Roteamento na Origem como Facilitador do Encadeamento Multi-nuvem de Funções de Rede: Proposta e Implementação

Rodolfo V. Valentim[1], **Cristina. K. Dominicini**[2], **Diego R. Mafioletti**[2], **Rodolfo S. Villaça**[1], **Moises R. N. Ribeiro**[1], **Magnos Martinello**[1]

[1] Núcleo de Estudos em Redes Definidas por Software (Nerds)
Universidade Federal do Espírito Santo (Ufes) – Vitória/ES – Brasil

[2]Instituto Federal do Espírito Santo (Ifes) – Serra/ES, Colatina/ES – Brasil

rodolfo.valentim@aluno.ufes.br, cristina.dominicini@ifes.edu.br

Abstract. *Network providers need to offer a wide range of services to their customers. Virtualization of network functions and software-defined networks make it easy to implement these services by virtualizing and chaining network functions. However, there are cases when these functions are in geographically distant data centers or in clouds in different domains. On the other hand, there are few works in the literature that focus on the mechanisms for the implementation of multiple cloud network function chaining. In this context, this work presents a solution for chaining network functions in multiple clouds. The article highlights KeySFC-based source routing as a facilitator of the proposed solution and has several advantages in managing flows and maintaining states on network devices. A prototype is implemented and evaluated in order to perform a functional evaluation of the proposal in multiple cloud chaining scenarios.*

Resumo. *Provedores de rede precisam oferecer uma grande variedade de serviços a seus clientes. A virtualização de funções de rede e redes definidas por software facilitam a implementação desses serviços por meio da virtualização e do encadeamento de funções de rede. Entretanto, há casos em que essas funções encontram-se em data centers distantes geograficamente ou em nuvens em domínios diferentes. Por outro lado há poucos trabalhos na literatura que focam nos mecanismos para implementação do encadeamento de funções de rede em múltiplas nuvens. Neste contexto, este trabalho apresenta uma solução para o encadeamento de funções de rede em múltiplas nuvens. O artigo destaca o roteamento na origem, baseado no KeySFC, como facilitador da solução proposta e apresentando diversas vantagens quanto ao gerenciamento dos fluxos e à manutenção dos estados nos dispositivos de rede. Um protótipo é implementado e avaliado, com o objetivo de se realizar uma avaliação funcional da proposta em cenários de encadeamento em múltiplas nuvens.*

1. Introdução

O advento do paradigma de Virtualização de Funções de Rede (NFV, ou *Network Functions Virtualization*) permitiu que provedores de serviços de telecomunicações possam migrar parte das funções de rede de equipamentos especializados para funções de rede virtualizadas (VNF, ou *Virtual Network Function*), instanciadas em servidores de prateleira localizados em nuvens públicas ou privadas [Mijumbi et al. 2016]. Em paralelo, a

popularização das Redes Definidas por Software (SDN, ou *Software Defined Networking*) permitiu maior programabilidade dos elementos de rede, que passaram a ter controle centralizado e separação entre o plano de controle e o plano de dados.

Com as possibilidades que a utilização dos paradigmas de NFV e SDN trouxeram, um conceito que vem ganhando popularidade é o Encadeamento de Funções de Rede (SFC, ou *Service Function Chaining*), que trata da instanciação de um conjunto ordenado de funções de rede e o subsequente direcionamento de fluxos de tráfego através dessas funções. Muitos trabalhos importantes têm estudado mecanismos para fornecer SFC em redes de *data center* e, mais recentemente, este conceito vem sendo expandido para múltiplas nuvens. Por nuvem entende-se um domínio bem definido e restrito de computação e rede que é, necessariamente, interligado à outras nuvens via protocolos de camada 3 (roteada).

Dessa forma, provedores de serviços poderão utilizar seus recursos computacionais distribuídos em diferentes domínios administrativos e/ou geográficos para fazer composição dinâmica de serviços. Por exemplo, compor aplicações de baixa latência, que dependem da alocação de parte das funções de rede em *data centers* de borda em diversos pontos de acesso, mais próximos aos usuários, com funções que requerem maior capacidade de processamento.

Entretanto, as principais soluções de SFC fazem uso de mecanismos tradicionais de encaminhamento baseados em tabelas, que possuem problemas relacionados à pouca escalabilidade e o alto grau de complexidade na gerência dos estados dos elementos de rede (entradas nas tabelas) devido à alta transitoriedade dos fluxos [Jin et al. 2016]. Uma consequência disso é o alto tempo de convergência necessário para reconfiguração dos estados de rede. Adicionalmente, outra consequência é que a engenharia de tráfego pode ficar restrita a um conjunto de caminhos mais curtos, impedindo que o orquestrador selecione caminhos otimizados para evitar congestionamentos ou falhas [Tso et al. 2016].

Esses problemas são agravados quando da implementação de uma cadeia de VNFs alocadas em múltiplas nuvens, pois se torna necessário orquestrar um maior número de recursos em diferentes domínios, gerenciar o tráfego e a conectividade inter e intra nuvem, e garantir a exposição dos serviços que suportam o acesso aos recursos de cada domínio. Assim, o custo para gerenciar a instalação e manutenção de estados pode tornar inviável a aplicação das soluções tradicionais para resolver o problema de encadeamento de forma ágil e escalável.

Uma alternativa para mitigar esses problemas são os mecanismos de encaminhamento com roteamento na origem (SR, ou *Source Routing*), que atenuam problemas de escalabilidade relacionados ao crescimento e ao gerenciamento das tabelas de encaminhamento nos equipamentos de rede [Jyothi et al. 2015] e reduzem a sobrecarga do plano de controle em comparação com os mecanismos tradicionais de encaminhamento [Martinello et al. 2014]. Uma subcategoria do SR é o roteamento rígido na origem (SSR, ou *Strict Source Routing*) onde, no momento da classificação dos fluxos, o nó de origem (ou *switch* de borda) insere no cabeçalho do pacote informações sobre o caminho desde a origem até destino. Dessa forma, não é necessário configurar tabelas de encaminhamento nos elementos de redes existentes no caminho.

Nesta linha, destaca-se o trabalho `KeySFC` [Dominicini et al. 2020], que propõe

uma solução de SFC baseada em infraestruturas de rede que utilizam SSR. O trabalho apresenta uma arquitetura de rede de *data center* que propõe dividir os dispositivos de rede entre borda e núcleo. Assim, mantém-se o núcleo da rede simples e a borda se responsabiliza em classificar, encapsular e desencapsular os fluxos. Todavia, o `KeySFC` também não faz nenhuma consideração sobre a aplicação da solução em múltiplos domínios.

Com essa motivação, este trabalho propõe, implementa e avalia uma solução de SFC para composição de serviços de rede distribuídos em múltiplas nuvens. A solução proposta pretende integrar a comunicação entre nuvens ao roteamento na origem de maneira transparente e, para que isso ocorra, é preciso propor uma arquitetura de interação entre serviços de modo a obter as informações necessárias para o roteamento e gerenciamento da infraestrutura. São abordados aspectos tanto do plano de controle, baseando-se no relatório da ETSI DGS/NFV-MAN001 [Mahmoodi et al. 2017], quanto do plano de dados, ao utilizar-se do `KeySFC` como mecanismo de encadeamento baseado em SSR.

A apresentação da solução dar-se-á da seguinte forma: a Seção 2 apresenta os principais trabalhos relacionados e destaca as principais contribuições deste artigo. Na Seção 3 o problema de encadeamento multi-nuvem será formalmente apresentado, junto com a proposta de solução, cuja implementação será apresentada na Seção 4. A Seção 5 apresenta resultados da avaliação da proposta e as conclusões e trabalhos futuros encontram-se na Seção 6.

2. Trabalhos Relacionados

Esta seção posiciona o trabalho com relação aos trabalhos relacionados, que estão subdivididos em três categorias: roteamento na origem, mecanismos de SFC em um única nuvem e, finalmente, SFC multi-nuvem.

2.1. Roteamento na Origem

A forma tradicional de executar SSR é inserir no cabeçalho do pacote uma pilha (ou uma lista ordenada) de portas ou endereços que detalha todo o caminho a ser percorrido pelo pacote na infraestrutura do *data center*. Assim, cada elemento de encaminhamento faz uma operação de *pop* nesta pilha pra descobrir a porta de saída, sem necessidade de consultar tabelas. Exemplos desta abordagem incluem: SecondNet [Guo et al. 2010], Segment Routing [Clad et al. 2018] e Sourcey [Jin et al. 2016].

Outra abordagem alternativa de SSR explora o conceito matemático de Sistema Numérico de Resíduos (RNS, ou *Residue Number System*). Nesta abordagem, a porta de saída em cada nó é definida por uma operação de módulo sobre um identificador de rota [Martinello et al. 2014]. Um esquema de SSR baseado em RNS pode explorar propriedades para fornecer funcionalidades de rede que não podem ser cobertas pelo método tradicional de lista, tais como encaminhamento sem reescrita de pacotes e resiliência.

Trabalhos relacionados já exploraram este tipo de abordagem de SSR para fornecer reação rápida a falhas [Gomes et al. 2016] e programabilidade de rede. Outros investigam técnicas para melhorar a escalabilidade do número de bits do identificador de rota e demonstraram a sua eficiência em cumprir restrições de latência para *multicast* em *data centers* [Jia 2014]. Por fim, o esquema `KeySFC` [Dominicini et al. 2020] aplicou SSR ao problema de SFC em uma única nuvem.

Assim, este artigo é o primeiro trabalho que propõe um esquema completo de SFC usando SSR baseado em RNS para encadeamento em múltiplas nuvens.

2.2. Mecanismos de SFC em uma única nuvem

As principais ferramentas de código aberto que implementam SFC, tais como OpenDayLight, ONOS, OPNFV, e Openstack Networking-SFC, são baseadas na arquitetura proposta pela RFC7665 [Pignataro and Carlos 2015] e no protocolo NSH (Network Service Headers) [Quinn and Guichard 2014].

O protocolo NSH especifica um cabeçalho que inclui dois identificadores de cadeia: o primeiro indica a cadeia selecionada e o segundo indica a posição atual na cadeia. Os elementos de encaminhamento no caminho usam as informações desses identificadores para pesquisar em uma tabela e decidir qual VNF deve ser executada a qualquer momento. Em seguida, o endereço do próximo elemento de rede é encapsulado em um cabeçalho mais externo, contando com os métodos de roteamento subjacentes para entregar o pacote entre esses elementos. Embora seja bastante flexível, esse esquema requer consultas em tabelas para cada salto tanto para descobrir a próxima VNF quanto para rotear entre as funções na rede física, o que leva a problemas de escalabilidade e agilidade relacionados ao número de estados na rede. Além disso, a inserção de cabeçalhos expande o tamanho do pacote, causando aumento de tráfego e possíveis problemas com fragmentação.

Outro método popular é o Segment Routing [Clad et al. 2018], em que uma lista ordenada de segmentos é codificada como uma pilha de *labels* no formato MPLS. O próximo segmento a ser processado é extraído do topo da pilha e uma operação de pesquisa em tabela de encaminhamento é executada em cada salto. Embora o nó de origem possa precisar enviar uma lista longa de segmentos para obter caminhos ótimos, a maioria dos equipamentos MPLS suporta um tamanho de pilha limitado (cerca de 3 a 5 *labels*), o que pode levar a uma distribuição ineficiente de tráfego [Abdullah et al. 2019]. Portanto, embora o esquema de Segment Routing possa ser usado para habilitar SFC com SSR, ele geralmente usa um esquema híbrido que especifica uma lista com apenas alguns elementos de rede pelos quais o pacote deve passar e delega o roteamento entre esses elementos à rede subjacente, que emprega caminhos mais curtos usando ECMP (*Equal-Cost Multi-Path routing*), por exemplo. Já o SRv6 [Lebrun and Bonaventure 2017] também depende de tabelas para realizar o roteamento, provocando efeitos de redução da escalabilidade da solução, e aumento do *overhead* nos pacotes, pois empilha cabeçalhos no formato IPv6.

Neste contexto este trabalho adota o esquema de SFC proposto por `KeySFC` [Dominicini et al. 2020], que habilita o encadeamento usando SSR baseado em RNS. Em contraste com os esquemas Segment Routing e NSH, o `KeySFC` não restringe a seleção de caminhos, diminui o número de estados na rede, especialmente em cenários com múltiplas nuvem, e elimina as tabelas presentes nos dispositivos no núcleo das redes. Mais detalhes sobre o funcionamento deste esquema serão apresentados na Seção 3.2.2.

Por fim, os trabalhos descritos nesta seção não exploram a aplicação desses mecanismos em cenários em que as VNFs estão distribuídas em múltiplas nuvens. A próxima seção detalha os trabalhos relacionados que exploram essa questão.

2.3. SFC em múltiplas nuvens

Vários trabalhos importantes já abordaram o problema do encadeamento de funções de rede em ambientes multi-domínios sob o ponto de vista de um problema de otimização. Pode-se citar [Bhamare et al. 2017], cujo foco é diminuir o fluxo entre *data centers* através do posicionamento eficiente das cadeias de VNFs. [Sun et al. 2019] também trata do problema do posicionamento usando Programação Linear Inteira (ILP, ou *Integer Linear Programming*) e otimiza o consumo de energia com foco no atendimento de requisitos dos encadeamentos. Já [Dietrich et al. 2017] resolve o problema do roteamento, através de otimização por ILP, propondo o particionamento de cadeias em múltiplos domínios conectados por *gateways* virtuais com o posicionamento estratégico das VNFs nestes domínios. [Gupta et al. 2017] também otimiza o posicionamento de VNFs em ambientes multi-nuvem usando um algoritmo preditivo com o objetivo de minimizar a latência das cadeias de funções. No entanto, embora algumas soluções de otimização para o problema de SFC multi-nuvem tenham sido propostas na literatura, poucos trabalhos focam no desenvolvimento dos mecanismos de SFC para permitir a implantação dessas soluções de alocação de recursos nas infra-estruturas de rede de *data center*.

Como a maioria das implementações de mecanismos de SFC em uma única nuvem utiliza NSH como protocolo de encapsulamento, uma abordagem para realizar passagem de contexto entre diferentes nuvens é utilizar o empilhamento de cabeçalhos NSH. Este é o caso do trabalho descrito em [Vu and Kim 2016], que implementa SFC Hierárquico (hSFC), um subproblema do encadeamento multi-nuvem. Contudo, esta abordagem carrega os mesmos problemas do protocolo NSH em relação à pouca escalabilidade e alta complexidade no gerenciamento de estados de rede, com adição de cabeçalhos extras para passagem de contexto entre nuvens e, consequentemente, maior sobrecarga no encaminhamento de pacotes e aumento no número de estados na rede.

Em resumo, considerando o estado da arte e a relevância dos trabalhos relacionados apresentados nesta seção, é possível confirmar a relevância e atualidade do problema abordado neste artigo. Entretanto, percebe-se que há uma lacuna na literatura no que se refere a proposição de mecanismos de encaminhamento que tornam viável a implementação de SFC em ambientes multi-nuvem, de forma flexível, ágil e escalável, sendo esta a principal contribuição deste trabalho.

3. Encadeamento de Funções de Rede entre Múltiplas Nuvens

Com o objetivo de apresentar e definir melhor o problema do encadeamento de funções de rede entre múltiplas nuvens, esta seção apresenta alguns casos de uso inspirados em [Dietrich et al. 2017, Kumar et al. 2017], onde destaca-se a necessidade da implementação de serviços virtualizados inter-nuvens. Além disso, motivados por esses casos de uso e baseando-se nas arquiteturas de planos de dados e de controle apresentados em ETSI [Mahmoodi et al. 2017], a solução proposta neste artigo é apresentada.

3.1. Casos de Uso

É possível pensar a tarefa de posicionar VNFs com requisitos de baixa latência, por exemplo, em *data center* de borda (mais próximos), enquanto funções de rede sem esse requisito poderão ser posicionadas em *data centers* maiores e mais distantes.

Há, também, em alguns casos, a necessidade de encaminhar fluxos *northbound* $\leftrightarrow$ *southbound*, ou seja, fluxos com origem ou destino externos ao data center, por uma SFC de Acesso. Dá-se o nome de SFC de Acesso para serviços que analisam fluxos de entrada ou saída do *data center*, conforme definição apresentada em [Kumar et al. 2017]. As SFC de Acesso são específicas para cada domínio. Neste caso, quando um fluxo de uma região administrativa A (*data center* de origem) precisa ir para outra região administrativa B (*data center* de destino), necessariamente irá precisar passar por duas SFCs de Acesso.

Pode-se tratar o encadeamento multi-nuvem como um caso de uso do Encadeamento Hierárquico de Funções de Rede (hSFC) [Dolson et al. 2018]. No hSFC, com o objetivo de diminuir a complexidade da implementação de longas cadeias de encadeamento, divide-se a cadeia em múltiplos domínios hierárquicos, cada um com sua autonomia administrativa. Este conceito pode ser aplicado em provedores de acesso que desejam particionar sua infraestrutura em regiões diferentes para descentralizar a tarefa de gerenciamento e ainda assim manter a possibilidade de consolidação de serviços compostos por VNFs instanciadas em diferentes domínios [Vu and Kim 2016].

Finalmente, quando funções de rede necessitem de recursos específicos (tais como suporte a SR-IOV, GPU, DPDK, por exemplo) pode-se precisar instanciar funções de rede em outros domínios, onde tais recursos estão disponíveis. Um exemplo são os *testbeds* especializados e autônomos, disponibilizados pelo consórcio BR-EU FUTEBOL [Both et al. 2019] em diferentes regiões do Brasil e da Europa.

3.2. Arquitetura

Seguindo os princípios de redes definidas por software e pensando nos casos de uso de encadeamento multi-nuvem, nesta proposta é realizada a separação entre o plano de dados e o plano de controle, com o objetivo de permitir a melhor programabilidade e flexibilização da tarefa de encaminhamento de funções de rede. A arquitetura resultante pode ser vista na Figura 1. A seguir, serão explicados cada elemento da proposta.

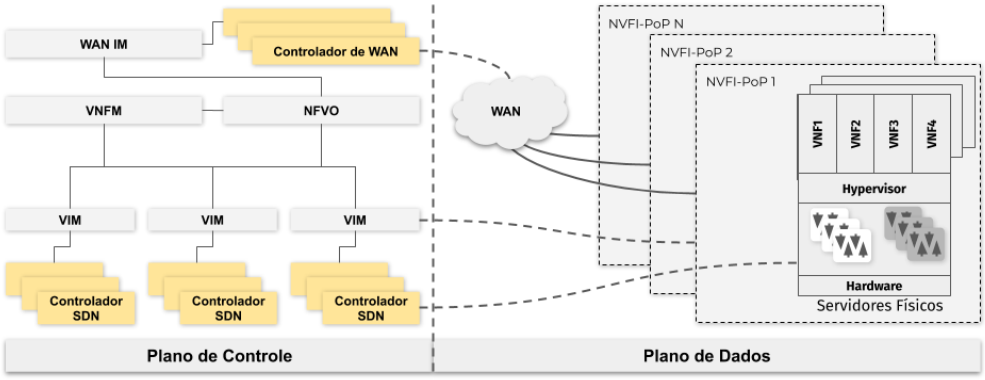

Figura 1. Arquitetura funcional da proposta de SFC multi-nuvem. Elaborada pelo autor e adaptada de [Mijumbi et al. 2016]

3.2.1. Plano de Controle

Atualmente as implementações de gerência e orquestração de funções de rede estão focadas em abordagens centralizadas, especialmente em cenários em que os serviços abran-

gem vários domínios administrativos [Mijumbi et al. 2016]. O ETSI propôs uma arquitetura para o plano de controle para a composição de serviços virtualizados. Os principais blocos que constituem o plano de controle são: NFVO, VIM, WAN IM, VNFM e os Controladores.

- **NFVO (NFV Orquestration)**: Responsável pela gerência e orquestração dos recursos das NFVI e do ciclo de vida dos serviços de rede;
- **VNFM (VNF Manager)**: Responsável pela gerência do ciclo de vida das instâncias VNF;
- **VIM (Virtual Infrastructure Manager)**: Responsável por controlar e gerenciar os recursos de computação, armazenamento e rede do NFVI (Infraestrutura NFV), geralmente dentro do domínio de infraestrutura de um operador;
- **Controladores de Rede**: Responsáveis por prover conectividade e abstração dos elementos de rede provendo uma interface programável para requerer serviços de conectividade. Geralmente os controladores de rede possuem visibilidade sobre os elementos que eles controlam;
- **WAN IM (Wide Area Network Infrastructure Manager)**: Responsável por prover conexões virtuais entre NFVI-PoPs. O NFVO pode requerer conexões virtuais para o WAN IM que estabelece a conexão e gerência o seu ciclo de vida.

Nesta arquitetura, para se implementar um encadeamento de funções de rede a requisição é feita ao NFVO. O NFVO possui uma visão de toda a infraestrutura de rede e é capaz de coordenar esforços de modo a ativar os atores corretos para cada tarefa. Por isso, o orquestrador precisa ter interface com as VIM e os controladores, uma vez que são estes que controlam os recursos a serem alocados.

Ao tomar a decisão de orquestração, ou seja, posicionamento e roteamento das instâncias VNF, o orquestrador requisita à VNFM a instanciação das VNFs. Já o VNFM utiliza sua interface com a VIM para alocar os recursos necessárias a essa tarefa. Uma vez instanciadas, é preciso redirecionar os fluxos correspondentes para essas funções. Para isso, o NFVO requisita a VIM que atue sobre os elementos de rede por meio dos controladores SDN.

Para o caso de tráfego inter *data center* é preciso requisitar a WAN IM para criar uma conexão virtual entre os NFVI-PoPs. É de responsabilidade do WAN IM solicitar ao controlador adequado para realizar a conexão. O controlador precisa estabelecer uma conexão virtual entre os *data centers* utilizando os recursos disponíveis de encapsulamento de pacotes ou particionamento (*slicing*) da infraestrutura.

Para realizar estas tarefas, é requisito que os blocos funcionais de gerenciamento e orquestração NFV que coordenam recursos virtualizados em um único NFVI-PoP ou em vários NFVI-PoPs, garantam a exposição dos serviços e o acesso aos recursos de cada domínio de uma maneira abstrata, aberta e bem definida, respeitando-se a autonomia de cada domínio administrativo. Na proposta apresentada neste artigo são implementados os blocos de NFVO, WAN-IM e os controladores de rede e WAN.

3.2.2. Plano de dados

A arquitetura de rede escolhida para o plano de dados é a `KeySFC` [Dominicini et al. 2020]. Esta arquitetura é compatível com o padrão

estabelecido pela ETSI e possui o diferencial de utilizar SSR como mecanismo de encaminhamento dos pacotes. No `KeySFC` utiliza-se dois tipos de *switches* para implementar a conectividade no *data center*: *switches* de borda e *switches* de núcleo. Enquanto na borda os dispositivos realizam a classificação do fluxo e a definição de rotas, no núcleo os dispositivos são elementos mais simples que realizam encaminhamento de pacotes sem a utilização de tabelas por meio de operações de módulo (resto da divisão). A utilização do `KeySFC` permite ao classificador de tráfego especificar a rota completa no ingresso do fluxo no *data center*.

O roteamento na arquitetura `KeySFC` utiliza o conceito matemático de Sistema Numérico de Resíduos (RNS, ou *Residue Number System*). Em resumo, considere $S = \{s_1, s_2, \ldots, s_N\}$ um conjunto de *N* identificadores de *switches* de núcleo. Considere, ainda, que estes N *switches* pertencem ao caminho que o pacote deve seguir de origem até destino e os valores devem ser co-primos entre si. Além disso, o conjunto $P = \{p_1, p_2, \ldots, p_N\}$ representa as portas de saída as quais os pacotes serão redirecionadas em cada *switches* de núcleo, onde p_i é a porta de saída do pacote quando em s_i. Então, diz-se que existe um número *R*, chamado de identificador da rota, que é função de um conjunto de portas de saída e *switches* de núcleo. O Teorema Chinês dos Restos afirma que é possível reconstruir R a partir dos seus resíduos.

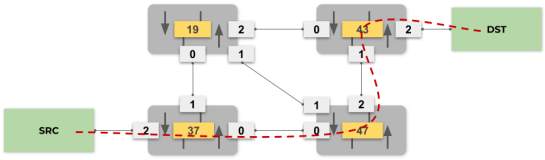

Figura 2. Exemplo de roteamento utilizando RNS.

Na Figura 2 vê-se um exemplo de roteamento no núcleo de uma rede. Assume-se que o pacote passe pela borda onde recebe a identificação de rota, feita após a coleta de informações topológicas. São necessários os *switches* que compõem a rota $S = \{37, 43, 47\}$ e as portas de saída em cada *switch* $P = \{0, 2, 2\}$. Para esse cenário o endereço de rota calculado é R = 32338.

Quando o pacote é então encaminhado para o núcleo da rede contém o endereço de rota no MAC de Destino. Ao chegar no *Switch* 37 a operação de módulo $< 32338 >_{37} = 0$ determina a porta de saída. Logo, o pacote é encaminhado a porta 0 que está conectada ao *Switch* 47. Ao chegar ao *Switch* 47 é realizada a operação $< 32338 >_{47} = 2$. Ao sair pela porta 2 do *Switch* 47 o pacote chega ao *Switch* 43. Novamente, é realizada uma operação de módulo para encontrar o próximo salto, o resultado de $< 32338 >_{43} = 2$ resulta no encaminhamento para a porta 2 que está conectada ao destino, que pode ser um switch de borda. Neste *switch* de borda o pacote precisa ser restaurado ao seu estado inicial. Ao sair do núcleo para a borda da rede, o MAC de Destino é restaurado para o valor original. Após restaurado, o pacote segue para o destino.

Ao não se utilizar tabelas, simplifica-se o gerenciamento do núcleo da rede, e esse é um argumento muito forte quando aplicado ao contexto de SFC. O protocolo recomendado pela ETSI para realizar o encapsulamento de fluxos é o NSH, que adiciona um cabeçalho extra aos pacotes e, para realizar o encaminhamento dos fluxos para as instancias VNF é preciso adicionar regras em todos os *switches* no caminho entre a origem e o

destino do fluxo. Essa característica sobrecarrega o plano de controle devido à alta transitoriedade dos estados nos dispositivos de rede e apresenta problemas de escalabilidade a medida que o número de fluxos aumenta [Dominicini et al. 2020]. Neste contexto o uso do `KeySFC` é vantajoso, pois não só elimina a necessidade de se gerenciar os estados nos elementos do núcleo da infraestrutura, como também facilita a execução de ações de engenharia de tráfego, tais como balanceamento de carga e mudança de rotas sem modificar o estado dos comutadores no núcleo da rede [Valentim et al. 2019].

De acordo com [Kumar et al. 2017], em *data centers* muito grandes ou distribuídos a classificação precisa ocorrer em diversos pontos de entrada, a depender do modelo de implementação da SFC de Acesso do *data center*. Deste modo, caso a classificação precise ocorrer em apenas um ponto, isso pode acarretar em uma implementação ineficiente já que o fluxo pode precisar percorrer longas distâncias até atingir o ponto de classificação.

A escolha do KeySFC como mecanismo de encaminhamento do plano de dados facilita a tarefa de estabelecer encadeamento inter *data center*, uma vez que, todo o contexto necessário para o roteamento encontra-se no pacote. Desta forma, evita-se reclassificação de fluxos ou empilhamento de cabeçalhos NSH.

4. Implementação

A implementação baseou-se nos conceitos e na proposta apresentada na Seção 3.2. Construiu-se um protótipo utilizando-se tecnologias largamente utilizadas no mercado com o intuito de demonstrar a viabilidade da solução e as vantagens da proposta quando comparadas às soluções similares disponíveis, que em sua maioria, utilizam NSH para realizar o encaminhamento.

Na implementação utilizou-se 5 servidores com processador Intel Xeon E5-2620 de 2,4 GHz, 16 GB de memória e 6 NICs Ethernet de 1 Gbps e sistema operacional CentOS versão 7. A representação do protótipo implementado é apresentada na Figura 3

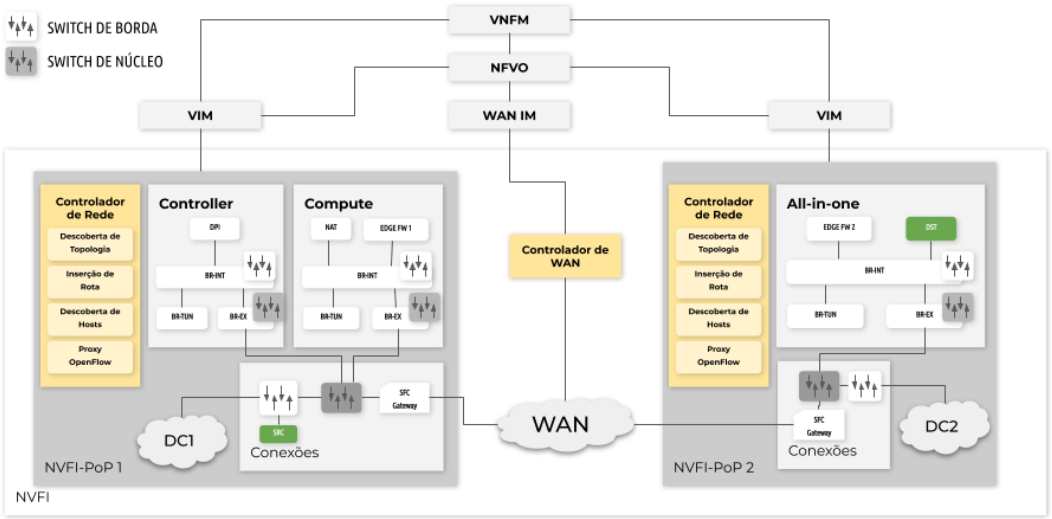

Figura 3. Representação do protótipo de solução SFC multi nuvem.

Considerando a premissa de utilizar-se de soluções disponíveis no mercado, um requisito de implementação foi a adoção do *OpenStack* como gerenciador da infraestru-

tura virtualizada (VIM). A escolha do *OpenStack* justifica-se por ser um arcabouço de nuvem, modular, que gerencia recursos de armazenamento, rede e computação. Através da interface de programação disponível na *OpenStack SDK* é possível acessar os recursos da infraestrutura do *data center* disponíveis no *OpenStack*.

Como quer-se demonstrar uma implementação multi-nuvem, foram instaladas duas nuvens diferentes: NFVI-PoP 1 e NFVI-PoP 2. Conforme é mostrada na Figura 3, o NFVI-PoP 1 possui dois nós hospedeiros e o NFVI-PoP 2 possui apenas 1. Os dois nós restantes na Figura 3 são utilizados para implementar conectividade entre os nós e os diferentes NFVI-PoPs.

A escolha do *OpenStack* como VIM é conveniente, já que a estrutura de comutadores virtuais utilizados para entregar conectividade as máquinas virtuais é compatível tanto com a arquitetura da ETSI quanto com a implementação do `KeySFC`. Conforme pode-se ver na Figura 3, os hospedeiros possuem instâncias de *switches* virtualizados e compatíveis com o mecanismo de encaminhamento proposto no `KeySFC`.

As funcionalidades dos *switches* virtuais são implementadas pelos controladores de rede. Foi necessária a implementação de alguns serviços para habilitar o SFC interdomínios: *Descoberta de Topologia*, *Inserção de Rota*, *Descoberta de Hosts* e *Proxy OpenFlow*. Para o controlador de rede escolheu-se o *Ryu*, que permite, com poucas linhas de código, implementar um controlador de rede totalmente funcional e adequado à proposta deste trabalho. Os *switches* virtuais são *bridges* do *Open vSwitch 2.12*, que possui suporte ao *OpenFlow*, desde a versão 1.0 até 1.5.

A *Descoberta de Topologia* é uma funcionalidade que deve ser implementada em todos os *switches* (borda e núcleo) e é uma tarefa necessária para implementar o SFC. A descoberta de topologia é possível devido ao protocolo LLDP (*Link Layer Discovery Protocol*). Cada *switch* divulga a todos seus vizinhos sua identificação. O controlador centraliza todas essas informações recebidas de todos os *switches* da rede e com isso é capaz de descobrir a topologia da rede.

A *Inserção de Rotas* é uma funcionalidade necessária apenas aos *switches* de núcleo. De acordo com o *KeySFC*, todo pacote ingressante em um *switch* de núcleo é encaminhado seguindo uma operação de módulo. O *Open vSwitch* não permite encaminhar pacotes desta forma e por isso, neste protótipo foi necessário utilizar o controlador para implementar essa funcionalidade. Importante frisar que essa é uma característica dessa implementação que quis utilizar *OpenvSwitch*, mas outras implementações, como exemplo *P4* e *SmartNICS*, são capazes de implementar encaminhamento por módulo livre de tabelas. Pacotes do tipo *KeySFC* são enviados ao controlador que, ao obter a identificação da rota do pacote, calcula a porta de saída e instala uma regra que encaminha pacotes deste fluxo encaminhando-o para a porta de saída correta.

Conforme apresentado na Seção 3, a origem dos fluxos de uma SFC pode ser tanto uma máquina virtual quanto um dispositivo físico dentro do *data center* (um hospedeiro ou um dispositivo de rede). Quando o fluxo não é proveniente de um elemento virtualizado, é necessário descobrir qual o *switch* de borda alvo responsável por classificar dos fluxos ingressantes. Desta forma, implementa-se através do controlador a funcionalidade de *Descoberta de Hosts*, que envia requisições ARP pesquisando pelos endereços IP de origem da requisição. Os *switches* que responderem serão os *switches* de borda alvo.

Finalmente, o *Proxy OpenFlow* é uma funcionalidade que permite a inserção de regras de fluxo nos switches da topologia via chamadas HTTP REST. O Proxy converte para *OpenFlow* as regras recebidas via chamadas REST e as envia para as *bridges* alvo.

Além destes elementos implementou-se o *Controlador de WAN* e o *SFC Gateway* para viabilizar SFC multi nuvem, inter *data center*. O *Controlador WAN* recebe requisições de conexões entre NFVI-PoPs e ativa os *SFC Gateway* associados. O *SFC Gateway* é um *switch* virtual que utiliza GRE (*Generic Routing Encapsulation*) para o encapsulamento dos fluxos. Como as pontas do túnel GRE são virtualizadas, não há necessidade de se configurar os equipamentos da infraestrutura de rede do *data center* de origem nem de destino. Assim, sob-demanda do NFVO, através do WAN IM e do Controlador de WAN, são criados túneis entre diferentes domínios NFV. O NFV-MANO (junção de NFVO e NFVM) é uma aplicação desenvolvida em Python que faz a interface entre o administrador e as (APIs) do *OpenStack* de cada domínio e do WAN IM.

5. Avaliação da Proposta

Para demonstrar o funcionamento da solução com base nos casos de uso descritos na Seção 3, foi instanciado o seguinte encadeamento de instancias VNF: SRC → DPI → NAT → Edge Firewall 1 → Edge Firewall 2 → DST, ilustrados na Figura 4. Por simplificação, e por não ser objetivo deste trabalho a avaliação de desempenho de diferentes funções de rede virtualizadas, as funções de rede usadas nesta avaliação apenas realizam a tarefa de reencaminhar os pacotes recebidos de volta à interface de rede de chegada, sem modificá-los.

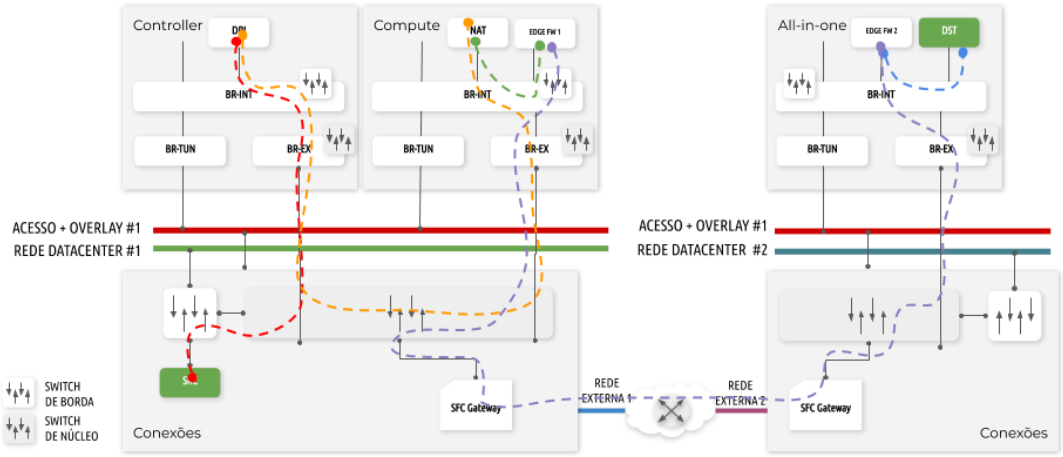

Figura 4. Diagrama SFC com destaque para cada salto. Elaborada pelo autor

Escolheu-se este posicionamento para avaliar o encaminhamento em diferentes cenários que exploram todos os aspectos da solução. São eles:

- Cenário I: SFC entre uma *Physical Network Function* e uma VNF (SRC → DPI);
- Cenário II: SFC entre VNFs no mesmo servidor (NAT → EDGE-FW-1);
- Cenário III: SFC entre VNFs em servidores diferentes (DPI → NAT);
- Cenário IV: SFC entre VNFs em NFVI-PoPs diferentes (EDGE-FW-1 → EDGE-FW-2) e;
- Cenário V: SFC entre uma VNF e o destino do fluxo (EDGE-FW-2 → DST).

Foram obtidas as métricas de vazão, latência e *jitter* para o encaminhamento SRC $\rightarrow$ DPI $\rightarrow$ NAT $\rightarrow$ Edge Firewall 1 $\rightarrow$ Edge Firewall 2 $\rightarrow$ DST. As medidas de latência e jitter são comparadas com as mesmas métricas medidas no salto SFC Gateway NFVO PoP 1 $\rightarrow$ Gateway NFVO PoP 2, com o objetivo de avaliar a contribuição deste salto nas medidas de latência e *jitter* fim-a-fim. Já a vazão foi comparada com os valores obtidos entre origem e destino sem encadeamento de funções, ou seja, sem usar a proposta apresentada neste artigo. Em tempo, caso haja interesse do leitor em avaliar o impacto da quantidade de saltos por VNF nas métricas de vazão, latência e *jitter*, sugere-se consultar [Dominicini et al. 2020], que discute esses casos no caso de SFC em uma única nuvem.

Para a medição da latência executou-se 30 vezes o *ping* entre origem e destino, e considerou-se que o RTT (*Round Trip Time*) obtido é uma medição indireta da latência da rede. Para a medição do *jitter*, executou-se 30 vezes a ferramenta `iperf` com tráfego UDP. Para a vazão máxima, novamente, utilizou-se a ferramenta `iperf` executando 30 vezes, também com tráfego UDP. Estes resultados estão apresentados na Figura 5.

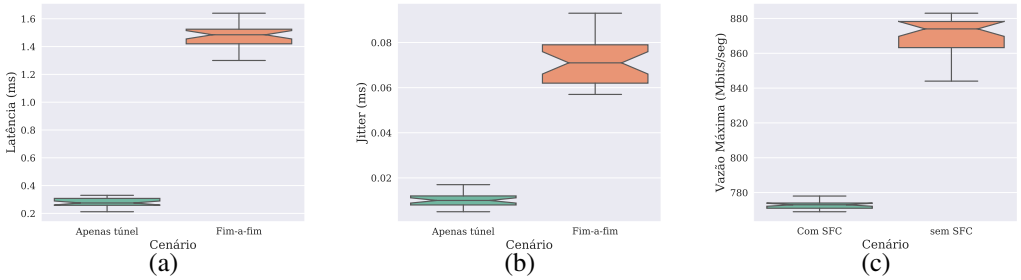

Figura 5. a) Latência sentida entre origem e destino do fluxo comparada com a latência medida apenas no túnel; b) *Jitter* sentida entre origem e destino do fluxo comparada com a *jitter* medida apenas no túnel; c) Vazão máxima em cenário passando pelas VNFs e sem passar pelas VNFs

A vazão máxima passando pela VNFs (com SFC) foi de 769 ± 13 Mbits/seg, menor que a vazão máxima sem SFC de 864 ± 26, que é muito próxima da capacidade nominal das interfaces de rede usadas no experimento (1Gbps). Já a latência média é de 1.5 ± 0.1 e a contribuição do túnel foi de 0.3 ± 0.1. Obviamente, quanto maior a distância entre os NFVI-PoPs, maior será a latência sentida pelo túnel. Mesmo assim, é importante avaliar que não é muito diferente do resultado obtido em saltos sem encapsulamento. Além disso, conforme apresentado na Seção 2, existem diversas publicações que estudam o problema do posicionamento de modo a diminuir a distância entre VNFs quando o serviço é sensível a latência, e esses resultados podem ser utilizados pelo orquestrador da rede. Já em se tratando dos resultados de *jitter*, o valor médio foi de 0.720 ± 0.010 e o *jitter* sentido apenas no salto entre PoPs é de 0.011 ± 0.003. Percebe-se que a contribuição do salto entre nuvens não destoa de um salto entre VNFs.

Em resumo, a vazão obtida com o protótipo é muito próximo do limite físico das interfaces de rede usadas no experimento. Além disso, os resultados permitem concluir que não há degradação de latência e *jitter*, portanto, afirma-se que proposta de realizar encadeamento *inter data center* (multi-nuvens) é viável e factível de ser implementada em situações reais, com software livre e de prateleira, amplamente utilizados no mercado.

6. Conclusão e Trabalhos Futuros

Este trabalho propôs, implementou e avaliou, de modo experimental, uma solução para facilitar o encadeamento de funções de rede em múltiplas nuvens. Partiu-se desta motivação e, levando-se em consideração os casos de uso apresentados, utilizou-se a arquitetura de plano controle proposta pelo ETSI e o `KeySFC` como solução para o plano de dados, que traz vantagens claras ao reduzir a complexidade dos dispositivos presentes no núcleo da rede, reduzir o gerenciamento dos estados nestes dispositivos e agilizar a criação de novos encadeamentos.

O presente trabalho apresenta um avanço significativo em relação ao `KeySFC`, uma vez que para habilitar roteamento entre nuvens é preciso propor uma arquitetura de interação entre serviços para obter as informações necessárias para o roteamento. Além disso, existe uma grande lacuna nos trabalhos relacionados na solução do problema de habilitar a consolidação de serviços de rede virtualizados entre domínios diferentes uma vez que boa parte da literatura foca em otimização e posicionamento de funções de rede em múltiplas nuvens, sem posicionar-se quanto aos mecanismos habilitadores para esta tecnologia.

O nosso trabalho mostra que SSR facilita a tarefa de estabelecer serviços em múltiplos domínios, uma vez que a conexão virtual entre as nuvens faz parte do esquema de roteamento através dos gateway de nuvens, além de possibilitar mecanismos eficientes de engenharia de tráfego. A contribuição inter nuvens será melhor explicada na versão final. Destaca-se, finalmente, que este é o primeiro trabalho a tratar do assunto usando uma solução baseada em SSR, mais especificamente o `KeySFC`.

Como trabalhos futuros destaca-se a necessidade de comparação formal e avaliação de desempenho da solução proposta com outros mecanismos habilitadores de SFC, tais como NSH. Outra possibilidade é incorporar soluções de engenharia de tráfego à solução proposta, visando garantir qualidade do serviço (ou *Quality of Service*, QoS) e requisitos de diferentes aplicações. Além disso estuda-se implementar desse mecanismo usando hardware programável (*switches*), incorporar soluções de otimização em orquestração de VNFs multi-domínios e estudo de soluções para melhorar a segurança da solução.

Agradecimentos

Este trabalho recebeu financiamento parcial proveniente de bolsas e projetos CNPq e FAPES, da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

Abdullah, Z. N. et al. (2019). Segment routing in software defined networks: A survey. *IEEE Communications Surveys Tutorials*, 21(1):464–486.

Bhamare, D. et al. (2017). Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications*, 102:1–16.

Both, C. et al. (2019). Futebol control framework: Enabling experimentation in convergent optical, wireless, and cloud infrastructures. *IEEE Communications Magazine*.

Clad, F. et al. (2018). Segment Routing for Service Chaining. Internet-draft, IETF.

Dietrich, D. et al. (2017). Multi-Provider Service Chain Embedding With Nestor. *IEEE Transactions on Network and Service Management*, 14(1):91–105.

Dolson, D. et al. (2018). Hierarchical Service Function Chaining (hSFC). 53:1–29.

Dominicini, C. K. et al. (2020). KeySFC: Traffic steering using strict source routing for dynamic and efficient network orchestration. *Computer Networks*, 167:106975.

Gomes, R. R. et al. (2016). Kar: Key-for-any-route, a resilient routing system. In *2016 46th Annual IEEE/IFIP DSN*, pages 120–127.

Guo, C. et al. (2010). Secondnet: A data center network virtualization architecture with bandwidth guarantees. In *Co-NEXT '10*, New York, NY, USA. ACM.

Gupta, L. et al. (2017). COLAP: A predictive framework for service function chain placement in a multi-cloud environment. *CCWC 2017*.

Jia, W. (2014). A scalable multicast source routing architecture for data center networks. *IEEE Journal on Selected Areas in Communications*, 32(1):116–123.

Jin, X. et al. (2016). Your data center switch is trying too hard. In *Proceedings of the Symposium on SDN Research*, SOSR '16, pages 12:1–12:6, NY, USA. ACM.

Jyothi, S. A. et al. (2015). Towards a flexible data center fabric with source routing. In *1st ACM SIGCOMM*, page 10. ACM.

Kumar, S. et al. (2017). Service Function Chaining Use Cases In Data Centers. Internet-Draft draft-ietf-sfc-dc-use-cases-06, Internet Engineering Task Force.

Lebrun, D. and Bonaventure, O. (2017). Implementing ipv6 segment routing in the linux kernel. In *Proceedings of the Applied Networking Research Workshop*, pages 35–41.

Mahmoodi, T. et al. (2017). Network Functions Virtualisation (NFV); Management and Orchestration. *IEEE Communications Standards Magazine*, 1(4):60.

Martinello, M. et al. (2014). Keyflow: A prototype for evolving SDN toward core network fabrics. *Network, IEEE*, 28:12–19.

Mijumbi, R. et al. (2016). Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 54(1):98–105.

Pignataro, J. M. H. and Carlos (2015). Service Function Chaining (SFC) Architecture. Technical Report 9.

Quinn, P. and Guichard, J. (2014). Service Function Chaining: Creating a Service Plane via Network Service Headers. *Computer*, 47(11):38–44.

Sun, G. et al. (2019). Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks. *Future Generation Computer Systems*, 91:347–360.

Tso, F. P., Jouet, S., and Pezaros, D. P. (2016). Network and server resource management strategies for data centre infrastructures: A survey. *Computer Networks*, 106.

Valentim, R. et al. (2019). RDNA Balance: Balanceamento de Carga por Isolamento de Fluxos Elefante em Data Centers com Roteamento na Origem. In *SBRC 2019*. SBC.

Vu, A. V. and Kim, Y. (2016). An implementation of hierarchical service function chaining using OpenDaylight platform. *IEEE NETSOFT 2016*, pages 411–416.