UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

GABRIEL DO AMARAL ALMEIDA

Estratégias de Balanceamento de Carga de Tráfego em Ambientes dotados de Funções Virtualizadas de Rede

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Prof. Dr. Luciano Paschoal Gaspary
Coorientador: Dr. Weverton Luis da Costa Cordeiro

Porto Alegre
dezembro de 2016

“Aprenda com o ontem, viva para o hoje, tenha esperança no amanhã.
O importante é não parar de questionar.”

— ALBERT EINSTEIN

AGRADECIMENTOS

Aos meus pais, pelo amor, dedicação, paciência, investimento e apoio ao longo de todos esses anos. Sem a ajuda deles, jamais teria chegado até aqui.

Ao meu orientador, Luciano Gaspary, pela paciência ao guiar meus primeiros passos no campo de pesquisa acadêmica, pelos conhecimentos adquiridos ao longo desse processo e pela oportunidade de fazer parte do Grupo de Redes do Instituto de Informática da UFRGS como bolsista de iniciação científica.

Ao meu coorientador, Weverton Cordeiro, pelo vasto conhecimento e experiência que pude tirar proveito ao longo do desenvolvimento deste trabalho.

Ao meu colega de laboratório, Gustavo Miotto, que gentilmente cedeu um componente crucial para a materialização deste trabalho.

RESUMO

Virtualização de Funções de Rede (*Network Function Virtualization*, NFV) é um novo 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). Dessa forma, neste trabalho propõe-se, desenvolve-se e avalia-se estratégias projetadas com o objetivo de balancear e encadear tráfego em ambientes dotados de funções virtualizadas de rede.

Palavras-chave: Redes de computadores. Software-Defined Networking. OpenFlow. Network Function Virtualization. Engenharia de tráfego.

ABSTRACT

Network Function Virtualization (NFV) is a novel computer 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 design, develop and evaluate strategies aimed at balacing load and steering network traffic in environments comprised of virtualized network functions.

Keywords: Computer networks. Software-Defined Networking. OpenFlow. Network Function Virtualization. Traffic engineering.

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ARP	Address Resolution Protocol
DPI	Deep Packet Inspection
ECMP	Equal-Cost Multipath Algorithm
GLP	Generalized Layering Problem
IP	Internet Protocol
MPLS	Multiprotocol Label Switching
NAT	Network Address Translation
NFV	Network Function Virtualization
N-PoP	Network Point-of-Presence
SDN	Software-Defined Networking
SFC	Service Function Chaining
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
VLAN	Virtual Local Area Network
VNF	Virtual Network Function
VoIP	Voice over IP

LISTA DE ALGORITMOS

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

As redes de computadores têm tido um aumento considerável em sua complexidade ao longo dos anos. Isso ocorre por uma série de fatores, dentre eles pode-se citar as funções de rede. Tais elementos compõem um agrupamento de diversas funcionalidades diferentes (*e.g. firewalling*, *caching*, *proxying*) que podem ser executadas sobre os dados que trafegam na infraestrutura de rede. Entretanto, esses dispositivos são proprietários, o que acarreta em uma baixa flexibilidade e em um alto custo de aquisição, manutenção e gerência.

Com o objetivo de simplificar os processos atrelados às funções de rede, foi proposta a Virtualização de Funções de Rede (*Network Function Virtualization*, NFV) (Network Functions Industry Specification Group, 2012). Trata-se de um paradigma que consiste na migração das funções de rede tradicionais para soluções apoiadas em *software* que executa em *hardware* de prateleira por meio de virtualização.

O surgimento de NFV alavancou inúmeros projetos de pesquisa nos últimos anos, justamente pela vasta abrangência de contextos em que pode-se utilizar as Funções Virtualizadas de Rede (*Virtualized Network Functions*, VNFs). De maneira semelhante, NFV beneficiou-se sobremaneira da arquitetura das Redes Definidas por Software (*Software-Defined Networks*, SDNs) (MCKEOWN, 2009), pois ambos os conceitos atuam de forma complementar.

A arquitetura SDN foi proposta com um propósito semelhante ao de NFV, pois visava a reduzir as dificuldades em gerência, administração e flexibilidade em redes de computadores. O conceito fundamental de SDN é o desacoplamento entre os planos de controle e dados. Em termos práticos, a função de encaminhamento dos dispositivos de rede (roteadores e *switches*) é mantida, mas o plano de controle é removido e atribuído a um elemento logicamente centralizado. Esse, denominado controlador de rede, é responsável pelas tomadas de decisões no encaminhamento dos dados. Dessa forma, a complexidade da infraestrutura de rede é simplificada, pois é desenvolvida em *software*. O protocolo de comunicação empregado para estabelecer a comunicação entre o plano de controle e o plano de dados é o OpenFlow (MCKEOWN et al., 2008).

Pode existir mais de uma VNF posicionada ao longo de uma infraestrutura e nem todos os dados que ingressam na rede são processados por alguma função. A definição dos fluxos de dados que devem transitar por quais VNFs antes de atingir o seu destino sucede-se por meio de um documento denominado Encadeamento de Funções de Serviço

(*Service Function Chaining*, SFC) (HALPERN; PIGNATARO, 2015), que deve estar disponível para o controlador direcionar os fluxos ao longo da infraestrutura. Dessa forma, o controlador deve respeitar as SFCs identificando fluxos sujeitos a VNFs e conduzir o tráfego ao longo dessas, o que é facilitado pela utilização de SDN.

Por atuar como um facilitador no processo de utilização de funções de rede, NFV permite diversas instâncias de uma mesma função posicionadas em diferentes pontos da rede, tendo, como consequência, mais de uma maneira de cumprir uma SFC. Consequentemente, verifica-se essencial a utilização equilibrada dos recursos de uma infraestrutura de rede a fim de garantir requisitos de latência e largura de banda mínima dos fluxos de rede, evitando congestionamentos dos enlaces que conectam os dispositivos de encaminhamento. Para evitar esses problemas e otimizar a utilização da infraestrutura, observa-se necessário o emprego de estratégias de engenharia de tráfego (AWDUCHE et al., 2002) pelo controlador de rede.

Existem estratégias de engenharia de tráfego já consolidadas em redes tradicionais, como *Equal-Cost Multipath Routing* (ECMP) (HOPPS, 2000) e *Multiprotocol Label Switching* (MPLS) (ROSEN; VISWANATHAN; CALLON, 2001), entretanto, tais estratégias são incapazes de estimar o estado global da rede em seus algoritmos devido à descentralização do plano de controle. Com o advento de SDN, permitiu-se uma maior maleabilidade para conduzir tráfego ao longo da infraestrutura, contudo, a presença de múltiplas VNFs introduziu um obstáculo na etapa de consolidação das rotas. Isso ocorre devido à necessidade de particionar uma rota considerando VNFs como nós intermediários que precisam fazer parte de um caminho fim-a-fim. Em caso de múltiplas instâncias de uma mesma função na infraestrutura, também é preciso realizar a opção por uma delas.

Este trabalho tem por objetivo principal propor e desenvolver estratégias de engenharia de tráfego aplicadas no contexto NFV. Também ambiciona-se estudar estratégias de engenharia de tráfego já propostas, bem como avaliar as estratégias desenvolvidas em situações específicas.

O restante deste trabalho está organizado como segue. O Capítulo 2 apresenta conceitos e fundamentos necessários à compreensão do trabalho, bem como uma síntese do estado da arte. O Capítulo 3 introduz as estratégias de engenharia de tráfego propostas. O Capítulo 4 apresenta a solução desenvolvida. O Capítulo 5 traz a metodologia de avaliação, assim como os resultados obtidos. Por fim, o Capítulo 6 conclui o documento com uma análise geral do desenvolvimento do trabalho seguido por planos de trabalhos futuros.

2 FUNDAMENTOS: REDES DEFINIDAS POR SOFTWARE E VIRTUALIZAÇÃO DE FUNÇÕES DE REDE

Este capítulo apresenta conceitos fundamentais relacionados a este trabalho. Inicia-se apresentando a arquitetura e o modelo de encaminhamento de SDN. Em seguida, expõe-se detalhes específicos de NFV. Por fim, é discutido o estado-da-arte no que diz respeito a estratégias de engenharia de tráfego em SDN e NFV.

2.1 Software-Defined Networking e o Protocolo OpenFlow

As redes de computadores tradicionais consistem em dispositivos de encaminhamento conectados entre si por meio de enlaces de rede. Esses dispositivos, representados por roteadores e *switches*, acumulam funções de controle (análise e decisão sobre o envio de dados recebidos) e de encaminhamento propriamente dito (envio dos dados recebidos por alguma de suas interfaces). Considerando que as funções de controle são repartidas entre os dispositivos de encaminhamento, pode-se verificar que a tomada de decisão é compartilhada entre esses, o que confere uma baixa programabilidade e flexibilidade à rede.

Dessa forma, pode-se constatar que a arquitetura tradicional de redes de computadores, em que ambos os planos são unificados em um mesmo equipamento, possui dificuldade em tratar os pacotes que trafegam na rede de forma mais minuciosa. Tendo em vista que a decisão de direcionamento dos pacotes é baseada majoritariamente na análise dos endereços IP de destino, existe um baixo potencial de customização na rede, principalmente no que refere-se a considerar outras peculiaridades de um fluxo de dados ao direcioná-lo ao longo da rede.

Com o objetivo de solucionar essa e outras limitações, foi proposta a arquitetura SDN, materializada por meio do protocolo OpenFlow. O protocolo foi proposto na universidade de Stanford e rapidamente foi "abraçado" pela indústria, tanto no que se refere à fabricação de dispositivos capazes de implementar o protocolo OpenFlow, quanto na utilização em suas respectivas infraestruturas de rede.

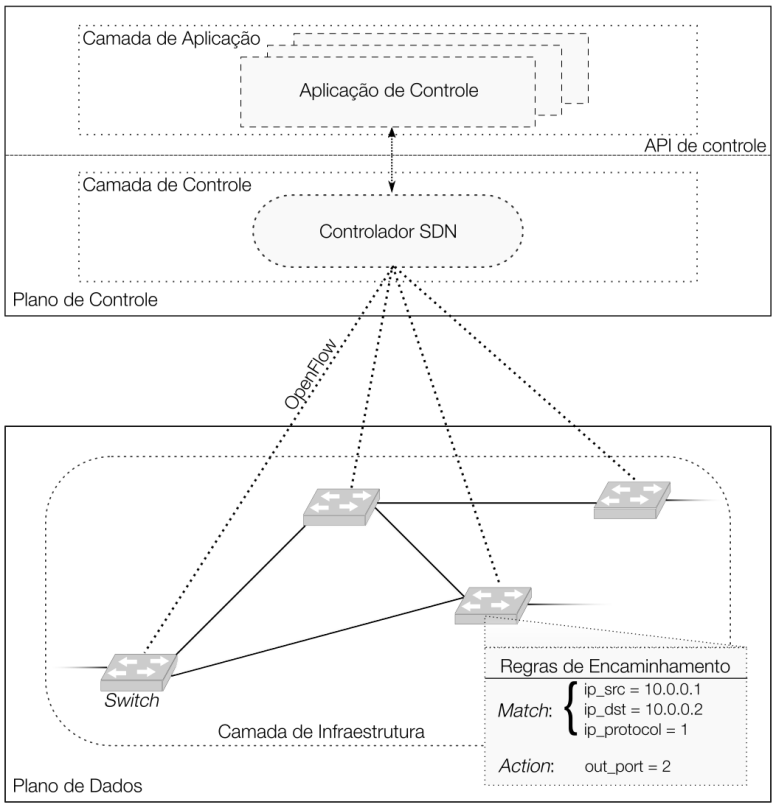
Em SDN é efetuado um desacoplamento do plano de controle, responsável pela tomada de decisões, do plano de dados, responsável pelo encaminhamento. O plano de controle é materializado através da presença de um controlador SDN e de uma ou mais

aplicações de controle, que se comunicam com o controlador SDN. Já o plano de dados é representado por *switches* OpenFlow e enlaces de rede.

Mais especificamente, os dispositivos de encaminhamento são responsáveis apenas pela análise e encaminhamento propriamente dito, relegando a tomada de decisão à aplicação de controle. O controlador, que possui visão global da rede, é responsável pela inserção de regras nas tabelas de fluxos dos *switches*. Sempre que um *switch* não possui uma regra de encaminhamento correspondente aos atributos de um fluxo de dados (representado por informações presentes no cabeçalho de um pacote), esse envia o pacote ao controlador, que o repassa à aplicação de controle. Ao analisá-lo, a aplicação de controle decide como pacotes pertencentes àquele fluxo devem ser tratados e repassa ao controlador uma regra de encaminhamento a ser inserida na tabela de fluxos desse *switch*.

A Figura 2.1 ilustra, com maiores detalhes, a arquitetura SDN. Dentro do plano de controle, existe uma divisão entre as camadas de aplicação e controle. A camada de aplicação é representada pelas aplicações de controle, que se comunicam, por meio de uma API, com o controlador SDN. Esse, por outro lado, atua na camada de controle e comunica-se com os *switches* do plano de dados por meio do protocolo OpenFlow, inserindo regras de encaminhamento nas tabelas de fluxos dos *switches* e obtendo informações (sobre configurações e estatísticas) da rede.

Figura 2.1: Detalhamento da arquitetura SDN



Detalhando-se um pouco mais o histórico e o funcionamento do protocolo OpenFlow, esse foi acolhido pela *Open Network Foundation* e teve sua primeira especificação elaborada em 2008, implementada através da versão 0.8. Após algumas versões de menor relevância, no final de 2009 foi lançada a versão 1.0, a primeira versão estável do OpenFlow. Passado pouco mais de um ano, foi desenvolvida a versão 1.1, onde houve a adoção de *tags* para fluxos, *multipath* e múltiplas tabelas. Oito meses mais tarde, foi lançada a versão 1.2, com suporte a IPv6 e reescrita de cabeçalhos, assim como uma extensão dos campos aceitos para *match*. Pouco mais de seis meses depois, a versão 1.3 foi concebida. Atualmente, essa é a versão mais utilizada. Possui suporte à obtenção de estatísticas de fluxos trafegando na infraestrutura e a *cookies*, dentre outras funcionalidades.

Em termos práticos, o protocolo atua por meio de regras *match+action*. *Match* indica os valores que campos do cabeçalho devem possuir para acionar a regra. A Tabela 2.1 expõe alguns campos disponíveis para *match* no OpenFlow versão 1.3. A ação (*action*) ocorre quando os campos de cabeçalho coincidem com aqueles especificados na regra. As principais ações possíveis são: reescrita do cabeçalho, encaminhamento do pacote por uma ou mais interfaces do *switch* ou descarte do pacote.

Tabela 2.1: Campos passíveis de *match*

Campo	Descrição
eth_src	Endereço físico de origem
eth_dst	Endereço físico de destino
eth_type	Código do protocolo encapsulado no cabeçalho Ethernet
ipv4_src	Endereço IP de origem
ipv4_dst	Endereço IP de destino
ip_proto	Código do protocolo encapsulado no protocolo IP
tcp_src	Porta origem do protocolo TCP
tcp_dst	Porta destino do protocolo TCP
udp_src	Porta origem do protocolo UDP
udp_dst	Porta destino do protocolo UDP

2.2 Network Function Virtualization

Além dos dispositivos de encaminhamento, podem existir dispositivos em uma rede responsáveis pela execução de operações diversas, denominadas funções de rede. Existem funções com finalidade de garantir segurança na rede (ex.: *firewalling*, *Deep Packet Inspection*), funções que buscam o aumento de desempenho da rede (ex.: *caching*), funções que atuam de forma a modificar o funcionamento natural da rede (ex.: *proxying*,

Network Address Translation).

Os dispositivos que executam as funções de rede, em sua forma tradicional, são desenvolvidos pela indústria e, por razões de mercado, possuem diversas informações de funcionamento suprimidas, tornando sua utilização semelhante a de uma "caixa preta". Além da falta de detalhes de funcionamento, possuem uma baixa flexibilidade (por apresentarem dificuldades de compatibilidade com produtos de outras empresas) e altos custos de aquisição, manutenção e gerência, por requererem mão-de-obra altamente especializada.

Todas essas dificuldades resultam em um engessamento na utilização de funções de rede. Com os objetivos de redução de custos e introdução de uma maior facilidade no processo de desenvolvimento e utilização de funções de rede, foi proposto o paradigma de Funções Virtualizadas de Rede. NFV consiste no desenvolvimento de *software* que atua da mesma forma que as funções de rede tradicionais, mas é executado sobre *hardware* de prateleira (*off-the-shelf*) por meio de virtualização. Esses equipamentos sobre os quais as funções virtualizadas de rede são executadas denominam-se Pontos de Presença da Rede (*Network Points-of-Presence*, N-PoPs).

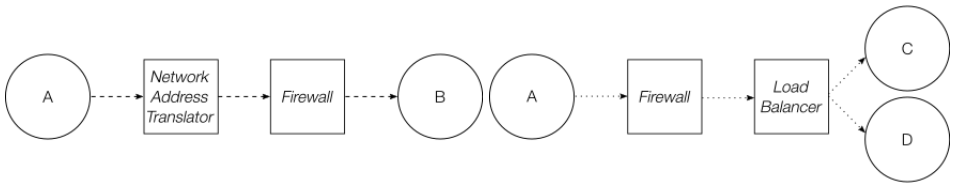
Ao empregar NFV em uma infraestrutura de rede, observa-se um potencial aumento da flexibilidade na utilização de funções, assim como uma redução de custos associados a aquisição, manutenção e gerência. Da mesma forma, constata-se uma melhoria na escalabilidade, tanto horizontal, por permitir um maior número de nós executando funções na infraestrutura, como vertical, por possibilitar mais de uma instância de função em cada um desses nós.

Existem fluxos de dados que devem ser processados por VNFs específicas (como tráfego UDP sendo processado por um *firewall*), da mesma forma que outros fluxos podem ser isentos de processamento de funções (como tráfego associado ao protocolo ARP). A maneira de definir quais fluxos devem ser tratados por quais funções é por meio das SFCs. Essas são especificações que definem o encadeamento necessário para cada fluxo. De posse das SFCs, uma entidade responsável pelo encadeamento de tráfego na rede deve materializar suas especificações por meio de regras de encaminhamento. O objetivo é fazer com que os fluxos especificados nas SFCs sejam direcionados ao longo das respectivas VNFs antes de alcançar seus destinos.

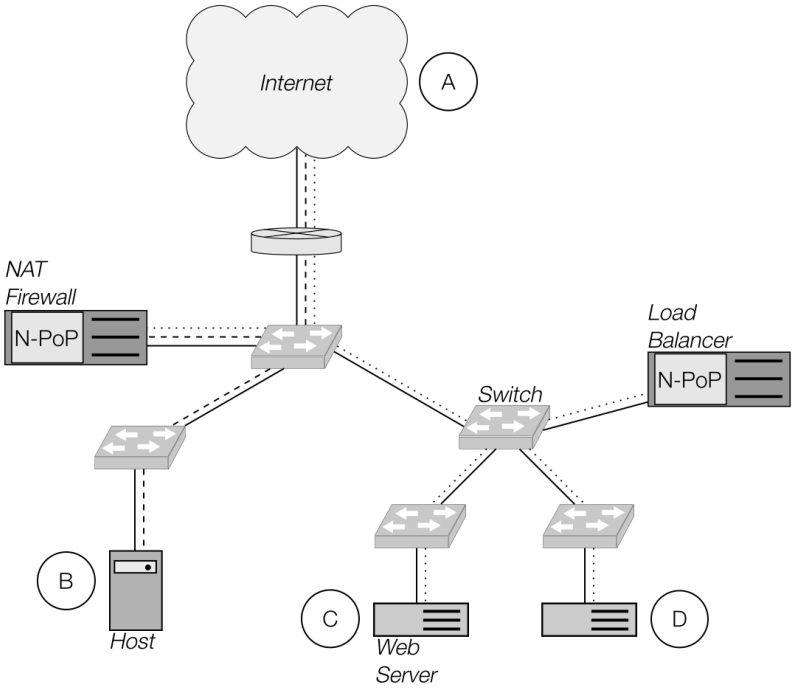
A Figura 2.2 ilustra a utilização de SFCs em uma rede. Foi especificada uma SFC para os fluxos destinados ao *host* (B) (Figura 2.2a), que devem ser processados por um NAT e um *firewall*. Também foi especificada uma SFC para os fluxos destinados ao con-

junto de *web servers* (C e D) (Figura 2.2b), que devem ser processados por um *firewall* e por um *load balancer*. O tráfego ingressante na infraestrutura (A) (Figura 2.2c) deve ser encadeado de acordo com as SFCs. Dessa forma, os fluxos direcionados ao *host* (representados pela linha tracejada) vão ser processados pelo N-PoP que implementa o *firewall* e o NAT e, em seguida, encaminhados ao *host* (B). De forma análoga, os fluxos direcionado ao par de *web servers* (representados pela linha pontilhada) vão ser processados pelo N-PoP que implementa o *firewall* e pelo N-PoP que implementa o *load balancer* previamente ao encaminhamento até o destino (C e D).

Figura 2.2: Exemplo de utilização de SFCs em uma infraestrutura de rede



(a) SFC do *host* (b) SFC do *web server*



(c) Materialização das SFCs na rede

Existem diversas formas de encadear tráfego na rede a fim de concretizar a utilização de uma SFC. Pode-se tirar proveito de protocolos como 802.1Q, que permite a criação de redes virtuais (*VLANs*), ou MPLS, que permite rotular pacotes. A forma mais comum, contudo, vem sendo pelo uso da programabilidade oferecida pelo protocolo OpenFlow. Quanto à efetivação de VNFs na rede, existem diferentes *frameworks* capazes de realizar

a instanciação das funções, como, por exemplo, *OpenBox* (BREMLER-BARR; HARCHOL; HAY, 2015), *OpenNF* (GEMBER-JACOBSON et al., 2015) e *ClickOS* (MARTINS et al., 2014).

2.3 Engenharia de Tráfego em SDN

Com o advento de uma maior capacidade de controle das redes de computadores, é natural almejar um aumento de desempenho. Esse aumento pode ser alcançado, principalmente, por meio de estratégias de engenharia de tráfego. A arquitetura SDN e o protocolo OpenFlow atuam como facilitadores no que refere-se à obtenção de informações que podem ser consideradas ao realizar engenharia de tráfego na rede, como a quantidade de fluxos, a quantidade de pacotes e, até mesmo, a quantidade de *bytes* recebidos e transmitidos por cada interface dos *switches*.

Como infraestruturas de rede têm testemunhado uma transição de redes tradicionais para SDN, funções de rede devem ser suportadas e também operar normalmente em ambientes que utilizam o paradigma. Tendo essa premissa em vista, pode-se tirar proveito da programabilidade que SDN oferece para realizar engenharia de tráfego considerando VNFs dispostas pela rede.

Dessa forma, realizou-se um levantamento e classificação das abordagens mais utilizadas para materializar engenharia de tráfego e balanceamento de carga em redes SDN, resumido na Tabela 2.2. Para fins de organização, a investigação está classificada considerando um conjunto de critérios dos trabalhos analisados. A primeira coluna indica o suporte a VNFs nos algoritmos propostos. A segunda coluna denota a generalidade do algoritmo no que refere-se a topologias de rede, podendo atuar em qualquer topologia ou apenas em um conjunto específico. A terceira coluna refere-se à capacidade, ou incapacidade, de realizar engenharia de tráfego independentemente da aplicação que percorre a rede. A quarta coluna identifica a métrica em que a tomada de decisão é fundamentada, podendo ser baseada na utilização dos enlaces, no tipo de fluxo que trafega na rede, numa combinação de ambos, ou na latência fim-a-fim. Finalmente, a quinta coluna identifica as abordagens que propõem cada combinação das especificações citadas.

Tabela 2.2: Classificação de trabalhos no contexto de engenharia de tráfego em SDN

VNFs	Topologia	Aplicação	Decisão	Trabalho
Não Suporta	Independe	Independe	Tipo de Fluxo	(SU et al., 2014)
				(KAMIYAMA et al., 2014)
			Tipo de Fluxo e Utilização	(LEE; HONG; LI, 2015)
		Específica	Tipo de Fluxo	(ADAMI et al., 2015)
			Utilização	(CRAIG et al., 2015)
	Específica	Independe	Utilização	(LI; SHANG; CHEN, 2014)
		Específica	Utilização	(HANDIGOL et al., 2009)
				(LI; PAN, 2013)
				(RODRIGUES et al., 2015)
Suporta	Independe	Independe	Latência fim-a-fim	(DWARAKI; WOLF, 2016)

A característica mais frequente nas investigações citadas é a falta de suporte a VNFs na rede. Apesar da abordagem proposta por Adami *et al.* (ADAMI et al., 2015) oferecer suporte a uma instância de DPI para classificar tráfego, trata-se de uma função tradicional de rede pré-posicionada e alocada na rede. Por outro lado, destaca-se o trabalho desenvolvido por Dwaraki e Wolf (DWARAKI; WOLF, 2016), que realiza engenharia de tráfego em qualquer infraestrutura com VNFs, contanto que especificadas previamente.

Quanto à especificidade das topologias que cada investigação suporta, pode-se observar trabalhos desenvolvidos que não estão restritos a apenas uma classe topológica, como a proposta de Dwaraki e Wolf (DWARAKI; WOLF, 2016). A abordagem citada foi avaliada em diversas topologias geradas aleatoriamente por meio dos modelos de grafos *Waxman*, GLP e BA-2. Em contrapartida, apesar de apresentarem resultados satisfatórios, algumas investigações limitam-se a topologias específicas, podendo-se citar a de Li e Pan (LI; PAN, 2013), que opera apenas em topologias *fat-tree*.

Analisando as aplicações-objetivo de cada proposta desenvolvida, observa-se uma grande variedade de finalidades. Constata-se aplicações de balanceamento de carga em servidores (HANDIGOL et al., 2009), assim como de engenharia de tráfego para *multicast* (CRAIG et al., 2015) e *streaming*, VoIP e jogos (ADAMI et al., 2015). Outras abordagens possuem cunho geral, tratando todo o tráfego que transita pela rede, com o objetivo de alcançar uma utilização equilibrada dos recursos independentemente da aplicação, como ocorre na proposição de Su *et al.* (SU et al., 2014).

Outro aspecto relevante nas propostas selecionadas é a métrica utilizada para definir as melhores rotas para novos fluxos na rede. De forma geral, percebe-se uma preferência por classificação de rotas fundamentada na utilização dos enlaces e *switches* da

rede, como na abordagem de Craig *et al.* (CRAIG et al., 2015). Nesse caso, os autores desenvolveram uma implementação que realiza uma atualização dos pesos de enlaces em tempo real. Alternativamente, existem proposições que consideram os tipos de fluxos que estão percorrendo a rede, como no trabalho de Su *et al.* (SU et al., 2014), que divide os fluxos em elefante (de maior ocupação de banda) e rato (de menor ocupação de banda). Complementarmente, Lee, Hong e Li (LEE; HONG; LI, 2015) utilizam uma classificação que considera o uso de banda em conjunto com os tipos de fluxos que trafegam pelo enlace. Por fim, a proposta de Dwaraki e Wolf (DWARAKI; WOLF, 2016) categoriza tráfego em função da latência fim-a-fim de uma rota específica.

Por apresentar uma maior semelhança com a proposição do trabalho realizado no âmbito deste Trabalho de Graduação, optou-se por avaliar mais detalhadamente a proposta de Dwaraki e Wolf (DWARAKI; WOLF, 2016). Primeiramente, apesar de apresentar suporte a VNFs dispostas na infraestrutura, deve-se questionar a baixa escalabilidade da solução proposta para uma grande quantidade de SFCs. Isso pode ser verificado pela forma com que o trabalho trata cada salto de uma SFC, replicando a quantidade de nós presentes na rede em uma nova partição do grafo que representa a rede. Para m SFCs, com n saltos cada, observa-se uma estrutura mn vezes maior do que a existente. Esse acréscimo na representação lógica da rede pode apresentar problemas no que diz respeito ao custo de processamento do cálculo da rota ideal. Devido ao grafo apresentar um múltiplo da quantidade real de nós presentes na rede, o custo de processamento ao avaliar todas as rotas possíveis entre origem e destino apresenta complexidade exponencial.

Tendo-se realizado essa análise, é possível afirmar que, apesar de engenharia de tráfego em SDN contar com o desenvolvimento e a proposição de diversas abordagens, a consideração da presença de VNFs nesse contexto ainda está em fase embrionária. Dentre os desafios que se apresentam nessa direção, pode-se citar a avaliação de propostas de engenharia de tráfego "mais baratas" com VNFs que optam pela melhor rota ponto-a-ponto (entre *hosts* e VNFs ou entre VNFs) ou fim-a-fim (entre *host* origem e *host* destino).

3 PROPOSTA

Este capítulo tem por objetivo o detalhamento das estratégias de engenharia de tráfego propostas e desenvolvidas neste trabalho. Primeiramente, são apresentados conceitos necessários para a melhor compreensão das proposições. Posteriormente, as estratégias propostas são aprofundadas e exemplificadas.

3.1 Conceituação e Classificação

Antes de aprofundar-se nas estratégias propostas, é preciso identificar os principais aspectos de classificação utilizados. Mais especificamente, a Seção 3.1 busca categorizar este trabalho em perspectiva à Tabela 2.2.

Ao realizar a proposição de novas estratégias de engenharia de tráfego, é necessário considerar *trade-offs* dentro do contexto de engenharia de tráfego e de SDN. Na proposta de Dwaraki e Wolf (DWARAKI; WOLF, 2016), por exemplo, optou-se por uma aplicação de controle mais robusta, em que foi feita a opção pela otimização de rotas fim-a-fim e pela utilização do atraso médio de cada rota como fator da tomada de decisão. Entretanto, a obtenção de rotas seguindo a otimização fim-a-fim tende a apresentar um *overhead* de processamento considerável, o que pode tornar a aplicação de controle um gargalo na rede, como já foi apurado por algumas investigações (SEZER et al., 2013; TOOTOONCHIAN et al., 2012).

Partindo do primeiro aspecto de caracterização da proposta, acredita-se que seja necessária a simplificação da aplicação responsável pelas tomadas de decisões, objetivando um menor custo de processamento proveniente do plano de controle. Desse modo, optou-se pela decisão de rotas fundamentada apenas na utilização dos enlaces de rede, com o objetivo de não sobrecarregar a aplicação de controle no que diz respeito à necessidade de processar as informações provenientes dos *switches*.

No que refere-se ao segundo aspecto de caracterização, é preciso ressaltar também a necessidade de suporte às VNFs presentes na rede. A aplicação de controle deve ser capaz de tratar as funções virtualizadas, desde que essas já estejam posicionadas na rede e a aplicação esteja ciente de suas respectivas localizações. Também é necessário que a aplicação esteja a par das especificações de encadeamento explicitadas nas SFCs, que regem o tráfego na infraestrutura, pois essas possuem papel-chave na tomada de decisão sobre cada rota.

Com o objetivo de caracterizar o terceiro aspecto desta proposição, é preciso considerar a forma de otimização das rotas. Mais especificamente, a opção por otimização de rotas *hop-by-hop* (entre um *host* e uma VNF ou entre duas VNFs) ou por otimização fim-a-fim (optando pela melhor rota entre dois *hosts* incluindo VNFs). A otimização de rotas fim-a-fim tende a apresentar um custo de processamento mais elevado, podendo, consequentemente, aumentar o atraso que a aplicação de controle introduz nos tráfegos da rede. Dessa forma, optou-se por otimização de rotas *hop-by-hop*, que, apesar de apresentar resultados inferiores à otimização fim-a-fim, oferece um custo procedural reduzido. No escopo deste trabalho, cada rota parcial encontrada entre dois *hops* especificados na SFC denomina-se sub-rota e, quando concatenadas, caracterizam uma rota completa.

Em um quarto aspecto de classificação, é preciso especificar a abrangência de operação da aplicação de controle no que refere-se a topologias de rede. Tendo em vista o objetivo deste trabalho, verifica-se crucial a capacidade de operação da aplicação de controle em qualquer topologia, pois isso confere generalidade da solução para uso em diversos cenários. Assim sendo, a aplicação de controle proposta neste trabalho deve ser capaz de operar sobre qualquer topologia de rede.

Por fim, o quinto e último aspecto que está passível de classificação é o suporte e tratamento a diferentes aplicações que trafegam ao longo da rede. Observando a generalidade que este trabalho visa a apresentar, tanto no plano de suporte a topologias quanto no plano de suporte a VNFs, acredita-se que suportar todas as aplicações e tratá-las igualmente vai ao encontro dos aspectos propostos citados. Portanto, a opção realizada é pelo suporte a qualquer aplicação de rede.

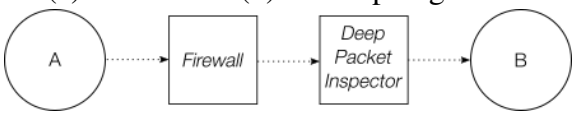
3.2 Estratégias de Engenharia de Tráfego

Dentre os pilares sobre os quais esta investigação está fundamentada, destaca-se a utilização de SDN/OpenFlow. Isso ocorre pela maior flexibilidade e programabilidade oferecida pelo paradigma. Tirando proveito de SDN/OpenFlow, pode-se obter visualização global da rede e coletar informações relativas aos enlaces e *switches* em intervalos de tempo da ordem de segundos. Com essas funcionalidades, oportuniza-se os processamentos dos dados citados e a respectiva atribuição de pesos lógicos aos enlaces. A aplicação de controle pode, então, consultar os pesos atribuídos aos enlaces sempre que for definir uma rota para um novo fluxo de rede.

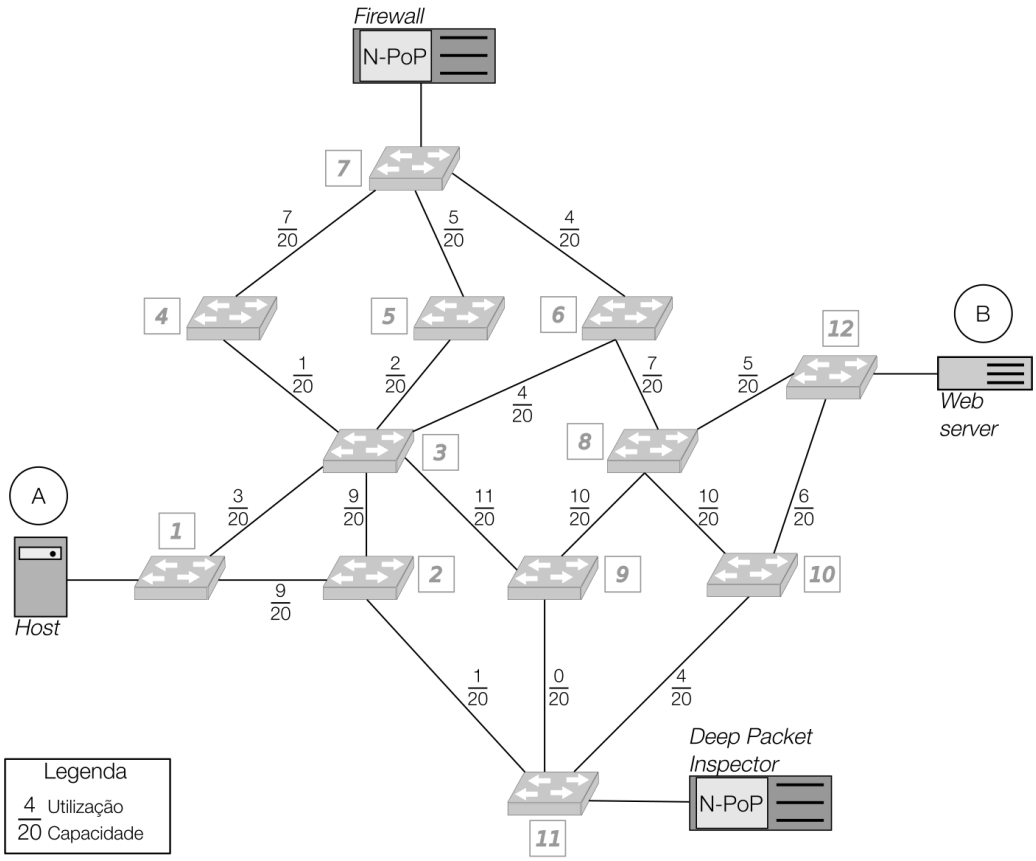
Com esse espaço de exploração em mente, propôs-se três estratégias de engenharia

de tráfego no contexto de NFV. Para exemplificar e facilitar a compreensão de cada uma das estratégias propostas, emprega-se a Figura 3.1. Pode-se observar na Figura 3.1a a única SFC a ser materializada na rede, onde é especificado que todo tráfego originário de A deve passar por um *firewall* e por um sistema de DPI antes de chegar até B. Na Figura 3.1b, visualiza-se que o nó A refere-se a um *host*, conectado ao *switch* 1, e que o nó B, conectado ao *switch* 12, refere-se a um *web server*. Também é possível verificar a existência de dois N-PoPs. O primeiro executa um *firewall* e está conectado ao *switch* 7. Já o segundo encontra-se conectado ao *switch* 11 e executa um sistema de DPI. Também é possível constatar os custos lógicos instantâneos atribuídos aos enlaces, em um dado momento, pela aplicação de controle, considerando que todos esses possuem a mesma capacidade máxima de transmissão.

Figura 3.1: Rede utilizada como exemplo para materialização das estratégias de engenharia de tráfego a partir de: (a) uma SFC e (b) uma topologia de rede



(a) SFC



(b) Rede física

3.2.1 Maior Disponibilidade

A primeira estratégia proposta visa à opção pela sub-rota que possuir o menor valor de utilização dentre os enlaces mais utilizados de cada sub-rota candidata. Isto é, em termos práticos, a sub-rota que possuir a maior disponibilidade de banda entre dois nós que compõem a SFC. Em caso de empate, a opção realizada é pela sub-rota que possuir a menor quantidade de saltos. Intuitivamente, essa estratégia tem por objetivo garantir a maior banda disponível para novos fluxos na rede.

Deve-se supor que existam m sub-rotas entre dois nós, representadas pelo conjunto $R = \{r_1, r_2, ..., r_m\}$, que cada uma dessas possua um conjunto de enlaces $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,k}\}$, de tamanho qualquer, e que cada enlace implemente função de custo do enlace $c(e_{i,j})$ para o j-ésimo enlace da i-ésima sub-rota. Para cada sub-rota r_i, o conjunto $C_{r_i} = \{c(e_{i,1}), c(e_{i,2}), ..., c(e_{i,k})\}$ caracteriza o custo atribuído a cada enlace que a compõe. Dessa forma, a sub-rota vencedora será aquela satisfizer a Expressão 3.1.

$$min(max(C_{r_i}), max(C_{r_2}), ..., max(C_{r_m})) \tag{3.1}$$

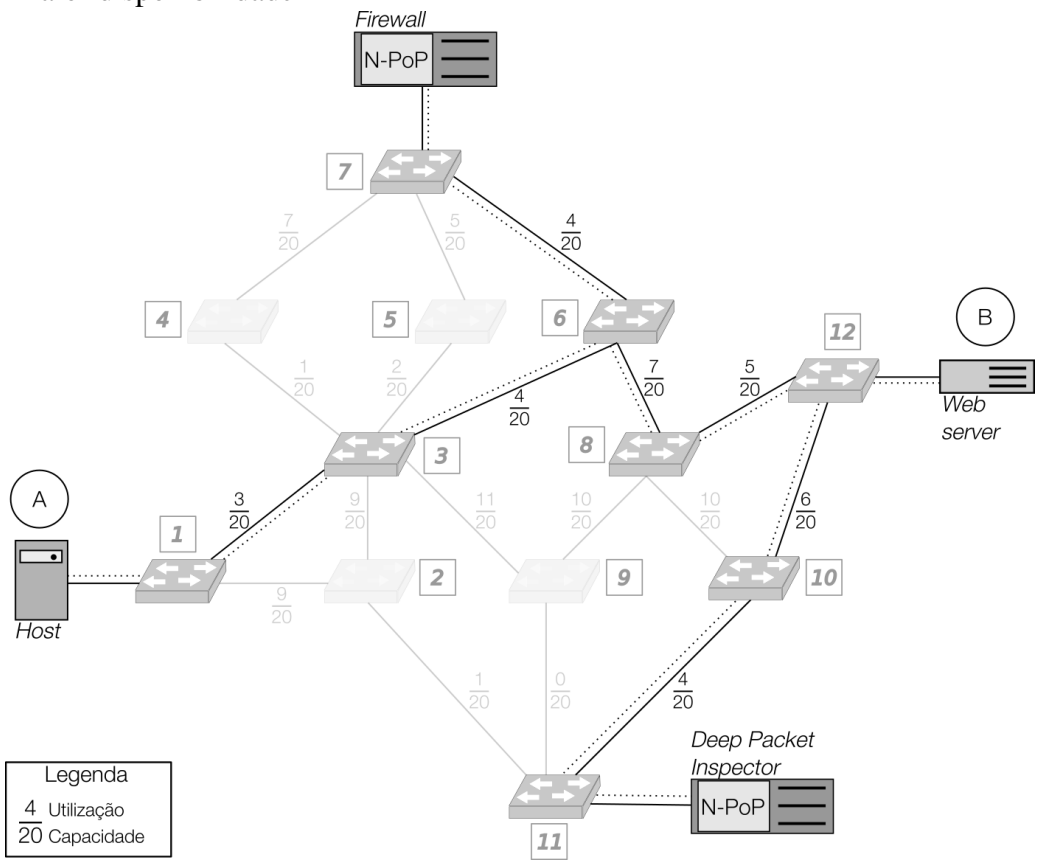
Para exemplificar o funcionamento desta estratégia, retoma-se a Figura 3.1. Ao aplicar a primeira estratégia no fluxo proveniente do *host* A e destinado ao *web server* B, os dados precisam, obrigatoriamente, trafegar pelo *firewall* e pelo sistema de DPI. Para identificar uma sub-rota entre o *host* e o *firewall*, verifica-se que, ao aplicar as sub-rotas candidatas à Expressão 3.1, obtém-se o valor 4, presente apenas na sub-rota $(1, 3, 6, 7)$.

Para o segundo passo especificado na SFC, é necessário encontrar uma sub-rota entre o *firewall* e o sistema de DPI. Nesse caso, o resultado da expressão é 7, presente, em menor quantidade de saltos, na sub-rota $(7, 6, 8, 12, 10)$.

Por fim, para completar a rota, é preciso encontrar uma sub-rota entre o sistema de DPI e o *web server*. A sub-rota que apresenta a maior disponibilidade é composta pelos *switches* $(11, 10, 12)$, com maior utilização igual a 6. Portanto, como rota entre *host* e *web server*, passando pelo *firewall* e pelo sistema de DPI, o resultado obtido através da aplicação da estratégia é $(1, 3, 6, 7, firewall, 7, 6, 8, 12, 10, DPI, 11, 10, 12)$.

A Figura 3.2 ilustra a rede resultante após a execução de um algoritmo que implemente a estratégia de maior disponibilidade de banda. Os *switches* e enlaces utilizados estão destacados em relação àqueles que foram preteridos durante a execução do algoritmo.

Figura 3.2: Topologia resultante após a execução da estratégia que obtém as sub-rotas com maior disponibilidade



3.2.2 Menor Média de Utilização

A segunda estratégia proposta é baseada no cálculo, por meio de média aritmética, da utilização de todos os enlaces de cada sub-rota candidata entre dois nós de uma SFC. Essa estratégia atua como desempate entre sub-rotas para um algoritmo de caminho mais curto. Tal opção foi realizada com o objetivo de evitar a opção por sub-rotas muito longas que possam apresentar um gargalo. Por exemplo, se duas ou mais sub-rotas forem detentoras da mesma quantidade de saltos, aquela que possuir a menor média de utilização é atribuída como sub-rota vencedora. O intuito desta estratégia é fazer com que a rede opere de forma equilibrada, sem que atue de forma gulosa ou complacente com os recursos da infraestrutura.

Novamente, supondo que existam m sub-rotas entre dois nós, representadas pelo conjunto $R = \{r_1, r_2, ..., r_m\}$, que cada uma dessas possua um conjunto de n enlaces $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,n}\}$, e que cada enlace implemente função de custo do enlace $c(e_{i,j})$ para o j-ésimo enlace da i-ésima sub-rota. O conjunto $C_{r_i} = \{c(e_{i,1}), c(e_{i,2}), ..., c(e_{i,n})\}$ caracteriza, para cada sub-rota r_i, o custo atribuído a cada enlace que a compõe. Dessa

forma, a sub-rota vencedora será aquela que satisfizer a Expressão 3.2.

$$min(\frac{\sum_{j=1}^{n} c(e_{1,j})}{|C_{r_1}|}, \frac{\sum_{j=1}^{n} c(e_{2,j})}{|C_{r_2}|}, ..., \frac{\sum_{j=1}^{n} c(e_{m,j})}{|C_{r_m}|}) \tag{3.2}$$

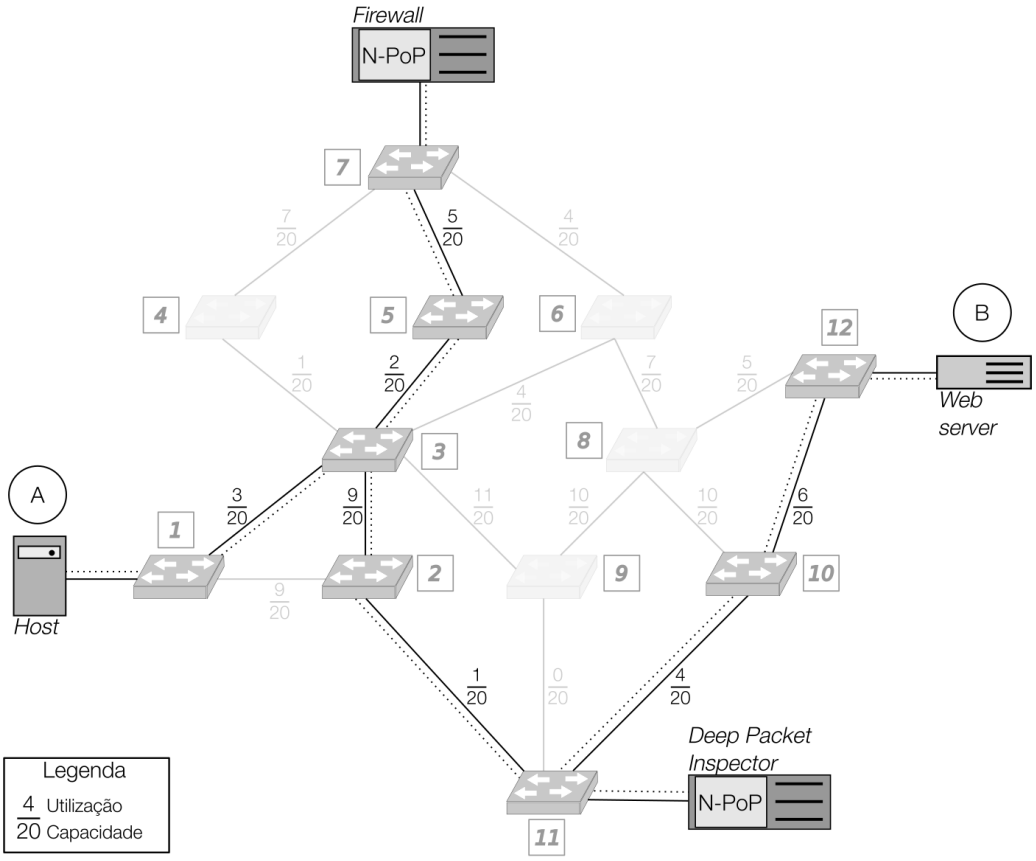
Analogamente à estratégia anterior, com o objetivo de exemplificação de funcionamento dessa estratégia, retoma-se a Figura 3.1. Objetiva-se utilizar a estratégia baseada nas sub-rotas que possuem menor média de utilização na infraestrutura referida. Nesse caso, o algoritmo implementado deve avaliar as sub-rotas com o menor número de saltos. Para o primeiro *hop* da SFC, em que o fluxo origina do *host* e é destinado ao *firewall*, visualiza-se três sub-rotas candidatas com 3 saltos nos *switches* da rede. São elas: $(1, 3, 4, 7)$, $(1, 3, 5, 7)$ e $(1, 3, 6, 7)$. Ao calcular a utilização média de cada uma dessas, obtém-se a rota $(1, 3, 5, 7)$ como vencedora, com uma utilização média de 3,33.

Com o intuito de obter a segunda fatia da rota completa, é preciso obter uma sub-rota entre o *firewall* e o sistema de DPI presentes na rede. Nesse caso, o número de sub-rotas candidatas é 5, entretanto, a rota vencedora está presente em $(7, 5, 3, 2, 11)$, com custo médio de 4.25.

Por fim, a sub-rota com o menor número de saltos entre o sistema de DPI e o *web server* está presente em $(11, 10, 12)$. Portanto, a rota completa entre *host* e *web server* é formada por: $(1, 3, 5, 7, firewall, 7, 5, 3, 2, 11, DPI, 11, 10, 12)$.

A Figura 3.3 apresenta a rede obtida após a execução de um algoritmo que encontra as melhores sub-rotas considerando a menor média de utilização dos enlaces. Os *switches* e enlaces destacados representam aqueles que seriam utilizados no cenário apresentado e na rede hipotética.

Figura 3.3: Topologia resultante após a execução da estratégia que obtém as sub-rotas com menor média de utilização



3.2.3 Enlace Menos Utilizado

Por fim, a terceira estratégia idealiza a proposição de uma estratégia que opta pela sub-rota que possuir o enlace de menor utilização. É importante notar que essa estratégia também atua apenas como forma de desempate entre sub-rotas para uma estratégia de caminho mais curto. Da mesma forma que a estratégia anterior, se duas ou mais sub-rotas possuírem a mesma quantidade de saltos, o desempate é realizado por meio desta estratégia e, assim, sucessivamente enquanto for necessário. Essa estratégia foi proposta com a intenção de equilibrar o uso das sub-rotas por meio da equiparação entre os enlaces menos utilizados dessas.

Supondo, novamente, que existam m sub-rotas entre dois nós, representadas pelo conjunto $R = \{r_1, r_2, ..., r_m\}$, que cada uma dessas possua um conjunto de n enlaces $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,n}\}$, e que cada enlace implemente função de custo do enlace $c(e_{i,j})$ para o j-ésimo enlace da i-ésima sub-rota. O conjunto $C_{r_i} = \{c(e_{i,1}), c(e_{i,2}), ..., c(e_{i,n})\}$ caracteriza, para cada sub-rota r_i, o custo atribuído a cada enlace que a compõe. Dessa

forma, a sub-rota vencedora será aquela que satisfizer a Expressão 3.3.

$$min(min(C_{r_1}), min(C_{r_2}), ..., min(C_{r_m})) \tag{3.3}$$

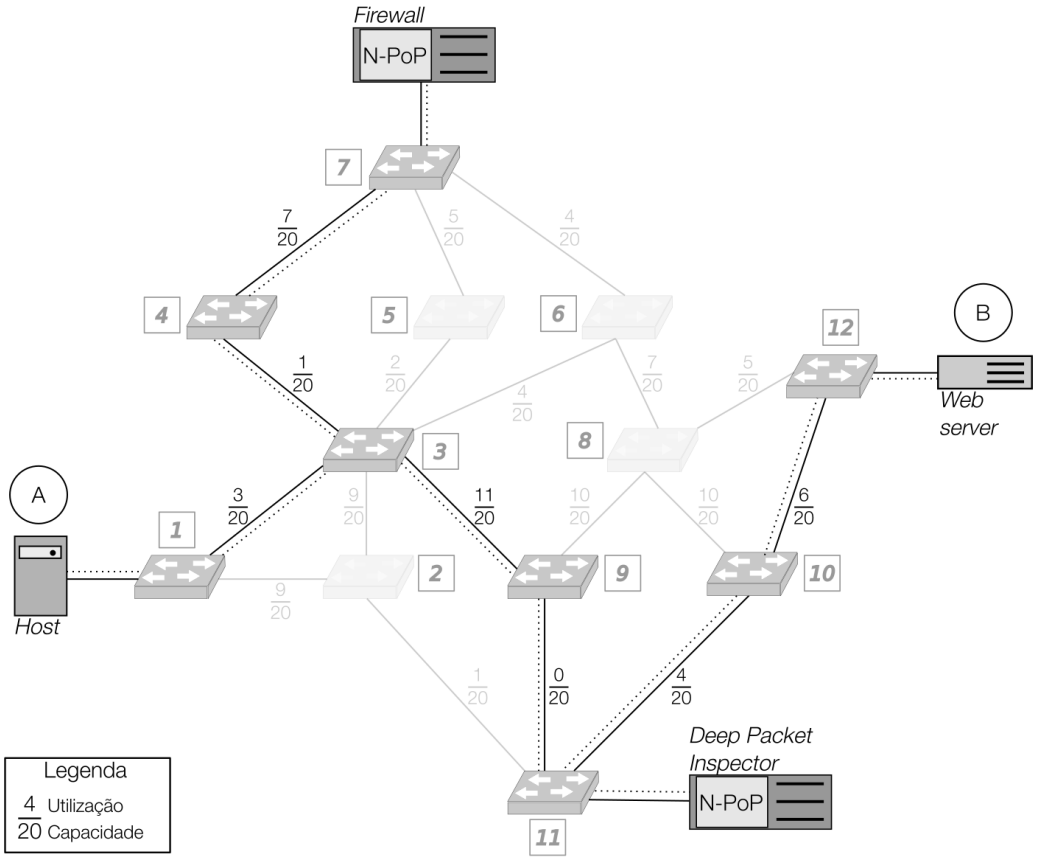
Finalmente, com o objetivo de demonstrar o funcionamento da estratégia, utiliza-se novamente a Figura 3.1. Caso almeje-se utilizar a estratégia que opta pela sub-rota que possui o enlace menos utilizado na infraestrutura referida, deve-se encontrar três sub-rotas que compõem a rota entre *host* e *web server*. Inicialmente, para encontrar uma sub-rota entre o *host* e o *firewall*, é necessário considerar as sub-rotas que possuírem a menor quantidade de saltos. No presente caso, observa-se três sub-rotas entre os nós citados. Dentre essas, aquela que possui o enlace com maior disponibilidade de banda está presente em $(1, 3, 4, 7)$, com o enlace $(3, 4)$ apresentando utilização 1.

O próximo passo consiste na obtenção de uma sub-rota entre o *firewall* e o sistema de DPI. Constata-se que as sub-rotas com o menor número de saltos entre as VNFs possuem quatro saltos (entre os *switches*), e que o enlace de maior banda disponível está presente em $(9, 11)$. Como esse enlace é compartilhado por mais de uma rota, é preciso encontrar o segundo enlace com maior disponibilidade de banda, resultando no enlace $(4, 3)$ e, consequentemente, na sub-rota $(7, 4, 3, 9, 11)$.

Para a obtenção de uma sub-rota entre o sistema de DPI e o *web server*, deve-se atentar que apenas uma sub-rota possui dois saltos. Portanto, essa sub-rota é a resultante: $(11, 10, 12)$. Dessa forma, pode-se constatar que a rota completa entre *host* e *web server* é composta por $(1, 3, 4, 7, firewall, 7, 4, 3, 9, 11, DPI, 11, 10, 12)$.

Com o objetivo de facilitar a verificação da rota obtida por meio da materialização da estratégia referida, lança-se mão da Figura 3.4. De maneira semelhante às Figuras 3.2 e 3.3 apresentadas anteriormente, esta apresenta os enlaces e *switches* utilizados no caso do emprego da estratégia fundamentada na utilização das sub-rotas com maior disponibilidade em um único enlace.

Figura 3.4: Topologia resultante após a execução da estratégia que obtém as sub-rotas que possuem o enlace menos utilizado



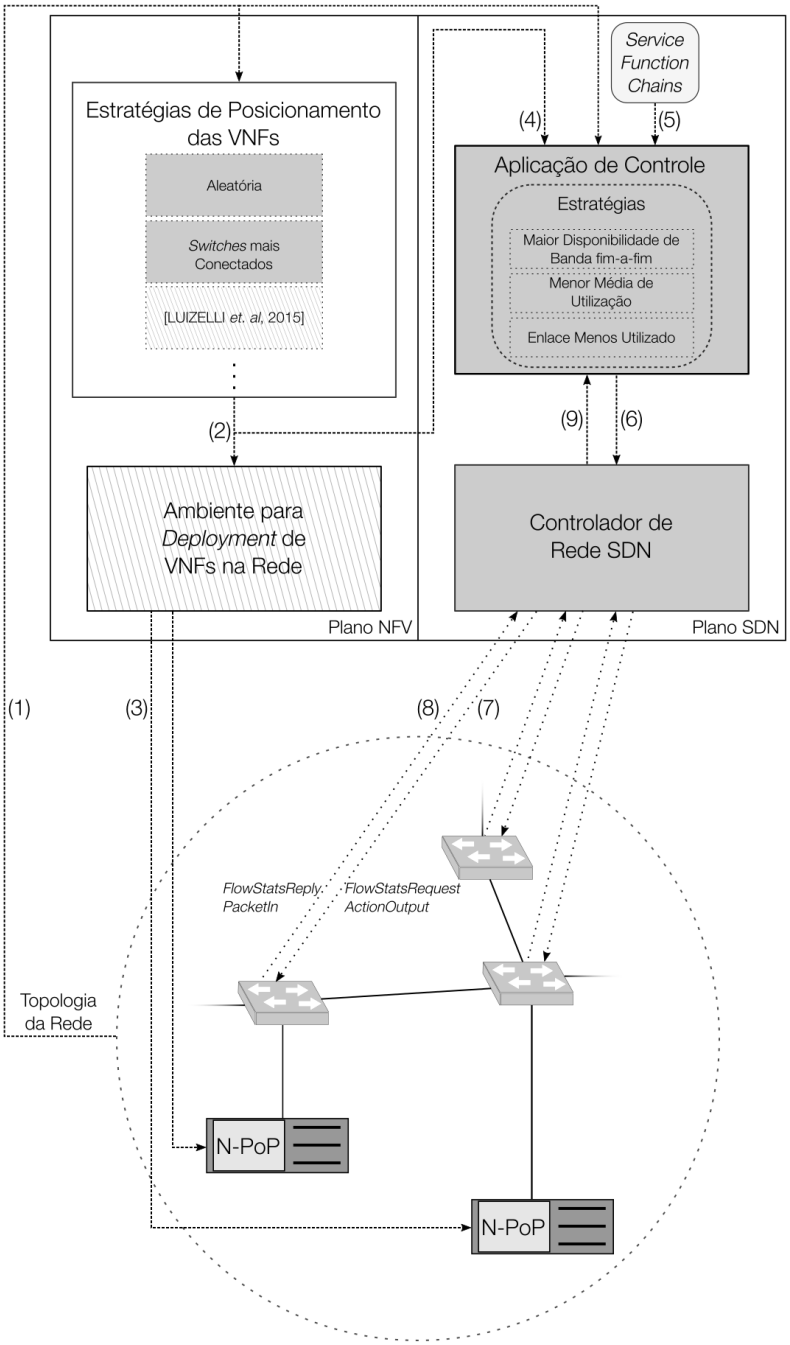
4 IMPLEMENTAÇÃO

Este capítulo apresenta detalhes da implementação que materializa os algoritmos propostos. Primeiramente, é apresentada uma visão geral da arquitetura proposta. Em um segundo momento, detalha-se o desenvolvimento dos algoritmos.

4.1 Visão Geral

Como parte importante do trabalho, desenvolveu-se uma arquitetura capaz de implementar as propostas de engenharia de tráfego expostas no Capítulo 3 em redes SDN/OpenFlow que possuam VNFs dispersas pela infraestrutura. A Figura 4.1 apresenta uma visão geral da arquitetura. Ela conta com uma separação clara entre dois planos existentes na rede, o plano NFV e o plano SDN. Os *switches* da rede interagem com o plano NFV, como indica o fluxo (1), a fim de posicionar as VNFs de acordo com alguma estratégia de posicionamento existente, como, por exemplo, aleatoriamente, posicionando as VNFs nos *switches* mais conectados ou a partir da proposta de Luizelli *et. al* (LUIZELLI et al., 2015). Após identificar os *switches* ideais para posicionar as funções, esse componente repassa os resultados calculados para a aplicação de controle SDN/OpenFlow e para o componente de instanciação de VNFs na infraestrutura, como indicado pelo fluxo (2). O fluxo (3) indica a materialização das funções virtualizadas de rede, onde o componente de *deployment* interage diretamente com os N-PoPs. Por sua vez, o plano SDN é responsável por tratar os fluxos que trafegam na rede. O controlador de rede atua de forma a estabelecer uma interface de comunicação entre a aplicação de controle, representada pelos fluxos (6) e (9), e os *switches*, representada pelos fluxos (7) e (8). Já a aplicação de controle é responsável por empregar as estratégias de engenharia de tráfego considerando: a topologia da rede (1), a posição das VNFs (4) e as SFCs especificadas (5). Essas informações são obtidas, respectivamente, por meio da rede propriamente dita, do componente de posicionamento e do usuário.

Figura 4.1: Visão geral da arquitetura proposta



Aspirando desenvolver o arcabouço proposto na Figura 4.1, observou-se necessário tomar decisões de projeto, principalmente no que refere-se às tecnologias que seriam utilizadas no ambiente de instanciação das VNFs, no controlador de rede e na aplicação de controle. Em seguida, será feita uma análise das opções empregadas em cada um dos componentes.

O ambiente de instanciação das VNFs foi desenvolvido pelo Grupo de Redes do Instituto de Informática da UFRGS e atua por meio de um *software* desenvolvido na linguagem Python que recebe como entrada a posição onde cada VNF deve ser instanciada. Primeiramente, é realizada a ativação de *containers Docker*, responsáveis por transformar *hosts* comuns em N-PoPs da rede. Em um segundo momento, são instanciados, nesses *containers*, *Click Routers*, encarregados de operar as funções de rede previamente desenvolvidas.

Analogamente ao ambiente de instanciação no plano NFV, deve-se atentar para a presença do controlador de rede no plano SDN. Existem diversas implementações de controladores de rede, como Floodlight, Ryu, Pox e Nox, para citar algumas. Optou-se pelo controlador Ryu, pois esse apresenta todos os módulos e ferramentas necessários para o desenvolvimento da aplicação de controle e por ter sido eleito o melhor controlador SDN em investigação conduzida por Khondoker *et. al*(KHONDOKER et al., 2014). Mais especificamente, as aplicações de controle desenvolvidas a partir do controlador Ryu foram desenvolvidas na linguagem Python.

Finalmente, a aplicação de controle é incumbida de executar toda a lógica para encadeamento e encaminhamento de fluxos na rede. Por estar estreitamente ligada ao controlador da rede, faz uso de sua API de comunicação e de suas diretrizes de desenvolvimento, responsáveis por fazer com a que a aplicação de controle seja orientada a eventos. Isso significa que, para cada evento disparado pelos *switches* ligados à rede, o procedimento responsável por tratar aquele evento será executado, podendo ou não resultar em uma resposta para os mesmos.

No escopo deste trabalho, existem quatro eventos de maior importância entre *switches* e a aplicação de controle, repassados pelo controlador, como pode-se visualizar na Figura 4.1. São eles:

- `PacketIn`: trata-se de um evento disparado por um *switch* que indica a entrada de um pacote por uma de suas interfaces. É a operação padrão do elemento de encaminhamento para um pacote que não possui nenhuma regra de encaminhamento associada. Em outras palavras, o *switch* não possui informações suficientes para

encaminhar o pacote, portanto, recorre à aplicação de controle para encontrar uma solução.

- `FlowMod`: é o evento mais habitualmente empregado pelo controlador no caso de pacotes sem regras, onde o *switch* insere uma regra em sua tabela de encaminhamento para as informações contidas naquele pacote. A aplicação de controle pode fazer a leitura dos campos presentes nos cabeçalhos do pacote e, após processar determinados campos, determina qual a ação que deve ser tomada pelo *switch* sempre que um pacote com aquele conjunto de informações for recebido. Também é possível utilizar esse evento para modificar regras já existentes nos *switches*.
- `FlowStatsRequest`: evento utilizado pela aplicação de controle com a finalidade de obter estatísticas dos dispositivos de encaminhamento da rede. Podem ser enviados em determinados intervalos de tempo para *switches* distintos e solicitar diferentes métricas, como quantidade de fluxos ou quantidade de *bytes* que trafegam por cada interface de um determinado *switch*.
- `FlowStatsReply`: é disparado quando um *switch* envia as informações solicitadas pela aplicação de controle no evento `FlowStatsRequest`. Ao receber as estatísticas de utilização, a aplicação pode reagir a elas e tratá-las da forma que considerar mais conveniente.

A atuação da aplicação de controle desenvolvida decorre, majoritariamente, por meio dos quatro eventos citados. Como a topologia da rede e a posição de cada VNF na mesma é conhecida, a aplicação deve tirar proveito dessas informações. Portanto, sempre que determinado *switch* encaminhar um pacote para a aplicação, essa consulta sua visão da rede, calcula a melhor rota que obedeça as SFCs especificadas e então encaminha a configuração de regras *match+action* para os *switches* que compõem a rota.

A visão da rede por parte da aplicação varia de acordo com a utilização dos enlaces da infraestrutura. Existe uma *thread* na aplicação de controle responsável por solicitar estatísticas (por meio do evento `FlowStatsRequest`) de todos os *switches* da rede de tempos em tempos (sendo esse valor configurável), permitindo a atualização de custos dos enlaces frequentemente.

4.2 Algoritmos Desenvolvidos

Partindo das estratégias propostas no Capítulo 3 e da arquitetura ilustrada na Seção 4.1, apresenta-se, agora, os algoritmos desenvolvidos para materializar as estratégias de engenharia de tráfego. Os algoritmos retratados a seguir são utilizados para encontrar uma sub-rota ideal para cada salto de uma (ou mais) SFCs. Em seguida, é preciso concatenar todas as sub-rotas resultantes para obter a rota completa de cada SFC. Destaca-se, ainda, que os custos atribuídos aos enlaces indicam o valor absoluto de utilização. Quanto à forma de apresentação dos algoritmos, esses são apresentados utilizando notação de pseudo-código.

4.2.1 Maior Disponibilidade

Com o objetivo de implementar a estratégia que seleciona a sub-rota com maior disponibilidade, foi desenvolvido o Algoritmo 1. Deve-se ressaltar que a estrutura `rede` está representada na forma de um vetor de listas. A estruturação foi realizada de forma que cada índice i do vetor represente o i-ésimo *switch* da rede e a lista associada a tal índice armazene os *switches* conectados ao mesmo. O primeiro passo do algoritmo consiste em efetuar uma cópia da topologia de rede (`rede`) em uma estrutura auxiliar (`copiaRede`), como indicado na linha 1, para fins de manipulação. Em seguida, inicia-se um laço `while` que será executado até que seja encontrada uma sub-rota vencedora. O laço principal inicia com dois laços *for* aninhados, presentes nas linhas 5 e 6, que são utilizados para realizar a varredura de todos os enlaces presentes em `copiaRede`. A função desse conjunto de *loops* é obter os enlaces mais custosos de `copiaRede`, que são armazenados na estrutura `enlacesMaisCustosos`. A variável `utilizacaoMaxima` pode não ter seu valor alterado ao longo da execução dos laços, o que indica que o maior custo de enlace presente em `copiaRede` é 0. Nesse caso, o condicional presente na linha 17 apresenta-se verdadeiro e é retornada a sub-rota mais curta dentre aquelas disponíveis em `copiaRede`. Caso contrário, considerando que existam enlaces de custo maior que 0, tais enlaces são removidos de `copiaRede`, como pode-se examinar no conjunto `else` presente na linha 20. Ao remover os enlaces de maior custo, é preciso verificar se ainda existem sub-rotas entre `origem` e `destino` em `copiaRede`. Essa verificação é realizada pelo `if` presente na linha 22 e, caso não exista uma sub-rota entre `origem` e `destino`, significa que foi encontrado o menor gargalo dentre todas as

sub-rotas candidatas entre os dois *hosts* especificados. Caso ainda existam sub-rotas candidatas, retoma-se a execução do laço principal a partir da remoção dos enlaces mais utilizados em `copiaRede`.

Algoritmo 1: Algoritmo que implementa a estratégia de maior disponibilidade para uma sub-rota

entrada: `origem`: refere-se ao *host* que inicia a conexão ou a uma VNF por onde a conexão trafega antes de alcançar seu destino
`destino`: refere-se ao *host* para o qual a conexão é destinada ou a uma instância de VNF candidata a processar o fluxo
saída : sub-rota vencedora para o par (`origem, destino`)

```
1  copiaRede[][] = rede[][];
2  while True do
3      utilizacaoMaxima = 0;
4      enlacesMaisCustosos.esvaziar();
5      for switch1 = 0 to tamanho(copiaRede) do
6          for switch2 = 0 to tamanho(copiaRede[switch1]) do
7              if custo(switch1, switch2) > utilizacaoMaxima then
8                  utilizacaoMaxima = custo(switch1, switch2);
9                  enlacesMaisCustosos.esvaziar();
10                 enlacesMaisCustosos.inserir(switch1, switch2);
11             end
12             if custo(copiaRede[i][j]) == utilizacaoMaxima then
13                 enlacesMaisCustosos.inserir(switch1, switch2);
14             end
15         end
16     end
17     if utilizacaoMaxima == 0 then
18         return caminho_mais_curto(copiaRede,origem,destino);
19     end
20     else
21         copiaRede.remover(enlacesMaisCustosos[]);
22         if
            tamanho(caminho_mais_curto(copiaRede,origem,destino))
            == 0 then
23             copiaRede.inserir(enlacesMaisCustosos[]);
24             return caminho_mais_curto(copiaRede,origem,destino);
25         end
26     end
27 end
```

4.2.2 Menor Média de Utilização

Já para a segunda proposição, a função desenvolvida opera de maneira distinta, pois é fundamentada no algoritmo de menores caminhos. Obtém-se todas as sub-rotas

candidatas através da função `obter_caminhos_mais_curtos`, presente na linha 4. Deve-se ressaltar que a estrutura `todasSubRotas` está representada na forma de um vetor de listas. A estruturação foi realizada de forma que cada índice i do vetor represente a i-ésima sub-rota candidata e a lista associada a tal índice armazene os *switches* que a compõem. Quanto ao laço `for`, que engloba as linhas entre 5 e 11, nele é realizada uma varredura das sub-rotas da estrutura `todasSubRotas`. O laço `for` aninhado, que envolve as linhas 7, 8 e 9, obtém a soma dos custos da i-ésima sub-rota, sendo essa armazenada na i-ésima posição do vetor `utilizacaoMedia`. Após a soma total ser calculada, obtém-se a média aritmética através da divisão dessa soma pela quantidade de saltos na sub-rota, valor também armazenado na i-ésima posição do vetor `utilizacaoMedia`. O último laço, iniciado na linha 12, visa a obter o identificador da sub-rota de menor utilização média, indicada pela variável `idSubRotaIdeal`. Após a obtenção de tal valor, retorna-se a sub-rota obtida a partir da aplicação da estratégia.

Algoritmo 2: Algoritmo que implementa a estratégia menor média de utilização para uma sub-rota

entrada: `origem`: refere-se ao *host* que inicia a conexão ou a uma VNF por onde a conexão trafega antes de alcançar seu destino
`destino`: refere-se ao *host* para o qual a conexão é destinada ou a uma instância de VNF candidata a processar o fluxo
saída : sub-rota vencedora para o par (`origem, destino`)

```
1  utilizacaoMedia[];
2  idSubRotaIdeal = 0;
3  menorUtilizacaoMedia = 'infinito';
4  todasSubRotas[][] =
     obter_caminhos_mais_curtos(rede,origem,destino);
5  for i = 0 to size(todasSubRotas) do
6  |   utilizacaoMedia[i] = 0;
7  |   for j = 0 to size(todasSubRotas[i]) do
8  |   |   utilizacaoMedia[i] += cost(todasSubRotas[i][j]);
9  |   end
10 |   utilizacaoMedia[i] = utilizacaoMedia[i]
   |    /size(todasSubRotas[i]);
11 end
12 for k = 0 to size(utilizacaoMedia) do
13 |   if utilizacaoMedia[k] < menorUtilizacaoMedia then
14 |   |   menorUtilizacaoMedia = utilizacaoMedia[k];
15 |   |   idSubRotaIdeal = k;
16 |   end
17 end
18 return todasSubRotas[idSubRotaIdeal];
```

4.2.3 Enlace Menos Utilizado

Por fim, o último algoritmo implementado segue o mesmo padrão apresentado pelo anterior. Inicialmente, as variáveis são inicializadas nas linhas 1 a 4, onde a estrutura `todasSubRotas` obtém o retorno da função `obter_caminhos_mais_curtos`. Deve-se ressaltar que a estrutura `todasSubRotas` está representada na forma de um vetor de listas. A estruturação foi realizada de forma que cada índice i do vetor represente a i-ésima sub-rota candidata e a lista associada a tal índice armazene os *switches* que a compõem. No primeiro `for`, com início na linha 5, são varridas as linhas da estrutura `todasSubRotas`, que contém as sub-rotas candidatas entre `origem` e `destino`. O segundo laço `for`, inicializado na linha 7, é responsável pelo armazenamento dos custos dos enlaces menos utilizados de cada sub-rota candidata no vetor `enlacesMenosUtilizados`. Por fim, para obter a sub-rota resultante, o último laço, que inicia na linha 13 e encerra na linha 18, atua de forma a identificar o enlace menos utilizado dentre todas as sub-rotas candidatas. A sub-rota vencedora é retornada na linha 19.

Algoritmo 3: Algoritmo que implementa a estratégia do enlace menos utilizado para uma sub-rota

entrada: `origem`: refere-se ao *host* que inicia a conexão ou a uma VNF por onde a conexão trafega antes de alcançar seu destino
`destino`: refere-se ao *host* para o qual a conexão é destinada ou a uma instância de VNF candidata a processar o fluxo
saída : sub-rota vencedora para o par (`origem, destino`)

```
1  enlacesMenosUtilizados[];
2  idSubRotaIdeal = 0;
3  menorUtilizacaoEnlaces = 'infinito';
4  todasSubRotas[][] = get_shortest_paths(rede,origem,destino);
5  for i = 0 to size(todasSubRotas) do
6  |  enlacesMenosUtilizados[i] = 'infinito';
7  |  for j = 0 to size(todasSubRotas[i]) do
8  |  |  if cost(todasSubRotas[i][j]) <
   |  |   enlacesMenosUtilizados[i] then
9  |  |  |  enlacesMenosUtilizados[i] =
   |  |  |   cost(todasSubRotas[i][j]);
10 |  |  end
11 |  end
12 end
13 for k = 0 to size(enlacesMenosUtilizados) do
14 |  if enlacesMenosUtilizados[k] < menorUtilizacaoEnlaces then
15 |  |  menorUtilizacaoEnlaces = enlacesMenosUtilizados[k];
16 |  |  idSubRotaIdeal = k;
17 |  end
18 end
19 return todasSubRotas[idSubRotaIdeal];
```

5 AVALIAÇÃO

Este capítulo apresenta informações relacionadas à avaliação do desempenho das estratégias propostas. Na Seção 5.1, realiza-se a descrição das tecnologias utilizadas e das métricas consideradas. Na sequência, a Seção 5.2 explana o cenário utilizado na avaliação. Finalmente, a Seção 5.3 explicita e discute os resultados obtidos.

5.1 Tecnologias Utilizadas e Métricas de Avaliação

Buscando relatar o aparato metodológico empregado na avaliação, descreve-se, a seguir, as tecnologias empregadas na prototipação de um ambiente propício para a avaliação, bem como uma justificativa para cada opção realizada. Na sequência, apresenta-se as métricas de avaliação escolhidas.

5.1.1 Tecnologias Utilizadas

Pela dificuldade de obter *switches* OpenFlow em grande escala, recorreu-se a simuladores capazes de instanciar a rede em ambiente emulado. Por se tratar do ambiente mais usualmente utilizado em situações de avaliação, o emulador *Mininet* (TEAM, 2012) foi considerado. Entretanto, esse apresenta dificuldades em instanciar *containers Docker* nos *hosts* virtuais da rede emulada. Com a finalidade de superar tal limitação, optou-se pelo ambiente *Containernet* (PEUSTER; KARL; ROSSEM, 2016), que se trata de uma variação do *Mininet* tradicional, diferenciando-se por apresentar suporte a *containers Docker* na infraestrutura.

Outro fator de grande importância nas proposições realizadas são as VNFs. Como já citado no Capítulo 4, recorreu-se ao ambiente de implantação de funções virtualizadas projetado pelo Grupo de Redes do Instituto de Informática da UFRGS. Entretanto, esse apresenta a limitação de poder instanciar apenas duas funções de rede diferentes (*firewall* e *load balancer*), pois encontra-se em fase de desenvolvimento. Tal fato não restringe o tamanho das SFCs, pois é possível instanciar múltiplos *firewalls* em diferentes *hops* da SFC. Isso se deve ao fato da função de encadeamento identificar cada função por onde fluxos devem trafegar por um número identificador, tornando possível a atribuição de um identificador diferente para cada instância da função.

Quanto à aplicação das estratégias desenvolvidas, apresenta-se a necessidade de gerar carga de trabalho consistente, tarefa que pode ser realizada por uma série de aplicações distintas. Desse modo, optou-se por utilizar a aplicação cliente-servidor com tráfego UDP da ferramenta Iperf (TIRUMALA et al., 2005). Essa escolha se deu por se tratar de uma aplicação amplamente usada no contexto de avaliação de redes.

A última ferramenta necessária para a avaliação do protótipo desenvolvido deve ser capaz de medir o atraso fim-a-fim entre dois *hosts* presentes na infraestrutura. Com esse requisito em vista, decidiu-se por utilizar a ferramenta Sockperf, que é capaz de medir o atraso de pacotes cursados entre quaisquer dois *hosts* com alta precisão (da ordem de nanossegundos).

5.1.2 Métricas

Engenharia de tráfego visa a utilizar a rede de forma mais equilibrada, sem que determinados enlaces sejam sobrecarregados e outros preteridos. Dessa forma, acredita-se que uma métrica essencial para avaliar as estratégias propostas é o *espalhamento* na utilização dos enlaces.

Outra métrica que precisa ser avaliada é a *latência* na comunicação entre dois *hosts*. Estratégias de engenharia de tráfego devem ser capazes de encontrar rotas alternativas, mas que não sejam onerosas em termos de latência.

Finalmente, deve-se considerar que este trabalho visa a desenvolver estratégias que possuam baixo custo de processamento e sejam escaláveis. Logo, a terceira e última métrica a ser avaliada é o *tempo médio de processamento* para a aplicação de controle obter uma rota entre dois *hosts*.

5.2 Cenário de Avaliação

A utilização de estratégias de engenharia de tráfego pode estar ligada a cenários e topologias específicas, como balanceamento de carga em servidores posicionados em uma topologia *fat-tree*, por exemplo. Da mesma forma, é preciso avaliar estratégias nos cenários para os quais elas foram propostas. Considerando que este trabalho visa a propor e desenvolver estratégias de engenharia de tráfego que podem ser aplicadas a quaisquer classes topológicas, é preciso gerar uma rede fundamentada em modelos aleatórios. A

forma mais conveniente de produzir essa geração referida procedeu-se por meio do gerador de topologias aleatórias aSHIIP (TOMASIK; WEISSER, 2010). A partir desse ambiente, tirou-se proveito do modelo de geração de grafos aleatórios Barabási-Albert (BARABÁSI; ALBERT, 1999) para conceber a topologia utilizada na avaliação das propostas desenvolvidas.

A prioridade das estratégias propostas é operar sobre infraestruturas que possuam VNFs distribuídas ao longo de seus *switches*. Assim sendo, é imprescindível escolher uma estratégia de posicionamento para essas funções na rede. Ambicionando atingir um maior desempenho, optou-se por posicionar as funções virtualizadas nos *switches* mais conectados da rede, de modo a torná-las mais acessíveis.

Organizou-se as especificações empregadas em cada uma das tecnologias utilizadas. O sistema de avaliação é utilizado da seguinte forma.

- Emulador *containernet* operando sobre uma topologia com 30 *switches*, 22 *hosts* reservados para comunicação e 8 *hosts* reservados para VNFs.
- Enlaces com taxa de transmissão igual a 100 Mbps, sem perda ou atraso acrescido.
- SFCs de tamanho 2, isto é, todo e qualquer tráfego proveniente de qualquer *host* deve passar por duas funções de rede antes de alcançar seu destino, independente de qual esse seja.
- 4 instâncias de cada VNF, permitindo que, caso uma instância esteja muito sobrecarregada, seja possível direcionar o tráfego a outra.
- Fluxos UDP unidirecionais de vazão igual a 20Mbps, onde todos os *hosts* origem e destinos são definidos aleatoriamente e estão sujeitos a apenas um fluxo. A cada 5 segundos, um novo par de *hosts* inicia um fluxo.
- Medições de latência em diferentes momentos da simulação realizadas pelos mesmos dois *hosts*. Esses são definidos previamente ao início da simulação e são diferenciados dos outros *hosts* por não gerarem carga de trabalho, apenas efetuar medições.
- Contabilização da utilização dos enlaces feita pela aplicação de controle ao fim da simulação.
- Medição do tempo de processamento de cada algoritmo desenvolvido pela aplicação de controle.

5.3 Resultados

Ao avaliar as estratégias de engenharia de tráfego propostas, é preciso considerar o estágio inicial em que se encontram os esforços de pesquisa para esse problema no contexto de NFV. Atentando para uma reconhecida escassez de trabalhos, estabeleceu-se, como estratégia a ser comparada, a de menores caminhos (usualmente denominada *Shortest Path*) (MOY, 1997). Entretanto, a implementação utilizada para a estratégia de menores caminhos é adaptada para redes com funções virtualizadas.

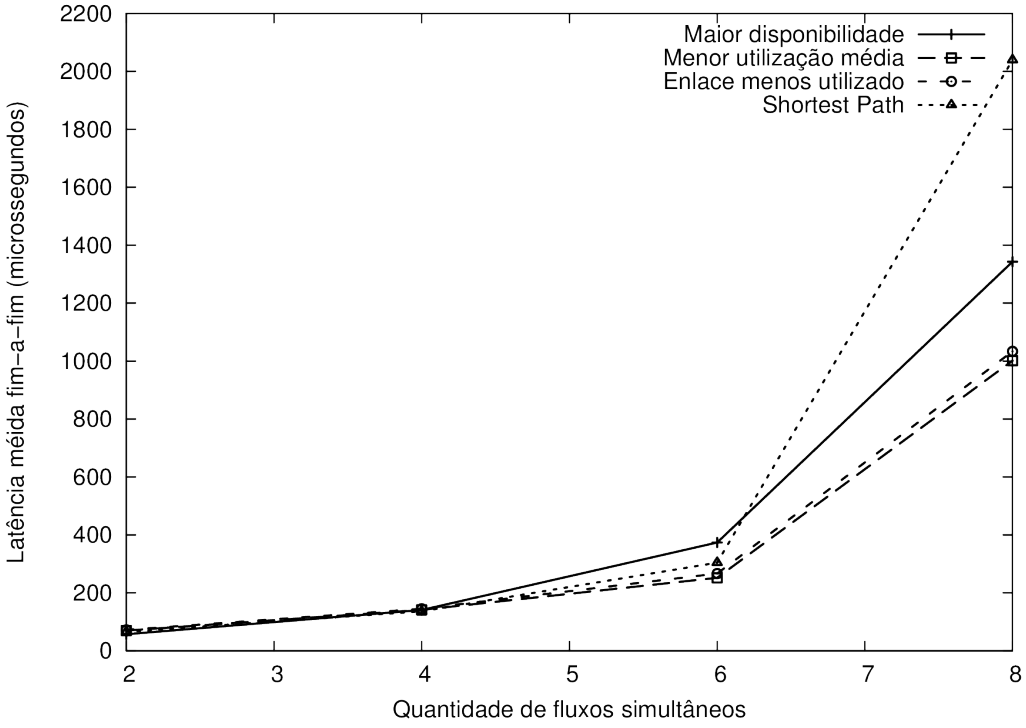
Quanto ao número de repetições, realizou-se 10 medições para obtenção da medida de atraso médio fim-a-fim e para a medida de espalhamento médio em momentos específicos da execução. Em relação ao tempo médio de processamento, baseou-se no tempo médio para encontrar rotas entre 300 *hosts* para cada tamanho de topologia apresentado.

5.3.1 Atraso Fim-a-Fim

Tendo como base o cenário de avaliação referido na Seção 5.2, a Figura 5.1 ilustra os resultados obtidos. Pode-se observar que todas as estratégias apresentam comportamento semelhante no início da simulação (2 conexões simultâneas). Isso se deve ao fato de que, com poucas conexões, há uma tendência a optar pelas rotas mais curtas, até mesmo para a estratégia de maior disponibilidade. Na segunda medição, onde há 4 conexões simultâneas, todas as estratégias ainda apresentam latência semelhante por ainda não haver congestionamento nos enlaces. Ao observar o momento em que existem 6 conexões simultâneas na rede, repara-se que a estratégia de caminho mais curto tradicional apresenta uma defasagem média de 12,5% em relação às estratégias que possuem uma política de desempate (seja ela pela menor média de utilização ou pelo enlace menos utilizado). Esse fato ocorre devido a essa estratégia desconsiderar a utilização da rede, enquanto as outras estratégias apresentam ciência e ajustam suas rotas para aquelas que estiverem menos congestionadas. Quanto à estratégia que fundamenta-se na maior disponibilidade, nota-se um aumento de latência em relação às restantes. A ocorrência desse fato se dá devido à opção por rotas mais longas, que apresentam maior atraso de transmissão e menor atraso de filas, já que essas apresentam menor congestionamento. Por fim, no momento em que existem 8 conexões simultâneas da rede, verifica-se que a estratégia de menores caminhos tradicional não toma nenhuma medida contra o congestionamento

da rede. Essa estratégia apresenta uma defasagem média de 34,2% em relação a aquela que visa a maior disponibilidade. Quanto às estratégias de menor média de utilização e enlace menos congestionado, observa-se um resultado bastante semelhante entre elas, apresentando um desempenho 23% superior à estratégia que garante a maior vazão.

Figura 5.1: Resultados obtidos a partir da medição de atraso fim-a-fim

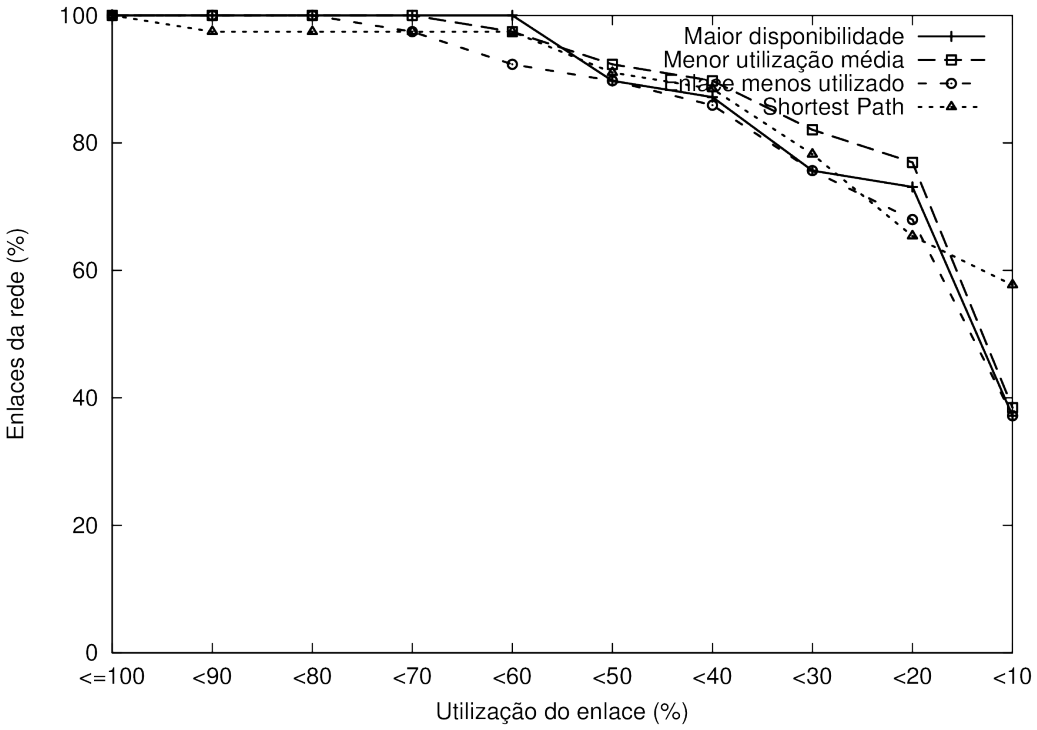


5.3.2 Espalhamento de Utilização dos Enlaces

Para avaliar o espalhamento na utilização da rede, avaliou-se dois cenários específicos. O primeiro, representado pela Figura 5.2, ilustra os resultados obtidos a partir da medição realizada na rede em um contexto sem contenção de recursos. O gráfico refere-se ao percentual dos enlaces que estão presentes em cada faixa de utilização dos mesmos. Inicialmente, pode-se observar que a única curva que indica que há enlaces com mais de 80% de utilização é a referente aos menores caminhos. Isso ocorre pois essa estratégia ignora as utilizações dos enlaces ao determinar uma nova rota. Analisando os enlaces com utilização entre 60% e 20%, nota-se que não existe muita distinção entre as estratégias nessa situação. Constata-se que mais de 80% dos enlaces da rede teve ocupação média inferior a 40% de sua capacidade para todas as estratégias aplicadas. Verificando os enlaces com utilização menor que 10%, percebe-se o quanto a estratégia de menores

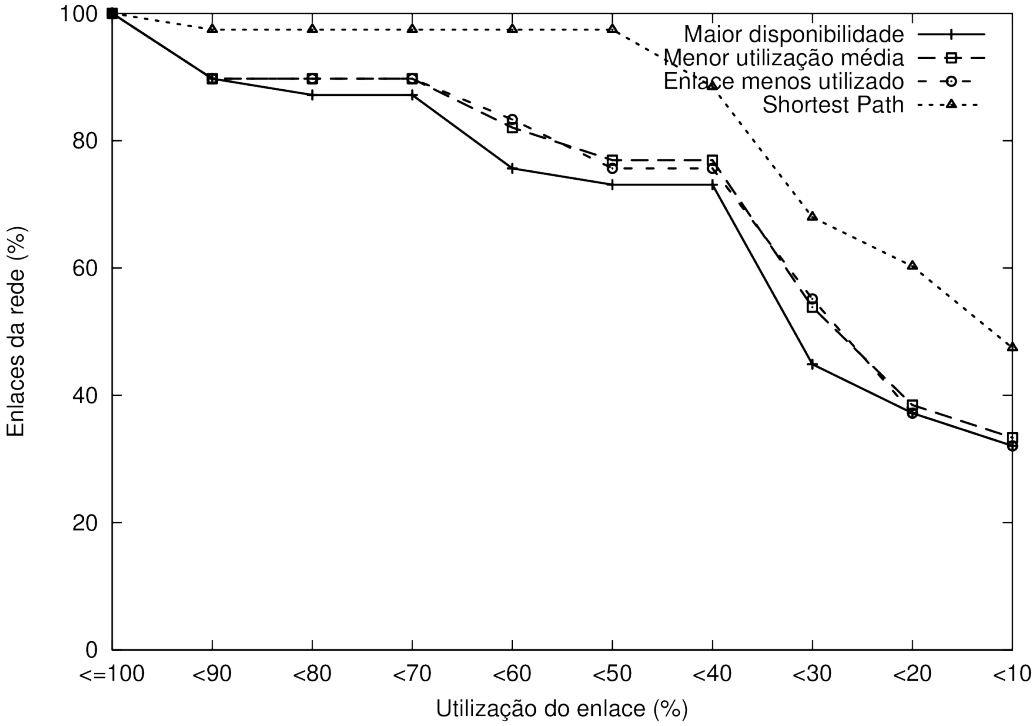
caminhos peca em equilibrar carga pela rede, pois mais de 57% dos enlaces estão praticamente inutilizados. Por outro lado, para todas as estratégias propostas, o percentual de enlaces com ocupação inferior a 10% gira em torno de 38%.

Figura 5.2: Espalhamento observado na rede em cenário sem congestionamento



Quanto ao segundo cenário avaliado, ilustrado na Figura 5.3, optou-se por avaliar o espalhamento no cenário em que a rede encontra-se congestionada. Existe uma clara diferença entre a Figura 5.2 e a Figura 5.3, pois verifica-se uma utilização mais equilibrada em todas as estratégias propostas. Examinando a estratégia de menores caminhos, constata-se que 2,5% dos enlaces possuem utilização maior que 90% e, então, o segundo conjunto de enlaces mais utilizados tira proveito de apenas 40% a 50% de sua capacidade. Ao estabelecer uma comparação com a estratégia de maior disponibilidade, 47,4% dos enlaces da rede que emprega a estratégia de menores caminhos possuem utilização menor que 10% de sua capacidade. No entanto, 55,2% dos enlaces sujeitos à estratégia de maior disponibilidade utilizam de 30% a 100% de sua capacidade. Ao tratar das estratégias menor média de utilização e enlaces congestionados, baseadas na estratégia de menores caminhos, é perceptível que ambas apresentam um comportamento semelhante. Ao analisar os enlaces menos utilizados para cada estratégia empregada, repara-se que mais de 47% dos enlaces presentes na rede com a estratégia de menores caminhos possuem utilização mínima. Já as estratégias propostas convergem para valores entre 32% e 33,5%, apresentando uma melhor distribuição de carga em situações de estresse da infraestrutura.

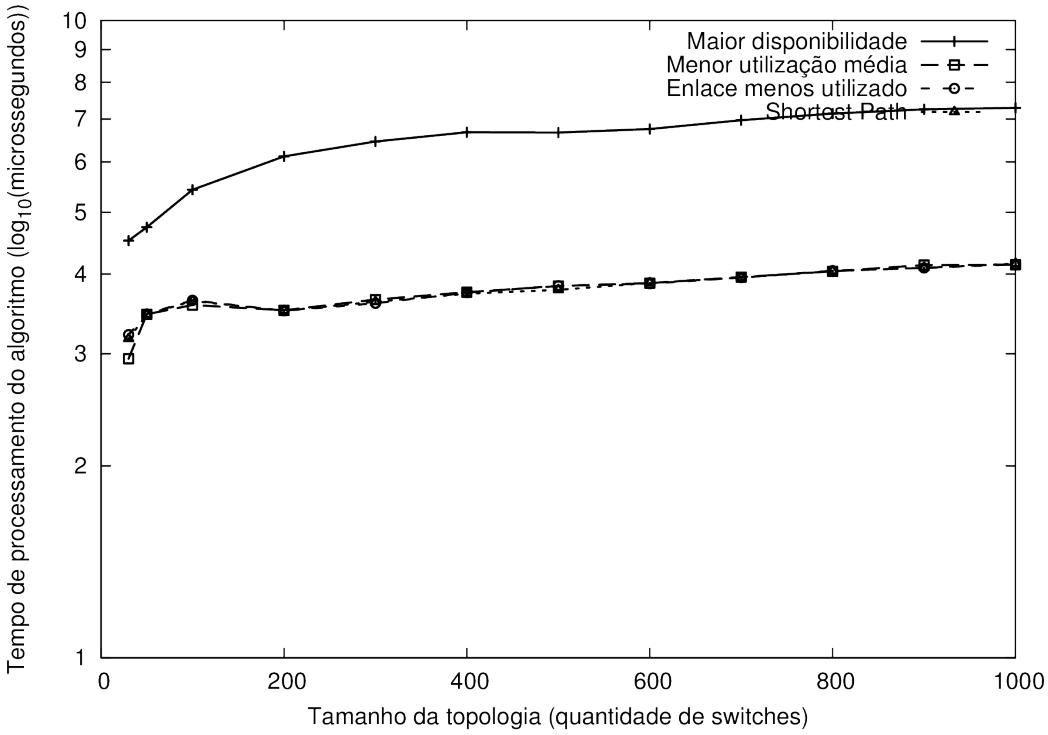
Figura 5.3: Espalhamento observado na rede em cenário com congestionamento



5.3.3 Tempo de Médio de Processamento

Esta métrica visa a avaliar a escalabilidade de cada estratégia para grandes topologias. A Figura 5.4 apresenta o tempo médio de processamento para obter uma rota entre dois *hosts* em topologias de diferentes tamanhos. A estratégia que obtém sub-rotas com a maior disponibilidade possui tempo médio de processamento bastante elevado, pois precisa considerar o estado global da rede e não apenas as sub-rotas com menor número de saltos. Em redes pequenas, de até 100 *switches*, o tempo médio para obter uma rota não ultrapassa a marca de 300 milissegundos. Porém, em topologias com 200 *switches*, já podem ser observados tempos da ordem de segundos. Para redes ainda maiores, com mais de 800 *switches*, são observados tempos médios da ordem de dezenas de segundos. Em relação à estratégia de menores caminhos e às estratégias propostas a partir dessa, constata-se tempos consideravelmente menores. Destacam-se os tempos médios de processamento para topologias de 800 *switches*, onde obteve-se medições da ordem de dezenas de milissegundos. Deve-se atentar que as estratégias propostas baseadas no *Shortest Path* não apresentam tempo de processamento consideravelmente maior em comparação à estratégia de menores caminhos sem desempate.

Figura 5.4: Tempo médio de processamento para cada estratégia em topologias de diferentes escalas



6 CONCLUSÃO

Nesse trabalho, realizou-se um estudo dos paradigmas de *Network Function Virtualization* e de *Software-Defined Networking*, seguido de um levantamento de estratégias de engenharia de tráfego capazes de operar em conjunto com as tecnologias citadas. Em seguida, alicerçado nos conhecimentos adquiridos, propôs-se três estratégias de engenharia de tráfego com capacidade de atuar em infraestruturas com funções virtualizadas de rede. Tais estratégias visam a promover uma abordagem inicial ao problema apresentado, haja vista que existem poucas propostas capazes de atuar nesse contexto. Posteriormente, desenvolveu-se um ambiente capaz empregar as estratégias de engenharia de tráfego considerando funções virtualizadas em redes SDN/OpenFlow. Por fim, avaliou-se as estratégias desenvolvidas.

Dentre os objetivos desse trabalho, almejava-se propor estratégias de engenharia de tráfego capazes de operar em qualquer rede e sobre qualquer aplicação. Dentre as métricas avaliadas como fundamentais para estratégias eficientes, estava a escalabilidade. Com base nos resultados obtidos a partir da avaliação, constatou-se que a estratégia baseada na opção pelas sub-rotas de maior disponibilidade apresenta dificuldade no que refere-se ao tempo de processamento em grandes infraestruturas. Dessa forma, acredita-se que tal proposta é incapaz de ser utilizada em redes complexas, limitando-se o uso para redes menores, onde apresenta bons resultados quanto ao atraso fim-a-fim e ao espalhamento na utilização dos enlaces.

Quanto às outras estratégias propostas, as quais eram fundamentadas na estratégia de menores caminhos, constatou-se ótima escalabilidade, apresentando tempo de processamento inferior a 100 milissegundos para topologias com 1000 *switches*. O mesmo pode ser dito a respeito dos resultados obtidos em relação ao atraso fim-a-fim, em que tais estratégias obtiveram os menores atrasos totais. No que refere-se à eficácia dessas estratégias em garantir alto espalhamento na utilização dos enlaces, os resultados obtidos foram inferiores aos da estratégia de maior disponibilidade. Entretanto, isso não invalida os resultados satisfatórios obtidos na métrica citada.

Por outro lado, verificou-se que a estratégia de menores caminhos por si só é incapaz de garantir a operação ideal da rede em situações de congestionamento. Confirmou-se a expectativa de que, com pequenas variações na forma de implementar o algoritmo, obtém-se uma utilização mais balanceada e eficiente da rede. As estratégias de enlace menos utilizado e menor média de utilização demonstram tal fato sem comprometer o

desempenho do algoritmo em termos de tempo de processamento.

Visando ao desenvolvimento de trabalhos futuros, almeja-se desenvolver estratégias que obtenham a melhor rota a partir de métricas fim-a-fim, em contraste à obtenção de sub-rotas ideais. Também aspira-se avaliar o comportamento de tais estratégias em comparação com aquelas apresentadas nessa investigação.

REFERÊNCIAS

ADAMI, D. et al. Towards an sdn network control application for differentiated traffic routing. In: IEEE. **Communications (ICC), 2015 IEEE International Conference on**. [S.l.], 2015. p. 5827–5832.

AWDUCHE, D. et al. **Overview and Principles of Internet Traffic Engineering**. IETF, 2002. RFC 3272 (Informational). (Request for Comments, 3272). Updated by RFC 5462. Available from Internet: <http://www.ietf.org/rfc/rfc3272.txt>.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999.

BREMLER-BARR, A.; HARCHOL, Y.; HAY, D. Openbox: Enabling innovation in middlebox applications. In: ACM. **Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization**. [S.l.], 2015. p. 67–72.

CRAIG, A. et al. Load balancing for multicast traffic in sdn using real-time link cost modification. In: IEEE. **2015 IEEE International Conference on Communications (ICC)**. [S.l.], 2015. p. 5789–5795.

DWARAKI, A.; WOLF, T. Adaptive service-chain routing for virtual network functions in software-defined networks. In: ACM. **Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization**. [S.l.], 2016. p. 32–37.

GEMBER-JACOBSON, A. et al. Opennf: Enabling innovation in network function control. **ACM SIGCOMM Computer Communication Review**, ACM, v. 44, n. 4, p. 163–174, 2015.

HALPERN, J.; PIGNATARO, C. **Service Function Chaining (SFC) Architecture**. IETF, 2015. RFC 7665 (Informational). (Request for Comments, 7665). Available from Internet: <http://www.ietf.org/rfc/rfc7665.txt>.

HANDIGOL, N. et al. Plug-n-serve: Load-balancing web traffic using openflow. **ACM Sigcomm Demo**, Barcelona, Spain, v. 4, n. 5, p. 6, 2009.

HOPPS, C. **Analysis of an Equal-Cost Multi-Path Algorithm**. IETF, 2000. RFC 2992 (Informational). (Request for Comments, 2992). Available from Internet: <http://www.ietf.org/rfc/rfc2992.txt>.

KAMIYAMA, N. et al. Flow aggregation for traffic engineering. In: IEEE. **2014 IEEE Global Communications Conference**. [S.l.], 2014. p. 1936–1941.

KHONDOKER, R. et al. Feature-based comparison and selection of software defined networking (sdn) controllers. In: IEEE. **Computer Applications and Information Systems (WCCAIS), 2014 World Congress on**. [S.l.], 2014. p. 1–7.

LEE, D.; HONG, P.; LI, J. Rpa-ra: A resource preference aware routing algorithm in software defined network. In: IEEE. **IEEE International Conference on Computer Communications (INFOCOM)**. [S.l.], 2015. p. 6.

LI, D.; SHANG, Y.; CHEN, C. Software defined green data center network with exclusive routing. In: IEEE. **IEEE INFOCOM 2014-IEEE Conference on Computer Communications**. [S.l.], 2014. p. 1743–1751.

LI, Y.; PAN, D. Openflow based load balancing for fat-tree networks with multipath support. In: **Proc. 12th IEEE International Conference on Communications (ICC'13), Budapest, Hungary**. [S.l.: s.n.], 2013. p. 1–5.

LUIZELLI, M. C. et al. Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In: **2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)**. [S.l.: s.n.], 2015. p. 98–106. ISSN 1573-0077.

MARTINS, J. et al. Clickos and the art of network function virtualization. In: USENIX ASSOCIATION. **Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation**. [S.l.], 2014. p. 459–473.

MCKEOWN et al. Openflow: enabling innovation in campus networks. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 2, p. 69 – 74, 2008. ISSN 0146-4833.

MCKEOWN, N. Software-defined networking. **INFOCOM keynote talk**, v. 17, n. 2, p. 30–32, 2009.

MOY, J. Ospf version 2. 1997.

Network Functions Industry Specification Group. Network function virtualisation (nfv): An introduction, benefits, enablers, challenges and call for action. In: **SDN and OpenFlow World Congress**. [S.l.: s.n.], 2012. p. 1–16.

PEUSTER, M.; KARL, H.; ROSSEM, S. van. Medicine: Rapid prototyping of production-ready network services in multi-pop environments. **arXiv preprint arXiv:1606.05995**, 2016.

RODRIGUES, C. P. et al. Avaliaç ao de balanceamento de carga web em redes definidas por software. 2015.

ROSEN, E.; VISWANATHAN, A.; CALLON, R. **Multiprotocol Label Switching Architecture**. IETF, 2001. RFC 3031 (Proposed Standard). (Request for Comments, 3031). Updated by RFCs 6178, 6790. Available from Internet: <http://www.ietf.org/rfc/rfc3031.txt>.

SEZER, S. et al. Are we ready for sdn? implementation challenges for software-defined networks. **IEEE Communications Magazine**, IEEE, v. 51, n. 7, p. 36–43, 2013.

SU, Z. et al. Cheetahflow: Towards low latency software-defined network. In: IEEE. **Communications (ICC), 2014 IEEE International Conference on**. [S.l.], 2014. p. 3076–3081.

TEAM, M. **Mininet: An instant virtual network on your laptop (or other PC)**. 2012.

TIRUMALA, A. et al. Iperf: The tcp/udp bandwidth measurement tool. **htt p://dast. nlanr. net/Projects**, 2005.

TOMASIK, J.; WEISSER, M.-A. ashiip: autonomous generator of random internet-like topologies with inter-domain hierarchy. In: IEEE. **2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems**. [S.l.], 2010. p. 388–390.

TOOTOONCHIAN, A. et al. On controller performance in software-defined networks. In: **Presented as part of the 2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services**. [S.l.: s.n.], 2012.

APÊNDICE A — TRABALHO DE GRADUAÇÃO I

Algoritmos de Balanceamento de Carga de Funções Virtualizadas de Rede

Gabriel do Amaral Almeida[1], **Weverton Luis da Costa Cordeiro**[1], **Luciano Paschoal Gaspary**[1]

[1]Instituto 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 novel computer 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 will investigate existing load balancing strategies, and propose the implementation of traffic engineering strategies for flow sterring, designed with the goal of linking the flows over the network functions. The proposed strategies will be implemented and evaluated during Undergraduate Thesis 2.*

Resumo. *Virtualização de Funções de Rede (Network Function Virtualization, NFV) é um novo 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. As estratégias propostas serão implementadas e avaliadas, no Trabalho de Graduação 2.*

1. Introdução

Virtualização de Funções de Rede (*Network Function Virtualization*, NFV) [Network Functions Industry Specification Group 2012] é um novo paradigma que

visa a conferir maior flexibilidade à gerência e 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.

O potencial de se utilizar NFV aumenta consideravelmente ao combiná-lo com Redes Definidas por *Software* (*Software Defined Networking*, SDN) [McKeown 2009]. 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, instaladas 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 (por meio 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 reune 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 fácilmente abordado com o uso de SDN, a virtualização de funções de rede torna mais complexo o problema de 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 presente artigo apresenta um levantamento das abordagens tradicionais de balanceamento de carga, engenharia de tráfego e de posicionamento de VNFs, visando à compreensão do problema e analisando possíveis formas de solução. Para o Trabalho de Graduação 2, serão desenvolvidas e comparadas estratégias para o problema de posicionamento de VNFs, bem como para o problema de balanceamento de carga em SDN com VNFs, respeitando os encadeamentos descritos nas SFCs a serem implantadas.

O restante deste 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 implementação projetados para o problema de engenharia de tráfego no âmbito de NFV. Na Seção 4 descreve-se a metodologia a ser utilizada no desenvolvimento do trabalho. Na Seção 5 apresenta-se o cronograma de atividades previsto para o andamento da pesquisa. Por fim, na Seção 6 são apresentadas as considerações finais.

2. Trabalhos Relacionados

Conforme mencionado anteriormente, um dos elementos-chave deste trabalho são as estratégias de balanceamento de carga. Em função disso, na Seção 2.1 serão discutidos pormenores das investigações que representam o estado da arte em estratégias de balanceamento de carga e engenharia de trafego. Vale ressaltar, entretanto, que pouco tem sido feito no sentido de desenvolver soluções que englobem redes com suporte à NFV. Em função disso, os trabalhos listados focam essencialmente em estratégias de balanceamento de carga e engenharia de tráfego no contexto de SDN. Na Seção 2.2 será discutido uma das investigações que oferecem alocação de VNFs, a qual deverá ser um dos alicerces sobre os quais o Trabalho de Graduação 2 será baseado.

2.1. Engenharia de Tráfego/Balanceamento de Carga

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 (*p.ex streaming*, aplicações de baixa latência, *web servers*, etc.), a técnica específica de balanceamento de carga (*p.ex* número de *hops*, 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 serão 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* e *elephant flows*, com o objetivo de isolar fluxos de maior ocupação dos enlaces (*elephant flows*) e agregar fluxos de baixa ocupação (*mice flows*). 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] propõe um mecanismo baseado na escolha de rotas com o menor número de hops. Para desempate, Lee *et al.* fundamenta-se em um cálculo de autovetores 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 autovetor 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] desenvolveu um controlador que opera sobre uma aplicação de engenharia de tráfego para aplicações diferenciadas em SDN. A estratégia 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 mencionada acima, 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.

2.2. Posicionamento de Funções Virtualizadas de Rede

O tópico de posicionamento (e encadeamento) de funções virtualizadas de rede possui diversos esforços de pesquisa publicados [Mehraghdam et al. 2014, Bari et al. 2015, Cohen et al. 2015, Moens and De Turck 2014, Addis et al. 2015]. Entretanto, tendo em vista que o presente trabalho visa a avaliação de algoritmos de balanceamento de carga aplicados à engenharia de tráfego em um contexto NFV, é necessário definir uma forma de realizar posicionamento e implantação de VNFs.

Como essa tarefa está, por ora, fora do escopo (isto é, será tratada no Trabalho de Graduação 2), optou-se por uma implementação de posicionamento de VNFs desenvolvida por Luizelli *et al.* [Luizelli et al. 2015]. Isso porque essa investigação foi desenvolvida no contexto do Grupo de Redes do Instituto de Informática. A implementação citada é sustentada em um modelo de *Integer Linear Programming* (ILP), responsável pelo posicionamento e encadeamento de funções de rede de forma quase-ótima.

3. Estratégias Propostas

Este trabalho tem por objetivo o estudo, a proposição e a avaliação de estratégias de balanceamento de carga aplicadas no contexto de engenharia de tráfego em uma infraestrutura SDN com suporte a VNFs. Para apoiar esse objetivo, visa-se desenvolver um arcabouço de *software* que permita, a partir de um *front-end* único, instanciar diferentes topologias, posicionar funções em locais distintos e avaliar as diferentes estratégias propostas e implementadas.

Uma vez que não foi encontrado um estudo que unificasse todas as tecnologias referidas, concluiu-se a inevitabilidade de unificar uma solução existente de posicionamento e encadeamento de VNFs com soluções existentes de engenharia de tráfego. Tal deverá ser feito através de algoritmos de balanceamento de carga, buscando ainda comparar a eficácia e eficiência das estratégias propostas.

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.

Tendo esses preceitos como base, apresenta-se três algoritmos de balanceamento de carga com o objetivo de avaliá-los ao longo do desenvolvimento do Trabalho de Graduação 2. Para ilustrar esses algoritmos, lança-se mão de três exemplos, que serão descritos nas seções subsequentes, embasados na SFC representada na figura 2 e na topologia ilustrada na figura 1, composta círculos representando *hosts*, quadrados representando N-PoPs e losangos representando comutadores. É importante observar a existência de enlaces conectando os elementos da topologia e seus respectivos pesos.

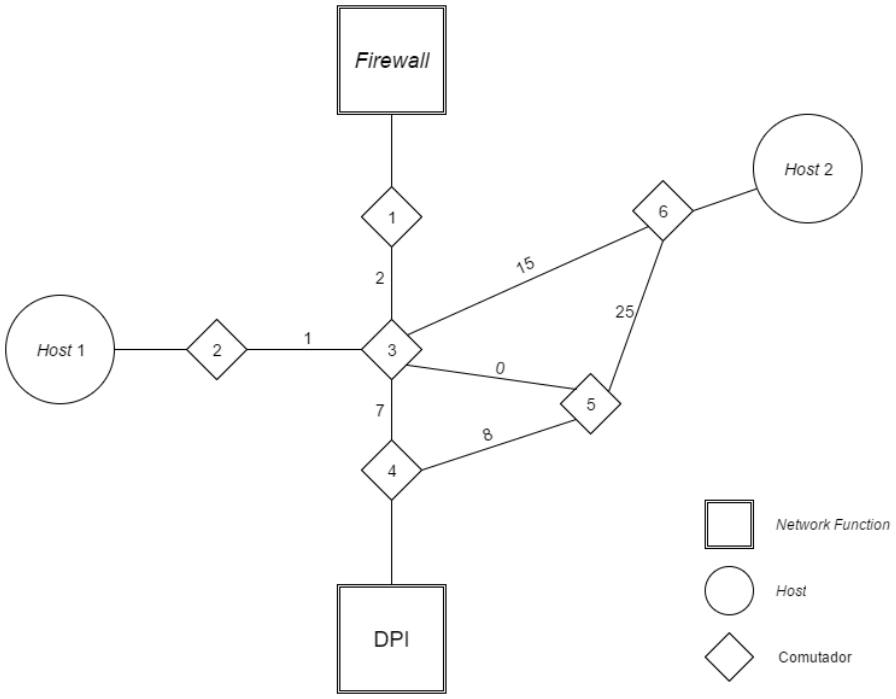


Figura 1. Exemplo de topologia com dois *hosts* e dois N-PoPs

Seguindo a noção de que uma rota é composta por $N+1$ sub-rotas, onde N indica o número de funções de rede presentes em uma SFC, é preciso calcular 3 rotas para a SFC referida. 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 inspetor

de pacotes (*Deep Packet Inspection*, DPI). Por fim, a terceira rota conduz o tráfego do DPI até o *host* destino.

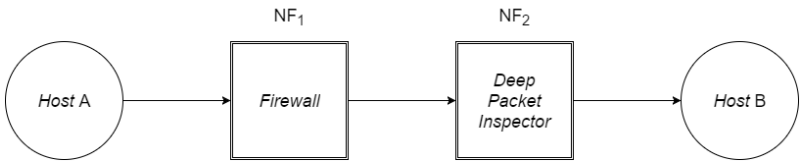


Figura 2. Exemplo de SFC que conduz o tráfego através de um *Firewall* e de um *Deep Packet Inspector* antes de encaminhá-lo ao seu destino

3.1. Evasão de Enlace Mais Congestionado

Esta estratégia 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.

1. Para a primeira sub-rota (*Host* 1→*firewall*), tem-se apenas um caminho a seguir, através dos comutadores $(2, 3, 1)$.
2. Observando a segunda sub-rota (*firewall* →DPI), existem três rotas 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 o algoritmo 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 $(1, 3, 4)$.
3. Por fim, a última sub-rota (DPI→*Host* 2) possui quatro alternativas: $(4, 5, 6)$, $(4, 5, 3, 6)$, $(4, 3, 5, 6)$ 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 $(4, 3, 6)$.

3.2. Contagem de Saltos, Opção por Enlace de Maior Banda Disponível

Com o objetivo de utilizar a rede de forma equilibrada, esta estratégia 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.

1. A primeira sub-rota (*Host* 1→*firewall*) não precisa ser calculada, pois existe apenas uma possível: $(2, 3, 1)$.
2. Com o objetivo de encontrar a segunda sub-rota (*firewall*→DPI), 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 $(1, 3, 4)$.

3. Por fim, as quatro sub-rotas (DPI→*Host* 2) aspirantes são $(4, 5, 6)$, $(4, 5, 3, 6)$, $(4, 3, 5, 6)$ e $(4, 3, 6)$. O algoritmo verifica 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 $(4, 3, 6)$.

3.3. Contagem de Saltos, Opção por Média de Utilização Mais Baixa

De forma semelhante à estratégia anterior, propõe-se uma estratégia baseada no número mais baixo de saltos entre origem e destino. 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 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.

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

1. Analogamente às estratégias anteriores, a primeira sub-rota (*Host* 1→*firewall*) possui apenas uma rota possível: $(2, 3, 1)$.
2. Devido ao fato desta estratégia utilizar como primeiro critério o menor número de saltos, tem-se apenas um caminho candidato para a segunda sub-rota (*firewall*→DPI): $(1, 3, 4)$.
3. Finalmente, existem quatro caminhos entre os comutadores 4 e 6 (DPI→*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-rota $(4, 3, 6)$ possui o menor peso médio, sendo escolhida como a rota percorrida pelos dados destinados ao *host* 2.

4. Metodologia

A metodologia a ser aplicada no desenvolvimento do Trabalho de Graduação 2 é apresentada a seguir.

1. Estudo: etapa que consiste na melhor compreensão do problema através de análise bibliográfica do estado da arte relativo a técnicas de engenharia de tráfego, funcionamento de SDN e NFV, além de algoritmos de balanceamento de carga.
2. Ambientação: etapa composta pela adaptação aos ambientes utilizados na etapa seguinte. Realiza-se a instalação, aclimatação e compreensão das ferramentas que apoiam a implementação, como *Mininet*, *Open vSwitch* e *iptables*.
3. Projeto: etapa em que são tomadas decisões relativas aos algoritmos a ser implementados, bem como seu resultado esperado.
4. Implementação: etapa de desenvolvimento da aplicação propriamente dita, que engloba o controlador *OpenFlow*, o software que executa sobre as VNFs e a ferramenta de *deployment* das funções.

5. Avaliação: a etapa final destina-se aos testes e à validação da solução proposta em ambiente simulado, à análise e comparação dos resultados obtidos em relação aos resultados estimados na etapa de projeto e à avaliação experimental do protótipo em ambiente real.

No presente momento, as etapas 1 e 2 foram concluídas e a etapa 3 encontra-se em andamento.

5. Cronograma

Alicerçado na metodologia apresentada na Seção 4, o cronograma de desenvolvimento deste Trabalho de Graduação compreende as seguintes tarefas:

1. Conclusão do projeto dos algoritmos.
2. Desenvolvimento do programa de controle e do ambiente de *deployment* de VNFs.
3. Integração das tecnologias desenvolvidas em um único ambiente parametrizável.
4. Avaliação experimental em ambiente simulado.
5. Avaliação experimental em ambiente real.
6. Redação do Trabalho de Graduação 2.
7. Apresentação do Trabalho de Graduação 2.

Tarefa	2016						
	Jun	Jul	Ago	Set	Out	Nov	Dez
1	**X**	**X**					
2		**X**	**X**	**X**			
3				**X**	**X**		
4				**X**	**X**	**X**	
5						**X**	
6			**X**	**X**	**X**	**X**	
7							**X**

Tabela 1. Cronograma de atividades

6. Considerações Finais

Este trabalho apresentou uma análise sucinta sobre desafios relacionados a técnicas engenharia de tráfego aplicadas ao contexto de Virtualização de Funções de Rede (NFV). Foram destacados ainda as principais estratégias de balanceamento de carga em SDN, por exemplo as estratégia de posicionamento, instanciação e encadeamento de N-PoPs dentro de uma infraestrutura de rede.

Além disso, foi apresentada a proposta do Trabalho de Graduação 2. O plano prevê o desenvolvimento de um ambiente SDN capaz de realizar engenharia de tráfego, e de uma ferramenta capaz de calcular o posicionamento quase-ótimo de VNFs. Por fim, objetiva-se a comparação das estratégias desenvolvidas e a verificação da escalabilidade da solução proposta.

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.

Addis, B., Belabed, D., Bouet, M., and Secci, S. (2015). Virtual network functions placement and routing optimization. In *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, pages 171–177. 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.

Bari, M. F., Chowdhury, S. R., Ahmed, R., and Boutaba, R. (2015). On orchestrating virtual network functions in nfv. *CoRR abs/1503.06377*.

Cohen, R., Lewin-Eytan, L., Naor, J. S., and Raz, D. (2015). Near optimal placement of virtual network functions. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 1346–1354. IEEE.

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-n-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).

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.

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 IFIP/IEEE International Symposium on*, pages 98–106. IEEE.

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.

Mehraghdam, S., Keller, M., and Karl, H. (2014). Specifying and placing chains of virtual network functions. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 7–13. IEEE.

Moens, H. and De Turck, F. (2014). Vnf-p: A model for efficient placement of virtualized network functions. In *Network and Service Management (CNSM), 2014 10th International Conference on*, pages 418–423. IEEE.

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.

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.