UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Gabriel Marchesan

UMA ANÁLISE COMPARATIVA ENTRE PARADIGMAS DE VIRTUALIZAÇÃO DE REDES

Santa Maria, RS
2018

Gabriel Marchesan

UMA ANÁLISE COMPARATIVA ENTRE PARADIGMAS DE VIRTUALIZAÇÃO DE REDES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC), da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação.**

Orientador: Prof. Dr. Carlos Raniery Paula dos Santos

Co-orientadora: Prof[a]. Dr[a]. Roseclea Duarte Medina

Santa Maria, RS

2018

Ficha catalográfica elaborada através do Programa de Geração Automática
da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Marchesan, Gabriel
Uma Análise Comparativa entre Paradigmas de Virtualização de Redes / Gabriel Marchesan.- 2018.
114 p.; 30 cm

Orientador: Carlos Raniery Paula dos Santos
Coorientadora: Roseclea Duarte Medina
Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Ciência da Computação, RS, 2018

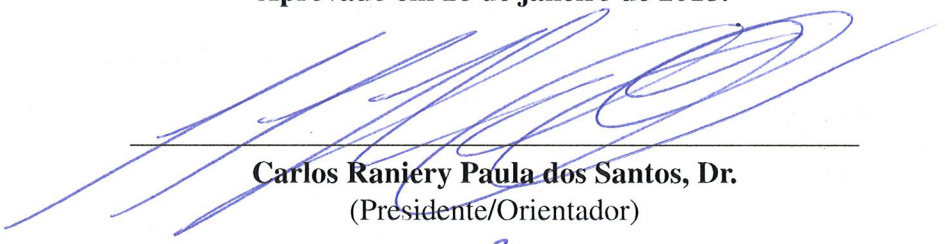
1. SDN 2. NFV 3. Virtualização 4. Funções de Rede I. Raniery Paula dos Santos, Carlos II. Duarte Medina, Roseclea III. Título.

Gabriel Marchesan

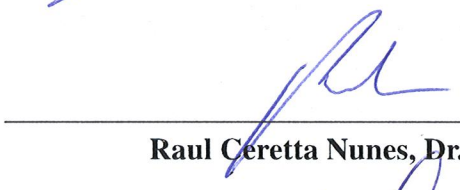
UMA ANÁLISE COMPARATIVA ENTRE PARADIGMAS DE VIRTUALIZAÇÃO DE REDES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PGCC), da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação.**

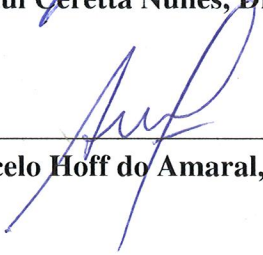
Aprovado em 26 de janeiro de 2018:



Carlos Raniery Paula dos Santos, Dr.
(Presidente/Orientador)



Raul Ceretta Nunes, Dr. (UFSM)



Érico Marcelo Hoff do Amaral, Dr. (UNIPAMPA)

Santa Maria, RS

2018

DEDICATÓRIA

Primeiramente dedico este trabalho à Deus pela saúde e sabedoria ao decorrer deste período, possibilitando assim que eu pudesse concluir mais uma etapa importante em minha vida acadêmica. Também dedico este trabalho aos meus pais Astrogildo Ari Marchesan e Sustene Tezsele Marchesan, e a minha irmã Fabríola Marchesan.

AGRADECIMENTOS

Agradeço primeiramente à Deus por ter me guiado durante todos esses anos de estudos em minha trajetória acadêmica, pela saúde e sabedoria em possibilitar a conclusão deste trabalho.

À minha família pelo apoio, ajuda e principalmente aporte financeiro para que eu pudesse realizar o Mestrado em Ciência da Computação mesmo sem ter ganhado a bolsa de estudos.

Ao meu orientador Profº. Drº. Carlos Raniery Paula dos Santos pelas suas orientações, revisões, conversas, reuniões e discussões acerca da realização deste trabalho.

À minha co-orientadora Profª. Drª. Roseclea Duarte Medina pelas suas sugestões, revisões, conversas e também pela sua atenção comigo desde a graduação.

Aos professores Drº. Raul Ceretta Nunes e Drº. Érico Marcelo Hoff do Amaral por aceitarem fazer parte da banca avaliadora deste trabalho.

Ao prof. Drº. Ricardo Tombesi Macedo por aceitar o convite como avaliador suplente deste trabalho.

Ao colega de mestrado e amigo Marcelo da Luz Colomé pelas nossas discussões e anseios compartilhados no decorrer do Mestrado. Além disso, pelas nossas revisões, considerações e consequentemente contribuições no trabalho um do outro. Também pelo bom humor e pelos aprendizados adquiridos no período que estive trabalhando contigo na Unidade de Tecnologia da Informação do Centro de Tecnologia (UTICT).

Ao amigo Diego Couto de Carvalho pela amizade construída, pelos conhecimentos aprendidos e compartilhados no período em que estive trabalhando contigo na UTICT.

Aos colegas de mestrado Anderson, Nilton, Silvio, Leonardo, Tavares e Vinícius do Grupo de Redes e Computação Aplicada (GRECA), aos amigos e demais parentes que de alguma forma também contribuíram para que eu pudesse chegar até aqui e concluir mais este trabalho.

“Se você não puder se destacar pelo talento, vença pelo esforço.”

(DAVE WEINBAUM)

RESUMO

UMA ANÁLISE COMPARATIVA ENTRE PARADIGMAS DE VIRTUALIZAÇÃO DE REDES

AUTOR: GABRIEL MARCHESAN
ORIENTADOR: CARLOS RANIERY PAULA DOS SANTOS
CO-ORIENTADORA: ROSECLEA DUARTE MEDINA

Novas tecnologias na área de redes de computadores vêm ganhando uma maior atenção da academia e da indústria, a exemplo de Redes Definidas por *Software* (SDN) e Virtualização das Funções de Rede (NFV). Enquanto SDN surgiu no intuito de tornar as redes mais programáveis e flexíveis, o paradigma NFV surgiu com objetivo principal de tornar funções de redes mais escaláveis. Ainda, com as funções de rede virtualizadas e executadas em dispositivos genéricos, NFV possibilita a redução expressiva dos custos de capital (CAPEX) e das operações (OPEX). Nessa perspectiva, embora funções de rede passaram a ser implementadas em ambos os paradigmas, a literatura carece de uma análise mais profunda de quais funções de rede são mais adequadas de se executar em determinado paradigma. Além disso, em quais casos, cenários e em quais situações seria mais vantajoso e adequado utilizar um paradigma SDN ou NFV para implementar determinada função de rede. Neste sentido, este trabalho apresenta um estudo sobre virtualização de redes onde observa-se qual paradigma apresenta um melhor suporte às necessidades de uma determinada função de rede. Para atingir este objetivo, implementou-se e testou-se as funções de *firewall, switching*, roteamento e *dhcp-server* em ambas as tecnologias. Ressalta-se que a presente pesquisa diferencia-se dos demais trabalhos disponíveis na literatura, já que tem como principal contribuição uma análise mais significativa dos aspectos qualitativos e quantitativos das funções de redes executando nas tecnologias SDN e NFV. Analisando-se os resultados obtidos com a realização deste trabalho, percebe-se que o paradigma NFV é mais adequado para a implementação de funções da categoria de aplicação. Para funções de interoperabilidade, SDN contempla maior embasamento teórico/prático e por isso entende-se que seja a mais adequada. No que tange a implementação de funções das categorias de otimização e de monitoramento/controle, nota-se que os paradigmas SDN e NFV podem ser substituídos um pelo outro, já que obtiveram resultados parecidos. Já para as funções de proteção, constata-se que NFV obteve melhores resultados, portanto sendo mais vantajosa e efetiva a implementação de funções desta categoria neste paradigma. Ainda, os resultados da pesquisa evidenciam que não existe uma única tecnologia exclusiva para ser implementada determinada função de rede. Tendo isto em vista, entende-se que ambos os paradigmas podem ser aplicados na implementação de funções de rede pertencentes a diferentes categorias.

Palavras-chave: SDN. NFV. Virtualização. Funções de Rede.

ABSTRACT

A STUDY ABOUT NETWORK VIRTUALIZATION

AUTHOR: GABRIEL MARCHESAN
ADVISOR: CARLOS RANIERY PAULA DOS SANTOS
COADVISOR: ROSECLEA DUARTE MEDINA

New technologies in the area of computer networks have been gaining greater attention from academia and industry, such as Software Defined Networking (SDN) and Network Function Virtualization (NFV). While SDN arose to make networks more programmable and flexible, the NFV paradigm has emerged with the primary goal of making network functions more scalable. Also, with virtualized network functions performed on generic devices, NFV enables expressive reduction of capital costs (CAPEX) and operations (OPEX). In this perspective, although network functions are implemented in both paradigms, the literature lacks in a deeper analysis of which network functions are more adequate to be executed in a given paradigm. In addition, in which scenarios and situations would it be more advantageous and appropriate to use an SDN or NFV paradigms to implement a particular network function. In this sense, this work presents a study on networks virtualization where it is observed which paradigm presents a better support to the needs of a given network function. To achieve this goal, the firewall, switching, routing and dhcp-server functions were implemented and tested in both technologies. There should be emphasized that the present research differs from the other works available in the literature, since its main contribution is a deeper analysis of the qualitative and quantitative aspects of the functions of networks running in SDN and NFV paradigms. Analyzing the results obtained with the accomplishment of this work, it is noticed that the NFV paradigm is more suitable for the implementation of functions of the application category. For interoperability functions, SDN encompasses a more theoretical / practical basis, and therefore is understood to be the most adequate in this scenario. Regarding the implementation of functions that belong to the optimisation and monitoring / control categories, it is noted that the SDN and NFV paradigms can be substituted for each other, since they obtained similar results. As for the protection functions, it is verified that NFV obtained better results, therefore it is more advantageous and effective the implementation of functions of this category in this paradigm. Furthermore, the research results show that there is no single technology to implement a particular network function. From this perspective, it is understood that both paradigms can be applied to implement network functions belonging to different categories.

Keywords: SDN. NFV. Virtualization. Network Functions.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BRAS	Broadband Remote Access Server
BSS	Business Support Systems
CAPEX	CAPital EXpenses
DDoS	Denial Distributed of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPDK	Data Plane Development Kit
EMS	Element Management System
ETSI	European Telecommunications Standards Institute
FG	Forwarding Graph
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
ISG	Industry Specification Group
MAC	Media Access Control
MANO	Management and Orchestration
NAT	Network Address Translation
NF	Network Function
NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
OPEX	OPerational EXpenses
OSS	Operational Support Systems
PoP	Point of Presence
RTT	Round Trip Time
SDN	Software Defined Networking
SLA	Service Level Agreement
SSL	Secure Socket Layer
VM	Virtual Machine
VNF	Virtual Network Function

SUMÁRIO

1 INTRODUÇÃO

Com a intensa utilização de novos recursos computacionais e um aumento na demanda por aplicações cada vez mais sofisticadas, as redes de computadores atuais já não conseguem mais atender satisfatoriamente as necessidades dos usuários. Este cenário deve-se a muitos fatores, tais como: limitação dos equipamentos proprietários que fazem uso de diversos protocolos, grande demanda das tabelas de roteamento, aumento exponencial do tráfego, etc (GUEDES et al., 2012). Assim, devido a todos esses fatores, o gerenciamento das redes atuais torna-se uma tarefa operacional complexa (HERRERA; VEGA, 2016).

Ainda, no decorrer dos últimos anos, equipamentos de rede dedicados que implementam diversas funções de rede, *e.g.*, *firewall, switches*, roteadores, balanceadores de carga, *Network Address Translation* (NAT), tornaram-se amplamente utilizados em ambientes corporativos para cumprir as demandas das modernas aplicações e serviços.

Estes equipamentos de redes, também conhecidos por *middleboxes*, têm desempenhado um papel essencial para a consolidação das redes de computadores, além disso, os mesmos vêm sendo bastante utilizados nas redes domésticas, onde um mesmo equipamento pode integrar diversas funcionalidades. Por exemplo, usuários residenciais, para acessar a Internet, utilizam um único equipamento que agrega várias funcionalidades, tais como: *modem, switch*, roteamento através do NAT e rede *wireless* (HERRERA; VEGA, 2016) (MATIAS et al., 2015).

Entretanto, apesar de sua ampla disseminação, o uso de equipamentos dedicados apresenta diversas desvantagens (MATIAS et al., 2015). Além de apresentarem um alto custo de aquisição, necessidade e disponibilidade de técnicos especializados, gastos relativos a manutenções preventiva e corretiva dos mesmos, o uso destes equipamentos dificulta a adequação das redes na realização de tarefas que necessitam de maior flexibilidade para satisfazer as novas demandas e exigências dos usuários (LIU et al., 2016).

Nesse contexto, percebe-se que a atual arquitetura da Internet atingiu uma saturação que a deixou pouco versátil. A complexidade para adoção, implementação e testes de novos protocolos e funções no núcleo da Internet define o processo conhecido como ossificação da rede (CHOWDHURY; BOUTABA, 2009). Reconhecendo este significativo problema, pesquisadores e especialistas na indústria e na academia, estão frequentemente buscando métodos e soluções novas e mais eficientes para planejar, implantar e manter as redes de computadores operacionais (MATIAS et al., 2015) (BATALLE et al., 2013).

Nesse sentido, a fim de mitigar estes problemas, algumas abordagens foram propostas, onde aquelas baseadas em técnicas de virtualização tiveram maior destaque (MATIAS et al., 2015). Com a virtualização foi possível que diversas aplicações executassem simultaneamente sobre o mesmo substrato físico compartilhado, *i.e.*, mesmo *hardware*, assim contribuindo para uma melhor utilização dos recursos computacionais. Dessa forma, roteadores e *switches* podem ser substituídos por seus correspondentes virtualizados implementados em *software*, sem que tais funções virtuais precisem executar em *hardware* especializado.

Nesse contexto, inicialmente surge um novo conceito de arquitetura de rede, chamado de Redes Definidas por *Software* (*Software Defined Networking*-SDN). Este novo paradigma sugere o desacoplamento do plano de dados, que é responsável pelo encaminhamento dos pacotes, do plano de controle, o qual possui a inteligência e visão global da rede e encontra-se logicamente centralizado em uma entidade externa denominada de controlador da rede.

Com a dinamicidade e adaptabilidade proporcionada pela arquitetura SDN, é possível introduzir programabilidade às redes de computadores. Além do mais, com este paradigma o gerenciamento torna-se fácil, maximizando a resolução de possíveis problemas. Dessa forma, o desempenho das aplicações é otimizado (LIU et al., 2016) (MCKEOWN et al., 2008).

Por trás da arquitetura SDN está o protocolo *OpenFlow*, o qual permite estabelecer a comunicação entre os elementos encaminhadores (*switches OpenFlow*) e o controlador da rede. Ademais, SDN permite que os administradores de rede possam definir fluxos de dados e determinar quais caminhos esses fluxos devem percorrer na rede, assim, criando as regras de encaminhamento do fluxo de dados (MCKEOWN et al., 2008).

Em virtude das inúmeras vantagens obtidas com o uso de SDN (*e.g.* flexibilidade, simplicidade e eficiência), diversos trabalhos publicados nos últimos anos como (TRAN; AHN, 2016), (BATALLE et al., 2013), (RAZA et al., 2016), (KIM; KIM; LEE, 2015), adotaram tecnologias baseadas neste paradigma para realizar determinadas funções de rede (*e.g.*, *firewall*, *proxy*, roteamento e NAT).

Entretanto, inicialmente com o surgimento desta nova arquitetura, acreditou-se que diversas aplicações de rede pudessem ser implementadas usando apenas SDN (MIJUMBI et al., 2016). Entretanto, como esse paradigma apresenta um processamento oneroso devido a constante comunicação dos comutadores com o controlador de rede, percebeu-se a necessidade de criar novas tecnologias que atendessem de forma mais adequada o desenvolvimento destas aplicações (MIJUMBI et al., 2016).

Nessa perspectiva, um novo paradigma complementar à SDN surgiu com o objetivo de separar o plano de dados do substrato físico, tecnologia conhecida por Virtualização das Funções de Rede (*Network Function Virtualization*-NFV) [ETSI, 2012] (CHIOSI et al., 2012). Nesse paradigma, funções de rede tradicionais são virtualizadas e executadas em equipamentos genéricos independentes de fabricantes (*i.e., commodity hardware*) através do uso de *hipervisores* (e.g., XEN, KVM, VirtualBox).

Além do mais, tais equipamentos são capazes de desempenhar as mesmas tarefas realizadas pelos dispositivos dedicados (CHIOSI et al., 2012). Diversos benefícios são esperados com a utilização de NFV, como a eliminação de dispositivos físicos dedicados, contribuindo para a redução nos custos de capital (CAPEX-*CAPital EXpenses*) e das operações (OPEX-*OPerational EXpenses*). Além disso, soluções baseadas em *software* possibilitam maior elasticidade, flexibilidade, escalabilidade e provisionamento da rede, permitindo que novas funções e serviços sejam rapidamente adotados.

Ainda é importante ressaltar que SDN e NFV são dois paradigmas independentes. Porém, podem ser complementares e mutuamente benéficos, pois os mesmos podem trabalhar de forma integrada. Desta forma os recursos da rede são melhor aproveitados, onde SDN geralmente ficará responsável pelo encaminhamento dos pacotes e NFV pela inteligência e pelo processamento dos pacotes como um todo. Nesse sentido, a integração destes dois paradigmas pode ser muito relevante, principalmente em tarefas de gerenciamento e orquestração de redes, permitindo um controle mais refinado para a infraestrutura computacional como um todo (LIU et al., 2016).

Também observa-se que SDN tornou-se gradualmente parte integrante da arquitetura NFV (LIU et al., 2016). Como já citado anteriormente, apesar de poderem trabalhar de forma integrada, a união de SDN e NFV resulta na utilização de duas camadas de *software*, sobrecarregando a função de rede e impactando negativamente no desempenho da mesma (WOOD et al., 2015). Além disso, essa integração possui alguns desafios para a sua efetiva adoção, tais como o gerenciamento consistente e a interoperabilidade com as infraestruturas legadas. Ademais, ambos os paradigmas podem ser utilizados para executar um mesmo conjunto de funções.

Nesta perspectiva, ainda não tem-se o conhecimento concreto de quais funções de rede são mais adequadas de se executar em determinado paradigma, isto é, em quais casos e em quais situações seria mais vantajoso utilizar uma arquitetura SDN ou NFV para implementar determinada função de rede. Deste modo, este trabalho tem como objetivo principal analisar

funções de rede executadas nos paradigmas SDN e NFV com o propósito de identificar em quais cenários e situações cada tecnologia é mais vantajosa de ser utilizada.

Em relação ao estado das funções, estas podem ser do tipo *Stateless* ou *Stateful*. As funções do tipo *Stateful* mantém o estado de conexão dos pacotes. Já as do tipo *Stateless* não armazenam o estado de conexão dos pacotes anteriores. Também destaca-se que neste tipo de função cada requisição é tratada como sendo uma nova sessão, onde nenhuma informação extraída dos pacotes é retida pelo remetente ou receptor. Ainda, funções desta natureza lidam com todas as entradas de pacotes de forma individual, onde todo pacote recebido é considerado novo, independente do fluxo de informação em questão, seja de uma requisição ou resposta (GUPTA; KAUR; KAUR, 2016) (KABLAN et al., 2015) (MANFREDI; CROVELLA; KUROSE, 2011).

Nas funções do tipo *Stateful*, também ressalta-se que em alguns casos a conexão é mantida aberta mesmo que dois sistemas finais não estejam transmitindo informações durante um determinado intervalo de tempo, ou seja, o estado de conexão entre os sistemas é mantido. Portando, funções desta natureza armazenam o estado de conexão do usuário durante toda a execução da aplicação (GUPTA; KAUR; KAUR, 2016) (KABLAN et al., 2015) (MANFREDI; CROVELLA; KUROSE, 2011). Dessa forma, justifica-se a escolha de funções do tipo *Stateless* para serem analisadas em ambos os paradigmas na presente pesquisa, pois funções do tipo *Stateful* já sabe-se, através da literatura, que são mais adequadas para serem executadas no paradigma NFV.

Os paradigmas serão observados e analisados sob parâmetros, métricas quantitativas e qualitativas a fim de auxiliar a escolha do paradigma mais adequado à implementação de determinada função de rede. Para atingir este objetivo, foram escolhidas especificadamente as funções *firewall, switching*, roteamento e *dhcp-server* para serem implementadas em ambos os paradigmas realizando-se então a partir disso análises e discussões dos resultados alcançados. Foram escolhidas estas funções pois pertencem a diferentes categorias e exercem finalidades distintas nas redes de computadores. Ainda, atuam em diferentes camadas do modelo de referência OSI. Assim, pode-se observar melhor o comportamento e tendências de funções com diferentes características e objetivos de atuação. Além disso, estas funções são frequentemente exploradas pela academia para avaliações e testes na área de redes.

1.1 MOTIVAÇÃO

A partir dos novos paradigmas de redes de computadores baseados em *software*, como SDN e NFV, gradativamente torna-se possível a utilização destes novos paradigmas na pesquisa e no desenvolvimento de novas soluções de redes. Nesse contexto, muitos trabalhos em (BATALLE et al., 2013), (OLAYA; BERNAL; MEJIA, 2016), (WANG; XU; LIU, 2017) podem ser encontrados na literatura com o objetivo de utilizar SDN e NFV para a implementação de diversas funções de rede. Estes trabalhos normalmente utilizam estes paradigmas de forma separada ou de forma integrada.

Entretanto, tais trabalhos não realizam uma análise profunda das funções de rede propostas e implementadas. Observa-se que há maior preocupação apenas com o desempenho alcançado, isto é, contemplando aspectos quantitativos. Tendo isto em vista, estes trabalhos não levam em consideração comportamentos apresentados, características qualitativas e discussões mais refinadas sobre os resultados obtidos na avaliação dos testes. Isso acontece pois geralmente os trabalhos possuem um foco maior na implementação de determinada arquitetura e na realização de testes.

1.2 PROBLEMA DE PESQUISA

Como por meio de um conhecimento científico, auxiliar o processo de escolha em quais situações ou cenários é mais adequado utilizar o paradigma SDN ou NFV para implementação e execução de funções de rede.

1.3 OBJETIVOS

Considerando que funções de rede podem ser implementadas em ambos os paradigmas, o objetivo principal deste trabalho é identificar quando e em quais situações um paradigma é mais adequado e vantajoso do que o outro para a implementação de determinada função de rede. A fim de contemplar este objetivo, será realizada a avaliação de funções de rede executando em ambos os paradigmas. As funções de rede avaliadas serão: *firewall, switching*, roteamento e *dhcp server*.

Ainda, para realizar esta avaliação levou-se em consideração aspectos quantitativos e qualitativos. Em relação aos aspectos quantitativos, o desempenho de tais funções de rede

será observado através dos resultados das métricas: atraso (*delay*), *jitter*, vazão (*throughput*) e perda de pacotes. Já em relação aos aspectos qualitativos, analisou-se parâmetros relacionados à segurança, restrições de implementação/programabilidade, gerenciamento, gestão dinâmica dos recursos, disponibilidade/resiliência e desempenho sob um ponto de vista mais qualitativo.

Assim, a fim de contemplar o objetivo geral, definiu-se alguns objetivos específicos:

a) Revisar e mostrar os principais conceitos dos paradigmas SDN e NFV;

b) Identificar categorias de funções de rede que possam ser implementadas nos dois paradigmas;

c) Propor uma taxonomia de funções de rede de forma a possibilitar a análise de quais dessas categorias são mais adequadas para serem utilizadas/implementadas em um determinado paradigma;

d) Pesquisar e identificar alguns parâmetros qualitativos e quantitativos entre os dois paradigmas;

e) Analisar o comportamento das funções de redes através da compreensão das métricas obtidas na execução dos testes.

Ainda, para que seja possível a realização desta pesquisa e para que os objetivos possam ser alcançados, foi necessário a revisão da literatura através da pesquisa nas principais bases de dados da área. Nesse sentido, foi explorado os temas e as principais características envolvendo os paradigmas de Redes Definidas por *Software* e Virtualização das Funções de Rede. Ainda, foram analisados os principais trabalhos correlatos com a proposta de pesquisa em questão.

Além disso, a partir dos conceitos explorados de SDN e NFV, identificou-se categorias de funções de rede do tipo *Stateless* que pudessem ser implementadas nesses dois paradigmas. A partir disso, passou-se a implementação da proposta e a realização dos testes a fim de minimizar o problema de pesquisa em questão e analisar os principais resultados alcançados pelo presente trabalho.

O texto desta dissertação está organizado da seguinte forma. O Capítulo 2 apresenta as arquiteturas e principais características dos paradigmas SDN e NFV, bem como o relacionamento existente entre os mesmos. No Capítulo 3 é proposta uma taxonomia de funções de rede e posteriormente discorre-se alguns exemplos práticos de utilização destas funções nos paradigmas SDN e NFV. O Capítulo 4 apresenta as dimensões qualitativas envolvendo ambos os paradigmas. O Capítulo 5 apresenta a análise das categorias e dimensões, onde criou-se uma tabela com níveis de aplicabilidade em SDN e NFV entre as categorias de funções de rede pelas

dimensões elencadas, após isso é realizada uma discussão geral dessa tabela. No Capítulo 6 apresenta-se os materiais e métodos utilizados para a realização deste trabalho, destacando-se os procedimentos de implementação e a elaboração dos cenários de testes. No Capítulo 7 tem-se a avaliação de desempenho das funções de rede executando em SDN e NFV. Além disso, trata da análise dos resultados, apresentando a tabela atualizada com a dimensão de desempenho, bem como discorre as considerações parciais alcançadas pelo trabalho em questão. Por fim, no Capítulo 8 apresenta-se as principais conclusões sobre a realização da presente pesquisa e também dá-se algumas sugestões de trabalhos futuros.

2 BACKGROUND

Dentre os novos paradigmas de redes propostos nos últimos anos, os paradigmas SDN e NFV foram os que tiveram maior destaque, onde estes utilizam os benefícios da programabilidade de rede e de técnicas de virtualização para a execução de funções de rede em *hardware* genéricos (MCKEOWN et al., 2008) (CHIOSI et al., 2012). Nesse contexto, esta seção tem o objetivo de mostrar as principais características e a arquitetura dos paradigmas SDN e NFV. Além disso, também contempla o relacionamento entre os paradigmas SDN e NFV, os trabalhos relacionados e uma discussão dos mesmos.

2.1 NOVOS PARADIGMAS DE REDE

Nos últimos anos, o aumento da conectividade à Internet revolucionou a maneira como a sociedade vive, trabalha, conduz diversos negócios, obtêm entretenimento, etc. Nesse contexto, a Internet passou a ser considerada parte da infraestrutura crítica de todos estes ambientes que fazem parte da sociedade (NUNES et al., 2014).

Ademais, com a significativa demanda por conectividade de boa qualidade e com o intenso surgimento de (*e.g.*, novas aplicações e recursos computacionais) que estão presentes no cotidiano de muitas pessoas, as redes atuais já não atendem de forma satisfatória a todos estes usuários (NUNES et al., 2014). Isso ocorre devido a uma série de fatores, tais como, limitação dos equipamentos que fazem uso de *hardware* e diversos protocolos proprietários, demanda das tabelas de roteamento, aumento significativo do tráfego de rede, etc.

Ainda, para que a rede funcione de acordo com as políticas desejadas, os operadores de rede precisam configurar cada dispositivo individualmente usando políticas de baixo nível através de comandos específicos de cada fabricante do equipamento (KREUTZ et al., 2015). Nesse sentido além da complexidade da configuração nos ambientes de rede atuais, estes ainda necessitam suportar falhas e adaptar-se às mudanças de carga do tráfego de rede.

Porém, mecanismos rápidos e confiáveis de reconfiguração e resposta a estas falhas são praticamente inexistentes e/ou muito onerosos para se implantar nas redes atuais. Além disso, o forte acoplamento dos planos de controle e de dados integrados no mesmo equipamento, utilizado pelas redes tradicionais, dificulta o desenvolvimento e a implantação de novos recursos de rede, como por exemplo, novos algoritmos de roteamento e protocolos de redes (KREUTZ

et al., 2015). Dessa forma, cumprir as políticas definidas é uma tarefa complexa e o gerenciamento das redes atuais tornou-se uma tarefa operacional complexa (HERRERA; VEGA, 2016).

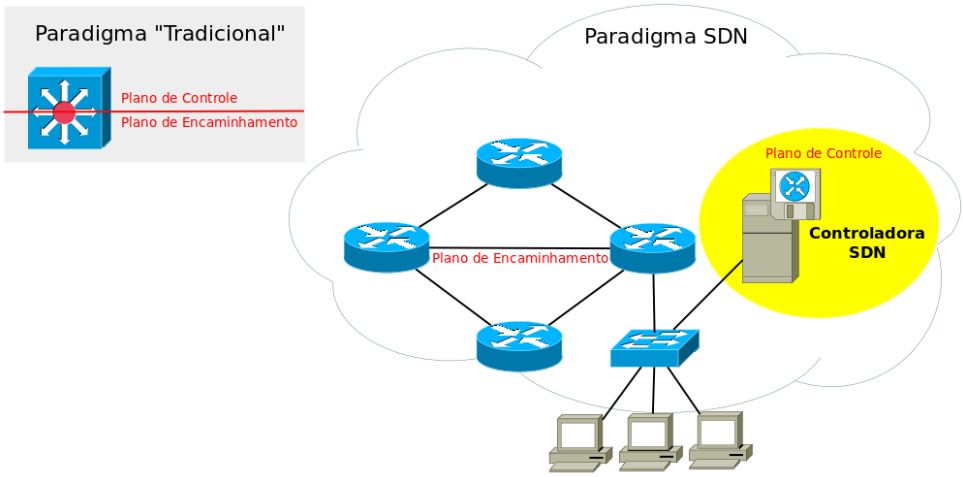
Em contraste a essa realidade, os pesquisadores de redes e a comunidade científica começaram a desenvolver propostas para a criação de novos paradigmas de redes de computadores, ou seja, novas arquiteturas de implementação no núcleo da rede (MCKEOWN et al., 2008). Dentre estas novas propostas, primeiramente surge o paradigma de Redes Definidas por *Software* (*Software Defined Networking*-SDN) (MCKEOWN et al., 2008). Além disso, mais recentemente é proposto o paradigma de Virtualização das Funções de Rede (*Network Function Virtualization*-NFV) (CHIOSI et al., 2012). É importante ressaltar que estes dois paradigmas são independentes um do outro, porém complementares, e cada um será melhor detalhado a seguir.

2.1.1 SDN

O paradigma de Redes Definidas por *Software* (MCKEOWN et al., 2008), vem adquirindo grande atenção e espaço da academia e das grandes indústrias da área de redes de computadores. O mesmo foi proposto para facilitar a evolução e a inovação da rede, permitindo a rápida implantação de novos serviços e protocolos (NUNES et al., 2014). Este novo paradigma apresentado na Figura 1, sugere o desacoplamento do plano de dados do plano de controle. Nesta arquitetura, o plano de dados é responsável por simplesmente realizar o encaminhamento dos pacotes. Já o plano de controle, é responsável pela inteligência e pela visão global da rede, sendo implementado de forma logicamente centralizada no controlador de rede, também denominado de sistema operacional de rede (XIA et al., 2015).

Ainda, o plano de controle tem como função o monitoramento da rede em geral, permitindo ao controlador o gerenciamento das entradas das tabelas de encaminhamento e das regras associadas ao tráfego desejado. Nessa perspectiva, é possível simplificar a configuração e execução das políticas definidas para a rede (MCKEOWN et al., 2008). Com SDN, é possível introduzir programabilidade às redes de computadores, além do mais, sua arquitetura é dinâmica, gerenciável e adaptável, assim, otimizando o desempenho das aplicações e tarefas, aumentando a agilidade na resolução de possíveis problemas (LIU et al., 2016).

Figura 1 – Paradigma Tradicional x Paradigma SDN.



Fonte: Adaptação de Brito (2013).

Ademais, SDN foi desenvolvida e pensada para facilitar a inovação e possibilitar o controle programático, através do *software*, dos elementos encaminhadores de pacotes, simplificando o gerenciamento da rede como um todo (NUNES et al., 2014). Nesse sentido, diferentemente das redes tradicionais, as quais os planos de controle e dados são implementados e executados no mesmo equipamento, impossibilitando qualquer alteração dos mesmos, em SDN devido a separação entre estes planos, essa limitação foi rompida, assim permitindo que os equipamentos de redes não fiquem limitados as implementações proprietárias, eliminando a necessidade de aquisição de *middleboxes* pelas organizações (NUNES et al., 2014).

Devido a todas estas mudanças introduzidas por meio deste novo conceito de rede, pode-se então citar algumas vantagens de SDN em relação as redes tradicionais (XIA et al., 2015):

a) **Desacoplamento do plano de dados do plano de controle**: Em SDN não é necessário mais a utilização de equipamentos dedicados com *software* e *hardware* proprietários, a partir desse paradigma as aplicações de rede passam a ser definidas por *software*, dessa forma propiciando a interoperabilidade entre as aplicações;

b) **Controle e gerenciamento centralizado**: SDN permite o controle logicamente centralizado da rede em tempo real, definição e a execução de políticas bem definidas;

c) **Configuração**: Em SDN é possível configurar os dispositivos de rede por meio de aplicações programadas em *software* de um único ponto automaticamente através do controlador da rede. Dessa forma, uma rede inteira pode ser configurada dinamicamente sem necessitar configurar individualmente cada equipamento;

d) **Desempenho**: SDN oferece oportunidades para melhorar o gerenciamento e o desempenho da rede devido a sua inteligência e a visão global que possui da mesma, assim muitos problemas de otimização de desempenho podem ser resolvidos através da configuração aprimorada e do desenvolvimento de aplicações de redes adequadamente projetadas;

e) **Inovação**: SDN incentiva a inovação e o desenvolvimento de novos mecanismos programáveis, ainda possibilita a experimentação destas novas soluções em ambientes de larga escala.

Nessa perspectiva, para estabelecer a comunicação entre o plano de controle e o plano de dados foi preciso criar e padronizar uma *Application Programming Interface* (API), sendo então estabelecido o protocolo *OpenFlow* (MCKEOWN et al., 2008). Este é um protocolo aberto que possibilita o desenvolvimento de mecanismos programáveis baseado em tabelas de fluxos em diferentes *switches OpenFlow* (MCKEOWN et al., 2008). Ainda, o mesmo protocolo estabelece uma comunicação segura entre os dispositivos de encaminhamento (*switches OpenFlow*) e o controlador, o qual utiliza esse canal para monitorar e estabelecer fluxos conforme a inteligência estabelecida pelo *software*. Este protocolo será melhor detalhado posteriormente na seção *OpenFlow*.

No que tange o funcionamento do paradigma SDN, de forma resumida, basicamente este segue um princípio simples, onde cada pacote recebido em uma das interfaces do comutador *OpenFlow* gera uma consulta à tabela de encaminhamento do mesmo. No caso de equipamentos de camada 2 (L2), como *switches*, essa consulta é baseada no endereço físico (MAC) de destino do pacote; em equipamentos de camada 3 (L3), como roteadores, em um prefixo do endereço IP de destino; então caso não exista, na tabela de encaminhamento, uma ação definida para este pacote, isto é, o dispositivo não sabe o que fazer com um novo fluxo de entrada ou porque há uma ação explícita para enviar o fluxo para o controlador, o equipamento encaminha o pacote para o controlador (KREUTZ et al., 2015).

Ainda, o controlador poderá inserir uma regra na tabela de encaminhamento do comutador, para que o próximo pacote deste tipo, que passar pelo equipamento, não precise ser encaminhado novamente para o controlador (MCKEOWN et al., 2008).

Algumas dessas regras, incluem:

a) encaminhar o pacote para uma porta específica do dispositivo;

b) alterar parte de seus cabeçalhos;

c) descartá-lo; ou

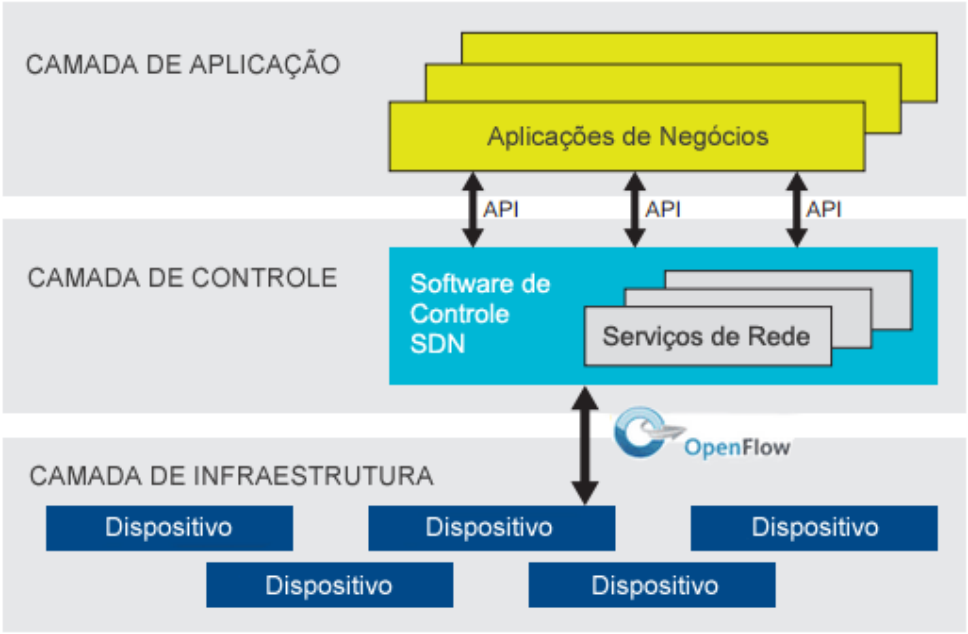
d) encaminhá-lo para inspeção por um controlador da rede.

Na próxima seção será apresentada de forma detalhada a arquitetura SDN e suas principais características.

2.1.1.1 Arquitetura SDN

Conforme pode-se observar na Figura 2, a arquitetura SDN é composta pelas camadas de Infraestrutura, Controle e Aplicação.

Figura 2 – Arquitetura SDN.



Fonte: Adaptação de Fundation (2012).

A seguir tem-se em detalhes as principais características de cada camada que compõem esta arquitetura:

a) **Infraestrutura**: A camada de infraestrutura SDN proporciona um acesso aberto programável por meio do protocolo *OpenFlow*. Ainda, esta camada é similar a equivalente nas redes tradicionais, sendo composta por equipamentos de rede. Em SDN estes são referidos como *switches OpenFlow*. Porém, a diferença é que em SDN estes dispositivos são simples encaminhadores de pacotes sem nenhuma inteligência para tomar decisões.

Estes dispositivos basicamente tem como principais funções coletar o *status* da rede armazenando-o temporariamente em dispositivos locais e enviá-los para os controladores. Também são responsáveis por processar os pacotes com base em regras fornecidas pelos controla-

dores. A inteligência da rede, agora está logicamente centralizada no controlador da rede e este é executado em uma plataforma de *commodity hardware* (XIA et al., 2015);

b) **Controle**: Nesta camada encontra-se o elemento crítico, considerado o "cérebro" da arquitetura SDN, isto é, o controlador SDN. Ainda, esta camada sobrepõe a camada de aplicação e a de infraestrutura através de duas interfaces. Para realizar a comunicação entre os controladores e os elementos encaminhadores, ela utiliza a API *Southbound*. Já para comunicar-se com a camada de aplicação ela utiliza a API *Northbound*. Além disso, através da camada de controle, as aplicações podem obter uma visão global instantânea da rede (XIA et al., 2015).

Ainda, por meio da inteligência do controlador, é facilitado o gerenciamento e a automatização da rede, tornando mais fácil a integração e a administração de aplicações (XIA et al., 2015). Destaca-se ainda que um controlador SDN também é responsável por gerar as regras de encaminhamento de pacotes, descrevendo as políticas e instalando-as em dispositivos de comutação *OpenFlow*, responsáveis por encaminhar os pacotes de acordo com as políticas estabelecidas pelo controlador (HU; HAO; BAO, 2014);

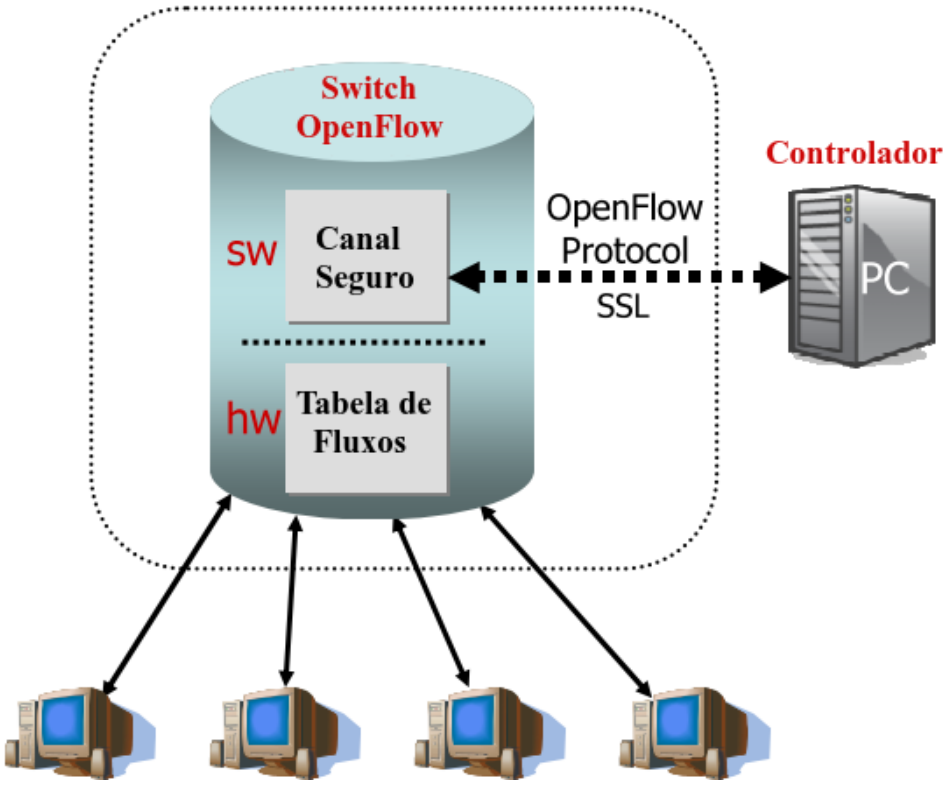
c) **Aplicação**: No topo está a camada de aplicação, que contém aplicações SDN que permitem automatizar tarefas de configuração, implementação e a criação de novos serviços de redes. Estas aplicações podem incluir as funcionalidades de *firewall, switch, router*, entre outras, e são capazes de acessar e controlar dispositivos de comutação localizados na camada de infraestrutura. Além disso, tais aplicações são projetadas para atender aos requisitos dos usuários (XIA et al., 2015), e desempenham um papel importante na medida em que podem englobar diferentes requisitos (HU; HAO; BAO, 2014).

2.1.1.2 OpenFlow

O protocolo *OpenFlow* foi proposto pela Universidade de *Stanford*, seu objetivo inicial foi atender à demanda de validação de novas propostas de arquitetura em redes de campus e protocolos de rede sobre equipamentos comerciais (MCKEOWN et al., 2008). Conforme pode-se observar na Figura 3, este protocolo permite a comunicação entre os elementos encaminhadores (*switches OpenFlow*) e o controlador da rede. Além disso, tal protocolo também teve como objetivo proporcionar o isolamento do tráfego de produção do experimental. Dessa forma, proporciona aos pesquisadores da área de redes a execução de novos protocolos e arquiteturas sem prejudicar redes em produção (MCKEOWN et al., 2008). Com isso, pode-se obter uma maior validação das pesquisas, e também promover a inovação no núcleo da rede por meio da execu-

ção de redes experimentais em paralelo com a infraestrutura operacional (MCKEOWN et al., 2008).

Figura 3 – Arquitetura SDN utilizando Protocolo OpenFlow.



Fonte: Adaptação de McKeown (2008).

Ainda, conforme a Figura 3 percebe-se que uma arquitetura SDN *OpenFlow* possui quatro componentes principais, estes são melhor detalhados a seguir:

a) **Tabela de Fluxos**: A tabela de fluxos consiste em regras, ações e contadores estatísticos. Através dessa tabela os *switches OpenFlow* tem a função de executar o encaminhamento de pacotes de acordo com as entradas que constam nessa tabela e que são periodicamente atualizadas pelo controlador;

b) **Canal Seguro**: Consiste em um canal que tem a finalidade de possibilitar a troca segura de informações entre os comutadores (*switches OpenFlow*) e o controlador da rede. Além disso, busca mitigar possíveis ataques de elementos mal intencionados e também atenua as taxas de erros na troca destas informações entre os mesmos;

c) **Protocolo OpenFlow**: Protocolo aberto que utiliza uma interface externa padronizada para realizar a comunicação e a troca de mensagens entre o controlador da rede e os comutadores. De forma mais prática, o protocolo *OpenFlow* determina como um fluxo pode ser definido,

quais serão as ações que podem ser realizadas para cada pacote pertencente a este fluxo;

d) **Controlador**: Também chamado de sistema operacional de rede, é o elemento que provê a inteligência à rede, responsável por tomar as decisões, como por exemplo, adicionar e/ou remover as entradas na tabela de fluxos dos comutadores, de acordo com o objetivo desejado. De forma resumida, realiza o gerenciamento e controle da SDN como um todo.

Ainda, o *OpenFlow* permite que os fabricantes de *hardware* possam inserir as funcionalidades deste protocolo nos seus comutadores sem a necessidade de expor o projeto desses equipamentos. Ademais, este protocolo oferece serviços de manipulação das tabelas de fluxos para o controlador, por exemplo, pode-se, inserir, excluir, modificar e procurar as entradas nas tabelas de fluxos através de um canal TCP seguro de forma remota (XIA et al., 2015).

Nesse contexto, conforme a Figura 4, cada entrada na tabela de fluxos do dispositivo de rede consiste em regras, ações e contadores estatísticos. A regra é formada com base na definição do valor de um ou mais campos do cabeçalho do pacote, sendo definido por meio dela como o fluxo será determinado. Ainda, através dos campos do cabeçalho é formado uma tupla de doze elementos. Essas tuplas podem ser formadas por meio das camadas de enlace, rede ou transporte, segundo o modelo TCP/IP.

As ações são associadas ao fluxo, definindo o modo como os pacotes devem ser processados e para onde devem ser encaminhados. As ações podem ser, por exemplo:

a) Encaminhar pacotes;

b) Enviar pacotes para o controlador;

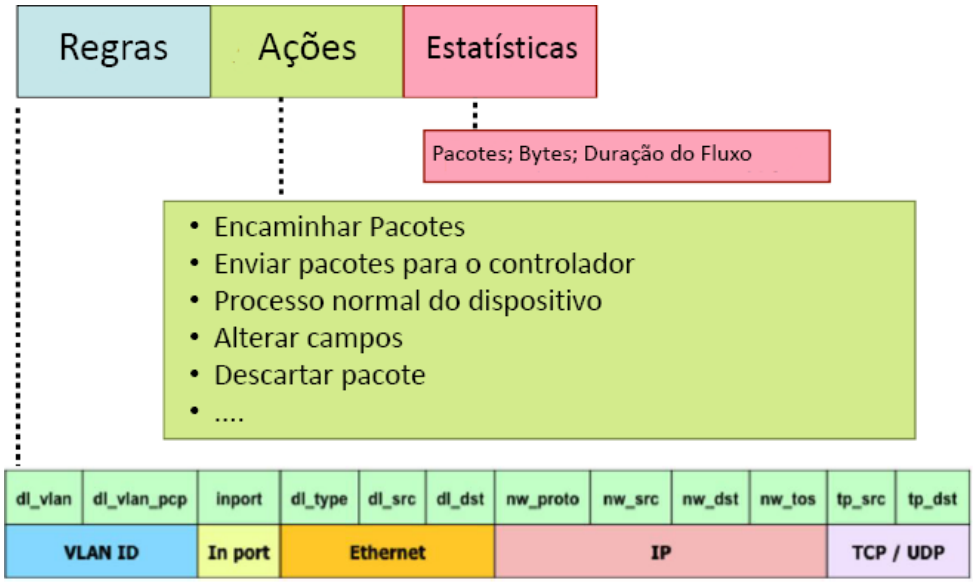
c) Processamento normal do dispositivo;

d) Alterar campos;

e) Descartar pacote.

Já, os contadores (controles estatísticos) são utilizados para manter e elaborar estatísticas de utilização do tráfego e para remover fluxos na rede que estejam inativos e/ou que não existam mais.

Figura 4 – Fluxo OpenFlow.



Fonte: Adaptação de Costa (2013).

Ressalta-se que além dos encaminhamentos dos fluxos poderem ser pelos endereços IP ou MAC dos pacotes, o mesmo pode se dar por outras características do pacote. Sendo assim, conforme pode-se observar na Figura 4, o encaminhamento também pode ocorrer através de portas de origem e destino dos protocolos de transporte.

2.1.2 NFV

O modelo tradicional de rede baseado em *software* e *hardware* verticalmente integrado e proprietário já não satisfaz mais a necessidade de escalabilidade da rede para atender os provedores de serviços de redes e de telecomunicações. Isso acontece frequentemente devido às limitações dos fabricantes de equipamentos de redes, onde os mesmos geralmente não possuem interoperabilidade com equipamentos de diferentes fornecedores, forçando a aquisição de quase toda a infraestrutura de rede de um mesmo fabricante (HERRERA; VEGA, 2016).

Além do mais, como as soluções proprietárias fornecem funcionalidades mais amplas, muitas vezes a necessidade de operacionalização de tarefas mais específicas e customizadas para determinado cliente acaba não sendo contemplada por tal equipamento, dessa forma não satisfazendo a necessidade do mesmo. Ainda, para a aquisição de equipamentos proprietários é preciso um investimento significativo por parte da organização.

Além disso, também é preciso contar com o espaço necessário para a instalação da

infraestrutura computacional e com o custo da implantação, manutenção e gerenciamento do ambiente. Ainda, necessita-se contratar mão de obra especializada de cada fabricante para realizar a correta configuração do parque computacional (HERRERA; VEGA, 2016).

Devido a todas estas dificuldades encontradas, várias empresas da área de telecomunicações se uniram e definiram um novo conceito para o provimento de novas soluções e serviços de redes de computadores. Então, a partir disso surge o paradigma NFV. Este propõe a virtualização das funções de rede em *hardware* de uso geral.

Ainda, com este paradigma é possível que as diversas funções de rede, como por exemplo, NAT, *proxy*, *firewall*, DNS, DHCP, *switching*, *routing*, *load balancing*, entre outras, até então implementadas em *hardware* especializado *(middleboxes)*, sejam executadas em *software* através de *Virtual Machine* (VM) controladas por um *hypervisor* instalado em um servidor de uso genérico.

Como estas VM são executadas em *hardware* de uso geral, desta forma é possível obter várias vantagens, como por exemplo, escalabilidade, elasticidade, flexibilidade, economia de energia elétrica. Também tem-se uma diminuição dos custos (CAPEX) e das operações (OPEX) com manutenções dos equipamentos da infraestrutura de rede (CHIOSI et al., 2012).

Para serem executadas em *software*, é necessário um grande esforço de programação para o desenvolvimento e execução destas funções em ambientes virtuais. Além disso, um grande desafio é fazer com que estas funções de rede virtuais tenham o mesmo desempenho dos dispositivos físicos que compõe a infraestrutura das redes atuais (CHIOSI et al., 2012).

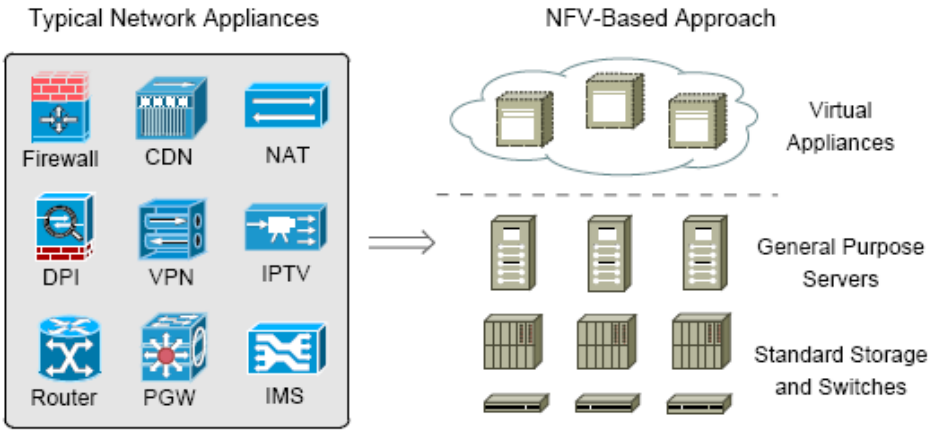
Este novo conceito, por ser uma das mais emergentes tecnologias da área de redes de computadores, começou a ganhar espaço em 2012 quando um grupo de operadoras de telecomunicações de todo o mundo propôs este novo paradigma sendo então publicado no mesmo ano pelo orgão *European Telecommunications Standards Institute* (ETSI), responsável pela normalização e padronização deste paradigma [ETSI, 2012].

A Figura 5 apresenta uma comparação da rede tradicional composta por diversos *appliances* de rede executando em *hardware* dedicado, onde cada um realiza uma funcionalidade específica e provê serviços para a rede. Em contrapartida, a abordagem baseada no paradigma NFV, agora mostra as funções de rede virtualizadas sendo executadas em *hardware* de uso geral *(commodity hardware)*.

Também é importante ressaltar que estes dispositivos virtuais podem ser instanciados sob demanda, sem a necessidade de instalação de novos equipamentos, como por exemplo, um

firewall baseado em *software* de código aberto pode ser executado em uma VM (HAN et al., 2015).

Figura 5 – Appliances de rede típica e abordagem baseada em NFV.



Fonte: (HAN et al., 2015, p. 91).

Ainda, através deste novo paradigma, torna-se possível que funções de rede virtualizadas, também chamadas de *Virtual Network Function* (VNF), sejam instaladas em diferentes pontos da rede. Estas VNF, podem estar no mesmo servidor, em servidores distintos ou em *data centers* independentes, desta forma, otimizando os recursos físicos de provedores, dos *data centers* e das redes corporativas em geral (HAN et al., 2015).

Ademais, com a escalabilidade e flexibilidade proporcionada por estas funções de rede virtualizadas, uma nova instância virtual pode ser instanciada. Isso pode ocorrer a medida que aumenta a criticidade das aplicações e dos requisitos de desempenho necessários para satisfazer as necessidades de negócio dos clientes. Em contrapartida, pode ser destruída quando não for mais necessária, liberando os recursos computacionais do equipamento hospedeiro e também provendo maior economia e eficiência energética (HERRERA; VEGA, 2016). NFV também aumenta a capacidade de implementar ou suportar novos serviços de rede mais rápidos e baratos. Algumas características deste paradigma são citadas a seguir (CHIOSI et al., 2012):

a) **Dissociação entre *software* e *hardware***: A partir da separação do *hardware* do *software*, a evolução, manutenção e o desenvolvimento de ambos é independente;

b) **Implantação flexível das funções de rede**: A separação entre *software* e *hardware* melhora o compartilhamento dos recursos de infraestrutura. Também quando trabalhando juntos podem executar funções diferentes em vários momentos;

c) **Dimensionamento dinâmico**: A separação da função de rede em componentes de

software instanciáveis proporciona maior flexibilidade e escalabilidade para dimensionar o desempenho de uma forma mais dinâmica e com maior granularidade.

Neste contexto, também pode-se citar alguns fatores que podem impulsionar ainda mais a utilização de NFV, tais como (CHIOSI et al., 2012) (MIJUMBI et al., 2016):

a) Otimização dos custos de operação e energia, pois NFV reduz a complexidade operacional do controle e gerenciamento das redes;

b) Agilidade na criação de serviços, trabalhando em conjunto com soluções da *cloud computing* e com SDN;

c) Escalabilidade e dinamicidade no gerenciamento dos serviços de rede;

d) Maior disponibilidade de compartilhamento de recursos de rede por meio de serviços e diferentes plataformas, permitindo assim melhor interoperabilidade;

e) Novo potencial de receita em virtude de inovação, a criação de serviços e tarifas *on-demand* gerará uma oportunidade adicional de receita para os provedores de serviço;

f) Melhor *time-to-market* (menor tempo de implementação de novas tecnologias para o mercado);

g) Habilidade para aumentar ou diminuir recursos computacionais muito rapidamente, provendo assim grande escalabilidade a rede.

Além destes fatores, a utilização deste paradigma apresenta alguns benefícios, tais como (CHIOSI et al., 2012):

a) Redução de custo de equipamentos e de consumo de energia;

b) Redução no tempo necessário para implementar novas abordagens/tecnologias de rede desacoplando o *hardware* no processo de inovação;

c) Direcionamento de serviços para um conjunto de clientes ou com base em posição geográfica tornando a escalabilidade dos serviços mais eficiente;

d) Incentivo à abertura de padrões, permitindo uma maior concorrência e encorajando a iniciativa de novas empresas na concorrência deste mercado.

Nessa perspectiva, as vantagens proporcionadas por NFV vão de encontro às limitações impostas pelas tecnologias proprietárias convencionais expostas anteriormente. Porém, novos requisitos são necessários e se fazem presentes aos seus desenvolvedores, como por exemplo, a necessidade de garantir interoperabilidade, alto desempenho e portabilidade das funções de rede. Estes requisitos devem ser superados para que esta tecnologia tenha cada vez mais ampla adoção pela indústria.

Neste contexto, além dos requisitos, NFV traz vários desafios para sua adoção em redes reais pelas empresas de telecomunicações. Como por exemplo, a garantia de desempenho da rede para *appliances* virtuais, instanciação e migração dinâmica das funções de redes virtualizadas, interoperabilidade entre plataformas de redes, desempenho, segurança, resiliência, entre outros (HAN et al., 2015).

Em suma, assegurar que o desempenho da rede continue a ser pelo menos tão bom quanto a de implementações em *hardware* dedicado deverá ser um dos principais desafios para utilização em larga escala desta nova tecnologia (HAN et al., 2015). Nesse sentido, observa-se que aos poucos este novo paradigma, ao lado de SDN, vem consolidando-se e já começa a ser adotado na implantação de soluções de infraestrutura de redes em alguns ambientes reais.

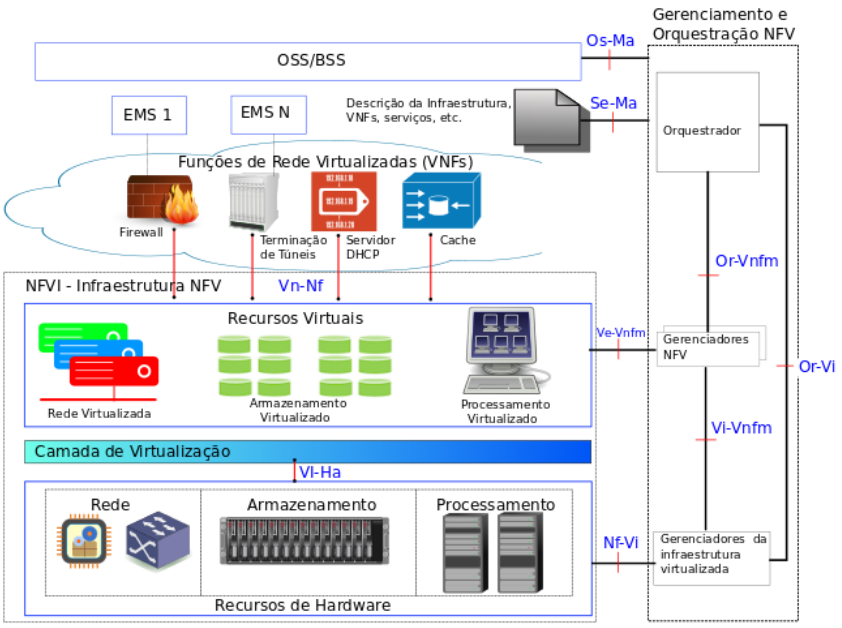
Ademais, é importante ressaltar que NFV e SDN são independentes uma da outra, porém são tecnologias complementares que podem trabalhar muito bem em conjunto de forma integrada. Nesta integração, SDN passa a ter como função prover a infraestrutura da rede e o encaminhamento dos pacotes. Já NFV, fica responsável pela inteligência e pelo processamento dos pacotes da rede. Ainda, com esta integração, NFV confere maior agilidade, versatilidade, elasticidade e escalabilidade para as redes, reduzindo as manutenções e os custos operacionais da infraestrutura de rede como um todo.

A seguir é apresentada a arquitetura NFV e as funcionalidades de seus principais componentes.

2.1.2.1 Arquitetura NFV

A arquitetura NFV, definida pelo órgão ETSI (CHIOSI et al., 2012), é dividida em alguns blocos funcionais. A mesma objetiva permitir a instanciação dinâmica de funções de rede virtuais, o relacionamento, as dependências e a conectividade entre as instâncias de VNFs e os blocos funcionais. Ainda, a conectividade entre diferentes VNFs pode gerar os grafos de encaminhamento (*forwarding graph-FG*), os quais têm a finalidade de prover serviços de rede. A Figura 6 ilustra a arquitetura NFV de alto nível, onde tem-se os principais blocos funcionais que serão detalhados a seguir:

Figura 6 – Arquitetura NFV em alto nível.



Fonte: (CHIOSI et al., 2012, p. 14)

a) **Funções de Rede Virtualizadas (VNF)**: Uma *Network Function* (NF) é um bloco funcional localizado dentro de uma infra-estrutura de rede que possui interfaces externas e funcionalidades e comportamentos bem definidos. Nesse sentido, VNF é uma implementação de uma NF que utiliza recursos virtuais executando em VM, onde uma única VNF pode ser composta de múltiplos componentes internos, e portanto pode ser implantada em várias VM, caso em que cada VM hospeda um único componente da VNF (MIJUMBI et al., 2016).

Nesse caso, o comportamento funcional de uma função de rede deve ser independente, isto é, seja se a função é virtualizada em múltiplas VM, em uma única VM ou não é virtualizada. Ainda, um serviço pode ser composto por uma ou mais NF. Exemplos de NF podem ser os elementos de uma rede doméstica, como por exemplo, *gateway* residencial, funções de rede convencionais, tais como, *switching, routing*, servidor DHCP, *firewall*, etc. Ressalta-se que sob perspectiva dos usuários, de forma transparente, os serviços de rede providos por essas funções que estão sendo executadas em equipamentos dedicados ou em VM tenham a mesma performance (CHIOSI et al., 2012) (ROSA et al., 2014);

b) **Infraestrutura NFV (NFVI)**: O bloco NFVI é a composição dos recursos de *hardware* e *software* necessários para a implementação, execução, e gerenciamento das VNFs e do ambiente NFV como um todo. Além do mais, este bloco inclui a conectividade entre *data*

centers e nuvens híbridas públicas ou privadas. Além do mais, conforme pode-se observar na Figura 6, os recursos físicos presentes neste bloco incluem *hardware* de rede (conectividade), armazenamento e processamento, onde os mesmos são fornecidos de forma virtual para as VNFs através da camada de virtualização que fica logo acima do *hardware* e abstrai os recursos físicos.

Já a sua infraestrutura para o provimento de NFVI pode ser distribuída em diferentes localidades (NFVI-PoP), de forma que a rede que provê conectividade entre os NFVI-PoP faz parte da NFVI. No entanto, a camada de virtualização para os recursos de *hardware* visto pelas VNF permite que a NFVI possa ser vista como uma entidade única (CHIOSI et al., 2012) (ROSA et al., 2014);

c) ***Framework* de Gerenciamento e Orquestração NFV (NFV MANO)**: O *framework* de Gerenciamento e Orquestração NFV é composto pelo: Orquestrador, Gerente de VNF e Gerente da Infra-estrutura Virtualizada. Este *framework* fornece a funcionalidade necessária para as tarefas de gerenciamento aplicadas as VNF, como por exemplo, provisionamento e configuração. O NFV-MANO inclui a orquestração e o gerenciamento do ciclo de vida de recursos físicos ou virtuais que suportam a virtualização de infraestrutura de VNF.

Também é de sua responsabilidade informar a localização das VNF na rede física. Além disso, o NFV-MANO concentra-se na gestão específica das tarefas de virtualização na estrutura NFV. Ademais, entre suas atividades, também define interfaces que podem ser usadas para realizar as comunicações entre os diferentes componentes deste bloco funcional. Ainda, realiza a coordenação com a rede tradicional através de sistemas de gerenciamento para permitir a operação de VNF executadas em equipamentos legados.

O gerente da infraestrutura virtualizada controla a interação das VNF com os recursos físicos, realizando o gerenciamento dos recursos, como por exemplo, alocação e desalocação. Além disso, realiza operações para garantir a visibilidade da infraestrutura, coleta de informações para gerência de falhas e desempenho. Já o gerente de VNF, é responsável pelo gerenciamento do ciclo de vida das VNF. Já o orquestrador, é responsável pelo gerenciamento dos serviços, coadunando recursos de infraestrutura e de *software* para as VNF.

Ainda, apresenta-se outras entidades envolvidas no NFV-MANO, dentre essas, inclui-se o *Element Management System* (EMS) que provê funções típicas de gerenciamento para uma ou mais VNF. Também há os Sistemas de Suporte Operacional (*Operational Support Systems* - OSS) e os Sistemas de Suporte ao Negócio (*Business Support Systems* - BSS). O primeiro é

responsável por processos internos da operadora de rede, como por exemplo, inventário de rede, provisionamento de serviços, configuração de elementos de rede e gerenciamento de falhas. Já o segundo, são os sistemas que lidam com solicitações dos usuários, suportando processos como processamento de cobranças, ordens de serviço, entre outros.

Além do mais, tem-se a Base de Configuração, o qual possui informações de configuração dos serviços, das VNF e da infraestrutura. Esta também inclui em seu escopo *templates* para a implementação de VNF, grafos de encaminhamento e informações relativas aos serviços e infraestrutura (CHIOSI et al., 2012) (ROSA et al., 2014);

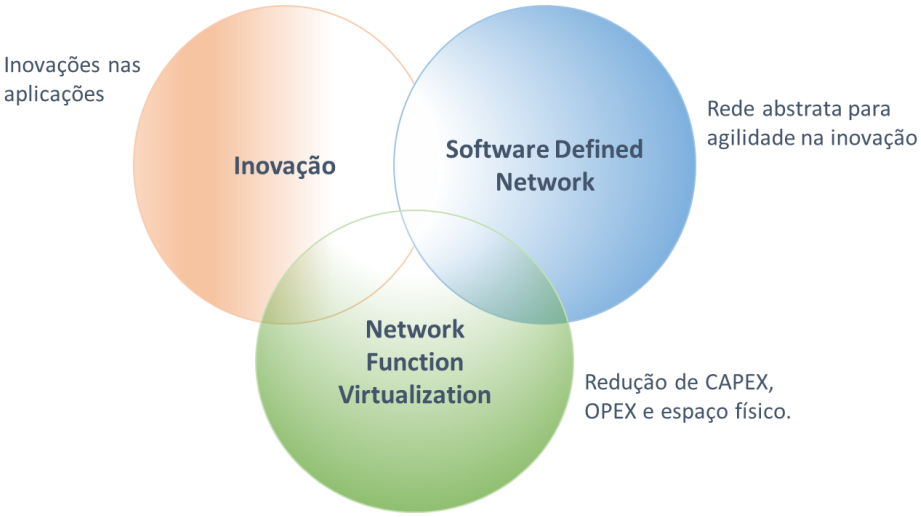
d) **Camada de Virtualização**: A camada de virtualização que está dentro do bloco NFVI, possui como função principal abstrair os recursos de *hardware*, desacoplando o *software* das VNF do *hardware* especializado. Para realizar esta abstração, nesta camada é realizado o particionamento lógico dos recursos *(slicing)* entre as VNF. Dessa forma, os recursos virtuais serão separados dos físicos, provendo assim os recursos lógicos para as VNF, tornando a rede mais flexível e dinâmica.

Exemplos de tais recursos incluem, conectividade, armazenamento e processamento, onde o desacoplamento é realizado através de virtualizadores ou *hypervisors*, como KVM, Xen, VMWare, etc, que permitem a execução de diversas VM em *hardware* de uso geral. Ademais, os virtualizadores são responsáveis pela alocação dos recursos físicos para a implementação das VM (CHIOSI et al., 2012) (ROSA et al., 2014).

2.1.3 Relacionamento entre SDN e NFV

No que tange o relacionamento entre SDN e NFV, conforme Figura 7, observa-se que estes paradigmas são altamente complementares, porém não dependem um do outro. Nesse sentido, NFV pode ser implementada sem SDN e vice-versa. Porém, quando estas tecnologias são combinadas, os recursos da rede são melhor aproveitados, possibilitando que os problemas sejam identificados e resolvidos mais rapidamente. Além do mais, pode-se obter maior eficiência e valor agregado nas soluções e arquiteturas propostas. Dessa forma, percebe-se que em muitos trabalhos integrando as duas tecnologias, frequentemente SDN fica responsável pelo provimento da infraestrutura de rede e NFV pela inteligência e pelo processamento dos pacotes (HERRERA; VEGA, 2016) (MIJUMBI et al., 2016).

Figura 7 – Relacionamento entre SDN e NFV.



Fonte: Adaptação de Herrera (2016).

Percebe-se que SDN pode ser entendida como uma tecnologia habilitadora de NFV. Como exemplo, SDN permite que os pacotes sejam roteados adequadamente para os servidores que executam o *software* das funções de rede virtualizadas. No que tange a parte de virtualização, NFV tem como um dos objetivos principais, substituir os dispositivos dedicados e de alto custo por funções programadas em *software* executando em VM implantadas em servidores genéricos. Dessa forma, ambos os paradigmas, permitem diminuir a quantidade de *hardware* proprietário necessário para operar serviços de rede (HERRERA; VEGA, 2016).

Devido a essa forte complementariedade, estes dois paradigmas estão estritamente relacionados, pois propiciam um ambiente com mais recursos de automação e virtualização da rede. Também possuem a capacidade de melhorar significativamente o desempenho da rede. Além do mais, SDN auxilia eficientemente no gerenciamento das cargas de tráfego da infraestrutura computacional e das funções de rede virtual (HERRERA; VEGA, 2016).

Ainda, SDN oferece aos usuários uma forma de gerenciar serviços de rede por meio de *software*, assim permitindo uma configuração mais rápida. Já NFV, devido a sua grande flexibilidade, possibilita respostas mais rápidas para atender as demandas da rede. Ademais, com SDN e NFV, pode-se implementar infraestruturas de rede mais centralizadas nos usuários finais, melhorando a escalabilidade e facilitando a customização de toda a rede (MIJUMBI et al., 2016).

A combinação destes dois paradigmas provê muitos benefícios em relação as redes tradicionais, tais como (HERRERA; VEGA, 2016) (HERRERA; BOTERO, 2016):

a) O *hardware* proprietário que frequentemente é muito oneroso e cumpre uma função específica, é substituído por *hardware* de uso geral, consequentemente diminuindo os custos de aquisição e manutenção da rede;

b) As funções de rede são implementadas em *software*, onde as mesmas possibilitam o desenvolvimento de funcionalidades mais específicas para atender as diferentes necessidades da organização;

c) O plano de controle abstrai a infraestrutura subjacente utilizada para enviar e receber dados, promovendo a inovação, criatividade, abertura e competitividade na rede e em suas aplicações, sem a necessidade de estar sempre atualizando e adquirindo novos equipamentos de rede;

d) A separação do plano de controle e de dados proposto pela SDN melhora o desempenho, simplifica a compatibilidade com implementações existentes, facilita a operação e os procedimentos de manutenção da infraestrutura de rede.

Ainda, conforme pode-se observar o relacionamento entre as tecnologias SDN e NFV mostrado na Figura 7, NFV é responsável pela virtualização dos dispositivos dedicados em funções virtuais e pela execução destas em *commodity hardware*. Já SDN tem como foco a programabilidade, o gerenciamento e orquestração dos recursos. Dessa forma, esta permite que os administradores de rede programem as melhores estratégias para as VNFs, como por exemplo, definir o plano de encaminhamento do tráfego de rede (MIJUMBI et al., 2016).

Por fim, como tanto SDN quanto NFV são paradigmas baseados em *software*, isso torna o seu uso atraente para pequenas e grandes empresas. O investimento para adoção destes dois paradigmas possui boa relação de custo-benefício, sendo benéfico para todos os tamanhos de organizações. Sendo assim, para as que possuem poucos recursos financeiros e para as grandes empresas que precisam escalar mais rapidamente os seus serviços. Em suma, com esta integração obtêm-se maior agilidade, inovação, redução dos custos (CAPEX e OPEX), automatização, gestão de tarefas, escalabilidade, etc (MIJUMBI et al., 2016).

2.2 TRABALHOS CORRELATOS ENVOLVENDO SDN E NFV

Pode-se encontrar alguns trabalhos relacionados envolvendo a integração dos paradigmas SDN e NFV. Uma parcela destes trabalhos estão publicados na área de telecomunicações, onde estes paradigmas, por serem complementares, tem o potencial de ajudar as operadoras a satisfazerem os acordos de níveis de serviços (*Service Level Agreement* - SLA). Desta forma, os

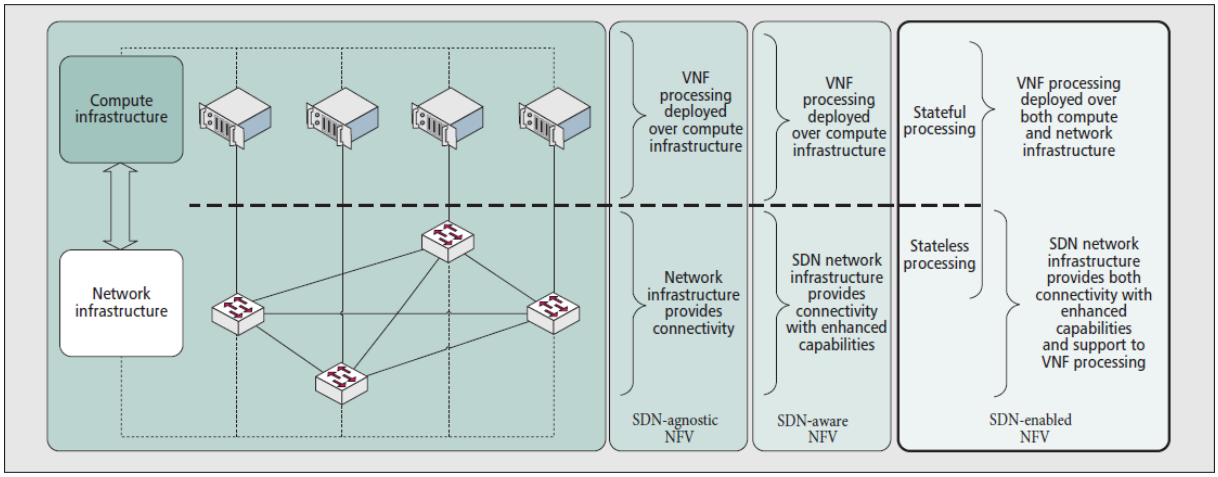
paradigmas SDN e NFV controlam e manipulam com maior precisão o tráfego das redes, assim minimizando as despesas com os custos (CAPEX) e com as operações (OPEX) da infraestrutura de rede.

O paradigma SDN não é apenas utilizado como suporte subjacente de infraestrutura de rede, mas também, dependendo do tipo de processamento do tráfego de rede, pode ser utilizado para auxiliar o paradigma NFV. Na pesquisa de (MATIAS et al., 2015) pode-se encontrar a classificação das arquiteturas que envolvem estes dois paradigmas trabalhando juntos de forma a se complementarem. Nesta pesquisa então propôs-se uma pequena taxonomia destes dois paradigmas trabalhando de forma integrada em três diferentes níveis de classificação: *SDN-agnostic NFV*, *SDN-aware NFV* e *SDN-enabled NFV*.

Na arquitetura *SDN-agnostic NFV*, o paradigma NFV funciona sem precisar saber o que o controlador SDN está realizando ou mesmo se ele existe na infraestrutura da rede, dessa forma o comportamento do controlador SDN nunca é alterado. Para a arquitetura *SDN-aware NFV*, as VNF têm o conhecimento que estão executando em uma rede SDN e podem obter diversos benefícios através da visão global que SDN possui da rede. Já em relação a arquitetura *SDN-enabled NFV*, o estado do controlador SDN é modificado pela funcionalidade NFV, nesse caso é permitido o processamento de pacotes em SDN quando não for preciso realizar análises profundas do tráfego de dados, isto é, quando o processamento for do tipo *stateless*.

Na Figura 8 é apresentada a classificação proposta por (MATIAS et al., 2015) mostrando a evolução das arquiteturas NFV trabalhando em conjunto com SDN.

Figura 8 – Classificação da Evolução das Arquiteturas NFV.



Fonte: Adaptação de Matias (2015).

Ainda, no trabalho de (MATIAS et al., 2015) é apresentada uma implementação de fluxo baseada em uma solução de controle de acesso a rede, onde SDN habilita NFV, estabelecendo serviços de autenticação de fluxos dos usuários finais. Nesta arquitetura, o acesso a rede é realizado nos comutadores de rede, enquanto o estado de autenticação e autorização é de responsabilidade do controlador. Com este projeto é possível obter uma escalabilidade elástica, uma granulação mais refinada, melhor adaptação as necessidades reais do projeto, dos usuários e total mobilidade de recursos.

No trabalho de (MASUTANI et al., 2014) é apresentada uma arquitetura flexível e elástica do serviço *Broadband Remote Access Server* (BRAS) executando de forma virtual. Também é utilizado a biblioteca *Intel Data Plane Development Kit* (DPDK) para aumentar a vazão e o processamento do fluxo dos pacotes no sistema, obter um controle de granulariedade mais refinado, e dessa forma obter um melhor desempenho do serviço. Nesta proposta é utilizada os paradigmas SDN e NFV, onde SDN suporta apenas a infraestrutura de rede subjacente programável e NFV é responsável por realizar o processamento dos pacotes. Este trabalho pode ser classificado como uma arquitetura do tipo *SDN-agnostic NFV*.

Na solução proposta de (ALHARBI; ALJUHANI; LIU, 2017), foi projetado uma arquitetura de mitigação de ataques do tipo *Denial Distributed of Service* (DDoS) através da integração dos paradigmas SDN e NFV. Em tal solução, determinados serviços de rede são inspecionados e é realizada a triagem por um analisador de tráfego que é responsável por determinar quais os fluxos de tráfego podem ser processados. Nesta arquitetura, NFV utiliza-se dos benefícios providos por SDN, onde a última é responsável por direcionar os fluxos de tráfego dinamicamente de acordo com a análise realizada pelo avaliador. Este trabalho pode ser classificado como uma arquitetura do tipo *SDN-aware NFV*.

No trabalho de (BOUBENDIR; BERTIN; SIMONI, 2016), é proposta uma arquitetura de rede como serviço, integrando os paradigmas SDN e NFV. A arquitetura de rede proposta atenua o elevado acoplamento entre serviços e infraestruturas de rede, sendo mais adaptável, dinâmica e resiliente, permitindo expressar ofertas e demandas de serviços. Nesta integração, SDN representa a parte da *Network Function Virtualized Infrastruture* (NFVI) e provê conectividade aos serviços.

Além do mais, um serviço NFV é construído baseado na combinação de VNFs usando o FG. Neste sentido, o controlador SDN implementa segmentos de rede virtuais para as cadeias de VNF provendo os serviços de rede. Além disso, o controlador SDN programa a rede usando

políticas para segregar o tráfego e direcionar o mesmo para ser processado por VNF particulares. Este trabalho pode ser classificado como uma arquitetura do tipo *SDN-enabled NFV*. Justifica-se isso pois o paradigma NFV aceita comandos de SDN para a implementação de segmentos virtuais para as VNF. Desta forma SDN participa efetivamente do provimento do serviço para a arquitetura de rede proposta no mesmo.

No trabalho de (DENG et al., 2015) é proposto o *framework VNGuard* que consiste em um *firewall* virtual implementado através da integração de SDN e NFV. Este *firewall* tem o objetivo de realizar a proteção de *Virtual Networks* (VN). Isso se faz necessário pois as VN enfrentam mudanças mais frequentes e exigem novos recursos de segurança que *firewalls* tradicionais não conseguem fornecer. Isso acontece já que estes em geral estão mais restritos a topologias de rede singelas e dependentes dos pontos de entrada da rede.

Indo além, *firewalls* tradicionais não possuem boa flexibilidade e adaptabilidade para atender às novas necessidades de segurança com relação as VNs. Em contrapartida, em NFV um *firewall* virtual é implementado como uma instância de *software* e oferece todas estas características necessárias para assegurar efetivamente maior proteção as redes virtuais. Já SDN é capaz de fornecer a engenharia de tráfego necessária para o correto funcionamento de tal função virtual. Também é relevante citar que funções virtuais, como o *firewall* desse exemplo, não possuem dependência de topologia fixa e de pontos de entrada. Nesse sentido, podem ser instanciados em qualquer VM com significativa flexibilidade, desde que a VM possa fornecer os recursos necessários.

Seguindo, para realizar a implementação da ferramenta proposta, os pesquisadores aproveitaram alguns recursos fornecidos pelo *ClickOS*. Ainda, *VNGuard* permite aos usuários definir suas políticas de segurança em alto nível, traduzir estas em regras para baixo nível, mantendo um desempenho satisfatório. Os resultados dos testes experimentais demonstraram a eficácia e eficiência do *framework* desenvolvido, onde em um dos experimentos realizados, verificou-se que o posicionamento do *firewall* virtual no *VNGuard* levou menos de 0,2 segundos para que este encontrasse uma posição ideal para aquela função na topologia de rede.

O trabalho de (BATALLE et al., 2013) apresenta a análise, projeto e implementação da função de roteamento virtualizada utilizando-se NFV sobre uma infraestrutura SDN através do protocolo *OpenFlow*. Por meio da função de roteamento virtualizada, evita-se a sobrecarga ocasionada por mensagens de sinalização a nível do plano de controle. Ainda, apresenta benefícios como a redução do OPEX e CAPEX. A redução de custos é alcançada através do gerencia-

mento logicamente centralizado e externo da funcionalidade de roteamento, diminuição dos *appliances* de roteamento, menor consumo de energia, etc.

Nessa perspectiva, reduz-se a configuração e o tempo de implantação e os custos de fabricação, pois o dispositivo físico agora fica completamente dedicado a encaminhar pacotes, enquanto a inteligência está localizada no controlador, o qual executa a função de roteamento. Além do mais, pode-se mover ou instanciar novas VMs com esta funcionalidade virtual, mantendo-se as tabelas de roteamento dinâmicas e simplificadas entre os diversos elementos que compõe a rede. Para validar e demonstrar a aplicabilidade da função de roteamento virtualizada foram realizados alguns testes experimentais, onde a rede foi simulada no *MiniNet*.

Nesse contexto, as abordagens reativas e pró-ativas de roteamento foram analisadas, onde os resultados obtidos podem variar dependendo da carga de tráfego, bem como alguns indicadores de desempenho, como por exemplo, o número de fluxos de entradas no *switch* ou a sobrecarga causada pela sinalização da troca de pacotes. Após a realização dos testes, observou-se que quando o número de entradas de fluxo aumenta, o *Round Trip Time* (RTT) mantém um comportamento estável.

Ainda, quando utilizado o mecanismo de roteamento pró-ativo, pode-se evitar o envio de muitas mensagens para o controlador executando o roteamento, reduzindo assim o atraso médio geral. Além disso, em tal trabalho, também verificou-se que quando o número de comunicações aumenta, o número de entradas de fluxo é reduzido, pois a função de roteamento responsável pelo cálculo das rotas consegue gerenciar melhor a entrada de fluxos devido à transferência da inteligência para uma máquina externa (controlador), dessa forma tornando a rede mais escalável.

2.3 DISCUSSÃO GERAL

Pode-se observar que muitos trabalhos encontrados na literatura envolviam a implementação de novas propostas de arquiteturas e funções de rede utilizando-se SDN e NFV de forma separada ou integrada. Além do mais, frequentemente estes trabalhos são destinados para a área de telecomunicações, envolvendo operadoras e prestadores de serviços da área de redes. Nesse sentido, mesmo estes paradigmas podendo trabalhar independentemente um do outro, a utilização destes dois paradigmas implementados de forma integrada possibilita maior eficiência e valor agregado às soluções de rede.

Através desta integração, é possível obter maior controle do tráfego de rede, diminuição

dos custos com manutenções e operações da infraestrutura de rede. Ainda, nos trabalhos analisados que utilizam estes dois paradigmas em conjunto, percebe-se que geralmente SDN fica responsável por prover a infraestrutura de rede e o encaminhamento dos pacotes. Também é possível analisar que NFV fica responsável pela inteligência e pelo processamento dos pacotes das soluções propostas.

A partir do conhecimento da classificação e da evolução da arquitetura NFV trabalhando com SDN, foi possível obter maior aprofundamento para realizar a avaliação e o entendimento dos trabalhos envolvendo estes dois paradigmas. Nas soluções propostas, desenvolvidas e avaliadas destes trabalhos, frequentemente é mencionado que a NFV confere maior agilidade, elasticidade e escalabilidade para a infraestrutura computacional.

Ademais, por meio da integração de SDN com NFV obtêm-se outros benefícios. Nesse contexto, também pode-se perceber que as arquiteturas tornam-se mais flexíveis e gerenciáveis. Assim, tem-se o controle com maior nível de granularidade e maior vazão no processamento dos fluxos dos pacotes. Também percebe-se uma melhor adaptação e mobilidade as reais necessidades do projeto. Dessa forma, viabiliza-se melhores resultados nas avaliações qualitativas e principalmente, nas quantitativas.

Também observou-se que a virtualização de funções de rede alcança bons resultados em relação as funções implementadas em *hardware* dedicado. Com a virtualização, os dispositivos físicos dedicados não são mais necessários. Dessa forma, também proporciona a diminuição dos gastos de aquisição e manutenção destes equipamentos. Além do mais, pode-se concluir que alguns dispositivos tradicionais não possuem significativa flexibilidade e adaptabilidade quando comparado ao que suas funções virtualizadas oferecem.

Seguindo este contexto, também pode-se constatar que SDN é considerada parte integrante, isto é, que compõe a arquitetura NFV, quando estes paradigmas estão trabalhando integrados. Acredita-se que isso é devido a SDN ser responsável pelo provimento da infraestrutura computacional, sendo capaz de tornar as redes corporativas mais ágeis e flexíveis, facilitando o gerenciamento através da visão global da infraestrutura de rede o qual possui. Já NFV, acaba tendo maior destaque pois fica responsável de fato pelo processamento dos pacotes, detendo o controle e a inteligência da rede.

Por fim, é importante ressaltar que embora SDN tradicionalmente seja responsável pelo encaminhamento dos pacotes e NFV ao processamento dos mesmos, ambas as tecnologias podem executar um mesmo conjunto de funções de redes. Nesse contexto, os trabalhos relaciona-

dos não realizam uma análise considerando em quais casos, situações e cenários cada tecnologia é mais vantajosa de ser utilizada para implementar determinada função de rede.

Também percebe-se que os trabalhos relacionados possuem um foco maior apenas na implementação de determinada solução, serviço ou arquitetura de rede utilizando-se SDN de forma separada ou integrada à NFV. Ainda, tais trabalhos visam mais a realização e a validação das soluções por meio de testes frequentemente levando em consideração apenas aspectos e métricas quantitativas. Assim, acabam deixando de abordar características importantes, aspectos qualitativos e discussões mais detalhadas dos resultados obtidos.

3 FUNÇÕES DE REDE

Com a evolução da Internet nos últimos anos, houve um crescimento exponencial dos equipamentos que realizam funções de redes, também chamados de *appliances* de rede ou simplesmente *middleboxes*. Estes dispositivos são considerados elementos integrantes da arquitetura tradicional das redes, sendo encontrados frequentemente na composição de sua infraestrutura computacional. Nesse contexto, pode-se citar algumas razões para o elevado número de *middleboxes* implantados nas redes atuais, tais como (MULLER, 2013):

a) Necessidade frequente de se implementar novas funcionalidades na rede sem precisar alterar os *hosts* finais, assim ocasionando o aumento dos *middleboxes* na estrutura computacional (MULLER, 2013) (EDELINE; DONNET, 2015);

b) Implantação e disponibilização de novos serviços de forma mais rápida através dos *middleboxes*. Neste sentido, ressalta-se como exemplo o uso da função NAT, o qual inicialmente surgiu como uma solução paliativa devido a escassez de endereços IPv4 públicos, uma vez que inovações significativamente são mais difíceis de se implementar apenas nos *hosts* finais. Contudo, com a adoção do IPv6 isso pode ser minimizado, já que agora não necessita-se mais fazer uso do NAT, possibilitando a comunicação fim a fim entre os *hosts* finais (MULLER, 2013);

c) Os *middleboxes* oferecem diversas funcionalidades com finalidades específicas para o funcionamento das redes, tais como, *switching*, roteamento, *firewall*, balanceamento de carga, etc, assim, estes já são parte integrante da infraestrutura da Internet atual (MULLER, 2013);

Ainda, estes desempenham um papel importante no provisionamento de recursos e serviços, principalmente para a diversidade de aplicações nas redes empresariais (POURNAGHSHBAND, 2014). Além disso, de acordo com a RFC 3234 (CARPENTER; BRIM, 2002), um *middlebox* pode ser definido como qualquer dispositivo intermediário que desempenha funções que vão além das funções tradicionais. Nesse contexto, estas funções estão além de funções conhecidas, como as de um roteador IP que realiza o roteamento de pacotes entre diferentes redes. Nesse sentido, estes dispositivos são capazes de melhorar o processamento, desempenho e segurança do tráfego, onde estas alterações refletem implicações significativas para os remetentes e receptores (CARPENTER; BRIM, 2002).

As funções de redes implementadas por tais equipamentos são fundamentais para atender as necessidades e demandas de novas aplicações, recursos e serviços inseridos na infra-

estrutura computacional. Além disso, também são importantes para satisfazer as demandas sociais, entretenimento e a necessidade de vários negócios. Indo além, os serviços fornecidos por tais funções, ajudam a estabelecer e coordenar comunicações seguras entre diferentes domínios de acordo com a definição de políticas de controle de acesso (POURNAGHSHBAND, 2014) (ZHANG et al., 2017) (MULLER, 2013).

Nesse contexto, neste capítulo será apresentada uma classificação das principais funções de redes utilizadas atualmente. Tais funcionalidades de redes foram agrupadas em classes, onde estas representam as suas principais finalidades e responsabilidades. Ainda, para facilitar tal classificação, foi proposta uma taxonomia destas funções de redes. Também será discutido de forma mais ampla algumas das principais características, exemplos e possíveis cenários de casos de uso destas funções de redes.

3.1 TAXONOMIA DE FUNÇÕES DE REDE

Conforme citado anteriormente, a presente pesquisa também propõe uma taxonomia que agrupa funcionalidades e responsabilidades de algumas das principais funções de redes utilizadas atualmente. Nesse contexto, ressalta-se a inexistência na literatura de uma taxonomia de funções de rede. Dessa maneira, encontrou-se disponível apenas a RFC 3234 *Middleboxes: Taxonomy and Issues* (CARPENTER; BRIM, 2002). Esta RFC tem como objetivo apresentar e discutir algumas funções de rede, bem como expor algumas de suas principais características. Além do mais, descreve e analisa o impacto atual dos *middleboxes* na arquitetura da Internet e suas aplicações. Com isso evidencia-se a importância de uma taxonomia de funções de redes, pois é uma taxonomia nova, já que a mesma não existe na literatura e por isso justifica-se como uma das principais contribuições desta dissertação.

Assim, a partir da leitura dessa RFC e através de um estudo mais aprofundado, elencou-se algumas dessas características que serviram como embasamento, de forma indireta, para a criação da taxonomia de funções de rede proposta neste trabalho. Além do mais, salienta-se que tais características foram muito importantes, pois também estão relacionadas com as funções avaliadas nesta pesquisa. Ainda, ajudaram a elucidar em quais categorias as funções de rede identificadas deveriam ser classificadas.

A seguir apresenta-se estas características:

a) **Camada de operação**: A função opera em qual camada do modelo OSI. Por exemplo, um *firewall* simples atua na camada de rede;

b) **Propósito do *middlebox***: O *middlebox* pode ser apenas uma otimização ou pode desempenhar uma função. Se operar como uma função, significa que o *middlebox* faz parte do funcionamento da aplicação e que não pode funcionar sem esta. Já caso seja uma otimização, significa que o mesmo tem como objetivo principal adicionar determinada funcionalidade a aplicação que deseja-se otimizar;

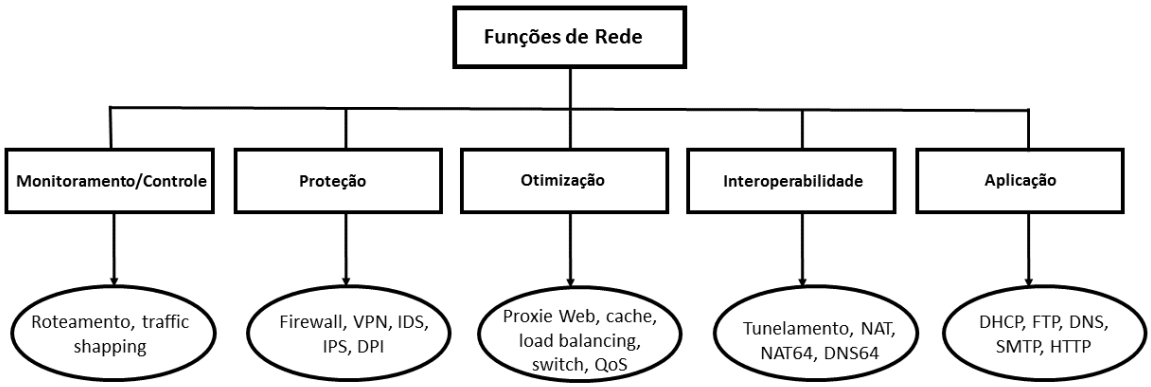
c) **Roteamento ou Processamento**: O *middlebox* pode escolher simplesmente o caminho que vai enviar os pacotes ou processar e realizar alguma forma de alteração nos mesmos;

d) **Estado de funcionamento**: O estado de funcionamento pode ser: *soft state* ou *hard state*. Se o *middlebox* perder as informações de seu estado, a sessão pode continuar sendo executada em um modo de degradação até recuperar o estado necessário (*soft state*) ou simplesmente falhar (*hard state*);

e) **Resiliência**: No caso de um *middlebox* falhar executando no modo *hard state*, a sessão é redirecionada para um *middlebox* alternativo que contenha a cópia da informação do estado atual, ou então é forçada a abortar à execução e reiniciar.

Ainda, conforme pode-se observar na Figura 9, esta taxonomia foi categorizada, extraindo propriedades em comum, em cinco áreas funcionais: Monitoramento/Controle; Proteção; Otimização; Interoperabilidade e Aplicação. Estas áreas serão melhor explicadas a seguir:

Figura 9 – Taxonomia de Funções de Rede.



Fonte: Autoria Própria.

a) **Monitoramento/Controle**: São representadas por funções de rede, tais como, roteamento, *traffic shapping*, etc. Funções desta categoria tem o objetivo de realizar o controle, monitoramento e gerenciamento de funções de rede. Neste sentido, os provedores de acesso podem usar ferramentas para realizar o controle de largura de banda de determinadas aplicações.

Nesse caso, restrições são aplicadas a determinados tipos de fluxos e tráfegos, tais como, serviços de *streaming* de vídeos, compartilhamento e *download* de arquivos *torrent*. (CARPENTER; BRIM, 2002);

b) **Proteção**: Esta categoria diz respeito as funções de rede orientadas para desempenharem o papel de segurança na rede. A mesma pode servir com o propósito de fornecer mecanismos contra ataques de negação de serviço centralizado (*Denial of Service*-DoS) ou distribuído (*Distributed Denial of Service*-DDoS), ataques *Man-in-the- Middle* (MITM), ataques de falsificação e fragmentação IP, entre outros exemplos. Funções de rede desta categoria podem ser representadas por *firewalls*, *Intrusion Detection Systems* (IDS), *Intrusion Prevent Systems* (IPS), *Deep Packet Inspecion* (DPI), *Virtual Private Network* (VPN) (CARPENTER; BRIM, 2002);

c) **Otimização**: Esta categoria visa melhorar o desempenho da rede, pode ser aplicada em *links* ao longo do caminho da rede, em conexões entre duas extremidades do caminho. Também pode ser usada para melhorar o desempenho de *links* de baixa largura de banda, utilizando técnicas de compressão e codificação. Exemplos de funções desta categoria pode-se citar: *proxies, cache, switch, load balancing, QoS*. Nesse contexto, a função *cache* de conteúdo é destinada a otimizar os tempos de resposta dos pedidos dos usuários, carregando as informações anteriormente já acessadas de forma mais rápida. Já o *switch* melhora o desempenho da rede já que não replica os pacotes para todas as portas, pois cada porta é um domínio de colisão e ainda pode realizar a segmentação lógica da rede através do uso de VLANs. A funcionalidade de *load balancing* ajuda a distribuir melhor o conteúdo, de forma eficiente, entre os servidores da aplicação requerida (CARPENTER; BRIM, 2002);

d) **Interoperabilidade**: Esta categoria diz respeito as funções de rede que realizam a tradução e a interoperabilidade entre diferentes protocolos. A funcionalidade de tradução de endereço (NAT) que é muitas vezes integrada a roteadores domésticos, permite que endereços IP privados sejam transformados em um endereço IP público para serem roteáveis na Internet. Dessa forma, permite o acesso à Internet aos usuários domésticos. Além disso, também pertence a esta categoria, mecanismos de tunelamento e tradução entre redes IPv4 e IPv6, tais como, NAT64 e DNS64. Ainda, visa a conexão de diferentes redes e a tradução de protocolos da camada de rede. Alguns problemas podem surgir com funcionalidades desta categoria, como por exemplo, tradução desatualizada ou possíveis falhas entre diferentes protocolos. Dessa maneira, isso pode levar a inconsistências e à incapacidade de usar determinados recursos da rede, assim, degradando o desempenho desta (CARPENTER; BRIM, 2002);

e) **Aplicação**: Esta categoria é responsável pelas aplicações utilizadas diariamente por todos os usuários. Nesta categoria pode-se incluir as funcionalidades de rede DHCP, DNS, etc. Ainda, estas funcionalidades podem ser utilizadas em cadeias de funções de redes, podendo ser utilizadas em conjunto com outras funcionalidades em um domínio específico de rede. Por exemplo, pode-se agregar e configurar várias funcionalidades, inclusive de outras categorias. Nesse sentido, pode-se ter em um mesmo *hardware commodity* as funcionalidades de DHCP, *switching*, roteamento/NAT provido pelas operadoras de acesso aos usuários residenciais (CARPENTER; BRIM, 2002). A seguir será discorrido de forma mais ampla alguns casos e cenários práticos de uso de algumas destas funções de redes executando nos paradigmas SDN e NFV.

3.2 FUNÇÕES DE REDE EM SDN

Ao utilizar o paradigma SDN para a implementação de funções de redes, é possível realizar a programabilidade e orquestração destas funções por *software*. Além disso, este novo paradigma possibilita o provisionamento dos serviços de rede conforme as necessidades requeridas para a concretização das tarefas. Ainda, possui significativa capacidade para automatizar e administrar as redes através de um controle logicamente centralizado. Nesse contexto, funções de rede bem conhecidas podem ser implementadas utilizando-se este paradigma. Alguns exemplos destas funções incluem NAT, *proxy*, *switching*, roteamento, *firewall*, balanceadores de carga, servidor DHCP, QoS, entre outras (NUNES et al., 2014).

As funções de rede, quando implementadas em SDN, possuem diferentes comportamentos, características, vantagens em relação as mesmas executando nas redes tradicionais. Nesse contexto, como exemplo da categoria de monitoramento/controle, a função de roteamento, quando implementada nas redes atuais torna-se uma tarefa operacional complexa por alguns fatores, por exemplo, a convergência das rotas é significativamente onerosa devido ao atual esquema descentralizado das redes. Em contrapartida, quando implementada em SDN diminui a sobrecarga causada por mensagens de sinalização a nível do plano de controle.

Além disso, devido a inteligência da funcionalidade de roteamento estar centralizada logicamente no controlador, os dispositivos físicos ficam apenas encarregados de realizar a tarefa de encaminhamento dos pacotes, ocorrendo uma maior flexibilização da seleção de caminhos. Ainda, o cálculo e a convergência das rotas torna-se mais rápido, com isso obtêm-se um melhor gerenciamento das entradas de fluxos, consequentemente otimizando o tráfego dos pacotes pela infraestrutura computacional. Além do mais, com esta função definida por *software* e vir-

tualizada, agrega-se maior escalabilidade, já que facilmente pode-se migrar e instanciar novas VMs com este serviço, mantendo-se as tabelas de roteamento atualizadas entre os elementos que fazem parte da rede (BATALLE et al., 2013).

Ainda, com a programabilidade fornecida pelo paradigma SDN, este também pode ser utilizado para fornecer um sistema automatizado para gerenciar a configuração dos dispositivos físicos habilitados com a função de roteamento. Através deste maior poder de monitoramento pelos operadores e provedores de rede, pode-se reduzir os dados replicados nas tabelas de roteamento, o qual é uma das causas do significativo crescimento destas tabelas nas redes atuais. Além do mais, através da comunicação do controlador com os dispositivos genéricos, intermediada pela API *southbound* e pelo protocolo *OpenFlow*, a aplicação de roteamento pode calcular, distribuir e configurar as tabelas de roteamento entre os diferentes elementos encaminhadores da rede (KREUTZ et al., 2015).

Outro serviço bastante utilizado em grandes redes e em *data centers* é o balanceamento de carga. Esta função de otimização utiliza algoritmos de enfileiramento, tais como, *First In First Out* (FIFO), *Round Robin*, etc, para definir a política de distribuição de carga entre os servidores disponíveis. Nas redes atuais este serviço possui pouca flexibilidade ocasionado pelas limitações proprietárias, não podendo ser realizadas customizações na aplicação. Indo além, geralmente estes equipamentos apresentam alto custo para serem adquiridos pelas organizações. Entretanto, ao utilizar o paradigma SDN, tal funcionalidade pode operar definindo-se a granularidade de fluxo desejada (OLAYA; BERNAL; MEJIA, 2016).

O balanceamento de carga em SDN também facilita a instalação de novos serviços na rede. Isso pode ser observado na instalação de um novo servidor para contemplar esta funcionalidade, onde são realizadas ações para distribuir o tráfego de forma transparente entre os servidores disponíveis. Para isso, leva-se em conta a carga de rede e a capacidade computacional fornecida pelos servidores. Assim, os operadores de rede conseguem maior flexibilidade no provimento de novos serviços, maximizando a utilização da rede com mínima sobrecarga. Em consequência disso, o processo de gerenciamento de rede é simplificado (KREUTZ et al., 2015).

Ainda, com o paradigma SDN na implementação do balanceamento de carga, não é necessário ter um dispositivo físico separado para realizar tal funcionalidade, pois a mesma vai executar no controlador e o equilíbrio de carga pode ser realizado com a programação de determinada política de balanceamento desejada. Além do mais, caso a demanda do tráfego

seja elevada e a política definida de balanceamento não esteja atendendo satisfatoriamente o equilíbrio de carga, então pode-se rapidamente mudar a programação da aplicação e executar outra política de balanceamento (OLAYA; BERNAL; MEJIA, 2016).

Em relação as aplicações sensíveis ao atraso executadas em SDN, através da função de otimização *Quality of Service* (QoS) pode-se implementar e definir políticas, estabelecendo o melhor caminho e o menor tempo de atraso na rede. Funções de QoS podem ser usadas para definir, em determinados períodos ou de forma fixa, maior aprovisionamento de largura de banda para determinadas aplicações e serviços, como por exemplo, videoconferências, em detrimento de outros tráfegos. Estas políticas também podem ser ajustadas dinamicamente por meio de funções de monitoramento/controle, a exemplo do *traffic shapping* que permite o bloqueio de certas aplicações, *e.g.*, *torrent* e permissão de determinados fluxos em detrimento de outros que deseja-se ter maior prioridade no tráfego da rede (KREUTZ et al., 2015).

Outro cenário favorável para utilização de SDN é na conexão de uma rede banda larga doméstica. Com SDN, este tipo de conexão à Internet pode simplificar a adição e programação de novas funções dentro de um mesmo dispositivo físico, sendo perceptível a melhora do desempenho ao acesso a este tipo de conexão. Ainda, permite que o sistema seja sensível ao contexto e reaja dinamicamente às mudanças ocasionadas pelas condições de acesso à rede doméstica (KREUTZ et al., 2015).

No que tange as funcionalidades de proteção inerentes ao paradigma SDN, como exemplos de funções desta categoria, têm-se *firewall*, IDS, IPS, DPI. Estas funções que desempenham a proteção das redes possibilitam um bom comportamento na reação, prevenção e detecção contra tráfegos anômalos, maliciosos, como ataques de DoS/DDoS. Ainda, para que a proteção seja efetiva, é preciso definir políticas refinadas e técnicas de segurança inteligentes, onde aplicações mal intencionadas são bloqueadas antes de obter acesso as regiões críticas da rede, como por exemplo, ao controlador.

Neste contexto, a definição de uma segurança realmente efetiva possibilita: a) Coletar dados de diferentes origens; b) Convergir para uma configuração consistente com as funções de segurança; c) Definir medidas para bloquear ou minimizar o efeito de ataques (YAN et al., 2016) (KREUTZ et al., 2015).

Também ressalta-se a importância da interface de programação centralizada. A fim de que simplifica a integração de mecanismos de detecção e prevenção de ataques. Para que esta proteção seja possível de ser realizada, os dispositivos de encaminhamento *OpenFlow*, além

de realizarem o tráfego dos pacotes, também ficam responsáveis por coletar informações e dados estatísticos da rede. Posteriormente, estas informações são analisadas por algoritmos especializados implementados no controlador, cabendo a este a responsabilidade pela execução das aplicações de proteção a fim de mitigar possíveis ataques (KREUTZ et al., 2015) (YAN et al., 2016).

3.3 FUNÇÕES DE REDE EM NFV

Um exemplo de uso do paradigma NFV pelo *Internet Service Provider* (ISP) é na implementação das funcionalidades de rede em equipamentos domésticos para acesso à Internet. Nas redes domésticas tradicionais, um mesmo equipamento pode integrar funcionalidades de otimização (*switch*) e interoperabilidade (NAT). Além dessas, também pode contemplar as funcionalidades de *modem* e acesso à Internet via *wireless*, etc. Porém, nestas redes, além de serem proprietários, tais equipamentos são extremamente limitados e restritos a funcionalidades específicas já pré-programadas.

Além disso, quando há necessidade de fazer alterações ou procedimentos no equipamento que não são resolvidos contatando o suporte, é necessário um técnico do ISP ir até a casa do cliente para resolver o problema. Nesse caso, dependendo do diagnóstico, pode ser necessário a substituição total do equipamento, sendo oneroso para o ISP e, dependendo do tipo de contrato, também para os clientes (MIJUMBI et al., 2016).

Com NFV algumas destas funções podem ser transferidas para a infraestrutura compartilhada do ISP ou para grandes estruturas de *data centers*. Nesse caso, ao mover tais funções para *data centers*, operadores de redes precisam apenas fornecer dispositivos de baixo custo aos clientes para que os mesmos possam realizar a conectividade física, assim, diminuindo procedimentos e requisitos de manutenção (HAN et al., 2015). Os dispositivos precisam fornecer apenas funcionalidades de otimização (*switching*) e interoperabilidade (NAT) para acesso à Internet. Já as funções de aplicação (DHCP e DNS) podem ser movidas para as estruturas dos operadores de rede (MIJUMBI et al., 2016).

Dessa forma, possíveis mudanças seriam mais fáceis de serem realizadas. Por exemplo, para atualizar o servidor DHCP para todos os clientes, envolveria apenas mudanças diretas em tal função de aplicação executando no ISP. Da mesma forma, também pode ser necessário a inclusão de uma nova funcionalidade, para isso, bastaria o ISP selecionar todos ou um subconjunto de seus clientes e instalar essa nova função de uma única vez. Em contrapartida, nas

redes tradicionais, cada vez que é necessário realizar algum procedimento funcional, como por exemplo, adicionar, remover ou atualizar, o mesmo precisa ser feito individualmente em cada cliente (MIJUMBI et al., 2016).

Serviços de multimídia, como IPTV e conteúdo sob demanda, podem ser disponibilizados pelos provedores utilizando-se NFV. Em redes tradicionais, este tipo de serviço é mais complicado de ser oferecido com boa qualidade devido as funções de controle de fluxo interativo, onde é preciso entregar conteúdo em diferentes períodos de tempo. Em contrapartida, como NFV proporciona boa disponibilidade e escalabilidade no provimento de soluções, a virtualização da rede doméstica reduz a complexidade na entrega de conteúdos sob demanda. Dessa forma, com o uso da NFV, evidencia-se a oferta, com melhor qualidade, em serviços de *streaming* de vídeos e IPTV (MIJUMBI et al., 2016).

No que tange as funcionalidades de proteção aplicadas com o uso do paradigma NFV, arquiteturas virtualizadas são utilizadas para proteção contra diversos ataques. Por exemplo, em ataques de DoS/DDoS, o administrador da rede pode instanciar e/ou migrar sob demanda novas VMs executando a função de *firewall*. Isso tem por finalidade bloquear ou pelo menos mitigar o tráfego malicioso que tem como objetivo impedir o correto funcionamento dos serviços da rede em questão. Assim, com esta flexibilidade oferecida por NFV, garante-se a disponibilidade da rede e o fornecimento dos serviços (LAL; TALEB; DUTTA, 2017).

Além do mais, funções da categoria de proteção quando virtualizadas e baseadas em *software*, podem ser implantadas em qualquer lugar da rede, proporcionando capacidades de análises mais avançadas do tráfego, bem como disponibilizam mecanismos mais escaláveis (HERRERA; VEGA, 2016).

Seguindo com os benefícios da virtualização provida por NFV, neste paradigma a divisão dos recursos pode ser definida de acordo com a implementação de VNF. Dessa forma, utilizando-se a função de otimização de QoS executando como uma VNF, é possível realizar tal procedimento através da definição de políticas e algoritmos apropriados. A identificação dos recursos disponíveis é fundamental para posteriormente poder ser realizado a distribuição das cargas de trabalho pelo escalonador.

O escalonador também fica responsável pela gerência das tarefas e pela maximização da utilização dos recursos disponíveis. Dessa forma, ele obtêm maior previsibilidade do comportamento dos recursos, assim facilitando a atribuição de tarefas a estes de acordo com requisitos pré-estabelecidos. Com a função de otimização QoS, pode-se definir e implementar políticas.

Como exemplo, pode-se disponibilizar com maior frequência a CPU ao processamento de aplicações prioritárias. Com isso, mantém-se um nível adequado de QoS necessário para a execução e conclusão das tarefas (HERRERA; VEGA, 2016).

3.4 DISCUSSÃO

Devido a inexistência de uma taxonomia de funções de redes na literatura, a presente pesquisa também propôs uma taxonomia de algumas das principais funções de redes utilizadas atualmente na infraestrutura das redes de computadores. Através desta taxonomia foi possível categorizar as funções de rede em cinco áreas funcionais. Esta categorização foi realizada de acordo com as principais características, funcionalidades e responsabilidades em comum que tais funções de redes apresentam. As áreas funcionais identificadas foram: Monitoramento/Controle; Proteção; Otimização; Interoperabilidade e Aplicação.

Ainda, com a identificação de categorias de funções de rede de diferentes finalidades, realizou-se uma revisão do estado da arte. Esta revisão consistiu principalmente em mostrar como estas funções estão sendo utilizadas em soluções de redes reais. Também elencou-se os principais benefícios que estas propiciam quando são implementadas em SDN e NFV em detrimento das redes tradicionais.

Nesse contexto, nas redes tradicionais, o alto custo de soluções e dos equipamentos de redes, dificulta a aquisição por parte das organizações. Ainda, soluções proprietárias frequentemente possuem pouca flexibilidade ocasionado pelas limitações tanto a nível de *hardware* quanto de *software*, inviabilizando a realização de customizações nestas aplicações. Em contrapartida, através dos paradigmas SDN e NFV, funcionalidades de redes, como por exemplo, funções da categoria de proteção, quando virtualizadas e baseadas em *software*, podem ser implantadas em qualquer lugar da rede, proporcionando maior escalabilidade e capacidades de análises mais avançadas do tráfego da rede.

Nessa perspectiva, dentro da realidade de cada paradigma, ao utilizar SDN para a implementação destas funções, é possível realizar a programabilidade e orquestração destas por meio de *software*. Dessa forma, pode-se obter maior flexibilidade, capacidade de provisionamento e automatização das tarefas através de um controle logicamente centralizado, simplificando o gerenciamento e a administração da rede como um todo.

Com o paradigma NFV, obtém-se maior disponibilidade e escalabilidade no provimento e execução das funções de redes. Além do mais, conforme os exemplos descritos anteriormente

utilizando-se este paradigma em infraestruturas reais, com a virtualização destas funções em equipamentos genéricos, a utilização de dispositivos físicos dedicados não é mais necessária. Dessa maneira, com a adoção deste paradigma é proporcionada significativa diminuição de gastos de aquisição e manutenção de equipamentos de redes pelas organizações.

Também com a implementação de NFV, operadores de rede podem mover funcionalidades para grandes estruturas de *data centers*, sendo necessário apenas disponibilizar dispositivos de baixo custo aos clientes para que os mesmos possam realizar a conexão física. Com esta praticidade, atualizações em tais funcionalidades podem ser realizadas de uma única vez ao invés de precisar ser realizada individualmente em cada cliente como ocorre nas redes tradicionais.

4 DIMENSÕES

Com o objetivo de complementar as métricas relacionadas ao desempenho na avaliação de funções de rede nos paradigmas de SDN e NFV, nesta seção será realizada uma discussão de dimensões qualitativas inerentes aos dois paradigmas. A escolha dessas dimensões é muito importante para proporcionar uma análise mais profunda destes paradigmas.

Os dois paradigmas por serem complementares e baseados em *software*, possuem características significativas para realizar a correta implementação, configuração e execução dos mesmos. Tendo isto em vista, elencou-se aspectos relacionados a segurança, implementação/programabilidade, gerenciamento, gestão dinâmica dos recursos, disponibilidade e resiliência e desempenho sob um ponto de vista mais qualitativo.

4.1 SEGURANÇA

O paradigma SDN, através da separação entre o plano de controle e de dados, permite implementar *software* de elevado nível de abstração, como por exemplo, aplicações para gerenciar a rede sem se preocupar com configurações da infraestrutura de rede física subjacente (DABBAGH et al., 2015). Além disso, através da visão global da rede proporcionada pelo paradigma SDN, um administrador de rede pode obter um maior controle da rede, como por exemplo, maior capacidade de detecção de intrusão em toda a rede.

O controlador recebe as estatísticas de tráfego coletadas de todos os *switches*, dessa forma pode-se detectar completamente a presença ou não de tráfego malicioso na rede. Essa vantagem é bastante perceptível em relação as redes tradicionais onde o sistema de detecção de intrusão (IDS) é um dispositivo que geralmente é instalado em uma certa parte da rede, e portanto, fornece capacidade de detecção limitada devido não possuir uma ampla visibilidade da rede como SDN possui (DABBAGH et al., 2015).

Tratando-se de SDN, o protocolo *OpenFlow* possui várias vulnerabilidades presentes em sua base de implementação. Embora seja o protocolo que viabiliza a SDN, não possui mecanismos nativos para proporcionar uma autenticação segura da conexão de origem. Diante disso, as estações finais se autenticam a uma rede *OpenFlow* por meio da validação de seus endereços MAC e/ou IP. Dessa forma, o protocolo *OpenFlow* torna-se uma ameaça frequente nas redes SDN, pois ele oferece várias opções de manipulação de pacotes. O atacante pode,

por exemplo, analisar e definir ações nos pacotes, enviando múltiplos comandos destrutivos para os equipamentos compatíveis com este protocolo podendo interromper o trabalho de uma aplicação e de toda a rede (MOSTOVICH et al., 2017).

Nesse sentido, ameaças significativas podem ser exploradas pelas vulnerabilidades deste protocolo. O canal de comunicação entre o controlador e os elementos de encaminhamento, previa na especificação inicial do *OpenFlow* o uso obrigatório do protocolo de proteção TLS. Este protocolo tem como finalidade proporcionar uma segurança efetiva e atuar como um canal de controle na comunicação entre o controlador e os *switches OpenFlow* da rede. Contudo, em versões mais recentes do *OpenFlow*, o TLS passou a ser um requisito opcional. Neste caso, a não obrigatoriedade de utilização do protocolo TLS possibilita que os atacantes se infiltrem em redes *OpenFlow* sem serem detectados. Na ausência do protocolo TLS, o controlador não pode ter a garantia de que a tabela de fluxos dos equipamentos encaminhadores da rede esteja configurada de acordo com as regras esperadas (MOSTOVICH et al., 2017).

Ainda neste cenário, em SDN o plano de controle logicamente centralizado permite obter vários benefícios, tais como: programabilidade e visão global da rede. Entretanto, o controlador e as aplicações ficam mais vulneráveis a ameaças de diversos ataques maliciosos e a interceptação do tráfego, ou seja, uma vulnerabilidade em um nó controlador torna toda a rede desprotegida. Isso ocorre já que o plano de controle centralizado pode representar um único ponto de falha (MOSTOVICH et al., 2017).

Seguindo neste cenário, ataques de DoS/DDoS são bem explorados em SDN, pois quando este tipo de ataque é empregado em direção a este paradigma, é difícil de ser detectado imediatamente. Isso pode ocorrer devido a falta de relacionamento mais refinado na autenticação dos comutadores *OpenFlow* com o gerenciamento das tabelas de fluxos destes dispositivos pelo controlador da rede. Ataques deste tipo podem ocorrer quando comutadores da rede enviam uma grande quantidade de pacotes maliciosos para serem processados pelo controlador. Assim, este acaba sendo sobrecarregado e tem seus recursos de processamento esgotados (KREUTZ et al., 2015) (MATTOS; DUARTE, 2014) (HAKIRI et al., 2014).

Ainda, ataques como *Man In The Middle* também são direcionados à SDN, onde um atacante pode se infiltrar no plano de controle. Com acesso a este plano, o atacante pode alterar as configurações das tabelas de fluxos dos comutadores *OpenFlow* que estão sob gerência do controlador dessa rede (MATTOS; DUARTE, 2014) (HAKIRI et al., 2014).

No que tange o paradigma NFV, este possibilita reduzir o uso de *hardware* dedicado,

melhora a escalabilidade e reduz os custos de implementação. Com isto, permite-se atualizações fáceis, menor consumo de energia e manutenção reduzida (LAL; TALEB; DUTTA, 2017). Além disso, a natureza escalonável da NFV ajuda a melhorar o tempo de resposta aos incidentes, proporciona uma melhor capacidade de resposta frente a problemas de segurança. Nesse sentido, em um ataque de negação de serviço distribuído (DDoS), o administrador da rede pode instanciar sob demanda novas VM executando a funcionalidade de *firewall*, a fim de bloquear ou mitigar o tráfego malicioso o quanto antes possível, dessa forma garantido a disponibilidade da rede (LAL; TALEB; DUTTA, 2017).

Entretanto, uma significativa ameaça presente nesta arquitetura diz respeito ao compartilhamento da infraestrutura subjacente e dos recursos entre múltiplos hospedeiros. Através deste compartilhamento entre diversas VM, atacantes podem aproveitar possíveis falhas de segurança, ocasionadas por isolamentos impróprios entre os elementos virtuais da arquitetura NFV, ou seja, ameaças de segurança podem estar presentes no relacionamento entre os *hypervisors* e as VNF (LAL; TALEB; DUTTA, 2017).

Como NFV também é um paradigma onde os componentes e recursos são baseados em *software*, estes podem conter falhas de segurança (*bugs*), principalmente quando várias VNFs são encadeadas formando sistemas virtuais complexos (HAN et al., 2015). Assim, permite ao atacante explorar tais brechas e obter o acesso total a todos os recursos do domínio da rede invadida. Após obter acesso completo aos recursos da rede, os atacantes podem realizar atividades maliciosas, consumindo os recursos computacionais, tais como, CPU, memória, disco, até saturar o sistema de virtualização e consequentemente comprometer a infraestrutura computacional (ALJUHANI; ALHARBI, 2017).

4.2 IMPLEMENTAÇÃO/PROGRAMABILIDADE

A programabilidade de rede é baseada na dissociação entre o plano de controle e o plano de dados. O primeiro é responsável pela inteligência da rede e pela programação das tabelas de fluxos dos comutadores. Já o segundo, fica encarregado de encaminhar o tráfego da rede, de acordo com as decisões programadas pelo plano de controle (HU; HAO; BAO, 2014). Além disso, uma rede programável é aquela em que o comportamento dos dispositivos de rede é gerenciado através de *software*, o qual opera independentemente do *hardware* subjacente.

Dessa forma, obtêm-se muitas vantagens, tais como, custo reduzido, gerenciamento da rede de modo consistente e holístico, capacidade das aplicações em manter informações dos

dispositivos de rede, melhor alocação de largura de banda e utilização dos recursos, etc (KAUR; SINGH; GHUMMAN, 2014).

Ainda em relação a SDN, com o benefício da separação entre os planos, o programador desse paradigma não precisa se preocupar com os detalhes de baixo nível do *hardware* subjacente dos comutadores. Os programadores SDN podem apenas preocupar-se com a escrita e com o comportamento desejado das aplicações, abstraindo os eventos e as regras de encaminhamento da rede. Dessa forma, um *switch OpenFlow* pode assumir o comportamento de diferentes funcionalidades, como por exemplo, roteador, *switch, firewall*, NAT, etc (FEAMSTER; REXFORD; ZEGURA, 2014). Além do mais, com um nível de intelectualização adicional, os programadores também podem suportar e monitorar aplicações e componentes *OpenFlow* (HU; HAO; BAO, 2014).

Em redes tradicionais, administradores de rede geralmente precisam configurar dispositivos de rede individualmente de forma estática, sem dinamicidade ou flexibilidade. Também é necessário utilizar diferentes interfaces e sintaxes de configuração específicas de cada fabricante. Este modo de operação aumenta a complexidade, tempo dispendido para realizar a configuração e diminui a inovação nas redes. Dessa forma esta operação torna-se muito complexa, sendo também necessário profissionais bem qualificados para realizar a configuração destes dispositivos. Por outro lado, em SDN a configuração dos comutadores é muito mais fácil e ágil de ser realizada. Sendo assim, basta executar a aplicação no controlador que dessa forma a mesma será aplicada em todos os comutadores da rede. Assim, não é preciso realizar a configuração em cada dispositivo individualmente como nas redes tradicionais, consequentemente facilitando o trabalho do administrador de redes (FEAMSTER; REXFORD; ZEGURA, 2014).

Os dispositivos de rede atuais possuem pouca flexibilidade para lidar com diferentes tipos de dados, pacotes e conteúdos. Isso ocorre devido a inflexibilidade das implementações fechadas baseadas em *hardware* proprietário. As redes que compõem o *backbone* da Internet precisam se adaptar as mudanças de maneira que não necessitem de muitos ajustes a nível de *hardware* ou *software*. No entanto, operações nas redes tradicionais frequentemente necessitam de muitas alterações e portanto, não são fáceis de serem configuradas.

Com a programabilidade de redes, este problema pode ser sanado utilizando-se regras de gerenciamento de dados através da implementação de módulos de *software* ao invés de configurações a nível de *hardware*. Sob este ponto de vista, o gerenciamento de dados permite aos administradores ter maior controle sobre os diferentes conteúdos e pacotes que trafegam

na rede. Ainda, propicia alterar dinamicamente as tabelas de roteamento, assim, modificando o encaminhamento dos pacotes. Também possibilita uma camada extra de controle sobre os dados da rede, onde o administrador pode atribuir prioridades altas/baixas ou permitir/bloquear determinados pacotes que fluem pela rede (HU; HAO; BAO, 2014).

Outra importante constatação, é que através da programabilidade e da virtualização proporcionada pelas tecnologias SDN e NFV, é possível interconectar recursos virtuais com redes físicas. Os benefícios introduzidos por estes dois paradigmas são muito importantes, por exemplo, a partir desse momento reduz-se a intervenção humana para realizar a configuração e a interconexão dos equipamentos. Ainda, por serem paradigmas complementares, SDN fornece o controle e a programabilidade das aplicações. Já NFV proporciona que funções de rede possam ser virtualizadas e executadas em equipamentos de *hardware* genéricos (OMNES et al., 2015).

Além disso, para que o comportamento da rede possa a partir de agora ser definido por *software*, deve ser observado os níveis de abstrações necessários para a implementação de determinadas aplicações virtualizadas de rede. Dessa forma, os utilizadores destes dois paradigmas precisam projetar aplicações otimizadas para que estas possam oferecer um desempenho e confiabilidade que seja próximo aos níveis alcançados pelos equipamentos dedicados (WICKBOLDT et al., 2015).

Os operadores de rede são responsáveis pela configuração de políticas para responder a eventos e aplicações de rede, assim como precisam transformar essas políticas de alto nível em configurações de baixo nível, enquanto ocorre mudanças nas condições da rede. Para realizar essas tarefas complexas, eles dispõem de ferramentas muito limitadas. Devido a estas questões, o gerenciamento e a configuração de redes tradicionais tornam-se altamente propensa a erros (NUNES et al., 2014). Neste sentido, soluções SDN e NFV trabalhando integradas, possibilitam que configurações possam ser automatizadas, assim minimizando a realização de tarefas repetitivas, consequentemente otimizando o trabalho dos administradores de rede (VENKATRAMAN et al., 2014) (MIJUMBI et al., 2016).

Ainda, com esta integração, NFV será responsável pela inteligência e pelo processamento da rede. Já SDN ficará encarregada com a parte de infraestrutura física, isto é, pelos equipamentos encaminhadores da rede. Nesse contexto, percebe-se que nos aspectos inerentes a programabilidade, as duas tecnologias também são altamente complementares, onde SDN fornece todas as abstrações a nível de *software* e NFV provê significativa capacidade de virtu-

alização da infraestrutura subjacente. Assim, considerando-se estas características, tanto SDN quanto NFV, através do controle a nível de *software*, simplificam e reduzem o custo de operação e manutenção da rede.

4.3 GERENCIAMENTO

A inflexibilidade do *software* e do *hardware* dos equipamentos proprietários instalados nas redes tradicionais, faz com que a introdução de novos recursos e o controle das mesmas seja extremamente difícil. Entretanto, com o aumento da flexibilidade, facilidade de gerência e programabilidade do paradigma SDN, torna-se muito mais fácil introduzir novas funcionalidades de redes nesses ambientes (KIM; FEAMSTER, 2013).

No que tange aos aspectos de gerenciamento, NFV também possui alguns desafios para serem resolvidos, como a posição dos Pontos de Presença (PoP). Os PoP da rede, no paradigma NFV, representam os equipamentos genéricos onde as funções de rede são executadas. As VNF que fornecem serviços ao clientes, podem ser espalhadas por diferentes servidores. Sendo assim, a dificuldade está em manter um nível aceitável de orquestração a fim de garantir que todas as VNF necessárias sejam instanciadas de forma coerente e sob demanda. Outro desafio, relacionado ao anterior, a ser sanado está relacionado a localização das VNFs na composição de um serviço, pois na implantação dessas, devem ser tomadas decisões onde as mesmas devem ser posicionadas entre os PoP disponíveis (MIJUMBI et al., 2016).

Também tem-se desafios de gerenciamento em SDN, tais como, o número de controladores presentes na rede, localização destes, a fim de evitar conflitos em casos onde mais de um controlador fique responsável por gerenciar um dado elemento de encaminhamento. O gerenciamento em SDN pode ser centralizado, distribuído ou ainda híbrido. O plano de controle centralizado apresenta apenas um único ponto de gestão e melhor controle sobre o estado da rede, porém isso incorre em algumas limitações. Nesse contexto, o controlador precisa atualizar mais frequentemente os comutadores para obter a visão global da rede e realizar a descoberta da topologia, gerando maior sobrecarga e aumentando o tempo de resposta do tráfego da rede (HAKIRI et al., 2014).

A simplicidade do modelo centralizado contrapõem-se ao custo da escalabilidade do plano de controle. Isso ocorre, pois todas as funcionalidades inseridas em um único nó, requerem maior poder de computação, armazenamento e vazão causando uma degradação no tempo de resposta na entrega dos pacotes. Além do mais, esse maior atraso também decorre deste

modelo centralizado, haja vista que o primeiro pacote de cada novo fluxo que é introduzido no sistema deve primeiro ser encaminhado para o controlador para ser inspecionado antes de trafegar pela rede. Assim, quando um novo fluxo deve ser programado, o controlador atualiza os estados da tabela de encaminhamento dos *switches OpenFlow* que estão inseridos na estrutura da rede. Este processo tem como consequência a possibilidade de falhas na rede, devido ao grande número de novos fluxos que foram programados, consequentemente também observa-se uma latência extra na rede (HAKIRI et al., 2014).

Já o modelo de gerenciamento SDN distribuído, tem como objetivo eliminar o único ponto de falha. O plano de controle distribuído, possibilita que as instâncias dos controladores compartilhem um grande volume de informações, dividindo a carga entre todos os controladores, consequentemente maximizando a escalabilidade da rede. Ademais, este modelo é frequentemente utilizado em grandes redes e infraestruturas de *data centers*, garantindo a consistência em toda a rede e um controle de granularidade mais refinado. Também salienta-se que a gerência da arquitetura SDN distribuída é facilmente capaz de se adaptar às aplicações e requisitos dos usuários.

O gerenciamento distribuído torna-se mais responsivo, robusto e pode reagir de forma mais rápida e eficiente aos eventos gerados da rede. Por outro lado, utilizando-se este modelo de arquitetura, o processo de depuração, resolução de problemas, monitoramento e cumprimento dos requisitos de segurança, são mais difíceis de serem controlados. Isso acontece pois diagnosticar possíveis gargalos sem possuir a visão global da rede e manter a consistência da estrutura computacional, torna-se uma tarefa operacional mais complexa. (HAKIRI et al., 2014).

Por fim, ainda existe mais uma possibilidade de gerenciamento em SDN, trata-se do modelo de gerência híbrido. Neste modelo, as funções de controle estão divididas entre o plano de controle e dados. Sendo assim, as funções de controle, como, a visão global da rede, gerenciamento de políticas e aprovisionamento de largura de banda estão concentradas no controlador SDN logicamente centralizado. Já as funções de controle que obtêm maiores benefícios sendo distribuídas pela rede, como, recuperação de falhas, monitoramento e segurança, continuam incorporadas nos elementos encaminhadores da rede. Dessa forma este modelo agrega os benefícios do controle simples e da visão global da rede fornecido pelo gerenciamento centralizado juntamente com a escalabilidade e resiliência do distribuído. Com estes benefícios, pode-se melhorar o desempenho da rede ao permitir o uso mais inteligente e eficiente dos recursos, possibilitando o ajuste fino e automatizando os aspectos da rede no nível de aplicação.

O modelo híbrido fornece políticas de gerenciamento para sanar a sincronização dos estados, com isso garantindo a estabilidade da rede. Também preocupa-se com questões relacionadas a sobrecarga do plano de controle, tratando de mitigar possíveis vulnerabilidades e ameaças de segurança. Indo além, este modelo permite a migração não disruptiva, atualizando a infraestrutura computacional sem a necessidade de alterar e interromper o processamento normal do sistema (HAKIRI et al., 2014).

4.4 GESTÃO DINÂMICA DOS RECURSOS

Em SDN e em NFV é adicionado um nível de flexibilidade que facilita a programação e o gerenciamento escalável da rede. Ainda, SDN pode agregar valor para NFV permitindo conectividade dinâmica através da programabilidade da rede com base no monitoramento e na análise do tráfego. Nesse sentido, NFV também proporciona benefícios à SDN, pois permite virtualizar os controladores SDN e dessa forma obter mobilidade dos mesmos para serem alocados rapidamente nos locais desejados da rede. Ambos os paradigmas permitem a interconexão de serviços e o agendamento/aprovisionamento de largura de banda necessários para melhor gerenciamento e automatização das redes (LAL; TALEB; DUTTA, 2017).

A dinamicidade citada é imprescindível já que muitas vezes os controladores podem tornar-se gargalos na operação da rede. Isso acontece devido ao aumento do tamanho da rede, onde muitos eventos e solicitações são enviados ao controlador, e este em algum momento não consegue mais controlar todas estas requisições de entradas, ou seja, frequentemente os controladores são programados para que os fluxos maiores sejam direcionados diretamente ao plano de encaminhamento. Já os fluxos menores podem ser enviados ao controlador, assim, reduzindo efetivamente a sobrecarga no controlador e melhorando a escalabilidade da rede (YEGANEH; TOOTOONCHIAN; GANJALI, 2013).

Em SDN, a dinamicidade fornecida pelo plano de controle provê a necessidade de requisitos escaláveis, onde aspectos de convergência e requisitos de consistência são inerentes a este paradigma. Um exemplo do uso desta tecnologia poderia ser na alocação de largura de banda para um grande evento com o objetivo de melhorar a transmissão deste, uma vez que o número de telespectadores assistindo ao evento aumenta substancialmente.

Já NFV, pode atender estes requisitos, fornecendo ambientes de processamento independentes e isolados, onde VMs podem ser instanciadas e/ou removidas sob demanda, dimensionadas para combinar com as constantes mudanças do tráfego. Para alcançar todos os benefícios, é

preciso configurar a rede de forma escalável e responsiva para atender as demandas que utilizam soluções deste paradigma. Como exemplo, embora uma única VM possa não ser capaz de satisfazer plenamente os requisitos de uma determinada função, pode-se tornar inviável implantar uma VM por NFV, pois isso resultaria em uma sobrecarga muito grande e poderia ocasionar problemas de escalabilidade na camada de virtualização. Além do mais, essa abordagem poderia se tornar um desperdício de recursos na execução de funções mais simples, como a de um servidor DHCP executando em uma rede doméstica, que não justificaria apenas uma VM dedicada a executar essa função (LAL; TALEB; DUTTA, 2017).

Ainda, em NFV, devido ao desacoplamento do *software* do *hardware*, funções de rede não necessitam mais ser implementadas em dispositivos dedicados. Isso justifica-se pela agilidade provida pela virtualização, onde funcionalidades de redes podem ser instaladas e executadas em equipamentos genéricos, pois como o *software* leva menos tempo para revisar e frequentemente é mais fácil para se modificar do que o *hardware*. Sendo assim, os operadores podem reagir de forma ágil e flexível às mudanças necessárias no ambiente de rede e também para atender às demandas dinâmicas dos usuários. Indo além, NFV deverá suportar grandes volumes de dados e a instanciação de novas funcionalidades e nesse caso, será preciso prover a escalabilidade necessária aos ambientes virtuais buscando sempre alcançar níveis ótimos de desempenho (KIM; KOO; PAIK, 2015).

4.5 DISPONIBILIDADE E RESILIÊNCIA

Nos paradigmas SDN e NFV, questões relacionadas a disponibilidade e resiliência da rede utilizando-se estes paradigmas são muito importantes para manter a rede operacional após o surgimento de possíveis falhas na infraestrutura das mesmas. Em SDN, o plano de controle é responsável pela lógica de encaminhamento dos pacotes pelos comutadores, e as notificações de falhas em enlaces da rede são enviadas diretamente ao controlador, o qual também fica encarregado por realizar a detecção e resolução de falhas do plano de encaminhamento, bem como atualizar as regras das tabelas de fluxos (LAL; TALEB; DUTTA, 2017).

O paradigma NFV pode trabalhar de forma integrada com SDN melhorando a resiliência da rede, assim como as VNF podem ser instanciadas sob demanda para resolver anomalias específicas da rede. Dessa maneira, uma VNF pode ser substituída por outra VNF quando anomalias não forem resolvidas corretamente. Como exemplo, um IDS/IPS pode ser substituído por um *firewall* se aquele não estiver sendo efetivo contra um ataque de varredura de porta

(MACHADO; GRANVILLE; SCHAEFFER-FILHO, 2016).

Em relação a NFV, para manter a resiliência da rede utilizando-se deste paradigma, em caso de falhas pode-se instanciar novas NF. Este processo pode ocorrer de forma manual ou automatizada de acordo com os requisitos da NF que deseja-se manter operacional. Ainda, é recomendado a replicação de dados e informações destas funcionalidades a fim de garantir a integridade e o desempenho das NF. Também é interessante, como mecanismo de proteção, não armazenar as funções que desempenham o mesmo papel nos mesmos recursos físicos, isto é, em um mesmo domínio (MIJUMBI et al., 2016). Indo além, também é necessário a definição de políticas e níveis de confiabilidade e disponibilidade para os serviços virtuais (MACHADO; GRANVILLE; SCHAEFFER-FILHO, 2016).

Ainda, em relação a NFV, também pode-se manter-se a rede resiliente através de estratégias, como resiliência do *link* e de VNF. Na primeira, é gerada uma comunicação adicional entre os caminhos entre as instâncias das VNFs alocadas, que podem ser utilizadas como recursos de retorno, em caso de falha do caminho primário original. Já na segunda, quando um nó físico falhar, instâncias de VNFs hospedadas nesse nó precisam ser migradas para nós diferentes fornecendo capacidade de processamento suficientes (BECK; BOTERO; SAMELIN, 2016).

No que tange aos aspectos de disponibilidade e resiliência em SDN, tanto o plano de controle quanto o plano de dados podem falhar, deixando a rede inoperante. Dentro deste contexto, falhas no plano de encaminhamento, como por exemplo, quedas ou rupturas de enlaces, devem ser detectadas e recuperadas pelo plano de controle. Isso ocorre pois caso a falha prejudique a comunicação entre os controladores e os comutadores, a operação da rede fica indisponível. Dessa forma, uma rede SDN deve ser planejada com tolerância a falhas nas entidades que compõe seus dois planos. Também deve-se observar e avaliar os aspectos de resiliência e as características inerentes a sua arquitetura, buscando sempre manter o pleno funcionamento e a correta interação entre os dois planos (MONTIBELER; FARIAS; ABELEM, 2017).

A eficiência da comunicação entre os planos é fundamental no tempo de recuperação de falhas da rede. Isso acontece pois o tempo de resposta do plano de controle é afetado pela latência introduzida na comunicação entre controlador e comutador. Nesse sentido, mesmo quando vários controladores são empregados no plano de controle, estes devem ser capazes de se comunicar para cooperar nas tomadas de decisões e agilizar a sincronização entre suas diferentes instâncias (MONTIBELER; FARIAS; ABELEM, 2017).

Diante dessa perspectiva, em SDN apenas uma instância de controlador na rede pode ser suficiente para a sua operação. Entretanto, com essa configuração tem-se apenas um único ponto de falha. Sendo assim, a falha desse dispositivo deixará a rede sem plano de controle. Portanto, para aumentar a tolerância a falhas no plano de controle logicamente centralizado, faz-se necessário que uma SDN possua múltiplos controladores em operação na rede, ou seja, possibilitar uma comunicação tolerante a falhas entre todos os dispositivos da rede SDN, independente de seu papel desempenhado, torna-se um quesito muito importante em ambientes programáveis (MONTIBELER; FARIAS; ABELEM, 2017).

Ao considerar num contexto de SDN, um plano de controle descentralizado, também é possível fornecer maiores garantias de resiliência, pois nesse tipo de arquitetura, devido a presença de controladores distribuídos, não encontra-se apenas um único ponto de falha na rede. Nesse contexto, cada controlador é responsável por um domínio da rede, que corresponde a um conjunto de *switches OpenFlow* do plano de dados (OBADIA et al., 2014).

Também é possível utilizar-se mecanismos de *failover* para migrar comutadores para outros controladores. Isso pode ser realizado quando controladores inoperantes tem comutadores alocados a seu domínio. Dessa forma, esta migração ocorre para outro controlador ativo, o qual ficará responsável, por determinado período, pelo plano de dados.

Outra possibilidade é utilizar algoritmos onde os controladores são programados para progressivamente gerenciar os *switches* que não possuem controladores em seu domínio. Também é possível ter situações que, em caso de falhas dos controladores, estes indiquem, através de uma ação programada, de forma proativa, os controladores vizinhos, solicitando para eles assumirem os comutadores da rede que estavam sob o seu domínio (OBADIA et al., 2014).

4.6 DESEMPENHO

A fim de verificar o desempenho nos paradigmas SDN e NFV, métricas quantitativas e qualitativas podem ser analisadas e aferidas de acordo com diferentes parâmetros e aspectos. Nesse sentido, estas métricas podem mostrar como está o desempenho de determinada função ou solução de rede (KIM; KOO; PAIK, 2015). As métricas quantitativas podem ser: *delay, throughput, jitter*, perda de pacotes, etc. Além disso, em SDN podem ser utilizadas outras métricas, tais como: tempo de descoberta da topologia da rede, tempo de provisionamento do caminho, tempo de detecção quando ocorre mudança na topologia da rede, etc. Já em NFV, tem-se as métricas: velocidade da placa de rede virtual, capacidade de processamento, tamanho

de memória da VM, tamanho do pacote, utilização de uma placa virtual por VM ou múltiplas VM compartilhando uma mesma placa virtual (KIM; KOO; PAIK, 2015).

Além dessas métricas, o projeto e a capacidade da infraestrutura subjacente de SDN influencia no desempenho de tarefas comuns de redes em comparação com soluções dedicadas, pois a infraestrutura deste paradigma pode ser ajustada para as aplicações específicas requeridas pelos usuários. Sendo assim, por meio do plano de controle é melhorado o nível de desempenho da rede e a experiência do usuário como um todo (GELBERGER; YEMINI; GILADI, 2013).

Já no paradigma NFV, ao implementar instâncias de VNF, é possível dividir a carga da rede com outras VM, buscando assim manter os requisitos de latência. Ademais, a infraestrutura NFV subjacente, deve ser capaz de reunir informações de desempenho da rede em diferentes níveis, como por exemplo, nos *hypervisors* (HAN et al., 2015).

No que tange o desempenho de NFV, um dos principais desafios está em garantir um desempenho comparável com as funções executadas em dispositivos especializados. Além disso, também é desejável que as VNF sejam portáteis entre os servidores de uso geral. Obter bom desempenho em altas velocidades não é um desafio somente de funções virtualizadas, já que também é de funções não virtualizadas. Diante disso, técnicas de aceleração de *hardware* mostram-se importante para NFV, já que melhoram o desempenho de algumas VNFs. Nesse contexto, soluções foram desenvolvidas para alavancar melhorias no desempenho de instâncias virtualizadas. Como por exemplo, a Intel lançou o *Data Plane Development Kit* (DPDK), cujo principal objetivo é melhorar a velocidade de comunicação entre as VMs, e entre estas e a placa de rede. Além disso, o DPDK também fornece um ambiente simples e completo que suporta o processamento rápido de pacotes para aplicações que necessitam de alto desempenho (HAN et al., 2015) (ROSA et al., 2014).

Apesar de SDN possuir muitos benefícios no que tange ao desempenho, a mesma também apresenta algumas desvantagens, como por exemplo, uma vez que um controlador logicamente centralizado é usado para programar toda a rede, é preciso ter uma visão global sobre a carga em cada comutador do caminho. Ainda, o controlador se comunica com os comutadores, através do protocolo *OpenFlow*, onde este é responsável por coletar estatísticas, erros, falhas de cada dispositivo da rede e reportar ao plano de controle. Neste plano está o controlador, o qual pode executar algoritmos analíticos a fim de detectar as sobrecargas e prever quando as mesmas possam ocorrer na rede futuramente (HAKIRI et al., 2014). Também observa-se que a adição de flexibilidade e inserção de novas funcionalidades exigem sobrecarga adicional sobre os equi-

pamentos. Como resultado, o desempenho em termos de vazão e velocidade de processamento podem ser minimizados (GELBERGER; YEMINI; GILADI, 2013).

Seguindo nesta perspectiva, com a adoção de SDN, alguns fatores críticos devem ser levados em consideração a fim de lidar com o intenso tráfego inerentes a estes ambientes. Tais fatores dizem respeito a quantidade de fluxos que o controlador pode processar por segundo e aos elementos encaminhadores programados em *software*. Estes afetam sensivelmente a latência e o desempenho das aplicações. Por fim, outro fator importante refere-se ao gargalo de desempenho introduzido na comunicação entre os comutadores e o controlador da rede (ALIYU; BULL; ABDALLAH, 2017).

5 ANÁLISE DAS CATEGORIAS E DIMENSÕES

O objetivo deste capítulo é fazer o relacionamento das categorias de funções de rede, elencadas na taxonomia apresentada no capítulo 3, pelas dimensões qualitativas, discorridas no capítulo 4. Além disso, também tem por finalidade, identificar qual dos paradigmas é mais adequado para ser implementada e executada funções de rede pertencentes as categorias mapeadas. Destaca-se que a dimensão de desempenho será avaliada em detalhes no capítulo 7.

Para o preenchimento da tabela utilizou-se como base pesquisas e trabalhos disponíveis na literatura. Também levou-se em consideração as dificuldades e os desafios enfrentados para a realização do presente trabalho, em especial nas questões inerentes a implementação das funções de rede nos paradigmas SDN e NFV. Além do mais, para que a tabela proposta pudesse alcançar melhores índices de mensuração e avaliação, definiu-se níveis de aplicabilidade para os dois paradigmas avaliados. Tais níveis são apresentados a seguir:

a) Não Aplicável (0): Ocorre quando os paradigmas SDN e NFV não são usados para a implementação de funções de rede de determinada categoria;

b) Fracamente Aplicável (1): Ocorre quando a implementação de funções de rede de determinada categoria nos paradigmas SDN e NFV traz mais desvantagens do que vantagens;

c) Aplicável (2): Ocorre quando a implementação de funções de rede de determinada categoria nos paradigmas SDN e NFV as vantagens e desvantagens são iguais;

d) Fortemente Aplicável (3): Ocorre quando a implementação de funções de rede de determinada categoria nos paradigmas SDN e NFV traz mais vantagens do que desvantagens.

A seguir será apresentado uma discussão das dimensões dentro de cada categoria das funções de rede que compõem a taxonomia proposta.

5.1 APLICAÇÃO

No que tange a dimensão de segurança, devido a várias aplicações em SDN executarem de forma logicamente centralizada, ocorre um único ponto de falha, proporcionando diversos ataques maliciosos em funções de rede executando neste paradigma (MOSTOVICH et al., 2017). Em relação ao paradigma NFV, a segurança de funções de rede é frequentemente realizada através do isolamento entre o sistema de virtualização e as funções que deseja-se executar neste paradigma. Para tanto, é preciso que o sistema como um todo esteja realmente bem con-

figurado e com políticas de segurança bem definidas, assim, proporcionando maiores e efetivos níveis de proteção a este paradigma (LAL; TALEB; DUTTA, 2017).

Para a dimensão de implementação/programabilidade, utilizando-se o paradigma NFV, percebe-se que é preciso um menor nível de complexidade, abstração e codificação, para a implementação de algumas funções desta categoria, como por exemplo, de um servidor DHCP. Para o aspecto de gerenciamento, observa-se que os dois paradigmas possuem comportamentos semelhantes, facilitando o trabalho do administrador de redes no monitoramento e interpretação da operacionalização destas funcionalidades nestes dois paradigmas. Nesse sentido, pode-se definir mesmos níveis de aplicabilidade em ambos.

Quanto aos aspectos de gestão dinâmica dos recursos, tais como flexibilidade e escalabilidade, também observa-se bom relacionamento e níveis de aplicabilidade iguais nos dois paradigmas. Para as características de disponibilidade e resiliência, as mesmas podem ser consideradas aplicáveis. Isso ocorre devido a dependência significativa da programabilidade definida nas funções, em relação ao comportamento que as mesmas devem seguir após a ocorrência de possíveis falhas e/ou ataques contra a infraestrutura computacional.

5.2 INTEROPERABILIDADE

Em relação as funções de interoperabilidade, a implementação destas funções ainda é pouco explorada na literatura envolvendo os paradigmas SDN e NFV em comparação com funções de outras categorias. Nesse sentido, a caracterização de aspectos, tais como, segurança, gerenciamento, disponibilidade e resiliência torna-se mais difícil de ser corretamente mensurada. Dessa forma, tais aspectos foram classificados como sendo fracamente aplicáveis.

Ainda, a carência de uma abordagem mais prática para poder fundamentar e solidificar a implementação de funções desta categoria, principalmente no paradigma NFV, dificulta uma avaliação mais concreta de funcionalidades desta categoria. Para esta categoria implementou-se a função de NAT em SDN. Nesse contexto, até o presente momento, percebe-se que na literatura o paradigma SDN contempla uma abordagem um pouco mais inerente a estas funções, agregando melhor dinamicidade e flexibilidade dos recursos.

5.3 OTIMIZAÇÃO

Na categoria de otimização, em relação ao aspecto de segurança, observa-se na literatura uma maior preocupação com funções implementadas em SDN. Como um dos motivos principais, frequentemente está ligado ao fato das aplicações estarem, em boa parte das vezes executando de forma logicamente centralizada, representando um único ponto de falha (MOSTOVICH et al., 2017). Dessa forma, as mesmas ficam mais vulneráveis as ameaças, como por exemplo, em uma função de *switching*, o atacante pode modificar ou desabilitar a funcionalidade de *Spanning Tree Protocol* (STP), assim introduzindo ciclos fechados "*loops*" no domínio da rede e ocasionando o congestionamento desta. Nesse sentido, os dois paradigmas podem ser utilizados em funções de otimização, porém, entende-se que o paradigma NFV seja o mais adequado.

Para o aspecto de implementação/programabilidade, é possível observar na literatura mais exemplos práticos do paradigma SDN, assim, possibilitando maior embasamento para a implementação de funções desta categoria. Nesse contexto, além da funcionalidade de *switching*, o qual foi implementada e avaliada nas duas tecnologias, conseguiu-se em SDN também realizar a implementação de um balanceador de carga utilizando-se o algoritmo de enfileiramento *Round Robin* como política do equilíbrio de carga.

Em relação aos aspectos de gerenciamento, gestão dinâmica dos recursos, disponibilidade e resiliência, observou-se que funções desta categoria possuem comportamentos e características semelhantes quando avaliadas sob estes aspectos. Por exemplo, para a função de *switching* implementada em *software* e virtualizada, a inteligência e o gerenciamento de tal função passa a ser responsabilidade da programação a nível de *software*, onde nesta tecnologia o *hardware* fica responsável apenas pelo encaminhamento dos pacotes na rede. Dessa forma, o comportamento da aplicação torna-se mais refinado, possibilitando maior previsibilidade e controle do administrador de rede na execução destas funções em ambos os paradigmas (LAL; TALEB; DUTTA, 2017).

5.4 PROTEÇÃO

Na dimensão de segurança, devido ao significativo poder de escalabilidade inerente ao paradigma NFV, este possibilita uma ação mais rápida e eficiente na resolução de incidentes de segurança. Além disso, como NFV não está exposta a presença de um controle logicamente

centralizado que oferece um único ponto de falha, ao empregá-la, obtêm-se maiores garantias de disponibilidade da rede. Em relação aos aspectos de programabilidade e gerenciamento, ambos os paradigmas apresentam características semelhantes (MIJUMBI et al., 2016) (HERRERA; VEGA, 2016).

Para a implementação/programabilidade de funções de proteção, como por exemplo, de um *firewall*, encontram-se na literatura trabalhos utilizando-se os dois paradigmas. Além da parte científica e teórica, também é possível encontrar alguns exemplos e abordagens mais práticas que podem auxiliar na implementação de funções desta categoria.

No que diz respeito aos aspectos de gestão dinâmica dos recursos, tais como, flexibilidade e escalabilidade, bem como as questões inerentes a disponibilidade e resiliência, utilizando-se NFV, pode-se instanciar sob demanda novas funcionalidades de proteção. Nesse contexto, a qualquer momento pode-se instanciar funcionalidades desta categoria, como por exemplo, IDS/IPS e *firewall*, a fim de sanar ou mitigar prováveis anomalias que venham a ocorrer na rede (MACHADO; GRANVILLE; SCHAEFFER-FILHO, 2016).

No paradigma SDN, manter a disponibilidade e a resiliência da rede torna-se uma tarefa mais complexa. Tendo isto em vista, além do comportamento definido pela programabilidade da rede no plano de controle, o administrador também precisa se preocupar com os elementos encaminhadores do plano de dados. Isso envolve dentre outros aspectos, o nível de segurança do protocolo *OpenFlow* na comunicação entre o controlador e os *switches OpenFlow*. (MONTIBELER; FARIAS; ABELEM, 2017).

Nesse cenário, funções desta importante categoria desempenham um papel essencial na proteção das organizações. Sendo assim, sabe-se que a adição de novos recursos, buscando maximizar a flexibilidade e escalabilidade nestes paradigmas, incorre em sobrecargas adicionais. Estas são geradas nos equipamentos responsáveis pelo encaminhamento de pacotes na rede (GELBERGER; YEMINI; GILADI, 2013).

5.5 MONITORAMENTO/CONTROLE

Para a categoria de monitoramento/controle, implementou-se e analisou-se a função de roteamento. Na literatura, pode-se encontrar a implementação dessa função trabalhando de forma integrada com os paradigmas NFV e SDN. O primeiro fica responsável pela inteligência do roteamento dos pacotes e o segundo fornece a infraestrutura computacional através do protocolo *OpenFlow* (BATALLE et al., 2013).

Para as questões inerentes ao aspecto de segurança desta categoria, SDN apresenta mais chances de ocorrerem possíveis falhas de segurança. Nesse sentido, atacantes podem explorar as fragilidades expostas pelo controle logicamente centralizado (MOSTOVICH et al., 2017). Assim, os mesmos podem obter acesso total ao controlador e dessa forma alterar as tabelas de roteamento. Dessa maneira, significativas falhas de comunicação podem ser ocasionadas entre os dispositivos que compõe o domínio de rede. (MATTOS; DUARTE, 2014) (HAKIRI et al., 2014).

Em relação aos aspectos de implementação/programabilidade, SDN e NFV apresentam vantagens e desvantagens, sendo fortemente aplicáveis para essa dimensão. Nesse contexto, constatou-se na implementação da função de roteamento, que em NFV é preciso menos esforço de codificação. Além disso, em NFV também necessita-se de menor nível de abstração para o entendimento da lógica de funcionamento do código desta função. Em contrapartida, na literatura e na comunidade acadêmica encontram-se disponíveis mais exemplos práticos em SDN para fundamentar e guiar a implementação de tal funcionalidade.

É interessante ressaltar que para realizar o funcionamento desta função implementada em NFV, considerando o cenário desenvolvido e as ferramentas utilizadas, o administrador de rede precisa estar atento para algumas questões. Estas questões estão relacionadas a configuração dos endereços MAC das *bridges* da VM executora da função virtual no *script* de configuração da função de roteamento.

No que tange as dimensões de gerenciamento, gestão dinâmica dos recursos, disponibilidade e resiliência, observou-se comportamentos e características semelhantes dos dois paradigmas, sendo fortemente aplicáveis. Nesse contexto, os dois paradigmas oferecem bom poder de flexibilidade, escalabilidade e disponibilidade. Isso pode ser observado na função de roteamento, onde os dois paradigmas podem ser utilizados em conjunto para a implementação desta funcionalidade (BATALLE et al., 2013).

Ainda, diferentemente das redes tradicionais que encontram-se engessadas, como SDN e NFV são paradigmas baseados em *software* e virtualizados em equipamentos genéricos, pode-se facilmente migrar e instanciar novas VMs com este serviço. Isso pode ser realizado sem alterar as tabelas de roteamento dos elementos encaminhadores da rede (BATALLE et al., 2013).

A seguir apresenta-se na Tabela 1 os níveis de aplicabilidade dos paradigmas SDN e NFV entre as categorias e as dimensões. Ainda, esta tabela tem a finalidade de representar o cruzamento entre as categorias de rede, elencadas na taxonomia proposta apresentada no

capítulo 3, e as dimensões qualitativas, discorridas no capítulo 4.

Tabela 1 – Tabela das Categorias de Funções x Dimensões.

Categorias/ Dimensões	Segurança	Implementação/ Programabilidade	Gerenciamento	Gestão Dinâmica dos Recursos	Disponibilidade/ Resiliência
Aplicação	SDN:2 NFV:3	SDN:2 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:2
Interoperabilidade	SDN:1 NFV:1	SDN:2 NFV:1	SDN:1 NFV:1	SDN:2 NFV:1	SDN:1 NFV:1
Otimização	SDN:2 NFV:3	SDN:3 NFV:2	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:2
Proteção	SDN:2 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:3	SDN:2 NFV:3
Monitoramento/ Controle	SDN:2 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3

Fonte: Autoria Própria.

5.6 DISCUSSÃO

Analisando-se os resultados gerais da tabela das categorias de funções pelas dimensões, observa-se que os paradigmas SDN e NFV possuem no máximo um nível de diferença. Nesse sentido, percebe-se que não existe um único paradigma exclusivo para a implementação de funções de rede de determinada categoria. Indo além, ambos os paradigmas possuem vantagens e desvantagens, onde em algumas dimensões SDN acaba sendo melhor avaliada e em outras NFV mostra-se mais vantajosa. Dessa maneira, percebe-se que na média ambos os paradigmas se equiparam devido as semelhanças existentes em muitas de suas características.

Ressalta-se que para a categoria de interoperabilidade até o presente momento não encontram-se muitos trabalhos teóricos e práticos disponíveis na literatura. Nesse sentido, é bastante difícil encontrar trabalhos que investiguem questões envolvendo técnicas e mecanismos de tunelamento e tradução a fim de permitir a comunicação entre endereços IPv4 e IPv6.

Continuando, em interoperabilidade também constata-se que a tarefa de gerenciamento apresenta-se bastante onerosa e complexa de ser realizada em ambos os paradigmas. Entretanto, analisa-se que para todas as outras categorias, a gerência em ambos os paradigmas mostra-se fortemente aplicável. Justifica-se essa diferença já que para esta categoria existe uma carência

de abordagens práticas na literatura, principalmente em NFV, para fundamentar a programabilidade de funções de rede desta categoria.

Nesta perspectiva, destaca-se que para a categoria de interoperabilidade e otimização, as funções de NAT e balanceamento de carga, respectivamente, não foram implementadas em NFV devido a considerável limitação de embasamento prático. Além do mais, no que tange os aspectos de implementação, notou-se significativas especificidades da linguagem de programação do *Click Modular Router*, assim prejudicando o desenvolvimento e consequentemente o funcionamento destas funções. Sendo assim, entende-se que para a categoria de interoperabilidade as questões inerentes, principalmente, aos aspectos de segurança, gerenciamento, disponibilidade/resiliência sejam fracamente aplicáveis em ambos os paradigmas.

Para as categorias de otimização e monitoramento/controle, observa-se que ambos os paradigmas podem ser aplicáveis em alguns aspectos e fortemente aplicáveis em outros. Contudo, destaca-se que para as duas categorias em questão o paradigma NFV obtém uma singela vantagem em detrimento de SDN. No que tange as funções de proteção, observa-se que NFV obteve melhores resultados para esta categoria. Destaca-se esta diferença do paradigma NFV em detrimento de SDN para a categoria de proteção devido aquele também possibilitar o cascateamento de funções. Este cascateamento permite que várias funções colaborem entre si para atingir um único objetivo.

Em suma, observando-se detalhadamente a tabela, pode-se concluir que nenhum dos paradigmas foi considerado não aplicável. Tendo isto em vista, percebe-se que ambos podem ser aplicados na implementação de funções de rede pertencentes a diferentes categorias.

6 MATERIAIS E MÉTODOS

Neste capítulo serão apresentados os procedimentos de implementação e a elaboração dos cenários de testes para realizar a avaliação das funções de rede implementadas e executadas nos paradigmas SDN e NFV. Será explicado a configuração dos cenários e as ferramentas computacionais utilizadas para a realização dos mesmos. Também será dada uma breve explicação do funcionamento das funções implementadas, bem como é abordado os protocolos e aplicações utilizadas para a realização dos testes.

6.1 PROCEDIMENTOS DE IMPLEMENTAÇÃO

Para realizar a avaliação e os testes, definiu-se os cenários de rede para os paradigmas SDN e NFV. As funções de rede que serão avaliadas são as funções de *switching*, *roteamento*, *firewall* do tipo filtragem de pacotes e de um servidor DHCP. Essas funções foram escolhidas devido executarem nas principais camadas do modelo de referência OSI, tais como, camada de enlace de dados, rede, transporte e aplicação, sendo possível então observar tendências de comportamento das funções que atuam em diferentes camadas. Também justifica-se esta escolha porque são funções muito utilizadas diariamente por todos os usuários para se conectarem à Internet, sendo frequentemente exploradas na academia.

Dentro desta conjuntura, a função de *firewall* consiste em um filtro que analisa os pacotes que chegam até ele, tomando a decisão de permitir ou não a passagem dos mesmos, aplicando um conjunto de regras determinadas pelo operador da rede. O *firewall* do tipo filtragem de pacotes opera nas camadas de rede e transporte, tem como característica principal analisar somente os campos do cabeçalho dos pacotes recebidos, como por exemplo, porta e endereço de origem e de destino e tipo de protocolo. A função de *switching*, que opera na camada de enlace de dados, encaminha os pacotes de acordo com o endereço MAC de destino, sendo considerado um dispositivo “inteligente”, que tem a função de aprender com a rede e posteriormente apenas encaminhar os pacotes para à máquina de destino específica. O roteamento, o qual opera na camada de rede, tem o objetivo de encaminhar os pacotes de uma origem a um destino que encontra-se em uma rede diferente. Já o servidor DHCP, operando na camada de aplicação, tem por finalidade fornecer a configuração do endereçamento IP dinamicamente aos seus clientes.

A fim de auxiliar no entendimento das funções de rede implementadas, elaborou-se

pseudo-códigos, conforme pode-se observar a seguir. No pseudo-código do *firewall*, existe uma lista *(lista_de_regras)*, com as definições das regras de permissão ou negação de tráfegos. Estas regras podem ser definidas de acordo com endereços IP, tipo de protocolo ou ainda, porta de origem e destino. O *firewall* recebe um pacote de dados de uma de suas interfaces de rede, salvando-o na variável "pacote". Em seguida, através de um laço, busca na lista uma regra capaz de classificar o pacote, segundo os campos de cabeçalho previamente citados. Se a regra for do tipo "bloquear", o pacote é descartado. Caso seja do tipo "permitir", o pacote é encaminhado para a interface adequada. Por padrão, regras de bloqueio têm precedência sobre regras de permissão. Por fim, caso nenhuma regra se aplique, então o pacote é descartado.

Algoritmo 1 - Firewall

```
1:  pacote ← escuta_interfaces
2:  for each regra ∈ lista_de_regras do
3:      if pacote.Endereco_Origem == regra.Endereco_Origem &&
4:        pacote.Endereco_Destino == regra.Endereco_Destino then
5:          if regra.porta == pacote.porta &&
6:            regra.protocolo == pacote.protocolo then
7:              if regra.tipo == bloquear then
8:                  drop_pacote()
9:              end if
10:         else
11:             if regra.tipo == permitir then
12:                 encaminha_pacote()
13:             end if
14:         end if
15:     end if
16: end for
17: drop_pacote()
```

Para a função de *switching*, para cada pacote que chega na interface de rede desta função, esta realiza o processo de ler os primeiros bits do endereço de destino, após isso, realiza-se uma busca em um conjunto de portas disponíveis, listando os endereços MAC dos dispositivos que estão associados a estas portas. A partir disso, caso esteja armazenada em sua tabela MAC o mapeamento entre o endereço MAC e a porta do destinatário, então o pacote é encaminhado diretamente para a porta deste. Caso contrário, o pacote é então encaminhado para todas as portas.

Algoritmo 2 - Switching

```
1: pacote ← escuta_interfaces
2: pacote.Le_Primeiros_Bits_Endereco_Destino
3: for each portas ∈ lista_de_portas do
4:     Listar_end_MAC_associados_portas
5:     if num.porta == MAC then
6:         encaminha_para_porta_especifica()
7:     else
8:         encaminha_para_todas_portas
9:     end if
10: end for
```

Para a função de roteamento, para cada pacote que chega na interface de rede desta função, é realizado primeiramente a localização do endereço IP do destinatário, após isso, calcula-se o endereço da rede, com isso também obtém-se o endereço de *broadcast* da rede e consequentemente o *range* de endereços IP da rede em questão. Após isso, verifica-se a tabela de roteamento a fim de identificar o *gateway* e a interface de saída, também realiza-se o processo de ARP para a partir do endereço IP descobrir o endereço MAC do destinatário. Seguindo, o processo continua com a reescrita da camada de enlace especificando o endereço MAC do destinatário e finalmente encaminhando o pacote para a interface de rede do mesmo.

Algoritmo 3 - Roteamento

```
1: pacote ← escuta_interfaces
2: for each End_IP_Destinatario ∈ lista_End_IP_Destinatario do
3:     Calcular_End_IP_Rede()
4:     Verificar_Tab_Roteamento()
5:     Identificar_GW_e_Interface_Saida()
6:     ARP()
7:     Reescrever_Camada_Enlace_End_MAC_Destinatario()
8:     encaminha_pacote_interface()
9: end for
```

Para a função de *dhcp-server*, o processo de pedido de endereçamento dinâmico por parte do cliente inicia-se com a mensagem *DHCP DISCOVER*, onde este envia um quadro *broadcast* com um pedido DHCP informando seu endereço MAC. O servidor então responde com uma mensagem *DHCP OFFER* que inclui o endereço IP disponível e outros parâmetros, após isso o cliente responde confirmando a oferta do servidor com uma mensagem *DHCP REQUEST*. Então, o servidor confirma a oferta do endereçamento por meio da mensagem *DHCP ACK*. Desta forma, o cliente já pode utilizar o endereçamento atribuído dinamicamente pelo servidor.

Algoritmo 4 - dhcp-server

1: $pacote \leftarrow escuta\_interfaces$
2: **for each** $Cliente\_Encaminha\_Pedido\_End\_IP \in Server\_IP\_Escuta$ **do**
3: $DHCP\_DISCOVER()\ Cliente\_Faz\_Pedido$
4: $DHCP\_OFFER()\ Server\_Oferta\_End\_IP$
5: $DHCP\_REQUEST()\ Confirmacao\_Cliente$
6: $DHCP\_ACK()\ Confirmacao\_Server$
7: $Cliente\_Utiliza\_End\_IP$
8: **end for**

6.2 ELABORAÇÃO DOS CENÁRIOS DE TESTES

Para a elaboração dos cenários de testes nos paradigmas SDN e NFV, definiu-se que seria utilizado protocolos/aplicações atuantes em diferentes camadas, a fim de observar comportamentos, tendências e padrões destes quando utilizados em ambos os paradigmas. Neste contexto, para facilitar a análise, estes foram separados em protocolos e aplicações. Os protocolos utilizados foram o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP). Devido a natureza dinâmica dos tipos de tráfegos das redes e com o objetivo de verificar diferentes comportamentos, executou-se os testes destes protocolos variando-se o tamanho dos pacotes em 64, 128, 256, 512 e 1024 *bytes* [RFC 2544] (BRADNER; MCQUAID, 1999). Já as aplicações testadas foram: *Domain Name System* (DNS), *ping*, VOZ e tráfego do jogo *Battlefield*.

Justifica-se a escolha destas aplicações, pois o objetivo era analisar fluxos com diferentes características tanto em relação ao tempo de duração quanto ao tráfego gerado. Assim escolheu-se aplicações de natureza leve e densa com fluxos curtos e longos com pouco e muito tráfego. A fim de facilitar o entendimento destes diferentes padrões de fluxos de forma resumida, a seguir apresenta-se a Tabela 2.

Tabela 2 – Fluxos de Rede

Duração/ Volume	**Pouco tráfego**	**Muito tráfego**
Fluxos curtos	Mice flows (Navegação Web)	Download de arquivos
Fluxos longos	VoIP, Telnet/SSH	Elephant flows (Videoconf., VoD)

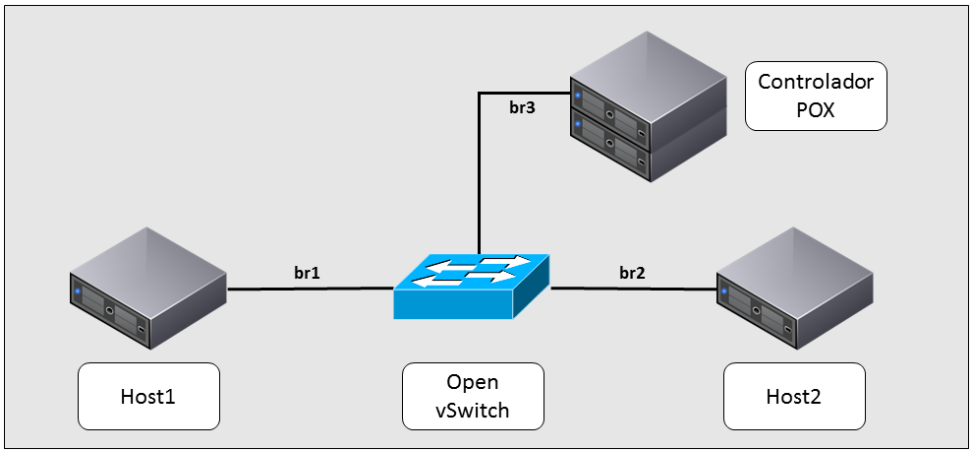
Fonte: Autoria Própria.

Para executar os testes, utilizou-se a plataforma de virtualização KVM, versão 1.4.0, executando juntamente com o gerenciador de máquinas virtuais *Virtual Manager*. As conexões entre as VM foram estabelecidas com *Linux bridges*, havendo uma *bridge* específica para cada ligação entre duas VM.

Em relação ao cenário SDN, ressalta-se que foi utilizado o controlador POX devido a sua consolidação na academia para a realização de pesquisas e experimentos de novos paradigmas na área de redes de computadores. Além disso, possui uma comunidade ativa e boa documentação possibilitando uma boa curva de aprendizado como um todo. Utilizou-se o *Open vSwitch* (OVS) para realizar a comunicação com o controlador SDN, pois o mesmo possibilita um processamento do tráfego dos pacotes de forma mais fidedigna com o mundo real.

Conforme pode-se ver na Figura 10, para o cenário SDN foi preciso criar 4 VM, sendo duas para os *hosts*, uma para o controlador POX e uma para o OVS. Criou-se uma *bridge* para conectar *host* 1 ao OVS, outra para ligar *host* 2 ao OVS e uma terceira ligando OVS ao POX.

Figura 10 – Arquitetura do cenário SDN.

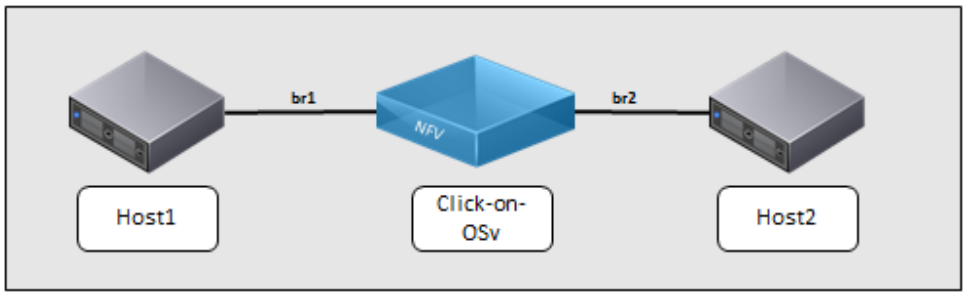


Fonte: Autoria Própria.

Já para o cenário NFV, conforme pode-se observar na Figura 10, foi preciso criar 3 VM, duas para os *hosts* e uma para o sistema de virtualização NFV. Criou-se apenas duas *brigdes*, cada uma ligando um *host* à VM executora da função virtual. Para o sistema de virtualização NFV foi utilizada a plataforma OSv [1] juntamente com o *Click Modular Router*.

[1] https://github.com/lmarcuzzo/click-on-osv

Figura 11 – Arquitetura do cenário NFV.



Fonte: Autoria Própria.

Todas as VM foram configuradas com 512MB de memória RAM e 1 vCPU. Ressalta-se que as interfaces de rede de todas as VM foram configuradas utilizando o modo *virtio* para obter um melhor desempenho no processamento dos pacotes trafegados na rede. Os *links* foram configurados usando uma rede *Gigabit*. Ainda, destaca-se que exclusivamente para a função do servidor DHCP executando no paradigma NFV, é preciso que as *bridges* sejam as mesmas tanto no servidor quanto nos clientes (*hosts*). Para as demais funções em NFV a configuração segue normalmente como mostrado na Figura 11.

Além do mais, as VM do OVS em SDN e do sistema de virtualização em NFV, foram configuradas com 3 interfaces de rede virtuais. Já os clientes foram configurados com apenas uma interface, onde cada adaptador de rede é conectado à sua *brigde* apropriada. Por fim, os cenários e os testes foram implementados em um notebook Dell Inspiron 15 série 5000 com processador Intel Core i7 de 7ª geração com 8 GB de memória RAM e 1 TB de disco, executando o sistema operacional Linux Ubuntu 16.04 LTS.

7 AVALIAÇÃO DE DESEMPENHO

Para realizar a avaliação do *firewall*, foram realizados testes dos protocolos TCP e UDP variando-se o tamanho dos pacotes. Além disso, também testou-se nesta funcionalidade as aplicações DNS, *ping*, VOZ e tráfego do jogo *battlefield*. Para a avaliação dos resultados obtidos nos testes, foram analisadas as seguintes métricas: *delay* (atraso), jitter, *throughput* (vazão) e perda de pacotes. Ressalta-se que as barras de erro nos gráficos estão representando a variação do atraso mínimo e máximo, isto é, o *jitter*. Estas métricas foram analisadas tanto nos testes dos protocolos quanto nas aplicações.

Ainda, com o *firewall* desabilitado, no *host* 1 foi utilizada a ferramenta *Hping3* onde foram gerados os tráfegos sintéticos desejados para realizar a avaliação. Ademais, estes tráfegos foram gerados com endereços IP de origem aleatórios com destino ao endereço IP do *host* 2. Neste último *host*, através da ferramenta *TCPdump* é capturado e gerado um arquivo do tráfego em formato *pcap*. Com a utilização da ferramenta *tcpreplay* e com o *firewall* habilitado, o tráfego é replicado novamente para a rede. A partir do início do teste, o tráfego gerado a partir do *host* 1 para o *host* 2 é salvo em formato *pcap*, onde as métricas resultantes dos testes de cada protocolo são melhor analisadas através do analisador *Wireshark*.

Na avaliação do servidor DHCP, foram observados o atraso (*delay*) desta função desde o seu instanciamento até a entrega do endereçamento IP dinâmico aos clientes. Além disso, também foi analisado a vazão (*throughput*) que esta função fornece na execução deste serviço. Para a função de roteamento, analisou-se o atraso e a vazão no processamento de pacotes ICMP com 64 bytes de tamanho (*ping*), variando-se o endereço IP de origem para o mesmo endereço IP de destino. Justifica-se essa variação já que tal funcionalidade tem como objetivo principal realizar o encaminhamento de pacotes entre diferentes redes. Destaca-se que para estas duas funcionalidades não foram observados perda de pacotes.

Já para a avaliação da função de *switching*, foram analisadas as métricas de atraso, vazão e perda de pacotes no processamento da aplicação *ping*. Esta função tem o objetivo de encaminhar pacotes dentro da mesma rede. Então para realizar a comutação dos *frames* manteve-se o mesmo endereço IP de origem para o mesmo endereço IP de destino. Ademais, tais testes nas funções de *firewall, switching* e roteamento foram executados durante um intervalo de tempo de 60 segundos [RFC 2544] (BRADNER; MCQUAID, 1999). Também destaca-se que os testes foram executados com um intervalo de confiança de 95%.

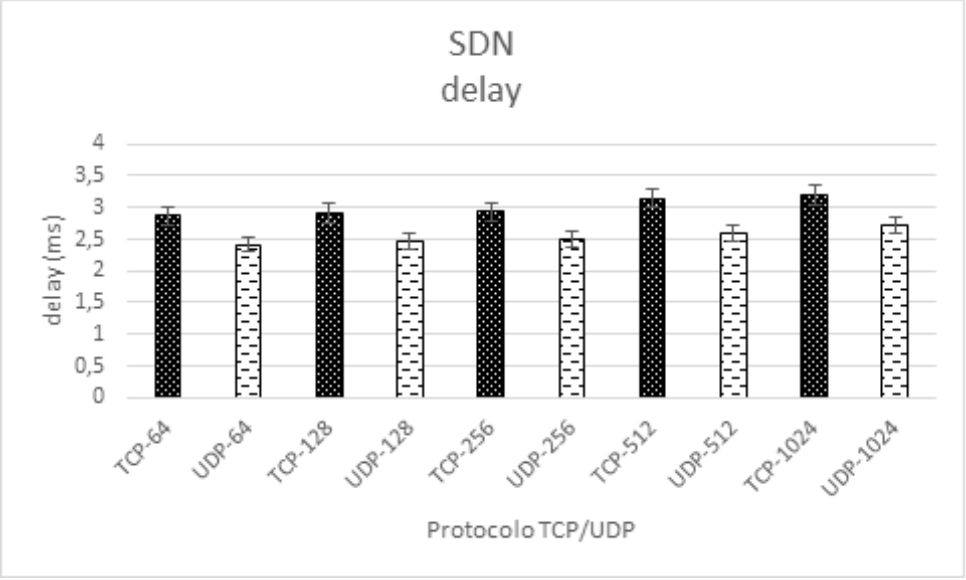
7.1 AVALIAÇÃO DAS FUNÇÕES EXECUTANDO EM SDN

Esta seção discute a avaliação das funções de *firewall*, *switching*, roteamento e *dhcp-server* executando em SDN.

7.1.1 Firewall

Conforme Figura 12 observa-se que nos protocolos TCP e UDP à medida em que aumenta-se o tamanho dos pacotes, também é perceptível um aumento do atraso no processamento dos mesmos. Além disso, também constata-se um maior atraso em todos os tamanhos de pacotes no protocolo TCP em detrimento do UDP. Nesse contexto percebe-se que o paradigma SDN necessita de maior capacidade de processamento na execução do protocolo TCP. Isso ocorre devido ao fato do protocolo TCP ser orientado a conexão e realiza retransmissões para garantir a entrega dos pacotes. Já como o protocolo UDP não é orientado a conexão e não necessita realizar a confirmação de entrega dos pacotes, então o seu tempo de processamento acaba sendo menor, consequentemente ocorrendo em menor atraso.

Figura 12 – Delay por Protocolo da função firewall implementada em SDN.



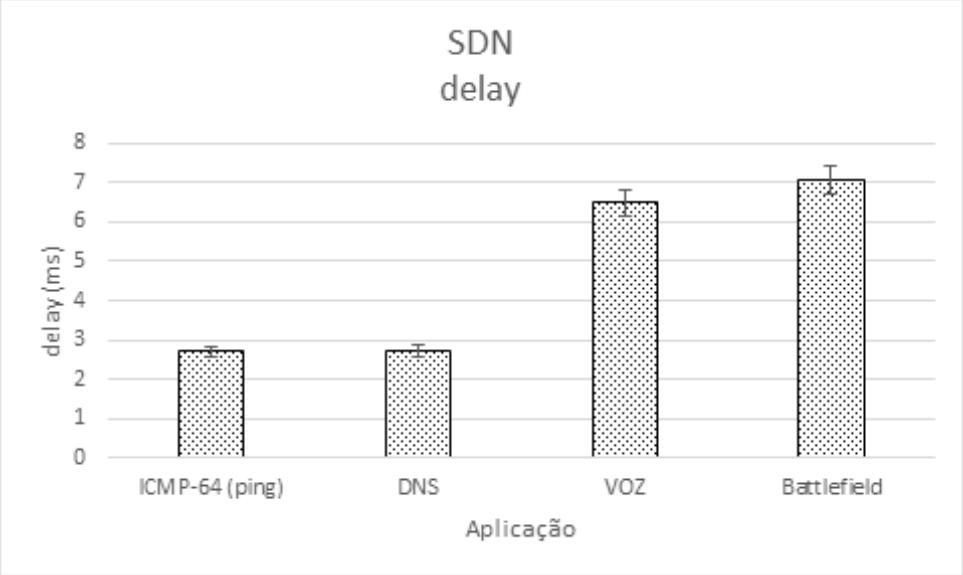
Fonte: Autoria Própria.

Com os resultados da análise do *delay* das aplicações avaliadas, observa-se, conforme Figura 13, valores semelhantes desta métrica para aplicações mais leves e para as mais pesadas. Como o tráfego das aplicações *ping* e DNS é considerado mais leve, logo estas aplicações são mais simples de serem processadas. Sendo assim, o atraso é significativamente menor em

relação as aplicações mais densas como é o caso dos tráfegos de VOZ e do jogo *battlefield*.

Ainda, dentre as aplicações avaliadas, constatou-se que o tráfego do jogo *battlefield* obteve o maior atraso. Nesse sentido, o atraso foi um pouco maior do que o tráfego de VOZ, pois aplicações desta natureza frequentemente demandam maior capacidade computacional. Dessa forma, necessita-se de maior processamento do controlador SDN na configuração dos fluxos no comutador virtual, consequentemente o atraso tende a ser maior. Nesse contexto, esta diferença pode ser ainda mais acentuada quando estas aplicações são configuradas para serem executadas em rede, envolvendo um grande número de usuários.

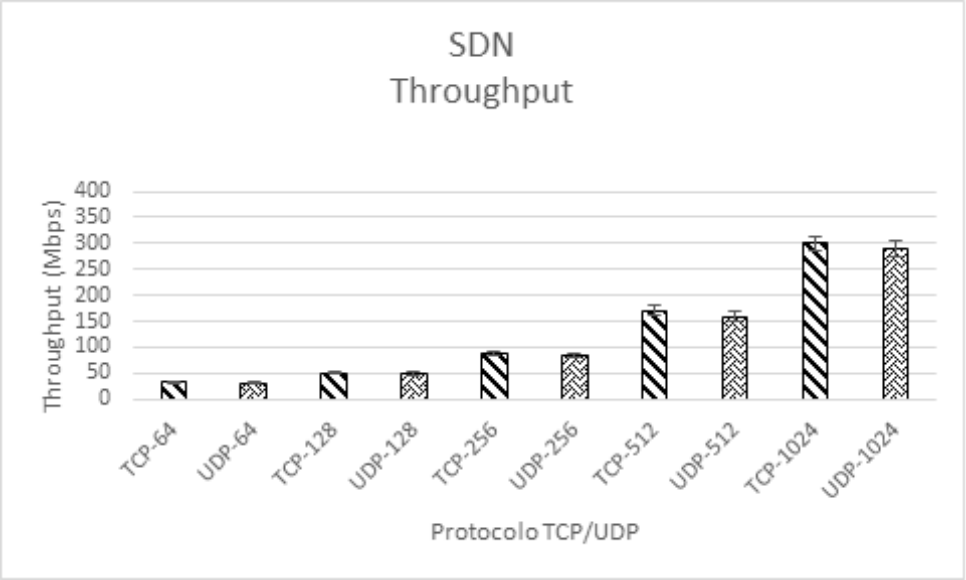
Figura 13 – Delay por Aplicação da função firewall implementada em SDN.



Fonte: Autoria Própria.

Em relação a análise do *throughput* dos protocolos TCP e UDP, observa-se através da Figura 14 um comportamento padrão dos mesmos. Indo além, a medida que aumenta-se o tamanho dos pacotes, automaticamente também há um incremento da vazão. Tal fato ocorre para facilitar o processamento e consequentemente o encaminhamento dos pacotes pela rede. Dessa forma, o controlador SDN, o qual possui a inteligência da rede, disponibiliza maior taxa de transferência e capacidade de vazão aos elementos encaminhadores da rede.

Figura 14 – Throughput por Protocolo da função firewall implementada em SDN.

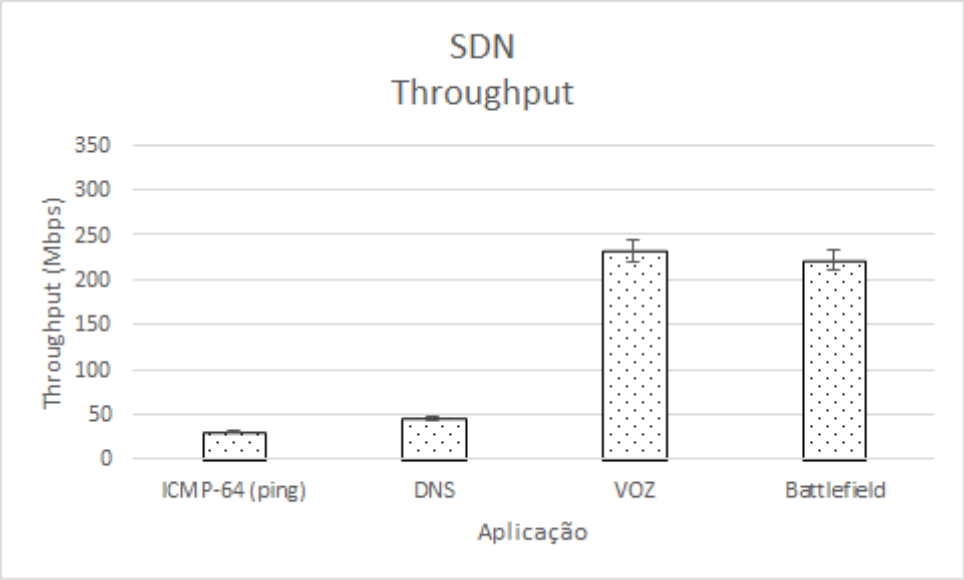


Fonte: Autoria Própria.

Nos valores obtidos na avaliação do *throughput* das aplicações, conforme apresentado na Figura 15, observou-se que para as aplicações mais leves (*ping* e DNS) obteve-se vazões pequenas, já para aplicações pesadas (VOZ e jogo), a vazão foi significativamente maior. Isto ocorre já que o paradigma SDN possui a inteligência para distinguir quais os tipos de aplicações necessitam de maior vazão para serem trafegadas pela infraestrutura computacional. Ainda, tal comportamento ocorre devido a natureza dinâmica e a programabilidade inerente a este paradigma.

Além disso, também observa-se um crescimento da vazão a partir dos tráfegos do *ping*, DNS e VOZ. Ainda, em relação a análise das aplicações de VOZ e do jogo *battlefield*, percebe-se que a primeira obteve uma vazão maior. Este resultado ocorreu já que a aplicação de VOZ obteve um atraso menor. Dessa forma o processamento pode ser realizado de forma mais rápida, assim, conseguiu-se uma vazão um pouco superior em relação ao jogo analisado.

Figura 15 – Throughput por Aplicação da função firewall implementada em SDN.

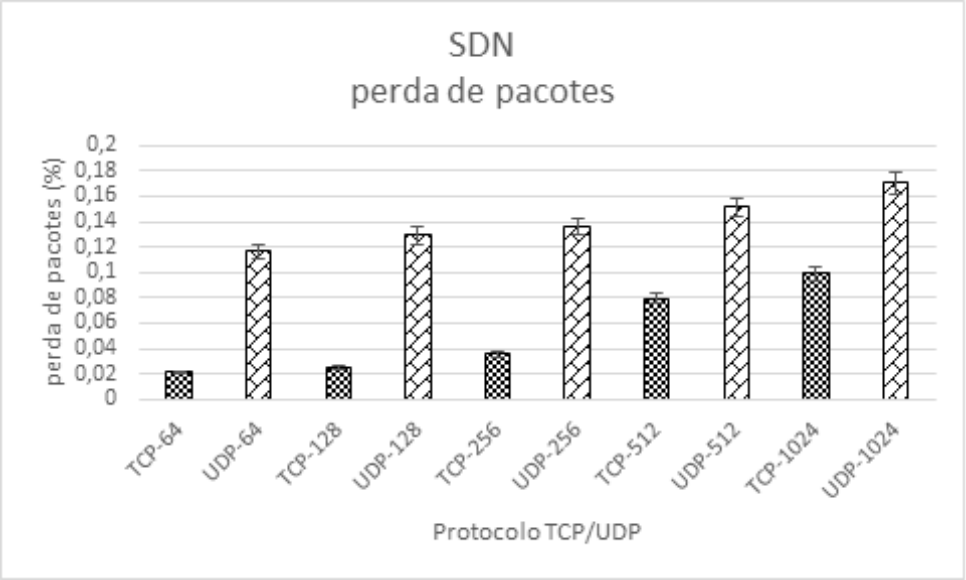


Fonte: Autoria Própria.

Constatou-se que a perda de pacotes dos protocolos TCP/UDP foi praticamente insignificante. Conforme a Figura 16, percebe-se o porquê tais perdas podem ser desprezíveis, já que não representam nem 0,2% do total de pacotes trafegados na função analisada em questão. Ainda, observa-se que conforme aumenta-se o tamanho dos pacotes, também aumenta-se a porcentagem das perdas.

Além disso, também percebe-se que as perdas são maiores em todos os tamanhos de pacotes do protocolo UDP. Tal fato acontece já que este protocolo não tem garantia de entrega dos pacotes. Assim, observou-se no analisador *Wireshark* que pacotes são perdidos por estarem fora de ordem, duplicados, mal formados, etc. Dessa forma, neste protocolo registrou-se maiores perdas de pacotes. Isso ocorre em menor quantidade com o protocolo TCP, pois este é orientado a conexão, e portanto necessita realizar a confirmação da entrega dos pacotes.

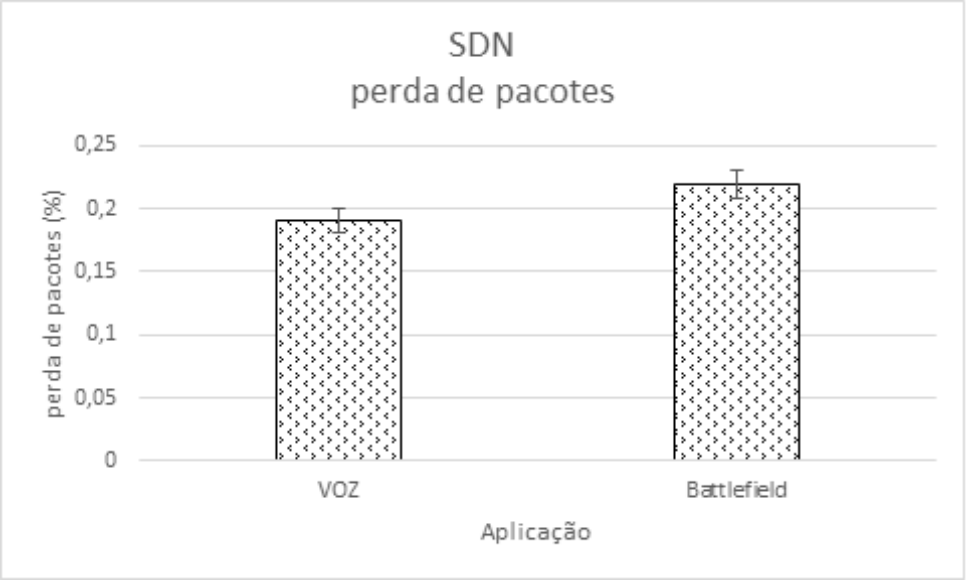
Figura 16 – Perda de pacotes por Protocolo da função firewall implementada em SDN.



Fonte: Autoria Própria.

Seguindo dentro deste contexto, porém agora analisando-se a perda de pacotes ocorrida nas aplicações, observa-se conforme Figura 17, que houve uma maior perda de pacotes destas em relação aos protocolos TCP/UDP. Analisou-se no *Wireshark* que boa parte das perdas são ocasionadas por pacotes mal formados. Estas perdas ocorrem em virtude que para realizar o processamento de aplicações mais densas, exige-se mais requisições ao controlador SDN a fim deste realizar o processo de tomada de decisão. Sendo assim, possivelmente maior quantidade de pacotes acabam sendo perdidos durante o decorrer deste processo. Ainda, ressalta-se que não houve perda de pacotes nas aplicações *ping* e DNS.

Figura 17 – Perda de pacotes por Aplicação da função firewall implementada em SDN.

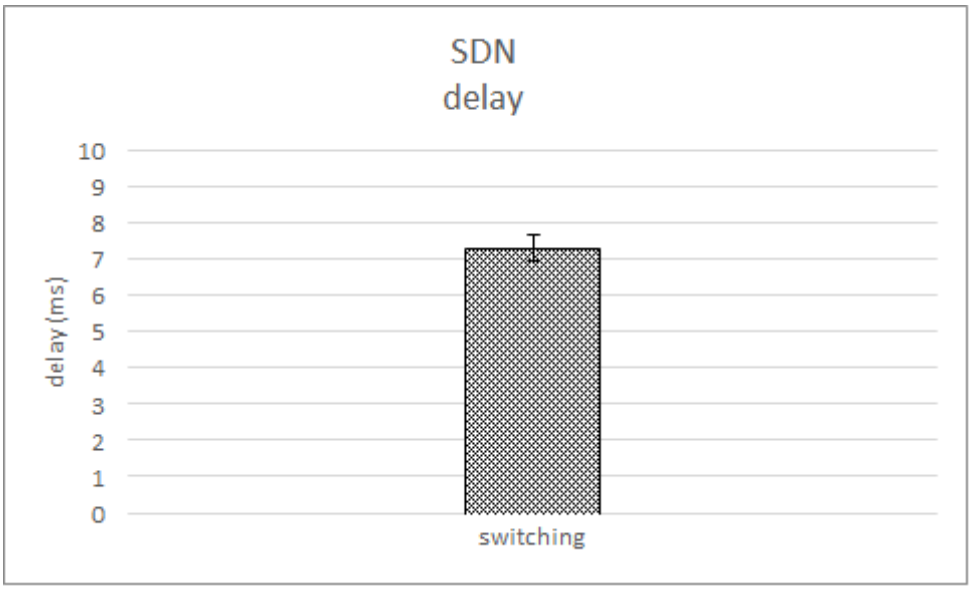


Fonte: Autoria Própria.

7.1.2 Switching

O atraso medido na função de *switching*, conforme apresentado na Figura 18, é um pouco menor que o valor obtido na função de roteamento. Isto acontece já que a função de *switching* tem como objetivo principal a comunicação de *hosts* dentro da mesma rede. Nesse contexto, o controlador SDN não necessita dispender significativo esforço em processamento a fim de aprender diferentes rotas para encaminhar os pacotes que chegam de diversos endereços IP. Sendo assim, a comunicação entre controlador e comutador para a instalação dos fluxos neste torna-se mais rápida de ser realizada. Dessa forma, verifica-se que para funções de comutação o atraso tende a ser um pouco menor do que para aquelas de roteamento.

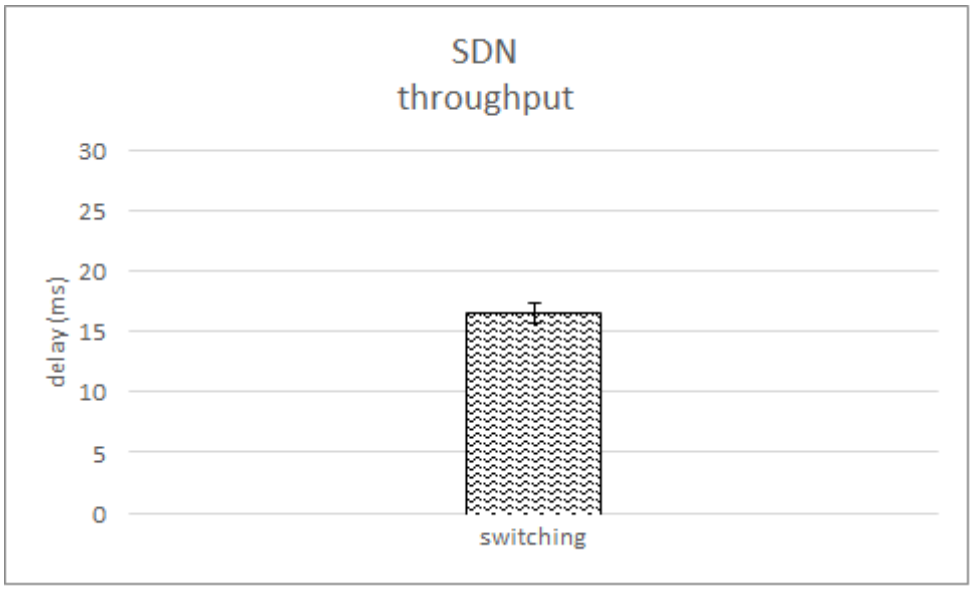
Figura 18 – Delay da função switching implementada em SDN.



Fonte: Autoria Própria.

A vazão medida nesta função é substancialmente maior quando comparada com a função de roteamento. Como o processamento dos pacotes dá-se dentro de mesma rede, o controlador não fica sobrecarregado. Sendo assim, este consegue proporcionar maior capacidade de taxa de transferência na comutação dos *frames* dentro do mesmo domínio de rede. Nesse sentido, funções da camada de enlace (*switching*) acabam obtendo maior vazão em detrimento de funcionalidades executando na camada de rede (roteamento).

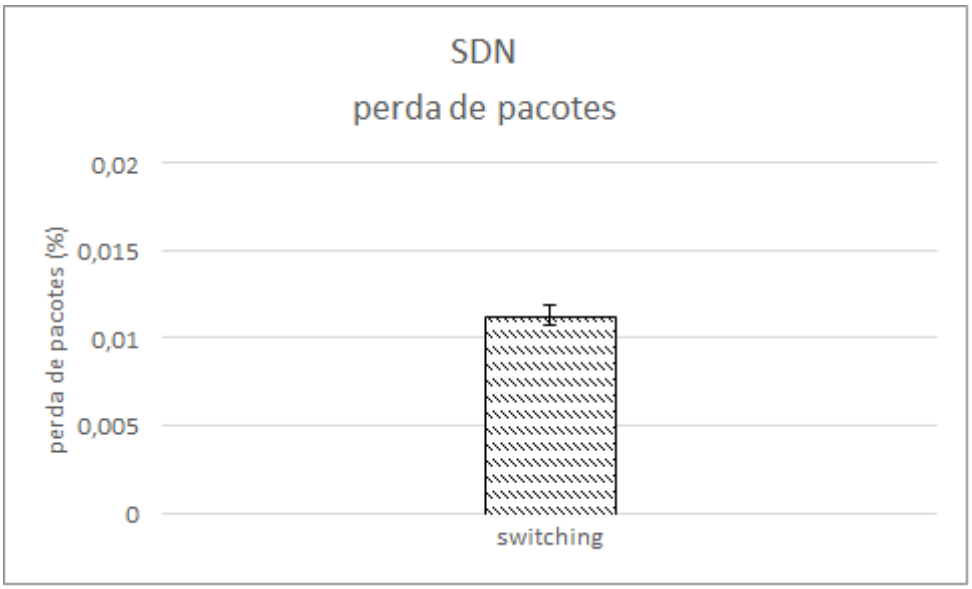
Figura 19 – Throughput da função switching implementada em SDN.



Fonte: Autoria Própria.

No que diz respeito a perda de pacotes desta função, observa-se que a mesma é mínima, resultando em um pouco mais que 0,01% do total de pacotes trafegados no teste desta função. Esta insignificante perda ocorre em razão que esta função obtém um bom valor de vazão. Diante disso, o processamento dos pacotes pelo controlador SDN tende a ser mais rápido. Dessa forma, observou-se que pacotes mal formados foram perdidos. Essa singela perda de *frames* ocorreu no processo de comutação realizado pelo controlador SDN.

Figura 20 – Perda de pacotes da função switching implementada em SDN.

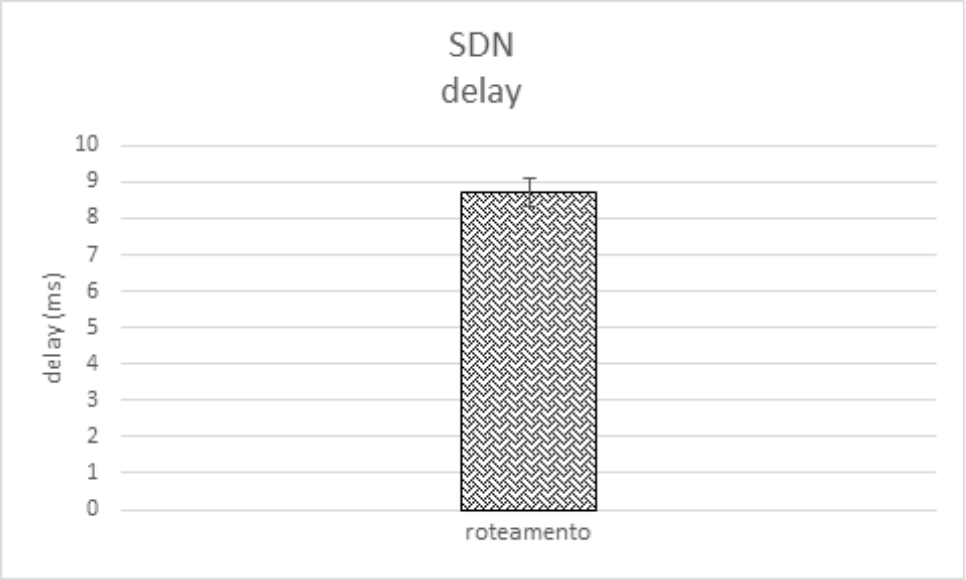


Fonte: Autoria Própria.

7.1.3 Roteamento

No que tange o atraso medido na função de roteamento, conforme pode-se verificar na Figura 21, percebe-se que este é significativamente maior que o atraso observado na função de *dhcp-server*. Esta substancial diferença é devido ao fato que o roteamento realiza a comunicação entre diferentes redes. Para isso é preciso que o controlador aprenda e calcule as rotas para posteriormente instalar as entradas de fluxos no comutador *OpenFlow* virtualizado. Dessa maneira, o atraso tende a ser substancialmente maior em funções de roteamento quando em comparação, por exemplo, a um servidor DHCP.

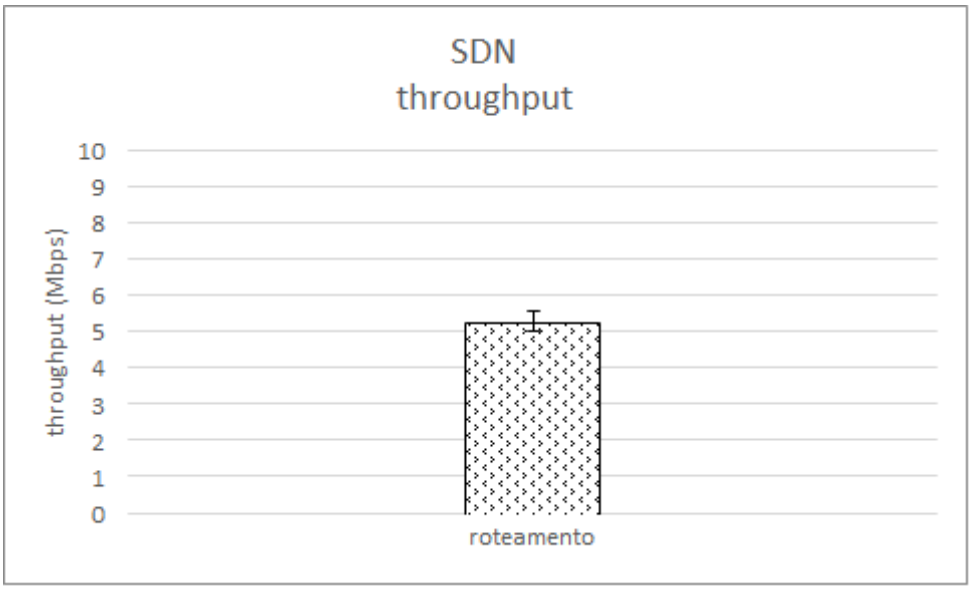
Figura 21 – Delay da função roteamento implementada em SDN.



Fonte: Autoria Própria.

De acordo com a Figura 22, como o processo de roteamento necessita de maior poder de processamento para ser realizado, a vazão observada nesta função foi maior em relação ao servidor DHCP. Através da inteligência e programabilidade fornecida pelo controlador SDN, cabe a este realizar a distinção e a priorização entre as necessidades e características de diversas funções que operam em diferentes camadas. Também salienta-se que não houve perda de pacotes na avaliação desta função.

Figura 22 – Throughput da função roteamento implementada em SDN.

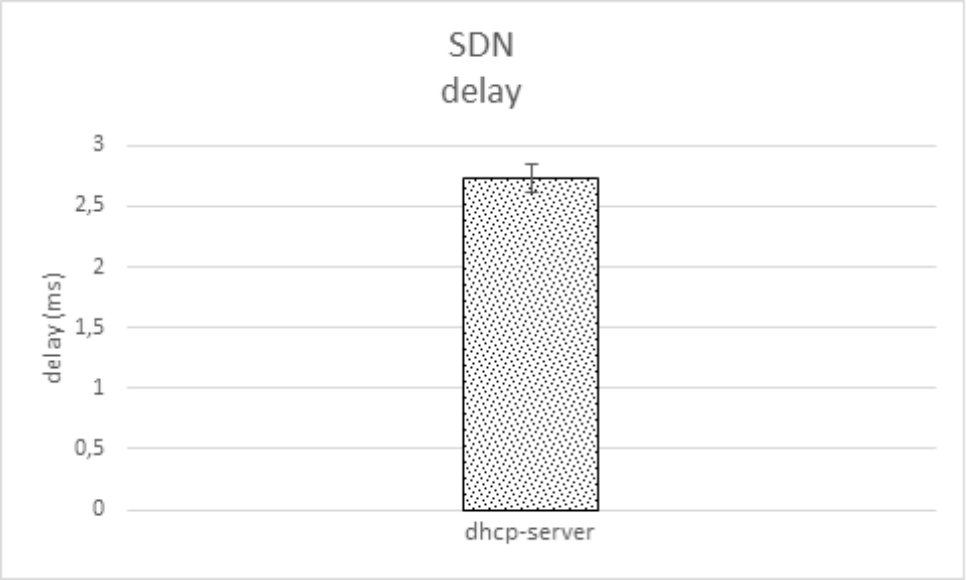


Fonte: Autoria Própria.

7.1.4 DHCP-Server

O atraso verificado na função de *dhcp-server* representa o tempo despendido entre a instanciação da VM que executa esta funcionalidade até o recebimento das configurações de endereçamento IP pelos clientes. Observando-se a Figura 23, percebe-se que este atraso é pequeno, portanto esta função executando em SDN possui bom desempenho.

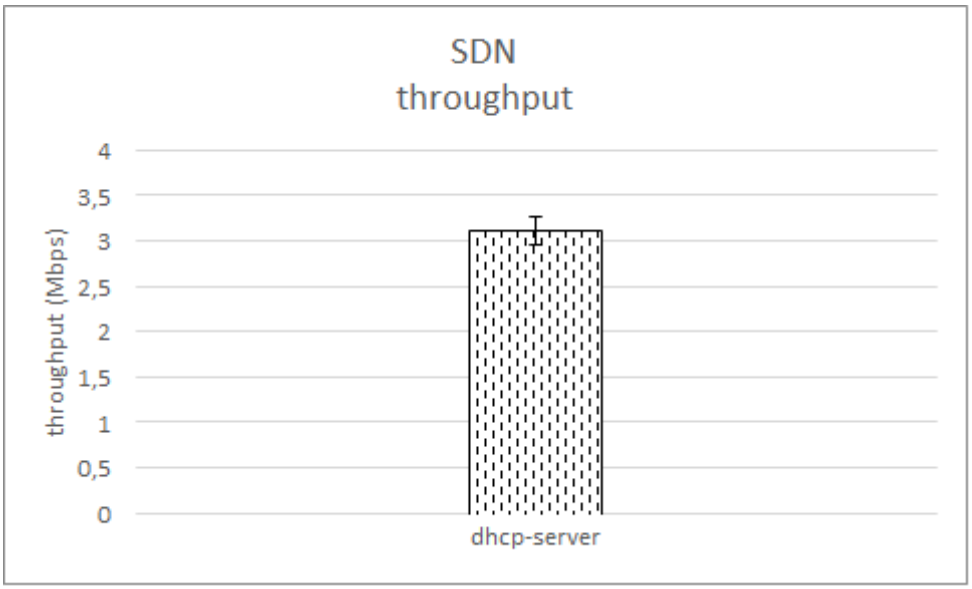
Figura 23 – Delay da função dhcp-server implementada em SDN.



Fonte: Autoria Própria.

Essa função serve basicamente para a configuração de endereçamento IP de forma dinâmica. Diante disso, conforme percebe-se através da Figura 24, a mesma não necessita de significativa vazão. Ainda, salienta-se que não houve perda de pacotes na avaliação desta função.

Figura 24 – Throughput da função dhcp-server implementada em SDN.



Fonte: Autoria Própria.

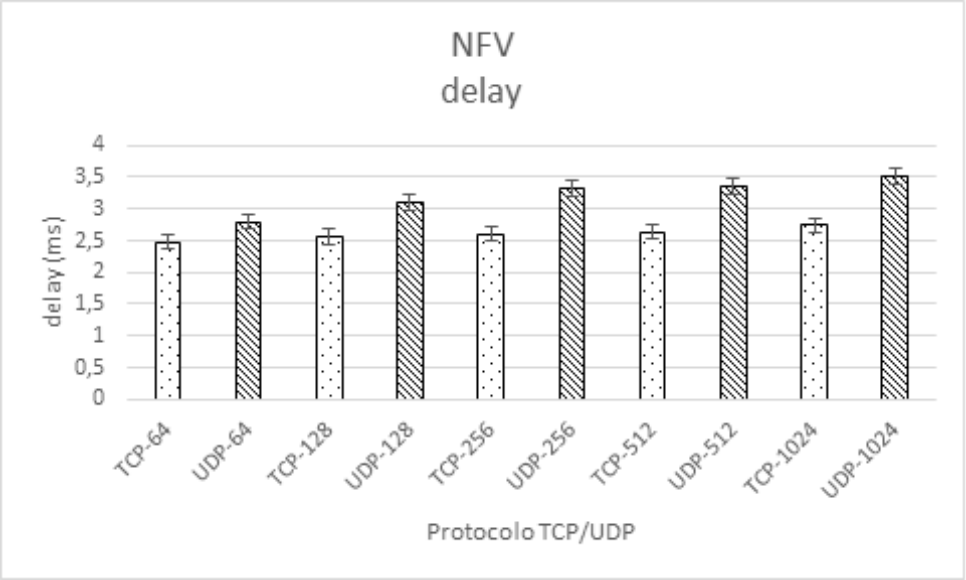
7.2 AVALIAÇÃO DAS FUNÇÕES EXECUTANDO EM NFV

Esta seção discute a avaliação das funções de *firewall*, *switching*, roteamento e *dhcp-server* executando em NFV.

7.2.1 Firewall

Conforme Figura 25, observa-se um comportamento semelhante dos protocolos TCP e UDP, pois à medida em que aumenta-se o tamanho dos pacotes, também é perceptível um aumento do atraso no processamento dos mesmos. Além disso, também constata-se um maior atraso em todos os tamanhos de pacotes no protocolo UDP em detrimento do TCP. Diferentemente de SDN, percebe-se que o paradigma NFV possui maior dificuldade para processar o protocolo de transporte UDP. Desta forma, tráfegos UDP exigem maior capacidade de processamento do paradigma NFV. Esse comportamento ocorre em função da natureza mais dinâmica do protocolo UDP.

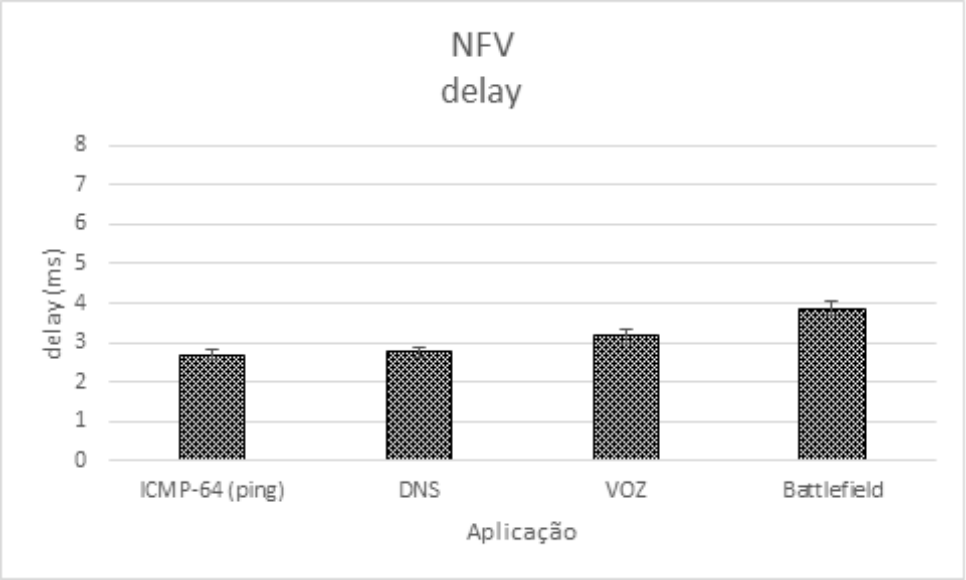
Figura 25 – Delay por Protocolo da função firewall implementada em NFV.



Fonte: Autoria Própria.

No que tange o atraso verificado nas aplicações, percebe-se, conforme Figura 26, um comportamento crescente desta métrica executando em NFV. Ainda, para as aplicações *ping* e DNS que são consideradas mais leves, observa-se resultados semelhantes na avaliação do atraso. Para as aplicações de VOZ e do jogo *battlefield* que são consideradas mais pesadas em relação ao tipo de tráfego, obteve-se maiores valores de atraso. Entretanto, como NFV possui a inteligência e a capacidade de processar tráfegos pesados, constata-se que a diferença entre o valor de atraso entre as aplicações leves e pesadas foi mínima, mais especificadamente em torno de apenas 1ms.

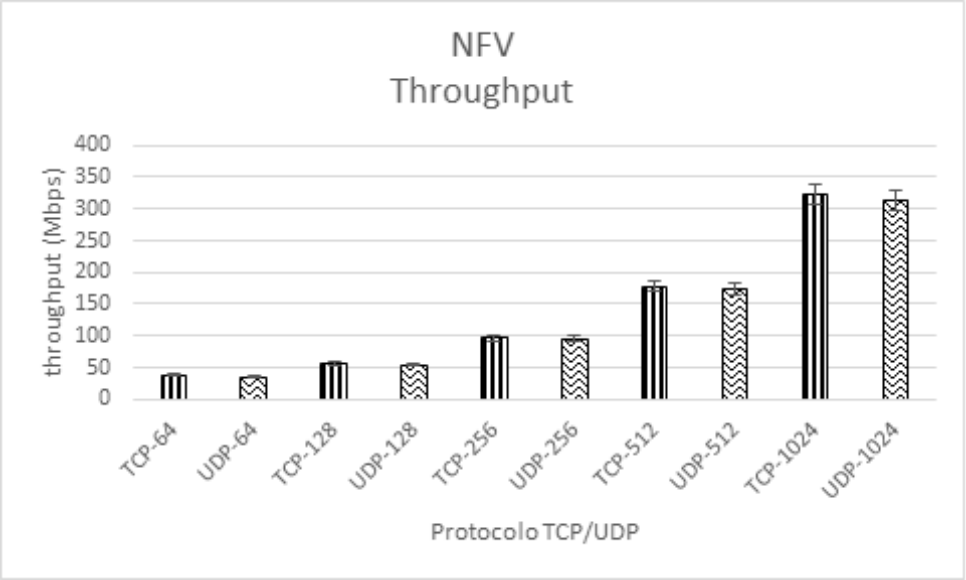
Figura 26 – Delay por Aplicação da função firewall implementada em NFV.



Fonte: Autoria Própria.

No resultado da vazão dos protocolos TCP e UDP, de acordo com a Figura 27, pode-se observar resultados semelhantes e um comportamento padrão dos mesmos. Ainda neste contexto, a medida que aumenta-se o tamanho dos pacotes, da mesma forma também há um aumento da vazão. Além do mais, também nota-se uma pequena diferença de vazão do TCP em relação ao UDP. Dessa forma, percebe-se uma tendência de linearidade nos valores em ambos os protocolos. Nesse sentido, apesar de uma leve diferença na vazão entre os protocolos, o paradigma NFV mantem um comportamento semelhante para esta métrica em relação aos dois protocolos da camada de transporte.

Figura 27 – Throughput por Protocolo da função firewall implementada em NFV.

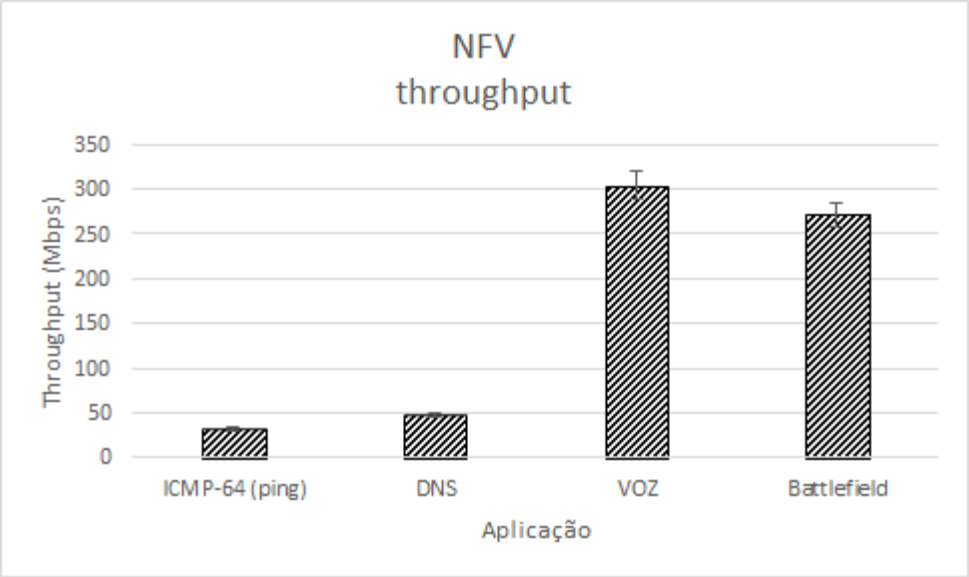


Fonte: Autoria Própria.

No que diz respeito aos resultados da vazão nas aplicações testadas, conforme visualiza-se na Figura 28, percebe-se comportamentos mais próximos entre as aplicações de natureza semelhantes. Ainda, em relação as aplicações leves, observa-se que o tráfego de DNS obteve uma vazão um pouco superior que a aplicação tradicional de *ping*. Este resultado aconteceu já que nota-se a tendência e a forte característica do paradigma NFV em priorizar, sempre que for possível, o processamento de aplicações mais densas.

Já nas aplicações mais pesadas, como o tráfego de VOZ e jogo em questão, a vazão observada foi significativamente maior em comparação as aplicações leves. O destaque foi a aplicação de VOZ, o qual obteve a maior taxa de vazão. Como esta aplicação obteve menor atraso que o jogo analisado, o paradigma NFV maximizou o processamento do tráfego de VOZ, consequentemente a vazão deste foi superior ao jogo em questão.

Figura 28 – Throughput por Aplicação da função firewall implementada em NFV.

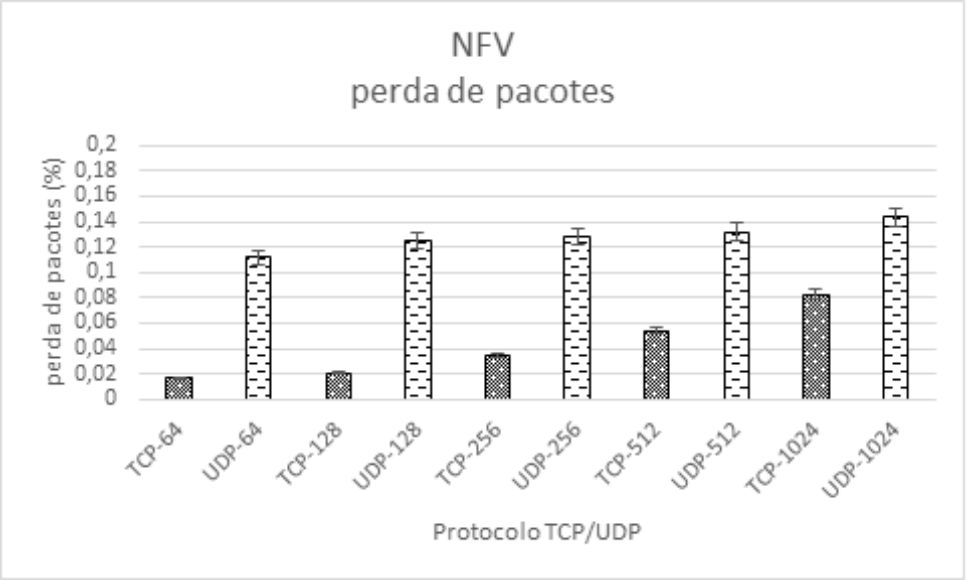


Fonte: Autoria Própria.

Conforme a Figura 29, constata-se que a perda de pacotes dos protocolos TCP/UDP foi desprezível. Ainda, também percebe-se que a medida que aumenta-se o tamanho dos pacotes, também há um aumento das perdas. Além disso, em relação ao protocolo TCP, observa-se uma tendência de linearidade conforme varia-se o tamanho dos pacotes.

Ainda, também verifica-se que as perdas são maiores em todos os tamanhos de pacotes do protocolo UDP. Isto decorre já que este protocolo não é orientado a conexão, portanto não necessita realizar a confirmação de entrega dos pacotes. Dessa forma, percebe-se comportamentos semelhantes dos protocolos TCP/UDP na função de *firewall* executando em ambos os paradigmas. Identifica-se isso através dos resultados obtidos na avaliação desta métrica em questão.

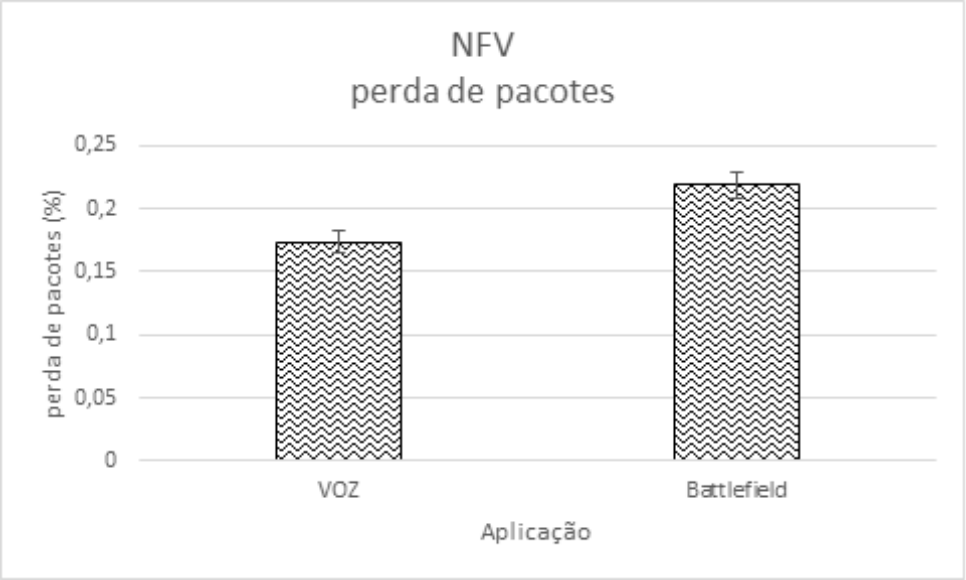
Figura 29 – Perda de pacotes por Protocolo da função firewall implementada em NFV.



Fonte: Autoria Própria.

Conforme a Figura 30, percebe-se que as aplicações obtiveram perdas de pacotes nos tráfegos de VOZ e do jogo *battlefield*. Tais perdas foram ocasionadas principalmente por pacotes duplicados e fora de ordem. Tais aplicações por serem sensivelmente mais difíceis de serem processadas, tendem geralmente a registrar perdas de pacotes. Nesse contexto, pode-se notar que as mesmas obtiveram perdas de pacotes superiores em contraste as registradas nos protocolos TCP/UDP. Também é importante mencionar que não foi registrado perdas de pacotes nas aplicações *ping* e DNS.

Figura 30 – Perda de pacotes por Aplicação da função firewall implementada em NFV.

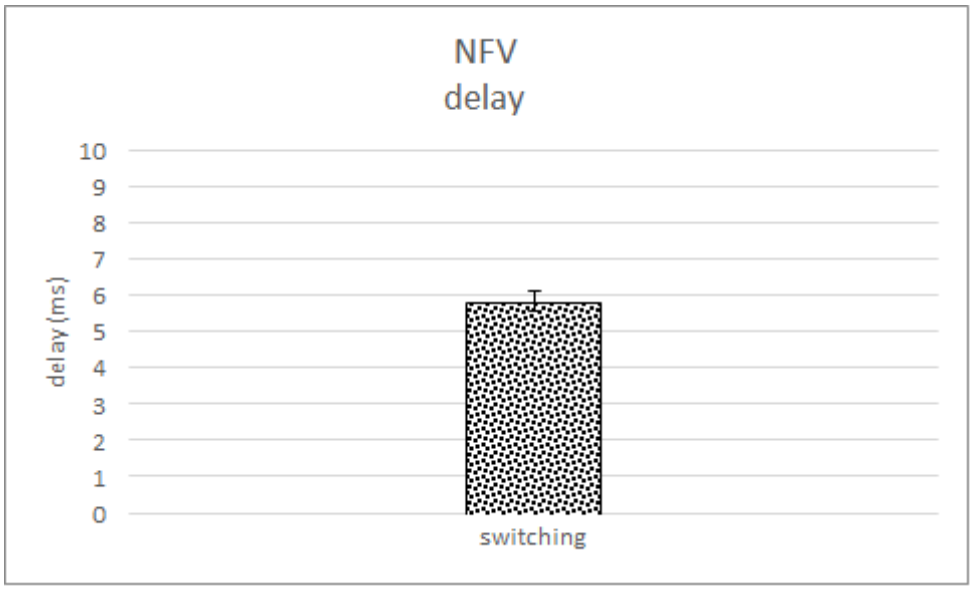


Fonte: Autoria Própria.

7.2.2 Switching

De acordo com a Figura 31, percebe-se que o paradigma NFV não apresenta diferença significativa na medição do atraso em relação a função de roteamento. Isso ocorre em função que no paradigma NFV não existe a questão do comutador *OpenFlow* precisar ficar consultando o controlador para este realizar a tomada de decisão. Dessa forma, o processamento de funções de comutação e roteamento possuem comportamentos semelhantes em relação a medição desta métrica.

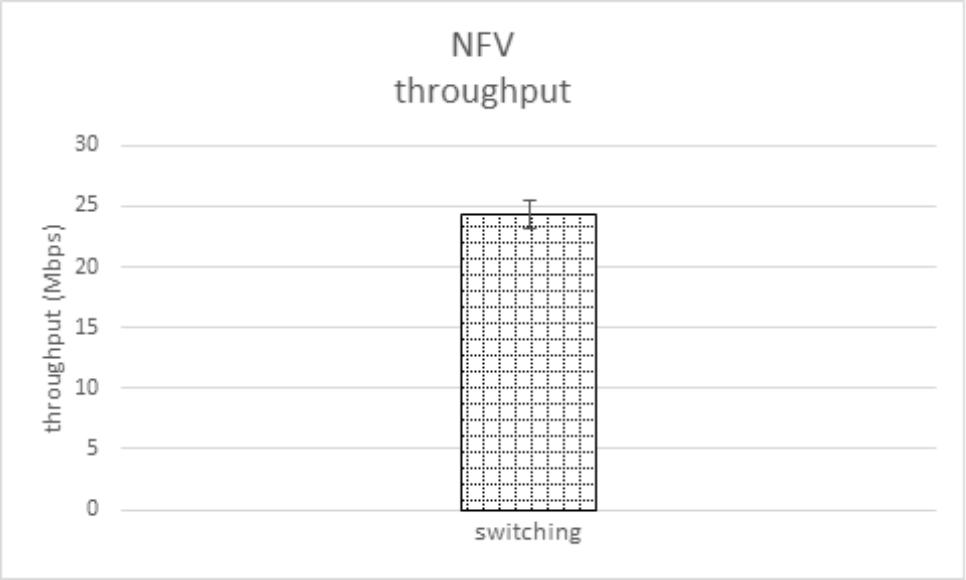
Figura 31 – Delay da função switching implementada em NFV.



Fonte: Autoria Própria.

A vazão medida nesta função é bastante superior quando comparada com o valor obtido desta métrica na função de roteamento. Nesse contexto, observa-se comportamentos semelhantes na avaliação desta métrica nos dois paradigmas em contraste ao roteamento. Isso também ocorre no paradigma NFV já que o processamento dos tráfegos das duas funções dá-se de maneira similar. Entretanto, como o processamento é realizado dentro do mesmo domínio de rede, a taxa de transferência de comutação acaba sendo consideravelmente maior. O paradigma NFV também proporcione uma vazão maior para funções de comutação em detrimento de roteamento.

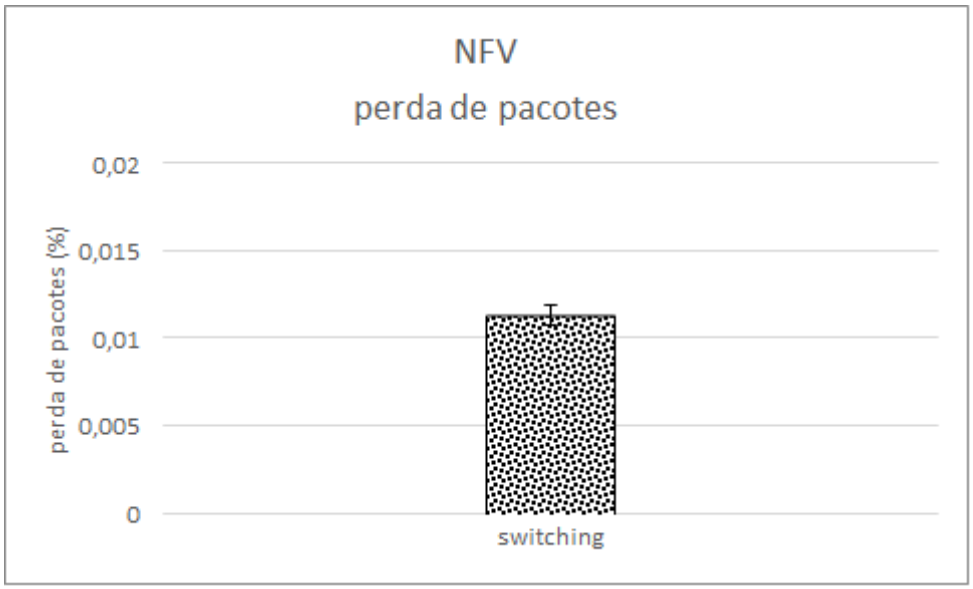
Figura 32 – Throughput da função switching implementada em NFV.



Fonte: Autoria Própria.

No que tange a perda de pacotes desta função, através dos resultados obtidos, percebeu-se comportamentos semelhantes em SDN e NFV. Constatou-se isso já que esta métrica resultou apenas em um pouco mais que 0,01% de pacotes perdidos. Sendo assim, pode-se dizer que esta porcentagem é praticamente insignificante. Essa perda ocorreu em função do atraso de resposta de alguns pacotes do tráfego ICMP. Nesse contexto, ressalta-se que por meio do *Wireshark* foram registrados alguns pacotes fora de ordem.

Figura 33 – Perda de pacotes da função switching implementada em NFV.

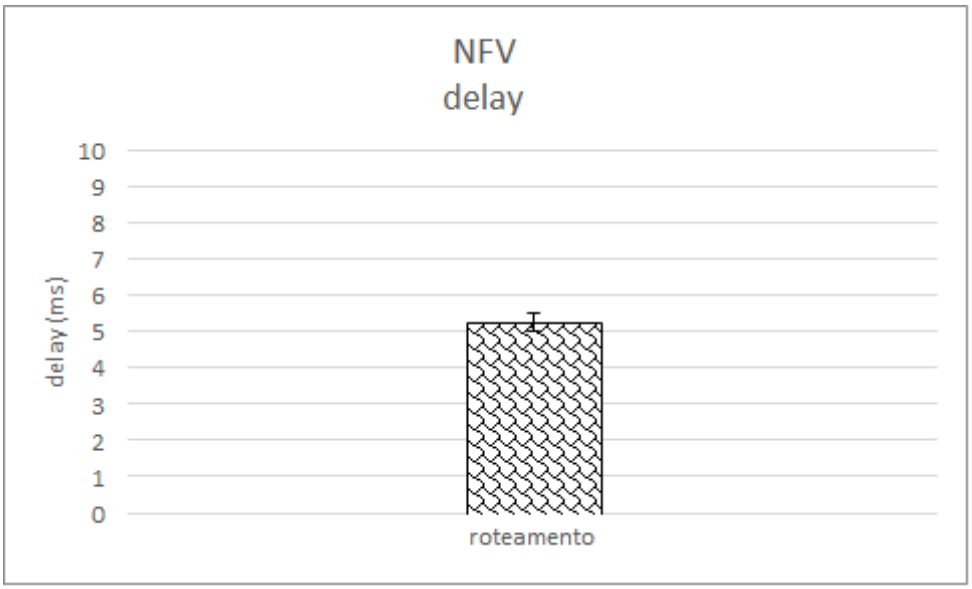


Fonte: Autoria Própria.

7.2.3 Roteamento

Na avaliação do atraso desta função, conforme resultado mostrado na Figura 34, percebe-se um atraso similar a função de *switching* e substancialmente mais elevado em relação ao *dhcp-server*. Dessa forma observa-se que os paradigmas SDN/NFV possuem comportamentos semelhantes em relação aos resultados das métricas avaliadas para essa função. Esta diferença também aconteceu em NFV já que é preciso que este paradigma demande maior inteligência e esforço computacional para realizar a comunicação entre redes diferentes. Consequentemente, o atraso da função, que executa na camada de rede, avaliada tende a ser superior a funções mais próximas do usuário, como por exemplo, a de um servidor DHCP, o qual executa na camada de aplicação.

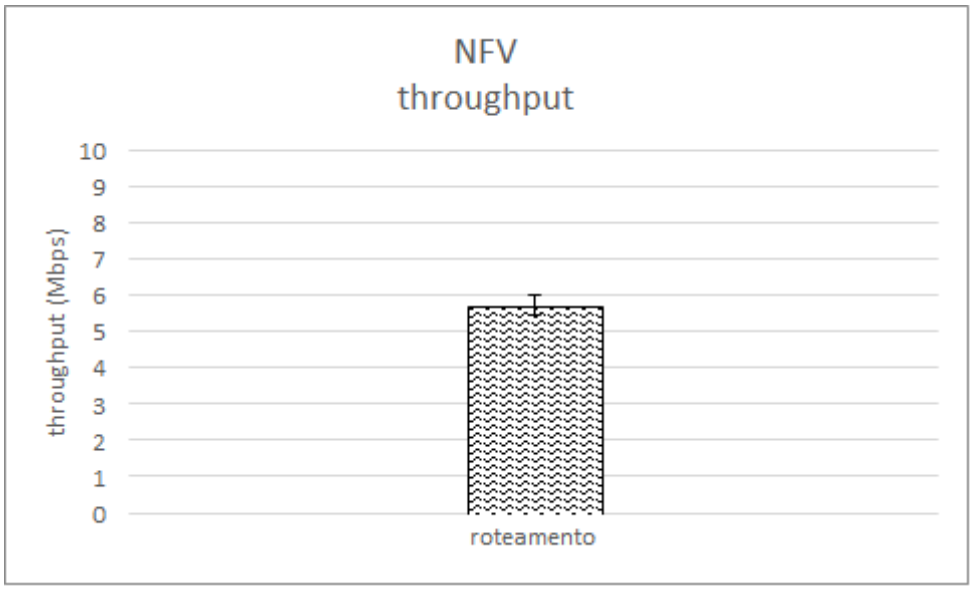
Figura 34 – Delay da função roteamento implementada em NFV.



Fonte: Autoria Própria.

Também observou-se comportamento semelhante entre os paradigmas SDN e NFV na avaliação da vazão desta função. Ainda, conforme Figura 35, observa-se que o resultado da vazão desta função foi maior que no servidor DHCP. Nesse caso, assim como em SDN, o paradigma NFV possui a inteligência em lidar com funções de diferentes camadas e finalidades. Além do mais, NFV aprovisiona maior taxa de transferência para as funções que necessitam de maior flexibilidade e poder de processamento. Dessa forma, o encaminhamento de pacotes torna-se mais ágil para estas funções.

Figura 35 – Throughput da função roteamento implementada em NFV.

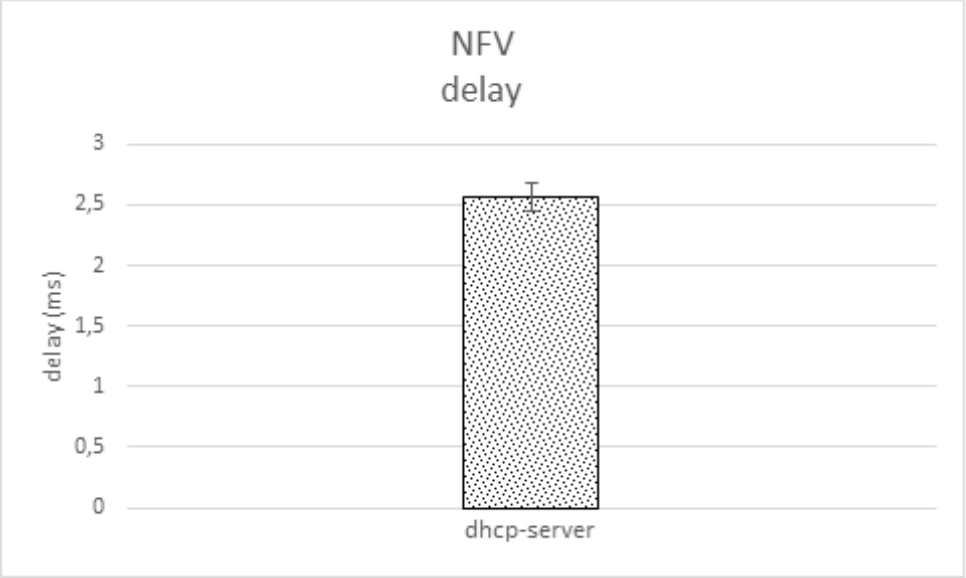


Fonte: Autoria Própria.

7.2.4 DHCP-Server

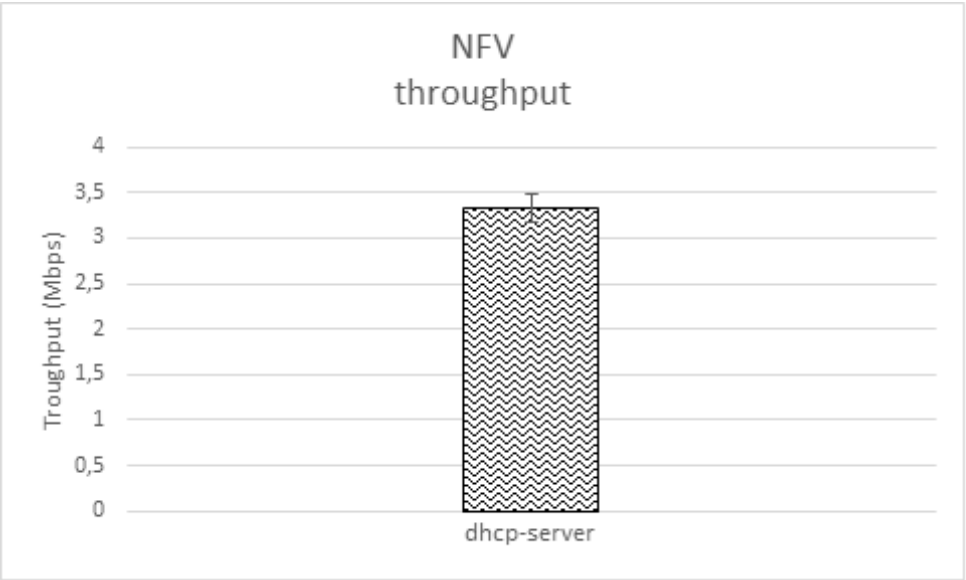
Conforme as Figuras 36 e 37 respectivamente, observa-se resultados semelhantes na avaliação do atraso e da vazão desta função executando em ambos os paradigmas. Ainda, mais especificadamente, verificou-se que em NFV o atraso foi um pouco inferior e a vazão um pouco superior em contraste a SDN. Dentro deste contexto, a execução desta função possui forte tendência em manter um mesmo comportamento em ambos os paradigmas. Nesta perspectiva, constata-se que esta função seja mais previsível para o administrador de redes quando implementada e executada em qualquer um dos paradigmas. Por fim, cabe mencionar que não houve perda de pacotes na avaliação desta função.

Figura 36 – Delay da função dhcp-server implementada em NFV.



Fonte: Autoria Própria.

Figura 37 – Throughput da função dhcp-server implementada em NFV.



Fonte: Autoria Própria.

7.3 ANÁLISE DOS RESULTADOS

Em relação aos resultados obtidos nas avaliações das funções de rede implementada nos paradigmas SDN e NFV, percebeu-se tendências de comportamentos semelhantes e por vezes diferentes na execução destas funções em ambos os paradigmas. Em relação ao *firewall*, na avaliação do atraso nos protocolos TCP/UDP, observa-se comportamentos semelhantes dos paradigmas SDN e NFV, pois a medida que aumenta-se o tamanho dos pacotes, também é visível um aumento do atraso no processamento dos mesmos.

Também percebe-se comportamentos distintos destes protocolos em tais paradigmas. Nesse contexto, em SDN constata-se maior atraso em todos os tamanhos de pacotes no protocolo TCP. Já em NFV o maior atraso em todos os tamanhos de pacotes dá-se no protocolo UDP. Sendo assim, o paradigma SDN necessita de maior esforço para realizar o processamento do protocolo TCP. Em contrapartida, NFV possui maior dificuldade para realizar o processamento do protocolo UDP.

A avaliação do atraso das aplicações testadas nos paradigmas SDN e NFV mostrou tendências similares, já que para aplicações mais leves (*ping* e DNS) o atraso foi menor do que o verificado para as mais pesadas (VOZ e jogo *battlefield*). Contudo, observou-se que em SDN a diferença do atraso entre os dois tipos de aplicações é significativamente grande. Já em NFV, esta diferença é mínima, pois como NFV possui maior capacidade de processar tráfegos pesados, logo o processamento dessas aplicações também pode ser realizado de forma mais rápida. Assim, a diferença do atraso entre as aplicações leves e pesadas foi aproximadamente de apenas 1ms.

Além disso, também observou-se que o atraso do jogo *battlefield* foi o maior entre as aplicações testadas em ambos os paradigmas. Nesse sentido, o atraso deste jogo foi um pouco maior do que o tráfego de VOZ, pois aplicações desta natureza costumam ser bastante dinâmicas e frequentemente demandam maior capacidade computacional. Dessa forma, necessita-se de maior processamento do controlador SDN para realizar a instalação dos fluxos no comutador virtual, consequentemente o atraso tende a ser maior.

Em relação a análise da vazão dos protocolos TCP e UDP executando em SDN e NFV, nota-se um comportamento padrão dos protocolos. Tendo isto em vista, a medida que aumenta-se o tamanho dos pacotes, também há um incremento da vazão. Tal fato aconteceu para auxiliar o processamento e consequentemente o encaminhamento dos pacotes pela infraestrutura de

rede. Além disso, em NFV percebe-se uma tendência de linearidade nos valores obtidos em ambos os protocolos. Nesse contexto, o paradigma NFV mantém comportamentos semelhantes para esta métrica em relação aos dois protocolos da camada de transporte verificados.

No que tange os valores obtidos na avaliação da vazão das aplicações testadas, observou-se comportamentos semelhantes entre as aplicações de mesma natureza executando nos paradigmas SDN e NFV. Nesse contexto, as aplicações leves (*ping* e DNS) obtiveram vazões baixas, já as aplicações mais densas (VOZ e jogo *battlefield*) obtiveram vazões significativamente maiores. Ainda, percebeu-se que o tráfego de VOZ obteve a maior vazão em ambos os paradigmas. Este resultado ocorreu já que em ambos os paradigmas o atraso do tráfego de VOZ foi menor em relação ao do jogo em questão. Sendo assim, o processamento nesta aplicação pode ser realizado mais rapidamente. Dessa forma, esta aplicação obteve uma vazão um pouco superior em relação ao jogo analisado.

Em relação a perda de pacotes dos protocolos TCP e UDP registrada nos paradigmas SDN e NFV, percebe-se que em ambos os protocolos esta perda pode ser considerada desprezível. Pode-se ponderar isso, já que a mesma não representa nem 0,2% do total de pacotes trafegados na função analisada em questão. Além disso, em ambos os paradigmas, também observa-se que conforme aumenta-se o tamanho dos pacotes, também há um incremento da porcentagem das perdas. Ainda, a medida que varia-se o tamanho dos pacotes, verifica-se uma tendência de linearidade desta métrica em ambos os protocolos, principalmente, no TCP.

Outro comportamento semelhante verificado nos dois paradigmas é que as perdas do protocolo UDP são maiores em todos os tamanhos de pacotes. Isto ocorre já que este protocolo não é orientado a conexão e portanto não tem garantia na entrega dos pacotes. Dessa forma, nota-se uma considerável semelhança do comportamento dos protocolos TCP/UDP testados na função de *firewall* em ambos os paradigmas no que tange a avaliação desta métrica.

Com a perda de pacotes verificada nas aplicações, observou-se comportamentos semelhantes nos paradigmas SDN e NFV. Isso foi constatado já que em ambos os paradigmas as maiores perdas de pacotes foram registradas nas aplicações em contraste aos protocolos TCP/UDP. Esse fato acontece já que estas aplicações são sensivelmente mais complexas de serem processadas. Diante disso, a tendência é que as mesmas acabem registrando maiores perdas de pacotes em comparação com os protocolos TCP/UDP.

Ainda, outro comportamento similar apresentado por ambos os paradigmas na avaliação desta métrica é que não registrou-se perdas de pacotes nas aplicações *ping* e DNS. Devido ao

tráfego dessas aplicações ser leve, assim torna-se mais fácil de ser processado. Dessa forma, a perda de pacotes em aplicações desta natureza é mais incomum de acontecer.

No que diz respeito aos resultados obtidos na avaliação do atraso da função de *switching*, observa-se diferentes comportamentos dos paradigmas SDN e NFV. Nesse sentido, em SDN percebe-se que o valor resultante desta métrica é um pouco inferior que o atraso verificado na função de roteamento. Esta diferença é verificada já que na função de *switching* a comunicação entre os *hosts* ocorre dentro da mesma rede. Assim, o controlador SDN não precisa dispender considerável esforço computacional a fim de aprender diferentes rotas para poder encaminhar os pacotes pelas diferentes redes.

Seguindo dentro deste contexto, com o paradigma SDN verifica-se que o processo de comutação tende a ser mais rápido que o de roteamento. Em contrapartida, de acordo com os resultados obtidos em NFV, percebe-se que este paradigma não faz distinção no processamento entre funções de comutação e roteamento. Dessa maneira, através desse comportamento semelhante percebido em NFV, não ocorre diferença significativa na medição do atraso entre estas duas importantes funcionalidades da área de redes de computadores.

Ainda em relação a função de *switching*, na análise da vazão observa-se comportamentos semelhantes entre os paradigmas SDN e NFV. Pode-se constatar isso já que em ambos os paradigmas percebe-se uma vazão significativamente maior em relação a vazão resultante da função de roteamento. Tal comportamento é analisado nos dois paradigmas, já que o processamento e encaminhamento dos pacotes dá-se entre *hosts* que pertençam a mesma rede. Dessa maneira, a taxa de transferência resultante deste processamento em ambos os paradigmas, acaba sendo mais expressiva para a função de comutação em detrimento de roteamento.

Em relação a perda de pacotes registrada no teste da função de *switching*, observou-se comportamentos semelhantes entre os paradigmas SDN e NFV. Isso foi possível já que em ambos os paradigmas a perda de pacotes registrada foi aproximadamente de apenas 0,01%, portanto podendo ser considerada desprezível. Nesse sentido, verificou-se em SDN através do *Wireshark* que pacotes mal formados foram perdidos. Em NFV, observou-se por meio do mesmo *sniffer* de rede, que pacotes foram perdidos pois estavam fora de ordem.

No que tange a avaliação do atraso na função de roteamento, observou-se comportamentos semelhantes nos resultados obtidos nesta função executando nos paradigmas SDN e NFV. Nesse sentido, percebe-se que o atraso nesta função é substancialmente superior quando comparado com o registrado no servidor DHCP. Esta importante diferença ocorre já que para

acontecer o processo de roteamento é preciso a comunicação entre *hosts* localizados entre redes diferentes. Assim, através de soluções baseadas em *software* e por meio da virtualização, é preciso maximizar questões inerentes a flexibilidade e escalabilidade a fim de dispender maior poder de processamento em ambos os paradigmas. Com isso, o atraso na função de roteamento é superior ao de funções mais próximas dos usuários finais, como é o caso do servidor DHCP.

Ainda no contexto da função de roteamento, em relação a avaliação da vazão nesta função, percebeu-se comportamentos semelhantes entre os paradigmas SDN e NFV. Dentro deste contexto, observou-se que o resultado da vazão desta função foi maior em comparação com o servidor DHCP. Este resultado aconteceu devido a inteligência que ambos os paradigmas possuem de priorizar determinados tráfegos em detrimento de outros. Além do mais, o paradigma NFV tem como importante característica o aprovisionamento de maiores taxas de transferências para maximizar a execução de determinadas aplicações. Por fim, também salienta-se que não houve perda de pacotes na avaliação desta função.

No que tange a avaliação do atraso e da vazão na função do servidor DHCP, notou-se comportamentos bastante semelhantes nesta função executando nos paradigmas SDN e NFV. Tendo isto em vista, observou-se que os valores de atraso obtidos em ambos os paradigmas são bastante similares. Ainda, relembra-se que a finalidade do servidor DHCP é apenas fornecer endereçamento IP de forma dinâmica aos clientes. Nesse sentido, não é necessário que esta função obtenha valores significativos de vazão. Por isso, os resultados obtidos na medição da vazão em ambos os paradigmas são pequenos. Diante das semelhanças já observadas na execução desta função em ambos os paradigmas, esta função acaba sendo mais previsível para o administrador de redes quando implementada em qualquer um dos paradigmas. Também ressalta-se que não houve perda de pacotes na avaliação da mesma. Desta forma, a seguir apresenta-se a Tabela 3 atualizada com a dimensão de desempenho. Na próxima seção será discorrido as principais considerações parciais do presente trabalho.

Tabela 3 – Tabela Atualizada das Categorias de Funções x Dimensões.

Categorias/ Dimensões	Segurança	Implementação/ Programabilidade	Gerenciamento	Gestão Dinâmica dos Recursos	Disponibilidade/ Resiliência	Desempenho
Aplicação	SDN:2 NFV:3	SDN:2 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:2	SDN:2 NFV:3
Interoperabilidade	SDN:1 NFV:1	SDN:2 NFV:1	SDN:1 NFV:1	SDN:2 NFV:1	SDN:1 NFV:1	SDN:2 NFV:2
Otimização	SDN:2 NFV:3	SDN:3 NFV:2	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:2	SDN:2 NFV:3
Proteção	SDN:2 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:3	SDN:2 NFV:3	SDN:2 NFV:3
Monitoramento/ Controle	SDN:2 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:3 NFV:3	SDN:2 NFV:3

Fonte: Autoria Própria.

7.4 CONSIDERAÇÕES PARCIAIS

De acordo com a tabela 3 pode-se observar que para a categoria de aplicação o paradigma NFV mostra-se mais adequado para a implementação de funções desta categoria. Em contrapartida, para interoperabilidade, SDN em detrimento de NFV contempla uma abordagem um pouco maior das dimensões avaliadas desta categoria. Todavia, funcionalidades inerentes a categoria de interoperabilidade são mais indicadas para a implementação no paradigma SDN.

Para as categorias de otimização e de monitoramento/controle, entende-se que os dois paradigmas podem ser substituídos um pelo outro, já que obtiveram resultados parecidos. Isso é perceptível analisando-se as diferentes dimensões avaliadas nestas duas categorias, onde os níveis de aplicabilidade em ambos os paradigmas em várias destas dimensões foram classificados como aplicáveis e fortemente aplicáveis.

No que tange as funções de proteção, observa-se que NFV obteve melhores resultados nas dimensões de segurança, gestão dinâmica dos recursos, disponibilidade/resiliência e desempenho. Principalmente para a dimensão de segurança, ressalta-se que NFV é mais adequada que SDN pois aquela permite o cascateamento de funções. O cascateamento permite que várias funções desta categoria colaborem entre si a fim de prover um nível de segurança mais efetivo para uma determinada aplicação ou serviço em específico. Como exemplo, um *firewall* pode trabalhar em conjunto com um sistema IDS/IPS, onde cada funcionalidade executa em uma

VM, sendo que o fluxo destas funcionalidades é encaminhado para as demais, comportando-se como um FG. Portanto, além da segurança, bem como as demais dimensões como um todo, a implementação de funções de proteção é mais vantajosa de ser realizada em NFV.

Outra constatação importante que deve-se ressaltar é que embora ambos os paradigmas possam trabalhar de forma integrada, ainda assim será responsabilidade do desenvolvedor o desafio de saber identificar qual paradigma é mais vantajoso, adequado e efetivo para a implementação de uma determinada função de rede em específico ou de forma mais abrangente, de uma categoria de função de rede.

Por fim, também destaca-se que as tendências e padrões de comportamentos verificados nesta pesquisa se mantém, mesmo quando os testes são realizados em equipamentos e ambientes diferentes dos disponíveis neste trabalho. Portanto, as constatações observadas na avaliação das funções de rede executando em ambos os paradigmas permanecem os mesmos, apenas percebe-se pequenas alterações nos valores finais das métricas avaliadas em questão.

8 CONCLUSÕES

O uso de equipamentos dedicados na implementação de funções de redes apresenta alto custo de aquisição, necessidade de mão de obra especializada, gastos relativos as operações e manutenções dos mesmos, etc. Além disso, devido ao *software* e ao *hardware* verticalmente integrados e proprietários, percebeu-se que o uso desses equipamentos dificulta a inovação nas redes de computadores. Nesse contexto, pode-se utilizar os paradigmas SDN e NFV para implementar tais funções em *software* por meio de VM controladas por um *hypervisors* instalados em servidores de uso genérico.

Ainda dentro deste contexto, soluções baseadas em *software* e virtualização possibilitam maior elasticidade, flexibilidade, escalabilidade e provisionamento da rede, permitindo que novas funções e serviços sejam rapidamente adotados. Tendo isto em vista, sabe-se que os paradigmas SDN e NFV podem ser utilizados para a implementação e execução de um mesmo conjunto de funções de redes. Contudo, não se tinha até então o conhecimento concreto de quais funções de redes de diferentes finalidades seriam mais adequadas e vantajosas de serem implementadas e executadas em determinado paradigma.

Dessa forma, para responder a esse problema de pesquisa, no presente trabalho foi apresentado e discutido diversas métricas qualitativas e quantitativas. Além do mais, foi proposta uma taxonomia para funções de rede. Tal taxonomia foi categorizada em cinco áreas funcionais. A partir disso, foi realizada a avaliação de diferentes funções de rede em ambos os paradigmas. As funções implementadas foram: *firewall, switching*, roteamento e servidor DHCP. Estas pertencem a diferentes categorias de funções de rede da taxonomia proposta, bem como atuam em diferentes camadas e possuem características e finalidades distintas. Em tal avaliação procedeu-se com uma significativa análise de tendências, comportamentos e padrões destas funções executando em ambos os paradigmas, sendo posteriormente apresentada uma discussão dos resultados alcançados.

Diante disso, muitos trabalhos podem ser encontrados na literatura envolvendo propostas e implementações de novas soluções de redes utilizando-se os paradigmas SDN e NFV de forma separada e/ou integrada. Entretanto, tais trabalhos disponíveis na literatura, não realizam análise e discussões aprofundadas sobre tais propostas. Ainda, observa-se que tais trabalhos possuem maior destaque para a implementação de determinada arquitetura e a realização de testes, portanto focando mais nos aspectos quantitativos.

A presente pesquisa diferencia-se destes trabalhos, já que tem como principal contribuição a análise qualitativa e quantitativa das funções de redes executando nos paradigmas SDN e NFV. Além disso, é levado em consideração em quais casos, situações e cenários cada paradigma é mais adequado e vantajoso de ser utilizado para implementar funções de diferentes categorias sob o ponto de vista das dimensões apresentadas.

No que tange a metodologia realizada para a concretização do presente trabalho, foi elaborado cenários de testes para os paradigmas SDN e NFV. Sendo assim, definiu-se que seria utilizado alguns protocolos e aplicações atuantes em diferentes camadas, a fim de observar-se padrões, comportamentos, tendências em ambos os paradigmas.

Os protocolos utilizados foram o TCP e o UDP. Já as aplicações testadas foram o DNS, *ping*, VOZ e tráfego do jogo *battlefield*. Para a execução das VM foi utilizada a plataforma de virtualização KVM. Para o cenário SDN criou-se 4 VM, sendo duas para os *hosts*, uma para o controlador POX e uma para o OVS. Já para o cenário NFV, criou-se 3 VM, duas para os *hosts* e uma para o sistema de virtualização NFV, o qual utilizou-se a plataforma OSv juntamente com o *Click Modular Router*. As conexões entre as VM foram estabelecidas utilizando-se *Linux bridges*.

Para a realização da avaliação do *firewall*, foram realizados testes dos protocolos TCP e UDP variando-se o tamanho dos pacotes. Ainda, também testou-se as aplicações DNS, *ping*, VOZ e o tráfego do jogo *battlefield*. Nesta função foram avaliadas as métricas de atraso, *jitter*, vazão e perda de pacotes. Para o servidor DHCP avaliou-se o atraso considerando-se o instanciamento da VM executando esta função até a entrega do endereçamento IP de forma dinâmica aos clientes. Além disso, também analisou-se a vazão que esta função fornece na execução deste serviço. Para a função de roteamento, verificou-se o atraso e a vazão no processamento da aplicação *ping*. Ressalta-se que para testar esta funcionalidade variou-se o endereço IP de origem com destino ao endereço do *host* 2. Destaca-se que para estas duas últimas funcionalidades não foram registradas perdas de pacotes.

Já para a avaliação de *switching*, analisou-se as métricas de atraso, vazão e perda de pacotes no processamento da aplicação *ping*. Salienta-se que para o teste desta função manteve-se o mesmo endereço IP de origem com destino ao endereço do *host* 2. Tais testes fazem parte da análise sob o ponto de vista quantitativo, onde o principal objetivo é avaliar o desempenho de tais funções de acordo com os resultados alcançados nos testes.

Contudo, com o objetivo de complementar estas métricas relacionadas ao desempenho,

elencou-se aspectos qualitativos. Tendo isto em vista, foi apresentado e discorreu-se sobre os seguintes aspectos: Segurança; Implementação/Programabilidade; Gerenciamento; Gestão Dinâmica dos Recursos; Disponibilidade/Resiliência e Desempenho sob um ponto de vista qualitativo.

No decorrer da implementação das funções de rede, algumas dificuldades foram encontradas para a realização das mesmas e consequentemente deste trabalho. Nesse sentido, em relação as funções de interoperabilidade, a implementação destas funções utilizando-se os paradigmas SDN e NFV ainda é pouco explorada na literatura em contraste com funções de outras categorias. Ainda, a carência de uma abordagem prática para poder embasar e fundamentar a programabilidade de funções desta categoria, principalmente no paradigma NFV, dificulta uma avaliação mais concreta e efetiva de funções desta categoria.

Dentro deste contexto, funções de otimização em NFV são fracamente exploradas pela literatura tanto a nível teórico quanto prático, sendo mais abordadas em pesquisas envolvendo o paradigma SDN. Todavia ressalta-se que embora não tenha sido realizado a avaliação da categoria de interoperabilidade, avaliou-se funções das demais categorias, o que propiciou uma análise significativa. Justifica-se essa constatação, pois foi possível avaliar e observar o comportamento de funções que operam em diferentes camadas com finalidades específicas.

Como lições aprendidas e novos conhecimentos agregados na realização deste trabalho, diferentemente do que poderia se pensar no início da pesquisa, percebe-se que não existe uma única tecnologia exclusiva para ser implementada determinada função de rede. Tendo isto em vista, ambos os paradigmas podem ser aplicados na implementação de funções de rede pertencentes a diferentes categorias. Através deste trabalho foi possível concluir quando de fato determinado paradigma é mais adequado/vantajoso para ser usado para a implementação e execução de uma determinada função de rede ou categoria de função de rede. Com isso, através dos resultados alcançados e discussões realizadas, acredita-se que o problema de pesquisa definido no início deste trabalho foi respondido com sucesso.

Além disso, foi possível constatar de forma teórica e prática que ambos os paradigmas possuem vantagens e desvantagens, onde em alguns aspectos um paradigma acaba sendo melhor avaliado em detrimento do outro. Ademais, foi possível notar que ambos os paradigmas possuem significativa semelhança em muitos aspectos avaliados. Dessa forma, para a implementação de funções pertencentes a categorias que obtiveram resultados semelhantes, pode-se substituir os paradigmas um pelo outro.

Em suma, apesar dos paradigmas SDN e NFV serem relativamente recentes, principalmente este último, pode-se observar nestes últimos anos o expressivo sucesso e expansão alcançados por ambos tanto na indústria quanto em pesquisas na academia. Nesse sentido, percebe-se que estes já estão propiciando muitos avanços para o desenvolvimento de novas aplicações, funções e arquiteturas de redes para a área da computação como um todo.

Por fim, como sugestão de trabalhos futuros, pode-se citar a implementação e a avaliação de novas funções de redes. Além disso, a realização de testes de funções de redes com a utilização de equipamentos reais. Ademais, também seria oportuno realizar os testes destas funções em ambientes de experimentação em larga escala (*testbeds*).

REFERÊNCIAS

ALHARBI, T.; ALJUHANI, A.; LIU, H. Holistic DDoS mitigation using NFV. In: IEEE 7TH ANNUAL COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE (CCWC), 2017. **Anais...** [S.l.: s.n.], 2017. p.1–4.

ALIYU, A. L.; BULL, P.; ABDALLAH, A. Performance Implication and Analysis of the OpenFlow SDN Protocol. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS (WAINA), 2017. **Anais...** [S.l.: s.n.], 2017. p.391–396.

ALJUHANI, A.; ALHARBI, T. Virtualized Network Functions security attacks and vulnerabilities. In: IEEE 7TH ANNUAL COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE (CCWC), 2017. **Anais...** [S.l.: s.n.], 2017. p.1–4.

BATALLE, J. et al. On the Implementation of NFV over an OpenFlow Infrastructure: routing function virtualization. In: IEEE SDN FOR FUTURE NETWORKS AND SERVICES (SDN4FNS), 2013. **Anais...** [S.l.: s.n.], 2013. p.1–6.

BECK, M. T.; BOTERO, J. F.; SAMELIN, K. Resilient allocation of service Function chains. In: IEEE CONFERENCE ON NETWORK FUNCTION VIRTUALIZATION AND SOFTWARE DEFINED NETWORKS (NFV-SDN), 2016. **Anais...** [S.l.: s.n.], 2016. p.128–133.

BOUBENDIR, A.; BERTIN, E.; SIMONI, N. NaaS architecture through SDN-enabled NFV: network openness towards web communication service providers. In: NOMS 2016 - 2016 IEEE/IFIP NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM. **Anais...** [S.l.: s.n.], 2016. p.722–726.

BRADNER, S.; MCQUAID, J. Benchmarking methodology for network interconnect devices. In: **Anais...** [S.l.: s.n.], 1999.

CARPENTER, B.; BRIM, S. Middleboxes: taxonomy and issues. In: **Anais...** [S.l.: s.n.], 2002.

CHIOSI, M. et al. Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action, [S.l.], p.1–16, 2012.

CHOWDHURY, N. M. M. K.; BOUTABA, R. Network virtualization: state of the art and research challenges. **IEEE Communications Magazine**, [S.l.], v.47, n.7, p.20–26, July 2009.

DABBAGH, M. et al. Software-defined networking security: pros and cons. **IEEE Communications Magazine**, [S.l.], v.53, n.6, p.73–79, June 2015.

DENG, J. et al. VNGuard: an nfv/sdn combination framework for provisioning and managing virtual firewalls. In: IEEE CONFERENCE ON NETWORK FUNCTION VIRTUALIZATION AND SOFTWARE DEFINED NETWORK (NFV-SDN), 2015. **Anais...** [S.l.: s.n.], 2015. p.107–114.

EDELINE, K.; DONNET, B. Towards a middlebox policy taxonomy: path impairments. In: IEEE CONFERENCE ON COMPUTER COMMUNICATIONS WORKSHOPS (INFOCOM WKSHPS), 2015. **Anais...** [S.l.: s.n.], 2015. p.402–407.

FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The Road to SDN: an intellectual history of programmable networks. **SIGCOMM Comput. Commun. Rev.**, New York, NY, USA, v.44, n.2, p.87–98, Apr. 2014.

GELBERGER, A.; YEMINI, N.; GILADI, R. Performance Analysis of Software-Defined Networking (SDN). In: IEEE 21ST INTERNATIONAL SYMPOSIUM ON MODELLING, ANALYSIS AND SIMULATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS, 2013. **Anais...** [S.l.: s.n.], 2013. p.389–393.

GUEDES, D. et al. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em redes de computadores. **Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC 2012**, [S.l.], v.30, n.4, p.160–210, 2012.

GUPTA, V.; KAUR, S.; KAUR, K. Implementation of stateful firewall using POX controller. In: INTERNATIONAL CONFERENCE ON COMPUTING FOR SUSTAINABLE GLOBAL DEVELOPMENT (INDIACOM), 2016. **Anais...** [S.l.: s.n.], 2016. p.1093–1096.

HAKIRI, A. et al. Software-Defined Networking: challenges and research opportunities for future internet. **Computer Networks**, [S.l.], v.75, n.Part A, p.453 – 471, 2014.

HAN, B. et al. Network function virtualization: challenges and opportunities for innovations. **IEEE Communications Magazine**, [S.l.], v.53, n.2, p.90–97, Feb 2015.

HERRERA, J. G.; BOTERO, J. F. Resource Allocation in NFV: a comprehensive survey. **IEEE Transactions on Network and Service Management**, [S.l.], v.13, n.3, p.518–532, Sept 2016.

HERRERA, J. G.; VEGA, J. F. B. Network Functions Virtualization: a survey. **IEEE Latin America Transactions**, [S.l.], v.14, n.2, p.983–997, Feb 2016.

HU, F.; HAO, Q.; BAO, K. A Survey on Software-Defined Network and OpenFlow: from concept to implementation. **IEEE Communications Surveys Tutorials**, [S.l.], v.16, n.4, p.2181–2206, Fourthquarter 2014.

KABLAN, M. et al. Stateless Network Functions. In: ACM SIGCOMM WORKSHOP ON HOT TOPICS IN MIDDLEBOXES AND NETWORK FUNCTION VIRTUALIZATION, 2015., New York, NY, USA. **Proceedings...** ACM, 2015. p.49–54. (HotMiddlebox '15).

KAUR, S.; SINGH, J.; GHUMMAN, N. S. Network programmability using POX controller. In: ICCCS INTERNATIONAL CONFERENCE ON COMMUNICATION, COMPUTING & SYSTEMS, IEEE. **Anais...** [S.l.: s.n.], 2014. n.s 134, p.138.

KIM, G.; KIM, J.; LEE, S. An SDN based fully distributed NAT traversal scheme for IoT global connectivity. In: INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2015 INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2015. p.807–809.

KIM, H.; FEAMSTER, N. Improving network management with software defined networking. **IEEE Communications Magazine**, [S.l.], v.51, n.2, p.114–119, February 2013.

KIM, T.; KOO, T.; PAIK, E. SDN and NFV benchmarking for performance and reliability. In: ASIA-PACIFIC NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (APNOMS), 2015. **Anais...** [S.l.: s.n.], 2015. p.600–603.

KREUTZ, D. et al. Software-Defined Networking: a comprehensive survey. **Proceedings of the IEEE**, [S.l.], v.103, n.1, p.14–76, Jan 2015.

LAL, S.; TALEB, T.; DUTTA, A. NFV: security threats and best practices. **IEEE Communications Magazine**, [S.l.], v.PP, n.99, p.2–8, 2017.

LIU, Y. et al. To Achieve a Security Service Chain by Integration of NFV and SDN. In: SIXTH INTERNATIONAL CONFERENCE ON INSTRUMENTATION MEASUREMENT, COMPUTER, COMMUNICATION AND CONTROL (IMCCC), 2016. **Anais...** [S.l.: s.n.], 2016. p.974–977.

MACHADO, C. C.; GRANVILLE, L. Z.; SCHAEFFER-FILHO, A. ANSwer: combining nfv and sdn features for network resilience strategies. In: IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATION (ISCC), 2016. **Anais...** [S.l.: s.n.], 2016. p.391–396.

MANFREDI, V.; CROVELLA, M.; KUROSE, J. Understanding Stateful vs Stateless Communication Strategies for Ad Hoc Networks. In: ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 17., New York, NY, USA. **Proceedings...** ACM, 2011. p.313–324. (MobiCom '11).

MASUTANI, H. et al. Requirements and design of flexible NFV network infrastructure node leveraging SDN/OpenFlow. In: INTERNATIONAL CONFERENCE ON OPTICAL NETWORK DESIGN AND MODELING, 2014. **Anais...** [S.l.: s.n.], 2014. p.258–263.

MATIAS, J. et al. Toward an SDN-enabled NFV architecture. **IEEE Communications Magazine**, [S.l.], v.53, n.4, p.187–193, April 2015.

MATTOS, D. M. F.; DUARTE, O. C. M. B. AuthFlow: um mecanismo de autenticaç ao e controle de acesso para redes definidas por software, [S.l.], 2014.

MCKEOWN, N. et al. OpenFlow: enabling innovation in campus networks. **SIGCOMM Comput. Commun. Rev.**, New York, NY, USA, v.38, n.2, p.69–74, Mar. 2008.

MIJUMBI, R. et al. Network Function Virtualization: state-of-the-art and research challenges. **IEEE Communications Surveys Tutorials**, [S.l.], v.18, n.1, p.236–262, Firstquarter 2016.

MIJUMBI, R. et al. Management and orchestration challenges in network functions virtualization. **IEEE Communications Magazine**, [S.l.], v.54, n.1, p.98–105, January 2016.

MONTIBELER, P.; FARIAS, F.; ABELEM, A. Fator de Resiliência para Aprimoramento Topológico em Redes Definidas por Software. **XXII Workshop de Gerência e Operação de Redes e Serviços (WGRS)-XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**, [S.l.], 2017.

MOSTOVICH, D. et al. High-level vulnerabilities of software-defined networking in the context of telecommunication network evolution. In: IEEE CONFERENCE OF RUSSIAN YOUNG RESEARCHERS IN ELECTRICAL AND ELECTRONIC ENGINEERING (EICONRUS), 2017. **Anais...** [S.l.: s.n.], 2017. p.184–186.

MULLER, A. Analysis and Control of Middleboxes in the Internet, [S.l.], 2013.

NUNES, B. A. A. et al. A Survey of Software-Defined Networking: past, present, and future of programmable networks. **IEEE Communications Surveys Tutorials**, [S.l.], v.16, n.3, p.1617–1634, Third 2014.

OBADIA, M. et al. Failover mechanisms for distributed SDN controllers. In: INTERNATIONAL CONFERENCE AND WORKSHOP ON THE NETWORK OF THE FUTURE (NOF), 2014. **Anais...** [S.l.: s.n.], 2014. v.Workshop, p.1–6.

OLAYA, M. E.; BERNAL, I.; MEJIA, D. Application for load balancing in SDN. In: EURO AMERICAN CONFERENCE ON TELEMATICS AND INFORMATION SYSTEMS (EATIS), 2016. **Anais...** [S.l.: s.n.], 2016. p.1–8.

OMNES, N. et al. A programmable and virtualized network IT infrastructure for the internet of things: how can nfv sdn help for facing the upcoming challenges. In: INTERNATIONAL CONFERENCE ON INTELLIGENCE IN NEXT GENERATION NETWORKS, 2015. **Anais...** [S.l.: s.n.], 2015. p.64–69.

POURNAGHSHBAND, V. End-to-End Detection of Third-Party Middlebox Interference, [S.l.], 2014.

RAZA, S. M. et al. Leveraging proxy mobile IPv6 with SDN. **Journal of Communications and Networks**, [S.l.], v.18, n.3, p.460–475, June 2016.

ROSA, R. et al. Network function virtualization: perspectivas, realidades e desafios. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, [S.l.], 2014.

TRAN, T. V.; AHN, H. Flowtracker: a sdn stateful firewall solution with adaptive connection tracking and minimized controller processing. In: INTERNATIONAL CONFERENCE ON SOFTWARE NETWORKING (ICSN), 2016. **Anais...** [S.l.: s.n.], 2016. p.1–5.

VENKATRAMAN, K. et al. Supervision of network through software defined networking. In: INTERNATIONAL CONFERENCE ON INFORMATION COMMUNICATION AND EMBEDDED SYSTEMS (ICICES2014). **Anais...** [S.l.: s.n.], 2014. p.1–7.

WANG, T.; XU, H.; LIU, F. Multi-resource Load Balancing for Virtual Network Functions. In: IEEE 37TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (ICDCS), 2017. **Anais...** [S.l.: s.n.], 2017. p.1322–1332.

WICKBOLDT, J. A. et al. Software-defined networking: management requirements and challenges. **IEEE Communications Magazine**, [S.l.], v.53, n.1, p.278–285, January 2015.

WOOD, T. et al. Toward a software-based network: integrating software defined networking and network function virtualization. **IEEE Network**, [S.l.], v.29, n.3, p.36–41, May 2015.

XIA, W. et al. A Survey on Software-Defined Networking. **IEEE Communications Surveys Tutorials**, [S.l.], v.17, n.1, p.27–51, Firstquarter 2015.

YAN, Q. et al. Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: a survey, some research issues, and challenges. **IEEE Communications Surveys Tutorials**, [S.l.], v.18, n.1, p.602–622, Firstquarter 2016.

YEGANEH, S. H.; TOOTOONCHIAN, A.; GANJALI, Y. On scalability of software-defined networking. **IEEE Communications Magazine**, [S.l.], v.51, n.2, p.136–141, February 2013.

ZHANG, X. et al. Scalable Network Function Virtualization for Heterogeneous Middleboxes. In: IEEE 25TH ANNUAL INTERNATIONAL SYMPOSIUM ON FIELD-PROGRAMMABLE CUSTOM COMPUTING MACHINES (FCCM), 2017. **Anais...** [S.l.: s.n.], 2017. p.219–226.