrentar as dificuldades apresentadas.

Como perspectivas de trabalhos futuros, visa-se o projeto e desenvolvimento de um ambiente capaz de empregar as estratégias apresentadas com o objetivo de avaliar e comparar o seu funcionamento. Dentre as métricas a serem avaliadas, devemos considerar, principalmente, eficiência e escalabilidade, pois são fatores-chave na decisão de adoção de uma estratégia em nível de grandes infraestruturas de rede.

Referências

- Adami, D., Antichi, G., Garroppo, R. G., Giordano, S., and Moore, A. W. (2015). Towards an sdn network control application for differentiated traffic routing. In *Communications (ICC), 2015 IEEE International Conference on*, pages 5827–5832. IEEE.
- Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. (2002). Overview and Principles of Internet Traffic Engineering. RFC 3272 (Informational). Updated by RFC 5462.
- Gember-Jacobson, A., Viswanathan, R., Prakash, C., Grandl, R., Khalid, J., Das, S., and Akella, A. (2014). Opennf: Enabling innovation in network function control. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 163–174, New York, NY, USA. ACM.
- Halpern, J. and Pignataro, C. (2015). Service Function Chaining (SFC) Architecture. RFC 7665 (Informational).
- Handigol, N., Seetharaman, S., Flajslik, M., McKeown, N., and Johari, R. (2009). Plug-serve: Load-balancing web traffic using openflow. *ACM Sigcomm Demo*, 4(5):6.

- Hopps, C. (2000). Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992 (Informational).
- Hwang, J., Ramakrishnan, K. K., and Wood, T. (2014). Netvm: High performance and flexible networking using virtualization on commodity platforms. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*.
- Lee, D., Hong, P., and Li, J. (2015). Rpa-ra: A resource preference aware routing algorithm in software defined network. In *IEEE International Conference on Computer Communications (INFOCOM)*, page 6. IEEE.
- Lewin-Eytan, L., Naor, J., Cohen, R., and Raz, D. (2015). Near optimal placement of virtual network functions. In *Computer Communications (INFOCOM 2015), 2015 IEEE International Conference on*, INFOCOM '15, pages 1–12, New York, NY, USA. IEEE.
- Li, Y. and Pan, D. (2013). Openflow based load balancing for fat-tree networks with multipath support. In *Proc. 12th IEEE International Conference on Communications (ICC'13), Budapest, Hungary*, pages 1–5.
- Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P., and Gaspary, L. P. (2015). Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *Integrated Network Management (IM 2015), 2015 IFIP/IEEE International Symposium on*, IM '15, pages 98–106, New York, NY, USA. IEEE.
- Matias, J., Garay, J., Toledo, N., Unzilla, J., and Jacob, E. (2015). Toward an sdn-enabled nfv architecture. *Communications Magazine, IEEE*, 53(4):187–193.
- McKeown, Anderson, Balakrishnan, Parulkar, Peterson, Rexford, Shenker, and Turner (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69 – 74.
- McKeown, N. (2009). Software-defined networking. *INFOCOM keynote talk*, 17(2):30–32.
- Network Functions Industry Specification Group (2012). Network function virtualisation (nfv): An introduction, benefits, enablers, challenges and call for action. In *SDN and OpenFlow World Congress*, pages 1–16.
- Rosen, E., Viswanathan, A., and Callon, R. (2001). Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard). Updated by RFCs 6178, 6790.
- Sherry, J., Gao, P. X., Basu, S., Panda, A., Krishnamurthy, A., Maciocco, C., Manesh, M., Martins, J. a., Ratnasamy, S., Rizzo, L., and Shenker, S. (2015). Rollback-recovery for middleboxes. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM '15*, pages 227–240, New York, NY, USA. ACM.
- Su, Z., Wang, T., Xia, Y., and Hamdi, M. (2014). Cheetahflow: Towards low latency software-defined network. In *Communications (ICC), 2014 IEEE International Conference on*, pages 3076–3081. IEEE.
- Xia, M., Shirazipour, M., Zhang, Y., Green, H., and Takacs, A. (2015). Optical service chaining for network function virtualization. *Communications Magazine, IEEE*, 53(4):152–158.