Utilizando Anycast para Filtragem de Pacotes para Funções de Rede Virtualizadas em Roteadores de Alto Desempenho

José Flauzino[1], Christian Lyra[2], Elias P. Duarte Jr.[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba – PR – Brasil

[2]Rede de Ensino e Pesquisa (RNP) – PoP-PR
Curitiba – PR – Brasil

{jwvflauzino,elias}@inf.ufpr.br

lyra@pop-pr.rnp.br

Resumo. *Os pontos de presença (PoPs) da Internet atualmente são baseados em roteadores que processam um grande volume de dados, da ordem de centenas de gigabits por segundo. Neste contexto, torna-se um desafio fazer a filtragem e encaminhamento de pacotes individuais para uma Função de Rede Virtualizada (VNF - Virtualized Network Function). No presente trabalho propomos uma estratégia que combina roteamento anycast com a classificação de pacotes para resolver o problema. Foi implementado um estudo de caso com uma VNF do tipo forwarder na rede de alto desempenho do PoP-PR da RNP. O forwarder encaminha pacotes para um servidor AWS remoto que compartilha o endereço anycast. Resultados demonstram a viabilidade da proposta, tendo surpreendido pelo baixo overhead imposto pela solução.*

1. Introdução

A demanda pelos mais diversos tipos de serviços de rede tem representado um desafio para o projeto de infraestruturas de rede capazes de atender aos seus requisitos. A revolução provocada pelos serviços baseados em Inteligência Artificial (IA) e *Machine Learning* (ML), além da proliferação e popularização em massa dos chamados "serviços OTT" (*Over-the-Top services*), tem forçado uma evolução contínua da infraestrutura. Há uma previsão de que as redes inteligentes em 2027 terão velocidade mínima dos enlaces de 800 Gbps, metade deles chegando a 1,6 Tbps [Boujelbene 2023]. Entretanto, aumentar as taxas de transmissão não representa uma solução completa para a evolução das redes em busca de atender às novas demandas. Entre as diversas tecnologias envolvidas, destaca-se neste artigo que a virtualização é uma das principais.

A virtualização permite que dispositivos de rede sejam executados como software [Chowdhury and Boutaba 2009]. Além das máquinas virtuais, os *containers* representam os mecanimos básicos para a execução de serviços virtualizados. A virtualização permite a criação de redes lógicas sobre um substrato físico. No contexto da Internet, a virtualização resolveu o problema da fossilização da Internet (*Internet ossification*), na medida em que permite que múltiplas soluções distintas entre si possam ser adotadas, sendo executadas simultamente em uma mesma rede física. A evolução da rede pode seguir múltiplos caminhos e tudo ao mesmo tempo. Não há como negar que a virtualização simplifica as operações de rede, na medida em que dispensam equipamentos de hardware

específicos, utilizando software, que pode ser inicializado, atualizado, terminado com muito mais facilidade. Uma das principais tecnologias que viabilizam a virtualização no contexto de redes é a NFV (*Network Functions Virtualization*), descrita a seguir.

Através do paradigma NFV [Hawilo et al. 2014], funções de rede tradicionalmente executadas em hardware especializado podem ser implementadas em software e instanciadas como VNFs (*Virtualized Network Functions*) sobre hardware de propósito geral [Tavares et al. 2018]. Exemplos destas funções includem *firewalls*, detectores de anomalia e intrusão, dispositivos NAT (*Network Address Translation*), até mesmo roteadores. Em comparação à solução clássica, baseada em *middleboxes* implementados em hardware especializado, as soluções NFV tendem a prover maior flexibilidade e facilidade de gerenciamento, além de reduções significativas de custos operacionais e de capital [Yousaf et al. 2017]. Além de funções de rede individuais (VNFs), é possível fazer a composição de serviços por meio de um processo chamado de *Service Function Chaining* (SFC), em que múltiplas VNFs são encadeadas em uma topologia de serviço [Fulber-Garcia et al. 2020].

De acordo com o documento do IETF (*Internet Engineering Task Force*) que especifica uma SFC [Halpern and Pignataro 2015], o tráfego de rede é direcionado para cada SFC através de uma classificação de tráfego baseada em políticas. Em especial, a classificação tem como objetivo permitir o redirecionamento de determinados tráfegos para serviços de rede específicos. Assim, cada pacote de rede é separado do tráfego agregado e direcionado para a SFC correspondente. Neste sentido, uma classificação pode ser ampla (como baseada apenas em endereços de portas TCP/UDP) ou granular (utilizando um conjunto de critérios). Além disso, pode ser realizada uma classificação inicial de pacotes no ingresso da rede e classificações mais específicas em pontos seguintes.

Contudo, em redes com tráfegos de grande volumes de dados, como em ISPs (*Internet Service Providers*) e seus PoPs (*Points of Presence*), a classificação de tráfego, que permite a filtragem de pacotes direcionados para VNFs específicas, torna-se um desafio significativo. Nestes ambientes, predominam os enlaces com largura de banda extremamente elevada, tendo capacidade para transferir centenas de gigabits por segundo. Redes desta magnitude exigem ativos de alto desempenho, como roteadores de camada 3 capazes de encaminhar na ordem de milhões de pacotes por segundo. Tendo em vista a carga destes dispositivos, é potencialmente inviável realizar inspeções adicionais em pacotes de rede para fins de classificação. Isso porque, mesmo uma classificação simples pode gerar gargalos, uma vez que requer checagem de cabeçalhos extras de cada pacote.

Neste trabalho, é proposta uma estratégia para classificação de tráfego em redes com grande volume de dados trafegados, visando redirecionar tráfegos específicos para funções e serviços de rede virtualizados individuais. A proposta aproveita a funcionalidade de maior desempenho dos roteadores: o roteamento de pacotes em camada 3. Em particular, a estratégia consiste em atribuir um endereço *anycast* para uma VNF, aproveitando a própria funcionalidade do roteador para a classificação. Um estudo de caso é apresentado, com uma VNF do tipo *forwarder* sendo executada na rede de alto desempenho do PoP-PR da RNP (Rede Nacional de Ensino e Pesquisa). No experimento conduzido, um tráfego direcionado a um destino específico na Internet é filtrado do tráfego agredado e redirecionado de modo a passar primeiro pela VNF *forwarder*. Resultados demonstram o baixo *overhead* imposto pela solução.

O restante deste trabalho está organizado da seguinte maneira. A Seção 2 descreve conceitos elementares e trabalhos relacionados. Na Seção 3, a estratégia proposta é descrita. Resultados experimentais são apresentados na Seção 4. Finalmente, a Seção 5 conclui o trabalho.

2. Fundamentos & Trabalhos Relacionados

Com o objetivo de padronizar a execução e a gerência dos serviços baseados em NFV, além de permitir a interoperabilidade de VNFs provenientes de diferentes desenvolvedores, a ETSI (*European Telecommunications Standards Institute*) vem coordenando o *framework* arquitetônico NFV-MANO (NFV - *MANagement and Orchestration*). Sua especificação inclui um bloco também chamado NFV-MANO, o bloco NFVI (NFV *Infrastructure*) e as VNFs propriamente ditas. O bloco NFV-MANO é responsável pelo ciclo de vida, orquestração e gerenciamento dos serviços virtualizados. O NFV-MANO fornece ainda interfaces de comunicação padronizadas, além de abstrair os recursos computacionais necessários para executar as VNFs. O bloco NFVI, representa a infraestrutura virtualizada na qual as VNFs são instanciadas, gerenciadas e executadas.

Recentemente proposta, a arquitetura NFV-COIN (NFV *COmputing In the Network*) [Venâncio et al. 2022] abre um caminho promissor para a tecnologia NFV, permitindo a construção de serviços arbitrários dentro da rede. Há vários serviços já implementados neste contexto, como para detecção de falhas de processos [Turchetti and Duarte 2015], consenso [Venâncio et al. 2021] e a difusão confiável e ordenada de mensagens [Venâncio et al. 2019]. Estão previstos ainda a execução dentro da rede de serviços como plataformas de agentes móveis [Duarte Jr and Cestari 2000] e sistemas para rastreamento de pacotes IP (IP *traceback*) [Hilgenstieler et al. 2010].

Trabalhos relacionados no contexto de uso de endereçamento *anycast* em NFV incluem [Wion et al. 2019]. A forma de utilização de endereços *anycast* entretanto é diferente: há múltiplas VNFs espalhadas pela rede, sendo que as VNFs de mesma funcionalidade (por exemplo, *firewalls*) compartilham o mesmo endereço *anycast*.

Melhorar o desempenho da classificação de pacotes direcionados para funções e serviços diferenciados é um problema relevante. Em [Yuan et al. 2019] é apresentada uma solução que faz uso de *caching* e paralelismo em hardware para acelerar a classificação. Outra solução para acelerar a classificação de pacotes para VNFs [Polverini et al. 2020] propõem uma solução baseada em otimização para gerenciar as entradas das tabelas de fluxos que chegam ao limite do número de entradas.

3. Filtragem de Tráfego NFV Inspirado em Endereçamento *Anycast*

A estratégia proposta para filtragem e redirecionamento de tráfego para VNFs/SFCs combina o roteamento *anycast* com a classificação de pacotes. Um endereço *anycast* é compartilhado por um grupo de nodos (*i.e.*, todos os nodos do grupo possuem o mesmo endereço IP) [Wion et al. 2019]. Quando um pacote é direcionado a um IP *anycast*, este é entregue ao nodo de menor custo no grupo (geralmente o mais próximo ao remetente, menos sobrecarregado, *etc.*). Em IPv4, um endereço *anycast* possui formato idêntico a um endereço *unicast*, por exemplo. O que torna o endereço em *anycast* é a forma que ele é tratado (roteado) pela rede. Considere como exemplo o caso de funções ou serviços virtualizados que devem encaminhar o tráfego para um determinado *host* de destino. Neste

caso, o ponto principal da estratégia é tratar como *anycast* um ou mais endereços IP públicos daqueles destinos. Assim, a rede mapeia um endereço *unicast* do destino como se fosse *anycast*. No caso de SFC, o endereço *anycast* é mapeado para a primeira VNF da SFC como membro de menor custo do grupo *anycast*. Desta forma, o tráfego é encaminhado primeiro para a VNF/SFC, ao invés de ser roteado diretamente ao *host* destino.

A Figura 1 apresenta a arquitetura da proposta. A solução considera que um dispositivo de origem se comunica com um de destino e o tráfego passa pela rede na qual estão instanciados os serviços de rede virtualizados. Para acionar a estratégia de filtragem basta adicionar regras apropriadas nos roteadores da rede, bem como as regras de classificação dentro da plataforma NFV. As regras de roteamento podem ser instaladas manualmente em cada roteador ou mesmo anunciadas por um protocolo IGP (*Interior Gateway Protocol*) - uma vez que tais regras devem ficar apenas dentro do domínio da rede, não podendo ser anunciadas entre diferentes domínios.

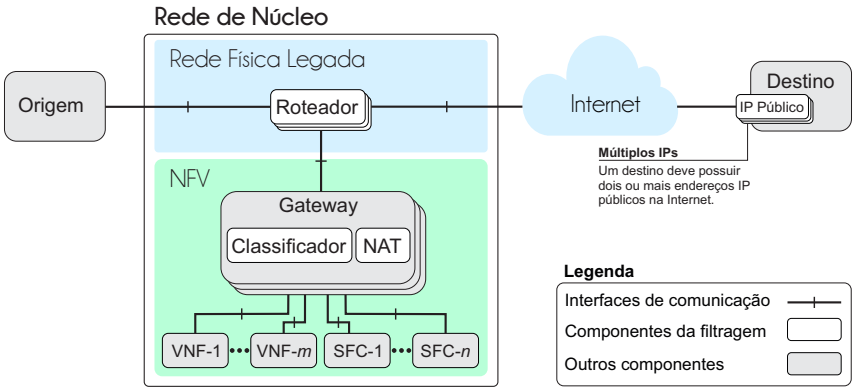

Figura 1. Arquitetura da filtragem de tráfego inspirada em *anycast*.

Uma regra de roteamento deste tipo indica que há um nodo membro do grupo *anycast* dentro da própria rede. Além disso, indica que o próximo salto (*next hop*) é a porta de entrada (ou *gateway*) da plataforma NFV existente dentro do domínio da rede. A VNF que tem o endereço *anycast* deve estar na mesma rede local do roteador. Naturalmente, ela sempre será o nodo de melhor custo, diante outros nodos espalhados pela Internet. Vale destacar que o presente artigo descreve a primeira versão da proposta, com apenas uma única VNF que também faz papel de *gateway*.

Em plataformas NFV (*i.e.*, sistemas que implementam o *framework* NFV-MANO) como o CloudStack/Vines [Flauzino et al. 2021], OpenStack/Tacker [Tacker 2024], *etc.*, ou ainda em nuvens como a AWS (Amazon Web Services), as instâncias virtualizadas (máquinas virtuais ou mesmo VNFs) são geralmente expostas à Internet através de algum tipo de *gateway* - *e.g.*, roteadores virtuais. Assim, endereços IP públicos não são atribuídos nas instâncias virtuais propriamente ditas, há apenas IPs privados em suas interfaces de rede. Os IPs públicos são mapeados pelos *gateways*. Portanto, sempre que recebe um pacote direcionado a um IP público de uma instância, o *gateway* redireciona o pacote para o IP privado correspondente (e vice-versa). A estratégia de filtragem proposta se beneficia desta característica.

O *gateway* da estratégia proposta faz um mapeamento do IP público do destino (IP este que já é tratado pelos roteadores como *anycast*). Assim, ao chegar um pacote no

gateway com destino a este IP público, ele classifica (com base em porta, protocolo, *etc.*) e encaminha o pacote para a VNF que corresponde às regras de classificação. Contudo, ressalta-se que aquela VNF não possui de fato o endereço IP *anycast*, trata-se apenas da regra de roteamento. No endereçamento/roteamento *anycast* tradicional, o endereço está efetivamente atribuído aos nodos correspondentes.

Após o tráfego ser processado pela VNF/SFC, ele é devolvido ao *gateway*. No entanto, se o *gateway* simplesmente encaminhar este tráfego da forma que está para a rede, ela o devolverá. Por isso, o *gateway*, na verdade, precisa fazer duas mudanças em cada pacote: altera a origem para seu próprio IP, e o destino para algum IP *unicast* público secundário do destino. Como resultado, a rede encaminha o tráfego normalmente até o destino. Assim, quaisquer pacotes de resposta são enviados pelo destino ao *gateway* (pois os pacotes que chegaram ao destino tinham como origem o IP do *gateway*), que re-encaminha os pacotes de resposta à origem. Por fim, para o dispositivo de origem, todo este redirecionamento fica transparente.

4. Experimentos e Resultados

A avaliação da estratégia de filtragem proposta neste trabalho requer implementação em uma rede com muitos fluxos diferentes e alto volume de tráfego de dados - na escala de dezenas ou centenas de gigabits por segundo. Tráfegos desta magnitude são geralmente vistos em centros de dados ou redes de núcleo. Uma vez que mudanças neste tipo de ambiente de produção exigem cautela, foi proposto inicialmente um cenário mínimo da estratégia para viabilizar uma análise preliminar de seu ponto chave: o redirecionamento de tráfego inspirado em *anycast*.

Neste sentido, esta seção apresenta resultados de experimentos conduzidos em um ambiente no qual o redirecionamento de tráfego inspirado em *anycast* foi implantado no PoP-PR da RNP. Como ilustrado na Figura 2, uma aplicação cliente instanciada na UFPR (Universidade Federal do Paraná), em Curitiba-PR, se comunica através da rede da RNP com um servidor instanciado na zona de São Paulo-SP da AWS. Dois cenários são avaliados, sendo o primeiro representado pela comunicação direta entre cliente e servidor, e o segundo pela comunicação que é redirecionada para passar por uma VNF do tipo *forwarder* instanciada no PoP-PR. Como forma de concentrar apenas no roteamento *anycast* aplicado à filtragem inicial do tráfego, neste experimento a VNF desempenha também o papel de *gateway*. Fazendo, em especial, as alterações de origem e destino de cada pacote.

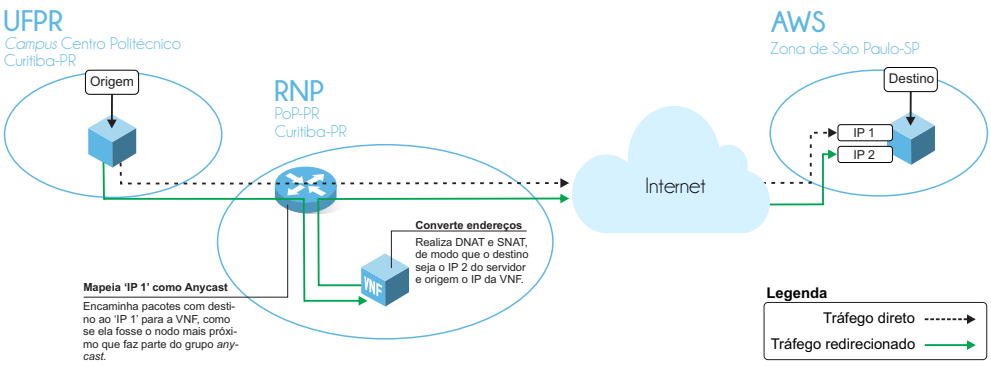

Figura 2. Redirecionamento de tráfego inspirado em *anycast* com a VNF *forwarder* executando no PoP-PR da RNP.

A Figura 3 apresenta o funcionamento do redirecionamento de tráfego do experimento. Neste caso, o servidor na AWS possui dois endereços IP, sendo que o endereço principal (*IP 1*) é tratado como *anycast* pelo PoP-PR, enquanto o secundário (*IP 2*) continua sendo *unicast*. Assim, quando o cliente envia uma requisição ao servidor via *IP 1*, a rede do PoP-PR reconhece a VNF *forwarder* como o nodo do grupo *anycast* do *IP 1* que está mais próximo, e encaminha os pacotes da requisição para ela. A VNF *forwarder* então altera cada pacote para que a origem seja seu próprio endereço IP, enquanto o destino passa a ser o *IP 2* do servidor na AWS (veja os passos 4 e 5 na Figura 3). Isso garante que (*i*) quando os pacotes transmitidos pela VNF *forwarder* voltam à rede do PoP-PR, eles são encaminhados diretamente ao servidor na AWS (passo 6); e (*ii*) a resposta do servidor também passará pela VNF *forwarder* (que reverterá as mudanças de endereçamento).

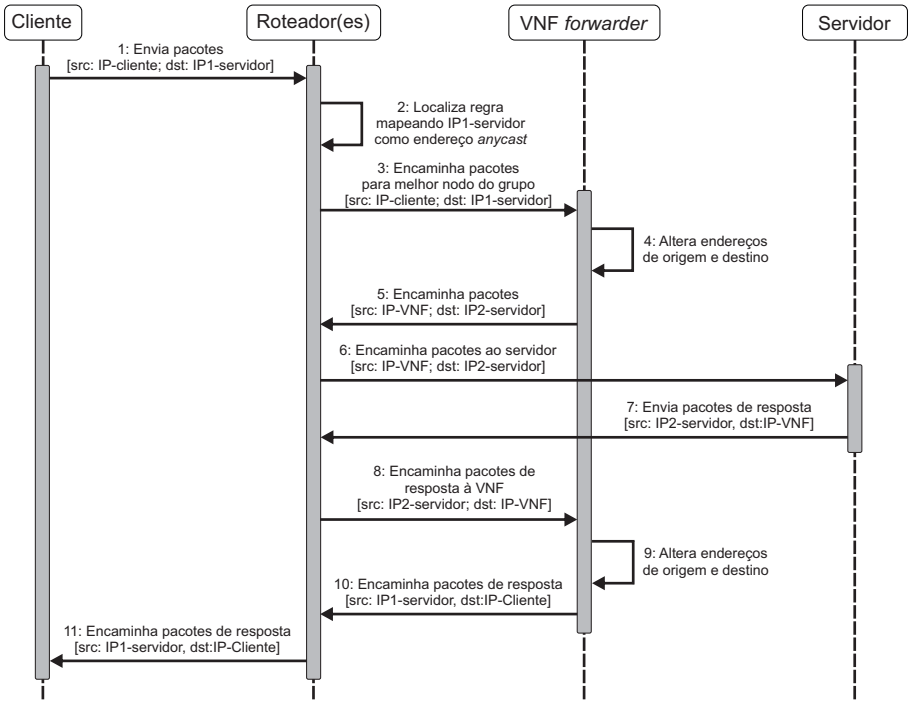

Figura 3. Diagrama de sequência ilustrando o redirecionamento de tráfego.

O gráfico na Figura 4 mostra o RTT medido a partir do cliente que executa na rede da UFPR (Laboratório LaRSiS). É preciso destacar que o baixo *overhead* produzido pela solução (mesmo com todo o processamento descrito na Figura 3) é extremamente reduzido. Por isso, para diferenciar o tráfego direto do tráfego redirecionado com a solução proposta, foi necessário apresentar os resultados da Figura 4 em microssegundos (μs). As distribuições acumuladas mostram o tráfego direto sem passar pela VNF (azul) e passando pela VNF *forwarder* (laranja). A maioria dos pacotes do gráfico azul (direto) teve RTT (latência) em torno de 7000 μs. Já no tráfego passando pela VNF *forwarder*, a maioria dos pacotes apresentaram uma latência um pouco abaixo de 8000 μs, ou seja, uma diferença pouco menor que 1 milissegundo. Este resultado demonstra que o tráfego foi filtrado e redirecionado (passando por um salto a mais, a VNF) com um custo mínimo.

Visando avaliar a diferença de desempenho da solução proposta para uma estratégia de classificação de pacotes tradicional, baseada no NetFilter do Linux, foram executados os experimentos de vazão da Figura 5. A figura mostra a vazão obtida sem

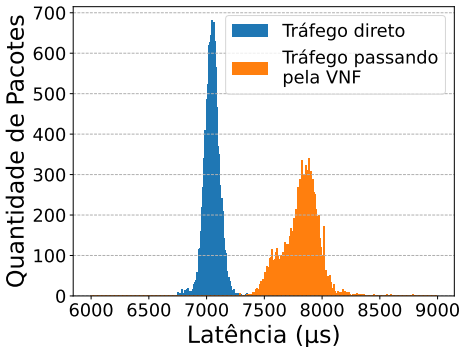

Figura 4. Distribuição da latência (RTT) do tráfego direto Vs. passando pela VNF.

passar e passando pela VNF *forwarder* em um cenário distinto daquele onde foi executado o primeiro experimento. Os experimentos foram executados entre duas máquinas (ambas com processador Intel Core i7-12700 e 16 GB de RAM) conectadas via Ethernet 10 Gbps. Através da ferramenta iPerf3[1], foi introduzido tráfego para saturar o enlace. Vale destacar que teria sido inviável executar esta solução tradicional no PoP-PR. Apesar da vazão média da presente proposta ter sido superior à da versão tradicional (8,17 Gbps versus 8,10 Gbps), os gráficos mostram que a diferença é estatisticamente insignificante. Contudo, estes resultados confirmam a viabilidade da proposta, a qual pode ser adotada em casos em que a estratégia tradicional não é viável.

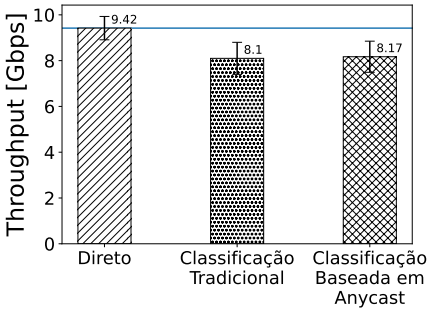

Figura 5. Diferença de vazão entre a classificação tradicional e a baseada em *anycast* em uma rede de 10 Gbps.

5. Conclusão

Neste trabalho apresentamos uma solução para a filtragem e encaminhamento de pacotes para VNFs individuais por roteadores de alto desempenho. A solução apresentada neste documento é a primeira versão da proposta, e visa VNFs que encaminham tráfego para um *host* destinatário. No caso, foi implementada uma VNF *forwarder* executada no PoP-PR. A solução é baseada em endereçamento *anycast*, utilizando a funcionalidade nativa do roteador para a filtragem. Experimentos mostram o excelente desempenho atingido. O principal trabalho futuro é consolidar uma arquitetura que permita a classificação de pacotes para múltiplas VNFs/SFCs de qualquer natureza.

[1]https://iperf.fr/

Agradecimentos

O presente trabalho foi parcialmente apoiado pela CAPES – Código de Financiamento 001; e o CNPq - projeto 308959/2020-5.

Referências

Boujelbene, S. (2023). Ocp 2023 key takeaways: The network is the computer for ai workloads. Accessado em 6 de abril de 2024.

Chowdhury, N. M. K. and Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, 47(7):20–26.

Duarte Jr, E. P. and Cestari, J. M. A. (2000). O agente chinês para diagnóstico de redes de topologia arbitrária. In *Anais do II Workshop de Testes e Tolerância a Falhas*, pages 88–93. SBC.

Flauzino, J. et al. (2021). Gerência e orquestração de funções e serviços de rede virtualizados em nuvem cloudstack. In *XXVI Workshop de Gerência e Operação de Redes e Serviços*, pages 82–95. SBC.

Fulber-Garcia et al. (2020). Cusco: a customizable solution for nfv composition. In *Int. Conf. on Advanced Information Networking and Applications (AINA)*, pages 204–216.

Halpern, J. M. and Pignataro, C. (2015). Service Function Chaining (SFC) Architecture. RFC 7665.

Hawilo, H., Shami, A., Mirahmadi, M., and Asal, R. (2014). Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc). *IEEE network*, 28(6):18–26.

Hilgenstieler, E. et al. (2010). Extensions to the source path isolation engine for precise and efficient log-based ip traceback. *Computers & Security*, 29(4):383–392.

Polverini, M., Galán-Jiménez, J., Lavacca, F. G., Cianfrani, A., and Eramo, V. (2020). A scalable and offloading-based traffic classification solution in nfv/sdn network architectures. *IEEE Transactions on Network and Service Management*, 18(2):1445–1460.

Tacker (2024). Tacker - OpenStack NFV Orchestration. Technical report, The OpenStack Project. `https://wiki.openstack.org/wiki/Tacker`. Acessado em abril de 2024.

Tavares, T. N. et al. (2018). Niep: Nfv infrastructure emulation platform. In *IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 173–180. IEEE.

Turchetti, R. C. and Duarte, E. (2015). Implementation of failure detector based on network function virtualization. In *IEEE Int. Conf. on Dependable Systems and Networks Workshops*, pages 19–25.

Venâncio, G. et al. (2019). Nfv-rbcast: Enabling the network to offer reliable and ordered broadcast services. In *9th Latin-American Symposium on Dependable Computing (LADC)*, pages 1–10.

Venâncio, G. et al. (2021). Vnf-consensus: A virtual network function for maintaining a consistent distributed software-defined network control plane. *Int. Journal of Network Management*, 31(3):e2124.

Venâncio, G., Turchetti, R. C., and Duarte Jr, E. P. (2022). Nfv-coin: Unleashing the power of in-network computing with virtualization technologies. *Journal of Internet Services and Applications*, 13(1):46–53.

Wion, A., Bouet, M., Iannone, L., and Conan, V. (2019). Distributed function chaining with anycast routing. In *Proceedings of the 2019 ACM Symposium on SDN Research*, pages 91–97.

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.

Yuan, Y. et al. (2019). Halo: Accelerating flow classification for scalable packet processing in nfv. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 601–614.