Uma Abordagem de Fatiamento de Rede entre Múltiplos Sistemas Autônomos

Rodrigo Moreira

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2021

Rodrigo Moreira

Uma Abordagem de Fatiamento de Rede entre Múltiplos Sistemas Autônomos

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Dr. Flávio de Oliveira Silva
Coorientador: Dr. Rui Luís Andrade Aguiar

Uberlândia
2021

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

M838 Moreira, Rodrigo, 1993-
2021 Uma Abordagem de Fatiamento de Rede entre Múltiplos Sistemas Autônomos [recurso eletrônico] / Rodrigo Moreira. - 2021.

Orientador: Flávio de Oliveira Silva.
Coorientador: Rui Luis Andrade Aguiar.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.te.2021.497
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Silva, Flávio de Oliveira,1970-, (Orient.). II. Aguiar, Rui Luis Andrade,1967-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Tese de doutorado, 18/2021, PPGCO				
Data:	09 de agosto de 2021	Hora de início:	08:30	Hora de encerramento:	12:56
Matrícula do Discente:	11723CCP008				
Nome do Discente:	Rodrigo Moreira				
Título do Trabalho:	Uma Abordagem de Fatiamento de Rede entre Múltiplos Sistemas Autônomos				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Sistemas de Computação				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, via videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Rafael Pasquini - FACOM/UFU; Rodrigo Sanches Miani - FACOM/UFU; Augusto José Venâncio Neto - DIMAp/UFRN; Diogo Nuno Pereira Gomes - IT-Aveiro/Portugal; Rui Luís Andrade Aguiar - IT-Aveiro/Portugal (coorientador) e Flávio de Oliveira Silva - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Augusto José Venâncio Neto - Natal/RN; Diogo Nuno Pereira Gomes e Rui Luís Andrade Aguiar - Aveiro/Portugal; Rafael Pasquini, Rodrigo Sanches Miani e Flávio de Oliveira Silva - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Flávio de Oliveira Silva, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

Ressalta-se que os examinadores Diogo Nuno Pereira Gomes e Rui Luís Andrade Aguiar por serem estrangeiros, residentes em outro país e não possuírem CPF registrado no Brasil não podem assinar a ata de defesa.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 30/08/2021, às 09:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Flávio de Oliveira Silva**, **Professor(a) do Magistério Superior**, em 30/08/2021, às 10:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rafael Pasquini**, **Professor(a) do Magistério Superior**, em 30/08/2021, às 10:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **AUGUSTO JOSÉ VENÂNCIO NETO**, **Usuário Externo**, em 30/08/2021, às 11:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2942677** e o código CRC **157978EE**.

Ao Espírito Santo do Senhor pelo consolo e alegria.
Aos meus pais por terem sido meu porto seguro.
À Larisa por ser o coração da minha razão e a razão do meu coração.

Agradecimentos

Esta tese é sobre um método de fatiamento de rede entre múltiplos Sistemas Autônomos. Eu acredito que ela é também sobre transformação. Perseguir o título Ph. D. é se transformar. A transformação profissional e pessoal são as conquistas mais significativas de todo o processo; e elas não se estabelecem somente na busca e realização dos meios de responder a um hipótese científica, elas requerem circunstâncias e pessoas que merecem apreciação e gratidão.

Primeiramente, agradeço a Deus, Autor da Vida, por ter me criado a sua imagem e semelhança com a capacidade de pensar, criar e avaliar.

Agradeço aos meus pais Donizete e Cláudia; e aos meus irmãos Thiago e Donizete-Jr pelo incentivo e amor incondicional.

Agradeço a Larissa pelo incentivo, apoio e pela compreensão nos diversos momentos dessa etapa.

Agradeço o professor e amigo Flávio de Oliveira Silva da Universidade Federal de Uberlândia por ter sido, durante toda a Pós-graduação, um agente de transformação profissional e pessoal. Gratidão pela oportunidade, orientação, suporte, conversas e ensinamentos.

Agradeço o professor Rui Aguiar do Instituto de Telecomunicações da Universidade de Aveiro pela valiosa coorientação nesta pesquisa.

Agradecimentos ao professor e amigo Pedro Frosi da Universidade Federal de Uberlândia pela ajuda, humanidade, conversas e direcionamentos ao longo da Pós-graduação.

Gratidão aos professores do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Uberlândia, em especial àqueles que me honraram com seus ensinamentos, professores: Maria Camila, Lásaro Jonas, Márcia Fernandes, Rodrigo Miani e Rafael Pasquini.

Agradecimentos à secretaria da Pós-graduação, em especial ao Erisvaldo e à Sônia pela condução diligente dos procedimentos acadêmicos.

Agradecimentos aos colegas e amigos da Pós-graduação da Universidade Federal de Uberlândia que tornaram a caminhada mais divertida: Acrísio, Pedro Damaso, Italo da Cunha e Hugo Valin. Gratidão aos amigos que a vida me deu por tornarem o tempo de descanso significativo.

Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Agradeço a Universidade Federal de Viçosa *campus* Rio Paranaíba pelo suporte funcional, pelo incentivo e viabilização da busca pelo aperfeiçoamento técnico.

Gratidão às circunstâncias, porque elas deixaram ainda mais claro que o Senhor é bom e que Sua misericórdia dura para sempre.

"God of mercy, sweet love of mine. I have surrendered to Your design.
May this offering stretch across the skies, and these Hallelujahs be multiplied."
(Nathaniel Rinehart/William Rinehart)

Resumo

O compartilhamento de recursos é comum no âmbito dos sistemas operacionais e em infraestruturas de *hardware* computacional. A comunidade científica, orientada por requisitos de flexibilidade e escalabilidade, tem expandido o conceito de compartilhamento de recursos para o âmbito das infraestruturas de redes de comunicação. Para isso, tecnologias habilitadoras como Redes Definidas por *Software*, Virtualização de Funções de Rede, Roteamento por Segmentos e paradigmas de computação em nuvem potencializam novas abordagens de compartilhamento de recursos. Compartilhar recursos de rede os organizando funcionalmente, logicamente em conectividade de rede pode ser percebido como fatiamento de rede. O fatiamento de rede recebeu esforços conjuntos da academia e indústria em virtude dos novos habilitadores tecnológicos como a programabilidade e a virtualização. O fatiamento de rede tem sido amplamente explorado no processo de especificação e padronização das novas arquiteturas de redes móveis, especificamente o *5G*. Por isso, as propostas da literatura concentram-se predominantemente em prover fatiamento de rede no contexto das redes móveis. As abordagens que propuseram fatiamento de rede para além de redes móveis mostraram-se funcionalmente incapazes de prover fatiamento de rede entre múltiplos Sistemas Autônomos (SAs). Nesse sentido, esta tese estrutura, constrói e avalia um método de fatiamento de rede entre múltiplos SAs, baseando-se nos algoritmos de roteamento da Internet. Esta tese responde a hipótese conjecturada adicionando dinamismo ao processo de fatiamento de rede ao habilitar o usuário ou administrador realizar definições

específicas da fatia de rede, como parâmetros de serviço e o caminho ao longo de roteadores da Internet para a fatia de rede. Além disso, avança o estado da arte com um arcabouço conceitual e tecnológico para realização de fatiamento recursivo. Experimentos que mensuraram a aplicabilidade, pertinência e desempenho da abordagem de fatiamento de rede foram conduzidos e contrastados com métodos tradicionais. Resultados experimentais em cenários significativos sugerem que o método desta tese supera funcionalmente e em desempenho métodos tradicionais do estado da arte. Além disso, experimentos demonstram que o dinamismo na escolha do caminho para a fatia de rede pode aprimorar a qualidade de experiência para aplicações de tempo real.

Palavras-chave: *SDN*. *NFV*. Computação em Nuvem. Fatiamento de Rede.

Abstract

Resource sharing is common within operating systems and computational hardware infrastructures. The scientific community, driven by flexibility and scalability requirements, has expanded the concept of sharing resources to the scope of communication network infrastructures. For this, enabling technologies such as Software-Defined Networking, Network Functions Virtualization, Segment Routing, and cloud computing paradigms leverage new approaches to sharing resources. Sharing network resources by organizing them functionally and logically in the network connectivity ecosystem can be recognized as a network slice. Network slicing has received joint efforts from academia and industry because of new technological enablers such as programmability and virtualization. Network slicing has been extensively explored in the specification and standardization process for new mobile network architectures, specifically *5G*. Therefore, the proposals in the literature focus predominantly on providing network slicing in the context of mobile networks. Approaches that proposed network slicing beyond mobile networks proved to be functionally ineffective in providing network slicing across multiple Autonomous Systems (ASs). Thus, this thesis builds and evaluates a network slicing method between multiple ASs based on Internet routing algorithms. This thesis answers the assumed hypothesis by adding dynamism to the network slice process by enabling the user or administrator to perform network slice specifies definitions, such as service parameters and the path along with Internet routers to the network slice. Besides, this thesis brings novelty with a conceptual and

technological framework for performing recursive slicing. Experiments measuring the applicability, relevance, and performance of the network slicing approach were conducted and contrasted with traditional methods. Experimental results in significant scenarios suggest that the method of this thesis functionally and in performance surpasses traditional state-of-the-art methods. Furthermore, experiments demonstrate that dynamism in choosing the network slice path can improve the quality of experience for real-time applications.

Keywords: *SDN. NFV. Cloud Computing. Network Slicing.*

Lista de ilustrações

Lista de tabelas

Lista de siglas

3GPP 3rd Generation Partnership Project

AS Autonomous System

ASs Autonomous Systems

ASN Autonomous System Number

API Application Programming Interface

BGP Border Gateway Protocol

BPF BSD Packet Filter

CSMF Communication Service Management Function

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

DNS Domain Name System

ETSI European Telecomunications Standards Institute

EoMPLS Ethernet over MPLS

FIB Forwarding Information Base

GRE Generic Routing Encapsulation

IX Internet Exchange

IANA Internet Assigned Numbers Authority

IP Internet Protocol

IPv6 Internet Protocol version 6

ISP Internet Service Provider

ISPs Internet Service Providers

IETF Internet Engineering Task Force

ICMPv6 Internet Control Message Protocol Version 6

IoT Internet of Things

KPI Key Performance Indicator

LLDP Link Layer Discovery Protocol

MANO Management and Orchestration

MPLS Multi Protocol Label Switching

MOM Micro-Orchestrator Manager

MO Micro-Orchestrator

MOs Micro-Orchestrators

MEC Mobile Edge Computing

NIM Network Infrastructure Manager

NBI Northbound Interface

NFV Network Functions Virtualization

NFVI Network Functions Virtualization Infrastructure

NSD Network Service Descriptor

NSTD Network Slice Template Descriptor

NANO Network and Orchestration

NGMN Next Generation Mobile Networks

NASOR Network and Slice Orchestrator

NASORs Network and Slice Orchestrators

NVGRE Network Virtualization using Generic Routing Encapsulation

NSH Network Service Header

NSMF Network Slice Management Function

NSSMF Network Slice Subnet Management Function

NECOS Novel Enablers for Cloud Slicing

OSM OpenSource MANO

OSPF Open Shortest Path First

OPI Open Policy Interface

OIB Orchestratio Information Base

DIB Domain Information Base

PNG Portable Graphics Format

QoS Quality of Service

QoE Quality of Experience

RIB Routing Information Base

RFC Request for Comments

SDN Software-Defined Networking

SDK Software Development Kit

SID Segment Identifier

SIDs Segment Identifiers

SRH Segment Routing Header

SR Segment Routing

SF Service Function

SFC Service Function Chaining

SLA Service-Level Agreement

WAN Wide Area Network

WIM WAN Infrastructure Manager

VNF Virtualized Network Function

VNFs Virtualized Network Functions

VPN Virtual Private Network

VLAN Virtual Local Area Network

VLANs Virtual Local Area Networks

VNFD Virtual Network Function Descriptor

VIM Virtualized Infrastructure Manager

VIMs Virtualized Infrastructure Managers

VxF Virtualized Everything Function

VxFs Virtualized Everything Functions

VxLAN Virtual Extensible LAN

VoIP Voice over Internet Protocol

XDP eXpress Data Path

XCP eXpress Control Plane

Sumário

Capítulo 1

Introdução

A virtualização de recursos computacionais permite que eles sejam, de forma independente, alocados a múltiplos inquilinos que se apropriam desses recursos subdivididos como fatias (JAIN; PAUL, 2013; HUSAIN et al., 2018). Essa dinâmica de compartilhamento exemplifica as capacidades do fatiamento de recursos, originalmente discutida no âmbito dos sistemas operacionais, a saber: nos sistemas operacionais de tempo compartilhado (CREASY, 1981; AHN; PARK; HUH, 2014).

A partir de 2003, em virtude da proposta *PlanetLab* de Anderson et al. (2005), Chun et al. (2003), o uso genérico e disjunto do termo "fatiamento" convergiu para um vernáculo comum, especialmente no que diz respeito a recursos computacionais e de rede (ANSAH et al., 2019). Inspirado pelas capacidades da virtualização, sobretudo da computação em nuvem, a comunidade científica e a indústria têm proposto mecanismos de fatiamento de rede sob a mesma ótica do fatiamento de recursos computacionais (AKYILDIZ; WANG; LIN, 2015; ROST et al., 2016; FOUKAS et al., 2017).

A dinâmica do fatiamento de redes considera que sejam atribuídas aos usuários parcelas de recursos físicos com isolamento garantido aos usuários, isso nos níveis de *hardware*, controle, gerenciamento e dados (SHERWOOD et al., 2010; NIKAEIN et al., 2015). Abordagens maduras como *Virtual Local Area Network (VLAN)* e *Multi Protocol Label Switching (MPLS)* se candidataram a materializar o conceito de fatiamento de rede (Afolabi et al., 2018), no entanto foram incapazes de prover a

programabilidade completa, isolamento a nível de plano de gerenciamento, controle e de dados para os usuários (BASTIN et al., 2014). Além disso, o fatiamento de rede tornou-se componente indispensável na concepção de novas redes, exigindo mecanismos e habilitadores tecnológicos de fatiamento de redes.

Inúmeras soluções de fatiamento de rede são encontradas no estado da arte, a tendência que se nota é que os esforços da comunidade foram primariamente direcionados pelas novas arquiteturas das redes móveis (RICHART et al., 2016; Afolabi et al., 2018). Por isso, são abordagens que concentram extensamente esforços no plano de dados e controle do núcleo ou acesso dessas redes (CABALLERO et al., 2019; MAHINDRA et al., 2013).

No entanto, com a mesma importância, o provimento do fatiamento de recursos de rede e computação em outros níveis da hierarquia das redes podem ser exploradas de maneira sistemática (BASIT et al., 2020). Adicionalmente, é bem estabelecido na comunidade que o gerenciamento de recursos de rede e computação não devem ser tratados de maneira singular (WUHIB; YANGGRATOKE; STADLER, 2015; MOREIRA et al., 2020), tratá-los de maneira integrada possibilita organizar os esforços para prover serviços mais específicos aos usuários.

Recair o olhar sobre esses aspectos de fatiamento, isto é, em outros níveis da hierarquia das redes, mostra-se indispensável, sobretudo no que diz respeito ao núcleo da rede *Internet Protocol (IP)*. A diversidade de *links*, métricas, atrasos, e parâmetros de confiabilidade inerentes a esse local da rede torna o processo da comunicação inter domínios, sobretudo o estabelecimento de caminhos desafiador (ZHOU; WEI; XU, 2007). Nesse sentido, prover um mecanismo de gerenciamento de conectividade entre vários domínios se mostra relevante de tratativa.

Assim, conforme a evolução do conceito de fatiamento de recursos, que recebeu expressivo impulso no *design* das redes móveis – em especial *5G* –, este trabalho propõe estender o conceito de fatiamento de rede concentrando esforços para além desse ecossistema, incluindo o núcleo da rede, notadamente a Internet.

1.1 Motivação

O núcleo da Internet precisa de novos formatos de concepção e gerenciamento. Conforme relatório da *Cisco*, até 2022, 71% do tráfego *IP* serão gerados por dispo-

sitivos móveis e *wireless*. Além disso, é previsto um expressivo aumento de usuários de Internet pelo globo, cerca de 66% de toda população, perfazendo uma cifra de 5.3 bi de assinantes com velocidade de acesso fixo médio de ≈ 110 *Mbps* (CISCO, 2018).

No mesmo sentido, a *Ericsson* assegura que em 2025 haverão 8 bi de assinantes de banda larga móvel dos quais 2.6 bi serão $5G$. O volume de dados gerado por esses assinantes será de 160 *EB* dos quais 76% é tráfego de vídeo. Por isso, a característica de orientação ao serviço e gerenciamento completo e automático da rede é mandatório em diversos níveis da rede, sobretudo porque as demandas de tráfego dos vários tipos de acesso convergem para o núcleo da rede. Assim, esforços para lidar com esse desafio são debatidos e desenvolvidos pela academia e indústria.

A infraestrutura dos *Internet Service Providers (ISPs)* é composta por uma variedade de dispositivos de fornecedores diversos colocados em domínios com níveis específicos como: *Core*, *Wide Area Network (WAN)*, Rádio e a Borda (MAYORAL et al., 2020; GARRICH et al., 2019). Além disso, a natureza do mercado de telecomunicações é fragmentada, de forma que existem diversos operadores de rede e de computação em nuvem (HILL, 2012; REFSLUND, 2016).

Essa heterogenidade de domínios e tecnologias trouxeram consigo os desafios de gerenciamento de recursos fim-a-fim (MECHTRI et al., 2017; MIJUMBI et al., 2016). Além disso, o formato de conectividade dessas infraestruturas é difícil e caro, além de propenso a consumir muito tempo na fase de estruturação e implantação dos serviços que perpassam por diversos domínios (CZIVA; PEZAROS, 2017; GUTIERREZ-ESTEVEZ et al., 2018).

Para lidar com esses desafios inúmeros habilitadores tecnológicas foram propostos. Para avançar em programabilidade, o conceito *Software-Defined Networking (SDN)* se estabeleceu em 2008 por meio do protocolo *OpenFlow* (MCKEOWN et al., 2008), inúmeros avanços ocorreram em diversos níveis da rede como a separação do plano de dados e controle. Posteriormente, o conceito de *NFV* acelerou o campo da entrega de serviços propondo um novo formato de hospedagem e entrega funções de rede sobre tecnologias de virtualização tradicionais (FIORE et al., 2017).

Também, tecnologias de computação em nuvem, e evoluções como computação de borda, passaram a compor o *framework* de concepção e padronização de

novas redes (TRAN et al., 2017). Isso se deu pois o gerenciamento de recursos de computação, anteriormente disjunto do gerenciamento de rede, passou a ser tratado de maneira holística com vistas a qualidade de serviço e programabilidade (MOREIRA et al., 2018).

O conceito de fatiamento de rede, em processo de discussão e padronização, é peça fundamental para prover características de flexibilidade e programabilidade completa para as aplicações que rodam sobre domínios e tecnologias heterogêneas. Fundamentalmente, conforme a *European Telecomunications Standards Institute* (*(ETSI)*) o fatiamento de rede refere-se a uma instância lógica da rede que possui gerenciamento independente da camada física. De forma análoga, a *Next Generation Mobile Networks (NGMN)* torna pragmático o conceito, subdividindo-o em três blocos com funcionalidades específicas (NGMN Alliance, 2016).

No mesmo sentido, a *Internet Engineering Task Force (IETF)* descreve o fatiamento de rede como "a coleção de tecnologias para criar redes lógicas e especializadas como serviço" onde cada fatia de rede lida com uma demanda de mercado específica, isto é, verticais (FOY; RAHMAN, 2017; GROUP et al., 2016; ITU, 2011).

O fatiamento de recursos de rede estabeleceu-se como uma tecnologia habilitadora para lidar com os desafios de projeto e implantação para aplicações especializadas sobre a rede, sobretudo no que se refere a escalabilidade e flexibilidade (Rost et al., 2017). O conceito de fatiamento de recursos de rede, previamente ensaiado nas propostas como a *PlanetLab* (PETERSON; CULLER; ANDERSON, 2002), *Emulab* (STOLLER et al., 2008), *X-BONE* (TOUCH et al., 2005), *OFELIA* (SUÑÉ et al., 2014) e outras (STIEMERLING et al., 2009), implementam o conceito de *testbed* geograficamente distribuído. Esses *testbeds* permitiam que fosse projetados, experimentados e avaliados novos serviços de rede sobre uma a mesma infraestrutura.

Motivado pelos recentes habilitadores tecnológicos como *SDN*, *NFV* e *Cloud/Edge Computing*, o conceito de fatiamento de rede tomou forma na especificação da rede móvel $5G$. No entanto, suas capacidades transcendem essa tecnologia de acesso, recaindo sobre o núcleo da rede de forma que podem ser explorados os conceitos de fatiamento e programabilidade nesse nível de arquitetural.

Dentre as propostas que se estabeleceram no estado-da-arte para prover o

fatiamento de rede, inúmeras basearam-se predominantemente nas tecnologias de tunelamento ou encapsulamento, quais sejam: *VLAN*, *MPLS*, *Ethernet over MPLS (EoMPLS)*, *Generic Routing Encapsulation (GRE)*, *Virtual Extensible LAN (VxLAN)*, *Network Virtualization using Generic Routing Encapsulation (NVGRE)*, *Network Service Header (NSH)* e outras (HUANG et al., 2017).

As soluções consultadas no estado da arte, provêm o isolamento do serviço de rede sob o argumento de estarem realizando o fatiamento de rede, mas possuem uma estrutura fortemente acopladas ao domínio político e tecnológico que eram implantadas. Assim, os serviços implantados sob *hardware* de rede de propósito geral em um domínio limitavam-se à fronteira tecnológica dele, ressaltando a necessidade de um mecanismo orientado a multidomínios.

Adicionalmente, os desafios inerentes ao novo formato de projeto, implementação e entrega de serviços, sob a ótica de fatias, pairam sobre a característica de múltiplos domínios, mas ainda não estão claramente resolvidos no estado-da-arte por diversas razões. Especialmente, porque cada domínio possui políticas e tecnologias próprias, isso torna a implantação do serviço sobre múltiplos domínios desafiadora.

Uma proposta de caminho para endereçar isso é: projetar e construir um mecanismo, concebido em formato de *framework*, capaz de prover, mediante programabilidade e configurabilidade, a implantação de fatias de rede que considere o cenário de conectividade inter domínios. Supõe-se que combinando tecnologias habilitadores do estado da arte para prover fatiamento de recursos torna-se possível entregar o serviço de conectividade fim-a-fim.

Com essa conectividade personalizada e fim-a-fim, na perspectiva de múltiplos *Autonomous Systems (ASs)*, espera-se habilitar a modificação no formato de concepção novos serviços e a monetização dos *ISPs*, que tradicionalmente é concentrada na comercialização de capacidade e conectividade (BANGERA; HASAN; GORINSKY, 2017; HANAFIZADEH; HATAMI; BOHLIN, 2019). As soluções do estado da arte obtiveram consideráveis avanços no fatiamento de recursos para as redes móveis, mas mostram-se tímidas no que no que se refere ao núcleo da rede *IP*.

Por isso, este trabalho propõe o *NASOR*, uma solução modular, compatível com o *framework ETSI*, com as redes legadas dos *ISPs*, orientada ao plano de

dados multidomínios e capaz de prover implantação de fatias de rede recursivas para conectividade de *Virtualized Everything Function (VxF)*.

Os mecanismos básicos do fatiamento de rede entre domínios consideram o comportamento do plano de controle dos algoritmos de roteamento, a saber: o *Border Gateway Protocol (BGP)* e o *Open Shortest Path First (OSPF)*, para o estabelecimento e configuração da fatia de rede salto por salto, e considera uma base de dados distribuída para anúncios de capacidade de rede e computação de cada domínio. Com isso, *NASOR* propõe oferecer um fatiamento de rede e torna possível a conectividade de *Virtualized Everything Functions (VxFs)* entre múltiplos domínios, resguardando a separação política e tecnológica dos domínios concebidos como *ASs*.

1.2 Objetivos e Desafios da Pesquisa

O objetivo geral desta tese é estruturar, construir e avaliar um mecanismo para o controle distribuído de fatiamento recursivo de recursos de rede e provisionamento de conectividade inter-domínios administrativos e tecnológicos da Internet.

A abordagem proposta avança o estado da arte no campo conceitual e funcional de fatiamento de rede, amplamente referenciado como *network slicing*, propondo uma alternativa às soluções que são fortemente acopladas ao domínio tecnológico sobre as quais estão inseridas. A construção da abordagem proposta orienta-se com os padrões bem estabelecidos na comunidade científica e indústria, no que diz respeito a linguagem de descrição do serviço e interfaceamento dos componentes.

Alcançar o objetivo genérico previamente descrito exige que objetivos específicos sejam granjeados de forma sistemática. Os objetivos específicos desta tese são:

- ❏ Objetivo 1: Um levantamento comparativo de propostas de fatiamento de rede;
- ❏ Objetivo 2: Propositura de um novo formato de fatiamento de redes entre múltiplos domínios;

- ❏ Objetivo 3: Estruturar um mecanismo de fatiamento de rede, como um *framework*, compatível com o plano de dados construído pelos algoritmos de roteamento da Internet;

- ❏ Objetivo 4: Construir um mecanismo, como prova de conceito, capaz de fatiar rede entre múltiplos *ASs* compatível com o *framework* de gerenciamento e orquestração da *ETSI*;

- ❏ Objetivo 5: Propor casos de uso significativos no âmbito do fatiamento de rede;

- ❏ Objetivo 6: Experimentar e Validar funcionalmente, qualitativamente e quantitativamente a prova de conceito do *framework NASOR*;

- ❏ Objetivo 7: Propor um mapeamento do fatiamento recursivo de rede sobre múltiplos *ASs* sob a ótica do *framework NFV* da *ETSI*.

Os desafios inerentes a esses objetivos são bidimensionais e podem ser descritos considerando as características fundamentais da proposta desta tese. O primeiro aspecto desses desafios, baseia-se no processo de delineamento do objeto de estudo, isso porque existem numerosas propostas no estado-da-arte que visam oferecer o fatiamento de rede.

A despeito de o conceito de fatiamento de rede ter ganhado expressivo *momentum* na especificação das redes *5G*, majoritariamente as propostas encontradas versavam sobre fatiamento de rede nesse domínio. Além disso, o processo de implantação de serviços, sobretudo a conectividade de serviços entre múltiplos domínios é genericamente abordada.

O segundo aspecto, de caráter tecnológico, refere-se à prospecção do mecanismo como candidato a resolver os problemas encontrados no estado da arte, sobretudo na sua estrutura tecnológica. Não há consenso na literatura sobre qual é o mecanismo de isolamento ideal para a separação lógica e física de fatias de rede. Nesse ponto, as propostas exploram e sustentam suas abordagens considerando diversos habilitadores tecnológicos de segregação de recursos.

Além disso, não é bem estabelecido na literatura como um mecanismo de fatiamento de rede recursivo de rede deve proceder, principalmente no que toca

delegação de competência e separação de controle. Portanto, ao driblar esses desafios a presente proposta traz um mecanismo de fatiamento recursivo de rede para prover conectividade de funções de rede em cenário de múltiplos domínios administrativos e tecnológicos. Além disso, esta tese propõe um mapeamento à luz do *framework NFV* da *ETSI*, apresentando como os componentes do *framework* poderiam ser estendidos para suportar o fatiamento de rede recursivo, que a *priori* pode ser compreendido como uma subdivisão da fatia de rede pai em sub fatias filhas.

1.3 Questões de Pesquisa

Conforme destacado anteriormente, existem inúmeras lacunas no estado-da-arte referentes ao fatiamento de rede em cenários de múltiplos domínios. Além disso, uma introdução concisa dos habilitadores tecnológicos e as abordagens da literatura permitem conjecturar uma proposta que cumpre resolver o problema em aberto.

Nesse sentido, para direcionar os esforços desta pesquisa, levantou-se questionamentos cujas respostas nortearam sua condução, assim a seguinte hipótese foi proposta: *um framework de controle distribuído, compatível com o modelo de orquestração ETSI, baseado no plano de dados BGP e no roteamento por segmentos para fatiamento de recursos de rede, é uma abordagem quantitativamente e qualitativamente apropriada para prover a implantação e conectividade lógica de funções de rede entre múltiplos domínios administrativos e tecnológicos.*

A hipótese levantada dá ensejo a questionamentos acessórios, quais sejam:

- ***Questão 1***: É possível implantar fatias de rede sobre múltiplos *ASs*?
- ***Questão 2***: Quais tecnologias para fatiamento de rede são mais adequadas para o *NASOR*?
- ***Questão 3***: Como permitir que o usuário defina parâmetros específicos para sua fatia de rede?
- ***Questão 4***: Como prover dinamicidade no processo de implantação de fatias de rede?

- ***Questão 5***: Como oferecer fatiamento recursivo de redes entre múltiplos *ASs*?

1.4 Contribuições

Esta tese apresenta um mecanismo de implantação de fatias de rede construído com habilitadores tecnológicos heterogêneos. As contribuições que este trabalho traz evidenciam-se em dois espectros: tecnológico e científico. O tecnológico diz respeito a verificação de aplicabilidade e viabilidade dos habilitadores tecnológicos que compõe presente solução. Como resultado técnico, houve a geração de bases de dados, artefatos em repositórios de versionamento e métodos inovadores como o *Packet Vision*.

Legitimado quanto a viabilidade técnica, o *framework NASOR* contribui cientificamente como um método adequado para a implantação de fatias de rede sobre roteadores da Internet baseado no roteamento por segmentos. Esta tese apresenta e descreve a avaliação do *NASOR* conduzida predominantemente por pares da comunidade científica. Tal aceitação, demonstra que o método é adequado para realizar fatiamento de rede entre múltiplos *ASs*.

No processo de especificação, construção e avaliação inúmeras contribuições são possíveis de serem pontuadas, quais sejam:

- ❏ Um mecanismo hierárquico-distribuído de implantação de fatias de recursos entre múltiplos *Autonomous System (AS)* é uma abordagem tecnológica satisfatória;
- ❏ A constatação da viabilidade técnica de se utilizar repositório de dados baseado em chave-valor como uma abordagem adequada para manter a consistência de informações de múltiplos *ASs*;
- ❏ A validação do mecanismo de roteamento por segmentos como um habilitador tecnológico adequado para o fatiamento de redes;
- ❏ Aferição da viabilidade técnica de implantar aplicações de rede de propósito geral e sua interconexão por meio do estabelecimento de redes lógicas, com vistas a privacidade e segurança;

1.5 Método

Esta seção descreve um método construído que organiza os passos adotados, os quais sustentam os argumentos corroborativos da hipótese. Nesse ponto, é necessário descrever o caráter e a natureza desta pesquisa, conforme: hipotético-dedutivo — configurado pela percepção de um problema, proposição de solução e experimentação; e quanto a natureza, original e com estilo de apresentação de algo presumivelmente melhor subjugado com procedimentos experimentais (WAZLAWICK, 2017; LAKATOS; MARCONI, 2007).

Conforme a definição de método proposto por Wazlawick (2017), a abordagem lógica construída nesta tese para alcançar os objetivos declarados no Capítulo 1 são ilustradas na Figura 1. Adicionalmente, nesta pesquisa foi adotada uma abordagem evolutiva de construção que se baseia em duas fases, quais sejam: primeira, a construção dos mecanismos de controle distribuído para estabelecimento de um plano de dados entre múltiplos domínios administrativos; segunda, aprimoramento dos mecanismos e exploração da solução com casos de usos significativos.

A primeira Fase, conforme ilustrado na Figura 1, consiste em etapas genéricas e micro atividades, que provêm saídas ao término de seu ciclo. Primeiramente, o sequenciamento lógico construído para esta tese considerou para a fase inicial uma Organização Sistemática dos Trabalhos Correlatos. Essa etapa considerou uma abrangente pesquisa bibliográfica com parâmetros específicos do campo de pesquisa dessa tese. Palavras-chaves, amplamente utilizadas em artigos de revistas e conferências, foram adotadas como entrada para os mecanismos de busca, que operam sobre bases de indexação largamente utilizadas pelo estado-da-arte. Para a escolha dos manuscritos conforme sua qualidade, observou-se: o quantitativo de citações, métrica *Qualis*[1], o fator de impacto da revista e a maturidade das conferências. Ao final, sistematizou-se, em formato de taxonômico, os trabalhos relacionados conforme as características relevantes apontadas pelo estado-da-arte. Dentre as características, foram tomadas as qualitativamente mais desejáveis para uma solução de fatiamento de rede em cenário multi domínios.

Na fase secundária, Desenho da Solução, atividades de engenharia de *soft-*

[1] O *Qualis* é um sistema de avaliação Brasileiro usado para classificar a produção científica dos programas de Pós-graduação no que se refere aos artigos publicados em periódicos científicos.

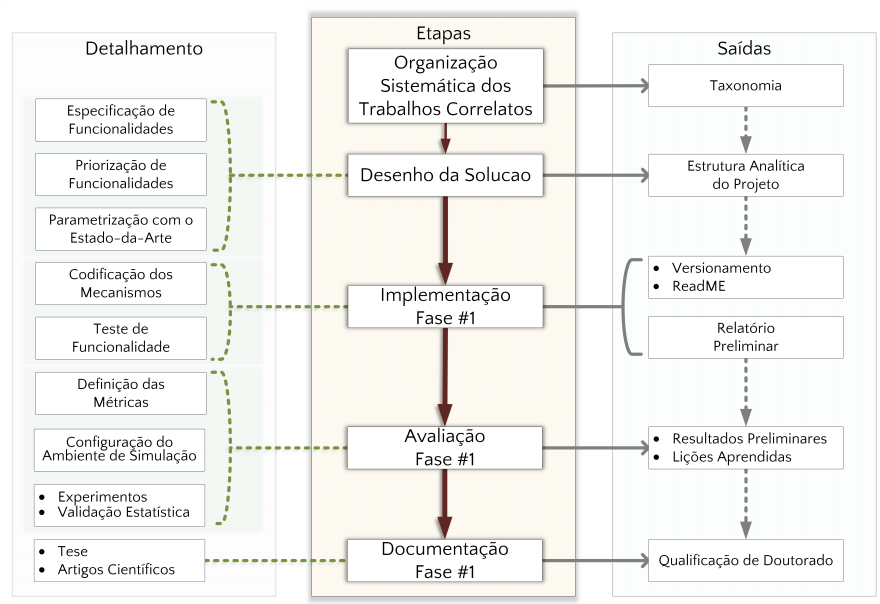

Figura 1 – Método de Pesquisa e Desenvolvimento – Fase 1.

ware são amplamente observadas, sobretudo no que se refere a especificação de funcionalidades e priorização. Nessas fases, os aspectos de desenvolvimento e a parametrização da solução com o estado da arte são amplamente analisados com vistas a produzir um documento lógico e estrutural da solução, a saber: a estrutura analítica do projeto e a diagramação das estruturas da solução.

A etapa de Implementação, compreendida na Fase 1 do desenvolvimento, consiste na codificação dos mecanismos projetados para lidar com o plano de dados e controle do fatiamento de rede. Paralelamente, são previstos nesta fase testes de funcionalidade que englobam: testes unitários, teste de integração e sistema. O término dessa etapa considera como entregáveis o versionamento do código fonte em repositório específico, bem como um documento *ReadME* que descreve a solução, seus componentes e o manual de instalação. Adicionalmente, essa fase provê um relatório preliminar que subsidia as fases seguintes de documentação.

A quarta etapa, Avaliação, consistem em testar as funcionalidades e a aplicabilidade da solução para lidar com os problemas apontados nas soluções do estado da arte. Nessa etapa, são definidas as métricas de comparação, configuração do ambiente de simulação, experimentação, validação, discussão estatística e

da adequabilidade da solução como algo presumivelmente melhor entre seus pares. O término dessa etapa provê resultados preliminares e a estruturação de lições aprendidas. Os entregáveis dessa fase são subsídios para a etapa subsequente, que se refere à documentação.

Então, a etapa de Documentação estrutura os entregáveis das fases preliminares e os organiza em documentos específicos: Tese e Artigo, ambos de gênero textual dissertativo-argumentativo. A estrutura do documento Tese é com vistas a um entregável compatível de Qualificação de Doutorado. Nele são abordados: hipótese, objetivos e resultados à sombra de experimentos preliminares que põe à prova a aplicabilidade da proposta em contraste com seus pares.

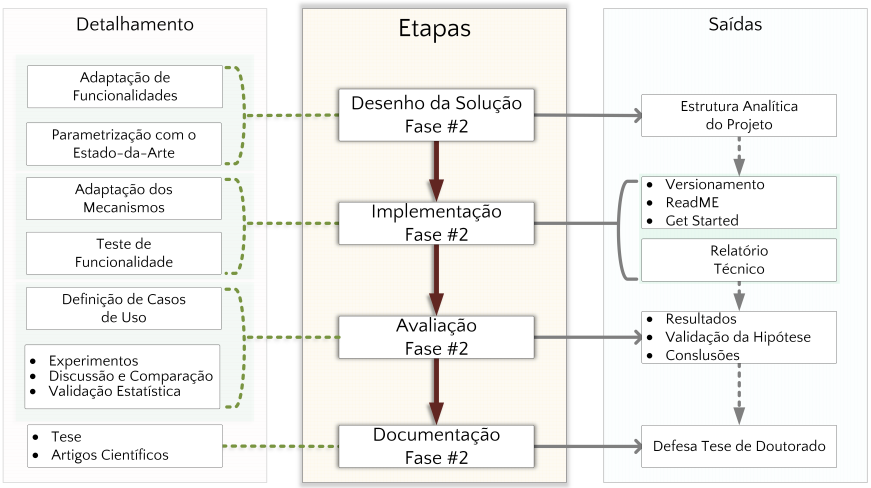

Figura 2 – Método de Pesquisa e Desenvolvimento – Fase 2.

A segunda fase prevista no desenvolvimento do projeto consiste em etapas adicionais, que são retroalimentadas. Essas etapas se prestam a prover um refinamento e adequação da solução para questões incipientes do estado da arte bem como ajustes remanescentes. A Figura 2 ilustra as etapas, os detalhamentos e entregáveis da segunda fase de desenvolvimento da solução.

A etapa Desenho da Solução prevê que sejam observadas micro etapas, quais sejam: Adaptação das Funcionalidades, que refinará os requisitos após retorno de avaliação de pares da proposta desta tese; a Parametrização com o estado da arte visa garantir que a solução lida com os desafios ainda não endereçados. Modifi-

cações pontuais nessa etapa podem provocar a alteração no entregável Estrutura Analítica do Projeto.

A etapa de Implementação, no âmbito da fase dois de desenvolvimento, prevê atividades como Adaptação dos Mecanismos previamente implementados. Essa adequação ocorre porque os requisitos podem eventualmente modificar, ou seu refinamento implicar na criação de outros requisitos; portanto, por questões de interoperabilidade e desacoplamento deve haver uma adaptação. Além disso, se eventualmente houver modificação na estrutura de codificação, são previstos uma bateria de testes de funcionalidade que incluem: unitários, integração e sistema.

Os entregáveis dessa etapa são: versionamento em sistemas especializados, construção de um documento *ReadME* que descreva os componentes e suas interações. Adicionalmente, está previsto um manual de criação e configuração do ambiente da prova de conceito. Outro entregável é o relatório técnico, está previsto que esse documento subsidiará a consolidação do fechamento de solução técnica adotada nesta tese.

Está incluída na fase dois a etapa de Avaliação que consiste no aprofundamento da experimentação e avaliação da aplicabilidade da solução para lidar com os desafios em aberto. Nesta fase, são detalhados os casos de uso sobre os quais a solução *NASOR* proverá o mecanismo de fatiamento de rede. Experimentos estatisticamente validados subsidiarão a discussão e a comparação da proposta com seus pares. O entregável dessa etapa culminará na discussão de resultados, argumentação da sustentação da hipótese, conclusões e limitações da tese.

Por fim, no sequenciamento lógico proposto está previsto a etapa de documentação que contêm micro atividades, a redação de um documento formal do tipo tese e artigos científicos para submissão em revistas e conferências relevantes com temáticas afins. O fim dessa etapa proporcionará o entregável Tese de Doutorado que consolidará, de forma estruturada, os entregáveis das etapas e fases preliminares.

1.6 Organização da Tese

Nesta seção são apresentados dois panoramas da organização da tese. A Organização Geral refere-se à disposição dos componentes textuais na estrutura do

documento. Por outro lado, a Organização Relacionada aos Objetivos apresenta os apontamentos da realização dos Objetivos em pontos específicos do documento.

1.6.1 Organização Geral

Esta tese está organizada focando na delimitação e contextualização do problema com destaque para propostas do estado da arte que propuseram resolver o problema de fatiamento de redes. A ordem e estrutura dos capítulos são resultado das saídas do método (Seção 1.5) adotado nesta investigação, assim a Figura 3 ilustra a composição e organização desta tese.

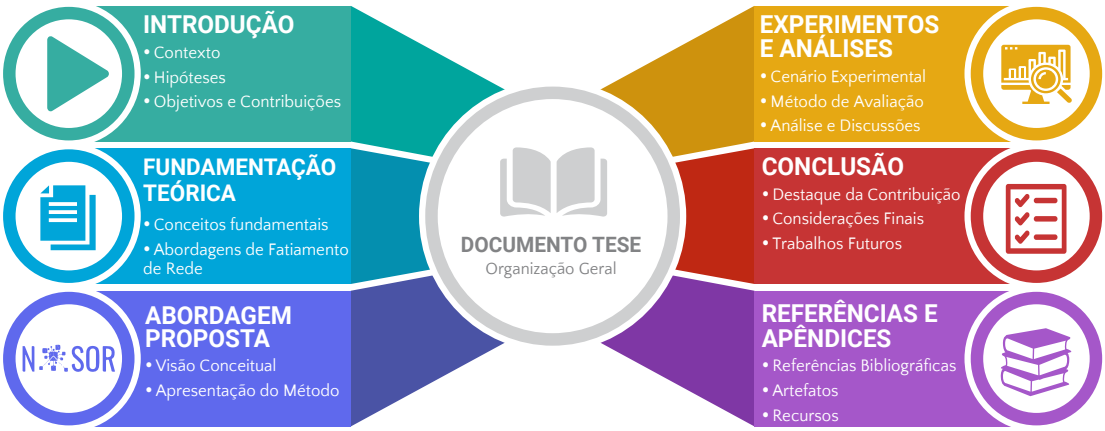

Figura 3 – Organização e Construção da Tese.

Além disso, o conteúdo dos capítulos são sumarizados conforme abaixo:

Capítulo 1 [**Introdução**]. Apresenta o contexto, motivação, contribuições, objetivos e organização do texto da tese.

Capítulo 2 [**Fundamentação Teórica**]. Apresenta os principais conceitos que permitem o compartilhamento de recursos e o estabelecimento de conectividade lógica entre entidades de rede sobre múltiplos *ASs*.

Capítulo 3 [**Abordagem Proposta**]. Contextualiza e delimita o problema de fatiamento de rede apresentando a proposta *NASOR* como uma entidade do plano de gerenciamento capaz de endereçar o problema de conectividade lógica entre múltiplos domínios políticos e tecnológicos da Internet.

Capítulo 4 [**Experimentos e Análises dos Resultados**]. São explorados cenários de uso onde se pretende validar a aplicabilidade e performance da solução

proposta para prover o fatiamento de rede sobre múltiplos domínios. Além disso, são apresentados experimentos que mensuram o desempenho das aplicações que utilizam o substrato de rede fatiado com uma perspectiva do usuário e do operador de rede.

Capítulo 5 [Conclusão]. Nesse capítulo são sumarizadas as principais contribuições para o estado da arte que a pesquisa propiciou. Além disso, contrasta-se a solução desenvolvida com seus pares, destacando os pontos endereçados e os em aberto. Também, são apresentadas as contribuições científicas e tecnológicas da tese e apontamentos para trabalhos futuros.

Capítulo 6 [Referências e Apêndices]. Neste parte da tese são apresentadas as principais referências que subsidiaram a solidificação da hipótese levantada e a construção do método para corroborar com a hipótese. Além disso, ao fim deste documento estão disponíveis artefatos e recursos organizados em Apêndices que foram construídos ao longo do desenvolvimento do método e da experimentação.

1.6.2 Organização Relacionada aos Objetivos

A realização dos objetivos são o ponto central desta tese. A Figura 4 mapeia a realização dos objetivos na estrutura da tese.

A construção e avaliação de um mecanismo de fatiamento de rede e os objetivos inerentes a essa tarefa estão organizados e agrupados em três (3) capítulos que se complementam e podem ser sumarizados da seguinte maneira:

- Capítulo 2: provê um levantamento do estado da arte e contrasta a funcionalmente o mecanismo proposto nesta tese com o estado da arte, realizando o objetivo preliminar desta tese;

- Capítulo 3: contêm a descrição da realização de objetivos como a visão conceitual do fatiamento de rede, o um arcabouço para o fatiamento de rede recursivo e os detalhes arquiteturas da prova de conceito do mecanismo proposta nesta tese;

- Capítulo 4: registra a realização dos objetivos como a proposição de casos de uso significativos, a implementação da prova de conceito e a experimentação e

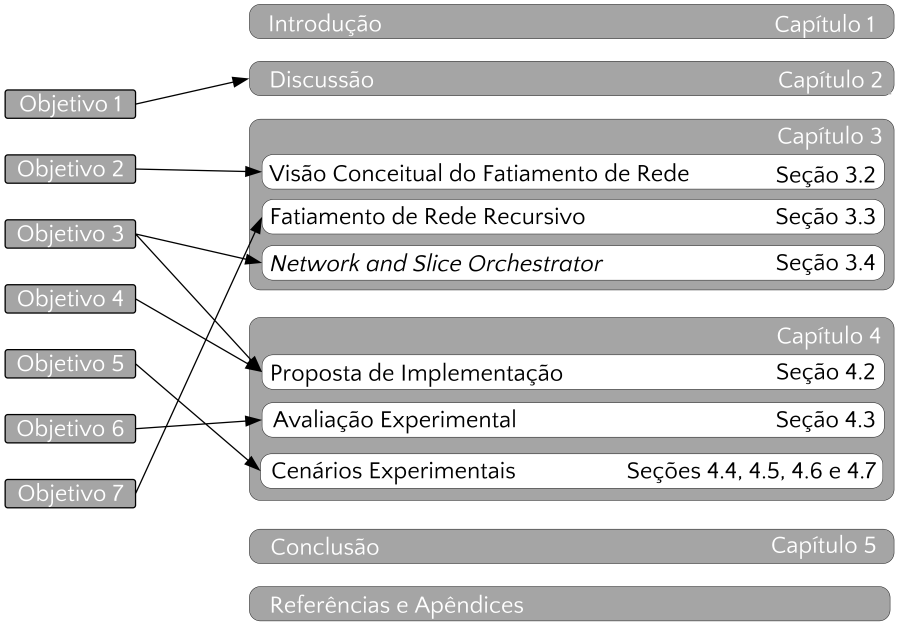

Figura 4 – Visão geral da estrutura da tese relacionada com seus objetivos e com os capítulos em que eles são realizados.

validação funcional, qualitativa e quantitativa do mecanismo proposto nesta tese.

Capítulo 2

Revisão da Literatura

2.1 Considerações Iniciais

Neste capítulo são apresentados conceitos que fundamentam a construção da solução que propõe responder às perguntas da hipótese. Os conceitos são apresentados detalhadamente conforme sua sequência e relevância histórica, além disso, são julgados quanto sua pertinência para lidar com o objeto de estudo.

Além dos conceitos, são discutidos trabalhos que possuem similaridade com o proposto. A discussão se estende para aspectos quantitativos e qualitativos de cada abordagem. Está proposto neste capítulo uma organização sistemática, em formato de tabela, das abordagens do estado-da-arte que versam sobre mesmo domínio de problema.

2.2 Redes Definidas por *Software*

O conceito *SDN* permeia os recentes avanços do estado da arte no que diz respeito a programabilidade de rede. A discussão em torno do termo data nos anos 90, tempo que a comunidade científica, em face da popularização da Internet, lidava com a criação de soluções e aplicações para uso generalizado. O processo de padronização dos protocolos demandava tempo, isso incentivou os pesquisadores a buscarem alternativas que viabilizassem a experimentação de suas propostas em cenários fiéis aos reais (FEAMSTER; REXFORD; ZEGURA, 2014). O re-

sultado disso deu forma ao paradigma de rede ativa — quando as entidades da rede possuem interfaces que expõe recursos, mecanismos e serviços customizáveis para especificação de alto nível, sobre os quais construíram-se mecanismos que materializaram o conceito de interface de rede (*network Application Programming Interface (API)*).

Inicialmente, da concepção de *network API* derivou-se dois formatos de prover programabilidade, compreendendo o modelo encapsulamento e o modelo de programabilidade pura (CALVERT et al., 1998). No modelo de encapsulamento, os códigos que influenciam o comportamento do nó de rede eram transportados em pacotes de dados, ao passo que no modelo de programabilidade, um canal fora da banda (*out-of-band*) era estabelecido com a entidade controlada o que permitia a inserção de códigos com vistas a um comportamento específico. Em outros termos, rede ativa diz respeito ao novo formato de gerenciamento, controle e orquestração unificada de entidades de rede específico, tornando as funções de controle da rede programáveis (NUNES et al., 2014).

Conforme Yang et al. (2004), Salim et al. (2003), Campbell et al. (1999), Doria et al. (2002), Toward (2004), Enns, Bjorklund e Schoenwaelder (2006) o conceito *SDN* deu forma aos objetivos preliminares de separação do plano de dados e controle por meio de interfaces padronizadas. O plano de controle representa a inteligência da rede, onde as instruções e decisões de encaminhamento são conjugadas. O plano de dados representa o encaminhamento dos pacotes, o comportamento das estruturas de enfileiramento e *buffers*.

A separação do plano de dados e controle da rede promove uma visão centralizada dos recursos de forma que as funções podem, flexivelmente, ser definidas ou modificadas após implantação da rede (LANTZ; HELLER; MCKEOWN, 2010). Dentre os fatores que impulsionaram significativamente a disseminação e o estabelecimento do conceito *SDN*, destacou-se o alvo de simplificar o *hardware* de rede e a incorporação da flexibilidade no nível de controle da rede (CASADO et al., 2012).

No conceito de *SDN* estão definidas interfaces inferiores e superiores, a inferior lida com aspectos de gerenciamento dos dispositivos físicos, suportando: programabilidade, reconfigurabilidade, compartilhamento de recursos e abstração da rede. Nesse nível, a interface que ganhou *momentum* foi a *OpenFlow*, ainda que, com

intrincadas características com a iniciativa *ForCES* que também materializava o conceito (YANG et al., 2004). A camada superior permite que as regras de encaminhamento, definidas por linguagem de alto nível, que consideram requisitos e políticas específicas (SHIN; NAM; KIM, 2012), sejam inseridas no plano de dados. Isso se dá, por meio do Controlador – entidade centralizadora que fornece interface aos requisitos programáveis de alto nível traduzindo-os em comportamentos do plano de dados.

A introdução e disseminação do protocolo *OpenFlow* em 2008 (MCKEOWN et al., 2008), como interface padrão do plano de controle, impulsionou o desenvolvimento do conceito por meio de vários controladores como *OpenDayLight* Medved et al. (2014), *ONOS* Berde et al. (2014) e *Ryu* Ryu (2014), além disso, abriu caminhos para inovação em outras áreas como computação em nuvem (AZODOLMOLKY; WIEDER; YAHYAPOUR, 2013).

A Figura 5 ilustra um típico *switch* com suporte ao protocolo *OpenFlow*, sua arquitetura consiste no Cliente *OpenFlow*, que implementa um canal seguro com o Controlador, e a tabela de fluxos com seus campos: Regras, Ações e Estatísticas (NUNES et al., 2014). Fluxo é o conjunto de pacotes que performam comportamentos similares, conforme seus cabeçalhos, em uma comunicação com origem, destino e duração específica. A programabilidade do plano de dados, consiste na capacidade de traduzir da linguagem de alto nível o tratamento a ser atribuído a um pacote de um fluxo conforme se cumprem as Regras, as quais levam em conta os campos do cabeçalho do pacote.

O campo Ação, considerado após uma determinada regra de fluxo ser satisfeita, determina o que ocorrerá com o pacote do fluxo em termos de encaminhamento. Há possibilidade de encaminhar o pacote de um fluxo para uma porta específica, ou enviá-lo ao controlador — típico de um cenário que há ausência de regras configuradas–, modificar o cabeçalho ou descartar os pacotes que constituem um fluxo (MCKEOWN et al., 2008).

O campo Estatísticas que permite adquirir detalhamento de pacotes, *bytes* e duração do fluxo. É certo que os campos que envolvem as Regras, os comportamento previstos nas Ações e as Estatísticas, variam entre as especificações do *OpenFlow*, no entanto, de maneira generalizada, a arquitetura de um *swtich OpenFlow* é conforme descrita (NUNES et al., 2014).

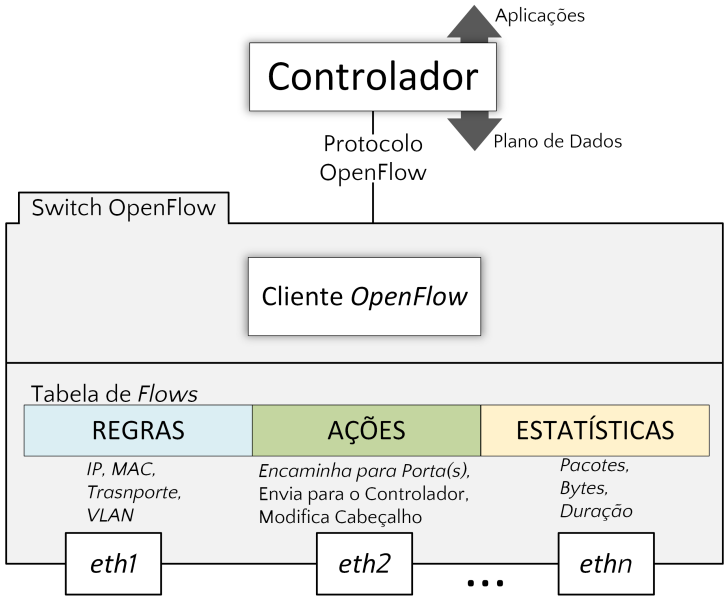

Figura 5 – Arquitetura de um *Switch Openflow*.

Fonte: Adaptado de Nunes et al. (2014)

A programabilidade dos comportamentos da rede é um habilitador tecnológico fundamental para prover serviços mais sofisticados e alinhados com as demandas das aplicações e dos usuários. Entregar recursos de rede programáveis e customizados aos usuários é oferecer-lhes uma fatia de rede. Por isso, nesta tese considera-se fundamental o conceito e as capacidades da programabilidade da rede baseada em *software* e suas tecnologias para o gerenciamento e oferta de serviços especializados.

2.3 Computação em Nuvem

Assim como na perspectiva de programabilidade das redes, o conceito de *cloud computing* (computação em nuvem) – estabelecido por John McCarthy (YU et al., 2012)–, como se apresenta hoje, é o resultado de um consistente amadurecimento científico. Com traços oriundos do *Unix Time-Sharing System*, que se apresentou como um sistema operacional de propósito geral e multi-usuário (RITCHIE; THOMPSON, 1978), o conceito preliminar de compartilhamento de re-

cursos direcionou-se para o aspecto massivamente escalável, com encapsulação na oferta dos serviços e orientação segundo perspectivas econômicas.

Depreende-se que *cloud computing*, além de ser uma sub-área de sistemas distribuídos, é fundamentalmente um conceito de orientação a serviços e compartilha características com o conceito de *Grid Computing* (FOSTER et al., 2008). Esse último, dentre outras definições disponíveis, pode sumariamente ser definido como uma infraestrutura de coordenação de recursos, os quais estão sob um controle descentralizado, que por meio de protocolos abertos e com interfaces de propósito geral, oferecem solução de problemas de forma dinâmica e com qualidade de serviço sobre recursos que se espalham sobre múltiplas organizações, possivelmente virtuais (CAFARO; ALOISIO, 2011; FOSTER et al., 2008).

O ponto de convergência dos conceitos de *Cloud Computing* e *Grid Computing* consiste na arquitetura e tecnologia, e as particularidades compreendem: segurança, modelo de programação, modelo de negócios, aplicações e abstrações. Ainda que date os anos 60, a tecnologia de máquinas virtuais é o habilitador expoente do conceito de *cloud computing* (CAFARO; ALOISIO, 2011). Máquina virtual é uma instância da máquina física que permite os usuários consumir recursos de *hardware* de forma transparente, provê à máquina inquilina a ilusão de rodar sobre uma máquina real (MERGEN et al., 2006). Essa característica da virtualização de *hardware* é consonante com a perspectiva de tempo compartilhado idealizada previamente nos sistemas operacionais (RITCHIE, 1980). Adicionalmente, a tecnologia de máquinas virtuais permite o compartilhamento de recursos de *hardware* computacional de alto custo.

A máxima da virtualização é permitir a utilização massiva de recursos computacionais, diminuindo os custos do serviço por meio de uma infraestrutura lógica subjacente que permeia o *hardware* e o processo computacional (TIANFIELD, 2011). Dentre outras tecnologias habilitadoras de *cloud computing*, a *Web services*, onde os serviços são oferecidos e consumidos sob demanda através da Internet, contribuiu significativamente para a pervasividade de *cloud computing*. Além disso, os sistemas de arquivo distribuídos e o modelo de programação conforme Dean (2007), Wang et al. (2010) são habilitadores destacáveis do conceito.

A camada de virtualização, ou *hypervisor*, permite que sistemas operacionais inquilinos sejam executados em paralelo sobre o mesmo *hardware*. As tecnolo-

gias que implementam o conceito são: virtualização completa, para-virtualização, virtualização na camada do sistema operacional, virtualização na camada de *hardware* e virtualização de aplicação; essas são abordagens bem estabelecidas no estado-da-arte. Na virtualização completa, o monitor de máquinas virtuais executa em formato de aplicação sobre o sistema operacional hospedeiro, assim as máquinas inquilinas consomem os recursos de *hardware* e executam operações de entrada e saída transparentemente (ABELS; DHAWAN; CHANDRASEKARAN, 2005; WALTERS et al., 2008).

Um ponto positivo desse formato de virtualização é sua facilidade a qual se estabelece pelo fato de uma aplicação que executa em nível de usuário ser capaz gerenciar as máquinas virtuais. No entanto, essa abordagem de virtualização possui desempenho inferir a abordagens de virtualização diretamente sobre o *hardware* (SAHOO; MOHAPATRA; LATH, 2010). No que se refere a virtualização na camada de *hardware* é uma abordagem que o *hypervisor* executa diretamente sobre o *hardware* sem a necessidade de um sistema operacional hospedeiro (PÉK; BUTTYÁN; BENCSÁTH, 2013; SAHOO; MOHAPATRA; LATH, 2010).

A virtualização em nível de sistema operacional executa mais de uma instância do mesmo sistema operacional em paralelo, isto é, cada máquina virtual usa a mesma imagem do sistema operacional (RESHETOVA et al., 2014; SHAN et al., 2012). Outra implementação é a paravirtualização, que considera modificações no sistema operacional inquilino, uma vez que a abstração do *hardware* para as instâncias é sutil (YOUSEFF et al., 2006).

Assim, as máquinas inquilinas são clientes que executam sobre um ambiente virtualizado mais enxuto e com performance melhor comparado com a virtualização completa (WALTERS et al., 2008). A virtualização de aplicações é um ambiente virtualizado onde as aplicações consomem apenas os recursos que lhes são necessários. Esse tipo de virtualização cria uma abstração de recursos e configurações entre as aplicações e o sistema operacional (LUNSFORD, 2009). Além disso, permite desacoplar as aplicações do sistema operacional hospedeiro (MILLER; PEGAH, 2007).

A *cloud computing* evoluiu o formato de construção e entrega dos serviços sobre sistemas distribuídos. Além disso, promoveu melhor uso de recursos e sua interconexão, buscando alcançar maior vazão e honorabilidade dos acordos de serviço

(BUYYA; YEO; VENUGOPAL, 2008).

Discute-se a computação em nuvem taxonomicamente conforme seu modelo arquitetural em camadas, o qual se subdivide em modelos de serviços, a saber: privado, público e híbrido. As nuvens podem ser privadas, quando implantadas exclusivamente dentro de uma organização para suporte de aplicações específicas. A nuvem pública permite a oferta de serviços de propósito geral para multi inquilinos na Internet, caracterizada principalmente pela possibilidade de pagar pelo o que se usa. Híbridas, quando combinadas nuvens privadas e públicas para oferta dos serviços (RIMAL; CHOI; LUMB, 2009). Os recursos podem ser ofertados conforme os modelos Plataforma, *Software*, Infraestrutura e *Hardware*, possibilitando o consumo sob demanda pelos usuários.

2.3.1 Tecnologias emergentes de Computação em Nuvem

Verbelen et al. (2012) destaca gargalo conceitual de manter a computação concentrada em infraestruturas de alto desempenho ao invés de migrá-la para dispositivos de borda, especialmente dispositivos móveis. Os autores levantam a hipótese de, apesar desses dispositivos terem restrições de capacidade, eles poderiam instanciar e suportar serviços de computação na borda.

A despeito de os serviços dos usuários serem construídos e predominantemente consumidos no formato cliente-servidor, suportados por uma infraestrutura de computação em nuvem, requisitos de latência e custo incentivaram comunidade científica propor uma camada intermediária: *mobile cloud computing* (nuvem computacional móvel) (GIURGIU et al., 2009). A definição de computação de nuvem móvel é concebida por três perspectivas conforme Fernando, Loke e Rahayu (2013), a primeira e tradicional é o dispositivo móvel que atua como cliente para consumir serviços hospedados em nuvem.

A segunda, considera dispositivos móveis como ofertantes de recursos para clientes conectados que servem-se de facilidades computacionais dos dispositivos móveis. A terceira refere-se ao conceito de *Cloudlet* que adiciona um nível intermediário entre a nuvem e o dispositivo móvel. Assim, o dispositivo móvel direciona carga de trabalho para nós remotos próximos. O conceito *cloudlet* se reformulou quando Satyanarayanan et al. (2009) propuseram uma solução confiável, com

boa capacidade computacional, disponível na Internet e, sobretudo, próxima dos usuários.

Outra arquitetura que estende os conceitos de computação em nuvem, que trouxe capacidades de computação para a borda da rede do operador, é *Mobile Edge Computing (MEC)* (MACH; BECVAR, 2017). Nessa arquitetura os recursos de computação são ofertados nas estações base e permite redução de latência para as aplicações dos usuários (LI et al., 2016). Outro conceito é o computação em névoa, computação em névoa, proposto pela Cisco em 2014; esse conceito incorporou as capacidades computacionais em dispositivos de rede como pontos de acesso ou *set-up-boxes*, tornando-os aptos a oferecerem serviços computacionais. Computação em névoa viabilizou o processamento mais próxima do usuário, de maneira o recurso esteja a uma distância de um salto (HABIBI et al., 2020) de se consumido. Diferente da computação em nuvem, o conceito de computação em névoa tem como alvo aplicações com restrições de latência, além disso possui orientação geográfica de demanda (BILAL et al., 2018).

Apesar das extensões do conceito de computação em nuvem, sobretudo os paradigmas de computação de borda, possuírem similaridades entre si, existem características centrais que diferenciam os conceitos. No que diz respeito a computação em névoa, o tipo de dispositivos, sobre os quais são ofertadas capacidades computacionais têm-se: roteadores, *switches*, pontos de acesso e *gateways* (GUSEV; DUSTDAR, 2018). Já o *Mobile Edge Computing* considera que os recursos computacionais, disponibilizados para servirem aos usuários, executam sobre *hardware* de propósito geral em estações base (HU et al., 2015; SABELLA et al., 2016).

Por outro lado, o *cloudlet* oferta seus recursos computacionais em equipamentos de baixo custo ou em computadores pessoais, conforme se observa em Lewis et al. (2014), Pang et al. (2015). A localização da oferta dos serviços leva em conta a proximidade do cliente. Em relação ao tempo de computação, tempo que a camada *edge* recebe e trata as tarefas atribuídas, tanto *MEC* quanto *Cloudlet* tendem a desempenhar melhor, dado sua característica de virtualização e os mecanismos de provisionamento. A capacidade de processamento de dispositivos de computação em névoa é convencionalmente menor, elevando os tempos de resposta computacional.

Além dessas características pontuadas, conforme Dolui e Datta (2017), os para-

digmas de computação em nuvem divergem quanto a localização do nó, arquitetura de *software*, orientação ao contexto, mecanismo de acesso, comunicação entre nós e proximidade.

Dentre as abordagens de computação em nuvem e suas particularidades, todas propõe a oferta de recursos para domínios e aplicações heterogêneas (SANAEI et al., 2014). Além disso, a consolidação das técnicas de virtualização e o formato de entrega de serviços em computação em nuvem, permitiu reconsiderar alguns paradigmas, como o da comunicação de dados (SHEA; LIU, 2012). Logo, as técnicas de virtualização encontraram espaço nas redes de comunicação, de forma que os as soluções de rede e comunicação invariavelmente incorporaram a computação como componente fundamental para a oferta e operação de serviços de rede (COSTA-PEREZ et al., 2013). Além disso, a virtualização de rede tornou-se tecnologia chave que facilita implantação, gerenciamento e garantia de receita (BARI et al., 2013; JAIN; PAUL, 2013; CHOWDHURY; BOUTABA, 2009).

O conceito de virtualização de recursos computacionais, que sustenta a computação em nuvem e seus serviços, pode ser transladado para o âmbito das redes de comunicação. O princípio basilar da virtualização é o compartilhamento de recursos, que constrói uma camada de virtualização para que sistemas apropriam-se do *hardware* subjacente com isolamento e transparência. No âmbito das redes de comunicação, têm havido esforços para que recursos de rede sejam também virtualizados e ofertados com isolamento e transparência. Esta tese, considera o conceito e as tecnologias de computação em nuvem como fundamentais para a realização do compartilhamento de recursos de rede.

2.4 Virtualização de Funções de Rede

Uma iniciativa da indústria propôs em 2012 reformatar o modelo convencional de entrega de equipamentos de rede e serviços, tradicionalmente como uma composição indivisível de *hardware* de propósito específico e *software* proprietário embarcado (HAWILO et al., 2014). A proposição e padronização da dissociação do serviço de rede do *hardware* conduzida pela *European Telecomunications Standards Institute* (*ETSI*), considerando as capacidades de virtualização, especificou o conceito de *NFV*, Virtualização de Função de Rede (ETSI, 2012). Essa proposta

consistiu em oferecer funções de rede implementadas como *software* que rodam sobre *hardware* de propósito geral (HERRERA; BOTERO, 2016).

Potencializado pelo conceito de *SDN*, a discussão em torno do formato de implantação e operação de rede ganhou novos argumentos no contexto de *NFV* (LI; CHEN, 2015). Os objetivos do *NFV* consistem em garantir equivalência funcional do virtualizado e do físico, baixa complexidade de suporte e gerenciamento e baixa sobrecarga na performance, quando comparado o físico e o virtualizado (SHERRY et al., 2012). Com *NFV*, a virtualização de funções de rede mostrou-se como uma forma de desacoplar as aplicações do fornecedor, de forma que, o provisionamento seja flexível e com custos reduzidos (MATIAS et al., 2015).

A *Virtualized Network Function (VNF)*, nome dado a função de rede virtualizada, possui seu ciclo de vida gerenciado pelo *Management and Orchestration (MANO)*, entidade descrita no *framework* da *ETSI* (ETSI, 2013b). Um importante aspecto habilitado por *NFV* é *Service Function Chaining (SFC)* que possibilita conexão em série de *Virtualized Network Functions (VNFs)* como *firewalls*, *Deep Packet Inspection (DPI)* e outros, de forma que sejam interligadas de maneira flexível no processo de implantação do serviço de rede (JOHN et al., 2013).

A estrutura da padronização, conforme se vê no *framework* ilustrado na Figura 6, delimita papéis específicos para as entidades, agrupadas em blocos, e fundamentam a comunidade e a indústria quanto a construção de novas *VNFs* (ETSI, 2013a). Os blocos que sistematicamente compõe *framework NFV* são o *Network Functions Virtualization Infrastructure (NFVI)*, que fornece o substrato de *hardware* para execução das *VNFs*, o principal papel desse bloco é fornecer uma camada de gerenciamento de *hardware* e tecnologias de virtualização.

O *Network Function Virtualization Orchestrator (NFVO)* lida com a orquestração e com o ciclo de vida dos recursos de *hardware* e *software*, isto é, os componentes desse bloco lidam com tarefas de gerenciamento do ciclo de vida das *VNFs*, através de entidade *Virtualized Infrastructure Manager (VIM)* e *Virtualized Network Fucntion Manager (VNFM)* (CUI et al., 2014). O terceiro bloco que compõe o *framework* compreende o conjunto das *VNFs* implantadas e suas interfaces de gerenciamento. Além disso, neste bloco estão convencionados os padrões de linguagem e formato de interconexão que descrevem o comportamento das funções

virtualizadas (ETSI, 2013b).

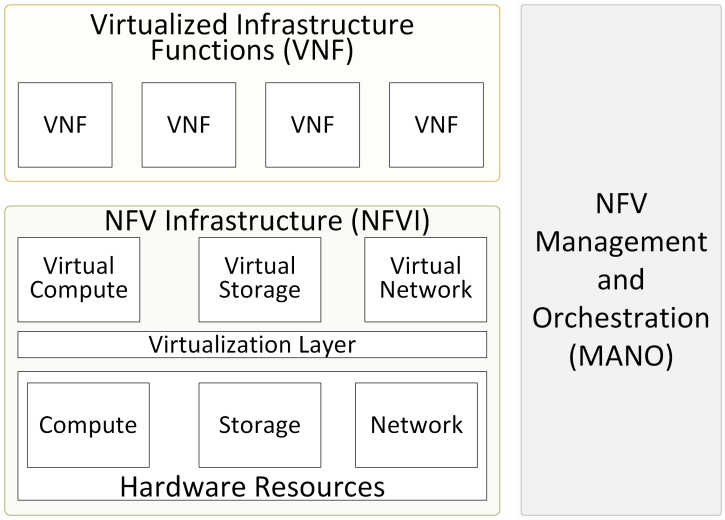

Figura 6 – *Framework* arquitetural *NFV ETSI*.

Fonte: Adaptado de ETSI (2013b).

Devido a agilidade, velocidade na entrega de serviços, e custos, a virtualização de funções de rede tornou-se componente fundamental no bloco estrutural de novas arquiteturas de rede (Wu et al., 2019). Agregada a essas capacidades, o conceito de compartilhamento de recursos de rede, no formato como se construiu o compartilhamento de recursos para máquinas virtuais, ganhou forma e traz consigo potencialidades que podem ser utilizadas para lidar com os desafios de fatiamento de redes e orientação a serviços. Por isso, a virtualização de funções de rede é um componente tecnológico fundamental o contexto de fatiamento de redes que versa esta tese.

Compartilhar recursos é fundamental para ofertar serviços customizados que executam de forma isolada e transparente sobre o *hardware* de rede subjacente. Além do compartilhamento de recursos, virtualizar funções de rede é fundamental porque adiciona-se ao contexto do serviço de comunicação funcionalidades excedentes ao que é oferecido por equipamentos físicos de fornecedores tradicionais. Com funções virtualizadas de rede, um novo contexto de serviços de comunicação

podem ser ofertadas. Assim, o conceito e tecnologias de *NFV* são amplamente considerados nesta tese.

2.5 Inteligência Artificial

O conceito de inteligência artificial foi proposto em 1955 por McCarthy, Minsky, Rochester e Shannon quando declaram: "...cada aspecto da aprendizagem ou qualquer outra característica da inteligência pode, em princípio, ser descrito com tanta precisão que uma máquina pode ser feita para simulá-lo" (RAJARAMAN, 2014; MÜLLER; BOSTROM, 2016). Desde a segunda guerra mundial, especialmente após a publicação do artigo *"Computing Machinery and Intelligence"* Turing (2009), a comunidade científica tem proposto métodos que replicam o aprendizado humano. Essa replicação tornou-se possível devido ao surgimento e popularização do computador moderno.

O aprendizado de máquina é uma abordagem de inteligência artificial que após analisar um conjunto de dados é capaz de construir modelos complexos e algoritmos capazes de fazer predições futuras (MITCHELL et al., 1997). Na inteligência artificial distingue-se o aprendizado de máquina em duas categorias: online e offline (BEN-DAVID; KUSHILEVITZ; MANSOUR, 1997). É possível esquematizar o aprendizado de máquina em categorias (LOVE, 2002) conforme abaixo:

- **Aprendizado Supervisionado**: O treinamento é realizado baseado em um *dataset* previamente categorizado (*labeled data*). O *dataset* é composto pelas entradas de dados e suas respectivas saídas esperadas. Algoritmos de aprendizado supervisionado são recomendados para classificação e extração de características em situações em que há dados históricos sobre o problema, por exemplo: problemas de processamento de sinais. Como técnicas de aprendizado supervisionado podemos citar: Redes Neurais Artificiais, Redes Neurais Profundas (*Deep Neural Networks*), Florestas Aleatórias (*Random Forests*).

- **Aprendizado Não-Supervisionado**: O treinamento é realizado com dados de teste que não foram rotulados, classificados ou categorizados previamente. O aprendizado não supervisionado identifica similaridades nos dados e reage com base na presença ou ausência de tais semelhanças em cada nova amostra.

Este tipo de técnica pode ser utilizada quando não há clareza sobre os dados a serem trabalhados. Como técnicas de aprendizado não-supervisionado podemos citar: Redes Neurais Artificias (também para não-supervisionado) e *K-means* (KOUR; GONDHI, 2019).

- **Aprendizado Semi-Supervisionado**: O treinamento é realizado com dados mistos, onde alguns contém a categorização prévia e outros são apenas amostras colhidas sem qualquer classificação (KOUR; GONDHI, 2019).

- **Aprendizado por Reforço**: A aprendizagem por reforço usa dados da implementação ao invés de dados históricos. O objetivo da aprendizagem por reforço é melhorar o desempenho de um agente em uma determinada tarefa por meio do *feedback* do ambiente. A aprendizagem por reforço não é supervisionada, no entanto, a forma de aprendizagem é diferente de outras técnicas de aprendizagem não-supervisionadas. Em vez de aprender a estrutura de alguns dados, o aprendizado por reforço tenta explorar as melhores ações no meio de operação. Portanto, a capacidade de capturar o ambiente por meio de *feedback* e realizar ações torna a aprendizagem por reforço adequada para problemas que envolvem uma série de decisões, ou seja, seguir uma política de ações de acordo com o estado do ambiente observado. Como principais técnicas de aprendizado por reforço pode-se citar as Redes Neurais Recorrentes (*Recurrent Neural Networks* (KAELBLING; LITTMAN; MOORE, 1996)).

O aprendizado *offline* fundamenta-se no aprendizado que parte de um conjunto de dados iniciais utilizado para capacitar o preditor para futuras predições. A despeito de as técnicas de aprendizado de máquina offline oferecerem uma boa performance para classificações convencionais, a técnica enfrenta resistência para classificação no contexto de redes (KATO et al., 2020). Especialmente no contexto de comunicação *wireless*, porque o aprendizado de máquina offline limita-se a quantidade e qualidade de dados pré-existentes para o processo de aprendizado, o que restringe a fase de treinamento e implica na qualidade das predições futuras.

As técnicas de aprendizado de máquina online fundamentam-se no aprendizado progressivo conforme mudanças e percepções do ambiente. No contexto de

redes de comunicação, o aprendizado *online* apoia-se nos conceitos emergentes como inteligência na borda (TALEB et al., 2020). A inteligência na borda prevê a inserção de recursos computacionais em elementos de rede para realização e suporte de requisitos específicos como o extrema baixa latência. Recomenda-se o aprendizado *online* quando o conjunto de dados na fase de aprendizado é grande inviabilizando computacionalmente o treinamento completo antes da fase de atuação do preditor (TALEB et al., 2020). No contexto de redes de comunicação, o aprendizado online é utilizado para previsão de utilização de canais de utilização wireless, consumo de energia e alocação de banda (ZHANG; SUN; YANG, 2021).

O conceito de redes neurais profundas veio do estudo de redes neurais artificiais (HINTON; SALAKHUTDINOV, 2006). As redes neurais profundas são um subcampo do aprendizado de máquina que vem alcançando excelentes desempenhos no processamento de imagem, linguagem natural e visão computacional (SAHU; DASH, 2021). Basicamente, as redes neurais possuem múltiplas camadas ocultas entre camadas de entrada e saída que são utilizadas para modelar e processar relacionamentos não lineares.

A Figura 7 ilustra uma arquitetura típica de *CNN*. A camada de convolução tem o objetivo de capturar o mapa de características de uma imagem. As imagens são *arrays* de duas dimensões que na camada de convolução são operadas por um *kernel* que realiza operações convolucionais para alimentar a próxima camada da *CNN* (SAHU; DASH, 2021).

Após, a saída da convolução alimentam a camada de *pooling* que é utilizada para reduzir o número de parâmetros e diminuir a amostra de cada característica do mapa de características. Alguns tipos de *pooling* conhecidos são do tipo *max pooling* que produz o valor máximo entre todos os valores na janela e *average pooling* que por considerar a média extrai características mais suaves do que o *max pooling* (SAHU; DASH, 2021; KOURETAS; PALIOURAS, 2019).

A camada *Fully Connection (FC)* captura como entrada a saída das outras camadas e transforma-as em números específicos de classes. A saída dessa camada permite calcular o gradiente de erros e pesos que retroalimentam a rede para compatibilizar os parâmetros das demais camadas.

Neste trabalho, um dos experimentos utiliza o conceito de *CNN* para a classificação de tráfego de rede a fim de inferir qual o tipo da aplicação corre predomi-

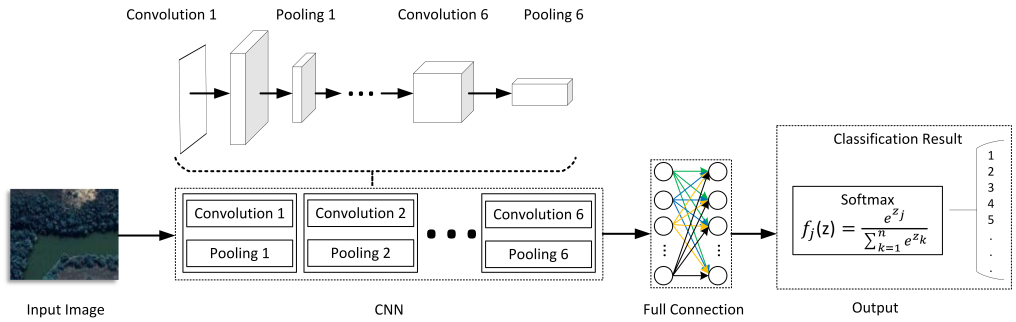

Figura 7 – Panorama Arquitetural de uma *CNN*.

Fonte: Adaptado de Sahu e Dash (2021).

nantemente sobre um enlace em que se implantará uma fatia de rede. As classes de tráfego podem ser Domain Name System (DNS), *IoT*, *BitTorrent* e VoIP. Baseado nesta classificação, o comportamento do solução proposta leva em conta o tipo de tráfego.

2.6 O Fatiamento de Redes

Tecnologias como *SDN* e *NFV*, por habilitar programabilidade, flexibilidade e modularidade nas redes, alterou o formato de compartilhamento dos recursos (KSENTINI; NIKAEIN, 2017). Esse compartilhamento, passa a permitir que instâncias independentes de rede, com métricas próprias, operem sobre *hardware* de rede comum (LI et al., 2017b). Um novo formato de instância lógica de rede tomou na literatura a forma de *network slice*, ou fatia de rede.

A perspectiva tenra de compartilhamento e gerenciamento de recursos tem se aprimorado desde o estabelecimento do conceito de redes programáveis na década de 90 (GALIS et al., 2004). Nos anos 2000, a programabilidade habilitou inúmeros avanços em termos de *testbeds*, os quais suportaram experimentos de novas aplicações e técnicas que, por questões de escalabilidade (HANDZISKI et al., 2006) e segurança (MCKEOWN et al., 2008), não se experimentava na rede em operação.

Em termos de compartilhamento de recursos e coexistência de múltiplas redes lógicas, a iniciativa *Global Environment for Networking Innovation (GENI)*,

proposta primária do atual entendimento de fatiamento de rede, introduziu as potencialidades da programabilidade e o conceito de *testbed* experimental (BERMAN et al., 2014). A despeito de não se observar um consenso na literatura acerca do sentido exato do termo "fatiamento de rede", sobretudo por já existirem iniciativas de separação lógica de redes bem estabelecidas, como *VPNs* ou *Virtual Local Area Networks (VLANs)*, algumas entidades propuseram definir o termo.

Assim, NGMN Alliance (2016) estruturou o conceito de fatiamento conforme um *framework*, que compreende três blocos: a camada de instância do serviço, instância da fatia de rede e a camada de recursos. Desta forma, uma instância da fatia de rede diz respeito a uma rede lógica com características, políticas e configurações específicas. Além disso, uma instância da fatia de rede pode ser: completa, parcialmente, logicamente ou fisicamente isoladas com políticas próprias (KAZMI et al., 2019).

A entidade *International Telecommunication Union Telecommunications (ITU-T)*, declara que o fatiamento de rede é uma partição lógica e isolada de rede que opera sobre recursos de rede, computação e armazenamento programáveis (ITU, 2011). Uma definição alternativa é dada pela entidade *3rd Generation Partnership Project (3GPP)* onde se declara que uma fatia de rede é uma rede lógica que provê facilidades e serviços de rede, tanto no acesso quando no núcleo, onde se deve habilitar o acesso simultâneo às múltiplas fatias (FOY; RAHMAN, 2017).

De forma mais abrangente, a entidade *5G Infrastructure Public Private Partnership (5G PPP)* declara que o fatiamento de rede é um subconjunto de infraestrutura de rede, virtual e fim-a-fim, que se estabelece por várias entidades, como: radio, acesso cabeado, núcleo, transporte e borda (GROUP et al., 2016). Recentemente, uma definição concisa é dada em Ordonez-Lucena et al. (2017) onde declaram que: fatia de rede representa uma rede lógica fim-a-fim criada sob demanda que executa sobre *hardware* comum.

Não há que se falar em fatiamento de rede sem mencionar as tecnologias chaves que habilitam o conceito. Os recursos de rede são habilitadores do conceito, eles se estabelecem e suportam funções virtualizadas que se prestam a realizar um papel importante no fatiamento de rede (FOUKAS et al., 2017). Além disso, uma fatia de rede abarca entidades de *hardware* e *software* heterogêneas (SINH et al., 2018) que são gerenciadas uniformemente (YOUSAF et al., 2017).

Outro habilitador do fatiamento de redes é o conceito de virtualização (ZHANG et al., 2017), como ocorre para recursos computacionais, a virtualização permite segmentar uma fatia de rede e seus recursos para entidades como: o provedor de infraestrutura, as redes lógicas justapostas sobre a infraestrutura e o usuário final.

Adicionalmente, um habilitador do fatiamento de rede é o conceito de orquestrador, cujo papel é coordenar o ciclo de vida e recursos associados a uma fatia de rede (ORDONEZ-LUCENA et al., 2017). Existem especificações de orquestradores para o lidar com o fatiamento de redes, entidades como *Open Network Fundation (ONF)* (PAUL et al., 2016) e *ETSI* (ETSI, 2016) propuseram modelos arquiteturais de orquestrador para fatiamento de rede.

Há de se mencionar também o conceito de isolação e compartilhamento inter fatias, como ocorre em (SAMDANIS; COSTA-PEREZ; SCIANCALEPORE, 2016), que é um aspecto tecnológico chave para conceito de fatiamento de redes. Além da separação do plano de dados, que inclui performance e métricas específicas de cada rede lógica, espera-se segurança, privacidade e gerenciamento para o fatiamento de rede (ORDONEZ-LUCENA et al., 2017).

Adicionar ao compartilhamento de recursos computacionais e de rede a orientação ao usuário e serviço dá forma ao conceito de fatiamento de rede. Uma fatia de rede pode conter recursos compartilhados de rede e computação, mas é a combinação lógica e funcional desses componentes e o formato da gerência e operação do serviço que fundamenta o termo. Assim, o problema fundamental que esta tese endereça reside na temática de fatiamento de rede.

2.7 Roteamento por Segmentos

A *Request for Comments (RFC)* 7855 introduziu o conceito de roteamento baseado na origem, onde o caminho do pacote, isto é, os nós que ele percorre para alcançar seu destino são predeterminados na origem (PREVIDI et al., 2016). Na *RFC* estão definidos termos como nó de ingresso, que além de ser o ponto de entrada dos pacotes, é onde se encapsula as instruções de nós a serem percorridos. As instruções são formatadas conforme uma lista de entidades, a qual é armazenada como campo no cabeçalho dos pacotes, e que instrui o nó, conforme ordenado na lista, a encaminhar o pacote para a entidade subsequente.

Segment Routing, ou roteamento por segmento, baseia-se no paradigma de roteamento centrado na origem (FILSFILS; FRANCOIS, 2016). Um segmento pode representar instruções de topologia e serviços. No que se refere a topologia, as instruções contidas no cabeçalho do pacote influenciam o comportamento da entidade a encaminhar o pacote para uma determinada interface em detrimento e outra.

De forma global, as instruções de topologia permitem instruir um pacote em um domínio de roteamento por segmento a tomar um caminho específico, ainda que haja outros disponíveis. O aspecto de instruções de serviços permite que o segmento esteja associado às políticas de *Quality of Service (QoS)* além de direcioná-lo a entidades como *containers* e máquinas virtuais ao longo do caminho. Além disso, o conceito permite explorar paradigmas de encadeamento de *VNFs* que desempenham comportamentos específicos ao longo da rede (FILSFILS et al., 2015).

A arquitetura *SR* possui um controle distribuído onde segmentos são alocados e sinalizados por nós individuais que os transmitem por meio protocolos como: *IS-IS*, *OSPF* ou *BGP*. O controle centralizado calcula políticas, aloca e instancia segmentos no controlador *SR* central. Pode de haver dois ou mais controladores no mesmo domínio *SR*, além disso, na implementação do conceito de *SR* não se estabelece o formato como o controlador lida com a rede. Assim, é possível, por exemplo, ser através de *NETCONF*, *Border Gateway Protocol (BGP)* ou *Path Computation Element Communication Protocol (PCEP)*. Está previsto para a arquitetura *SR* dois tipos de plano de dados, baseado em *MPLS* ou *IPv6* (FILSFILS et al., 2015; LEBRUN, 2016).

O *Segment Identifier (SID)* é responsável por identificar um segmento, eles podem ser atribuídos a várias instâncias do domínio *SR*. Uma adjacência local recebe o identificador *SID* quando representa o *link* entre dois nós. Um caminho no domínio *SR* associa um *SID* ao prefixo destino *IP* que é usado no processo de encaminhamento. Um *SID* associado ao nó permite que um segmento seja encaminhado a um determinado nó segundo a política de menor caminho calculada por outros mecanismos como *Equal-cost multi-path routing (ECMP)*.

Um *SID* associado a um segmento *anycast* é similar ao *SID* de Prefixo, salvo o aspecto de um-para-muitos do *anycast* que suporta alta disponibilidade. Um *SID* associado ao serviço permite aplicar um comportamento granular a cada segmento,

aproximando-o de um serviço específico, como encadeamento de funções de serviço (ABDULLAH; AHMAD; HUSSAIN, 2019; VENTRE et al., 2018).

A representação semântica e comportamental dos *SIDs* em cada nó torna-se efetiva com a premissa de prévia troca de informações entre os dispositivos da rede envolvidos na comunicação. Isto é, o plano de controle implementado pelo protocolo *Interior Gateway protocol (IGP)* anuncia os *SIDs* dos roteadores e dos *link*s pela rede – similar ao que ocorre com anúncios de rotas (ABDELSALAM et al., 2018). Protocolos do tipo *IGP* como *IS-IS* e *OSPF* implementam esse comportamento (FILSFILS et al., 2015).

Os *SIDs*, a depender do domínio, são concebidos como locais, quando possuem representação somente no nó que lida com ele, e global quando o segmento é conhecido por todos os nós pertencentes ao domínio. Além disso, segmentos do tipo global são: prefixo do nó, do segmento e *anycast*. A categoria local compreende o segmento de adjacência (ABDULLAH; AHMAD; HUSSAIN, 2019).

Definir o caminho que um pacote deve percorrer ao ingressar em um nó do domino *SR* depende da política implementada no plano de controle. Métodos como *Constrained Shortest Path First (CSPF)* computa o caminho mais curto a um destino considerando se as restrições do caminho combinam com o critério previsto para a comunicação. O método estático, definido previamente pelo operador, é uma abordagem simples, mas com utilização advertida devido a limitações de escalabilidade, resiliência e gerenciamento. Por outro lado, a abordagem baseada em *SDN* provê um plano de controle flexível, além de um plano de dados escalável e resiliente (FILSFILS et al., 2015).

Na arquitetura *SR* dois modelos de funcionamento do plano de dados são conhecidos, um baseado em rótulos *MPLS* e outro baseado em *IPv6* (FILSFILS et al., 2014; LEBRUN; BONAVENTURE, 2017). No *MPLS* cada nó *SR* mantém um conjunto de rótulos, locais e globais, que são atribuídos aos segmentos que ao percorrerem o domínio *SR* são tratados segundo sua especificidade. A lista de segmentos que acomoda esses rótulos, típicos do domínio *MPLS*, é influenciada por operações como: *PUSH*, *POP*, *SWAP* com objetivo de percorrer o caminho preliminarmente estabelecido na fonte. Por outro lado, nós orientados à *SR* operam as funções: *CONTINUE*, que encaminha o pacote baseado no segmento ativo da lista; *PUSH*, adiciona e define como primeiro um segmento no cabeçalho *SR* –

similar ao *MPLS*; e *NEXT*, marca o próximo segmento como o ativo.

No mesmo sentido, o plano de dados baseado em *IPv6* faz uso da lista de segmentos para acomodar endereços *IPv6* dos nós por onde os segmentos devem percorrer. Assim, um cabeçalho para controle de processamento da lista foi incorporado, a saber o *Segment Routing Header (SRH)*. O cabeçalho *SRH* contém uma lista de segmentos que compreende endereços *IPv6* onde o primeiro da lista será último da política de roteamento por segmento. O campo *Next Header* identifica o tipo de protocolo que virá imediatamente após o cabeçalho *SRH*. Em *Hdr Ext Len* é definido o tamanho do cabeçalho. O campo *Routing Type* permite identificar um cabeçalho *SRH* alternativo (LEE; SHEU, 2016; FILSFILS et al., 2015).

O campo reservado *Segments Left* armazena o número de segmentos pendentes de processamento. A entrada *type-specific* data refere-se ao comprimento variável de formato do protocolo de roteamento. A especificação do campo *Last Entry* contém o índice do último segmento da lista. A *Internet Assigned Numbers Authority (IANA)* dispõe do campo *Flags* como reserva para definições futuras. Pacotes que possuem o campo *Tag* podem ser classificados ou agrupados conforme características e políticas compartilhadas. Por fim, o campo *Type Length Value* permite definir meta-dados para o processamento do segmento (PREVIDI et al., 2017).

O roteamento por segmentos é um habilitador tecnológico considerado nesta tese porque ele possibilita que uma fatia de rede seja identificada e distinguida das demais em um mesmo *hardware* de rede subjacente. Posteriormente, serão apresentadas algumas tecnologias e métodos que distinguem e identificam redes lógicas, no entanto, dado a natureza do problema que esta tese ataca, a saber o fatiamento de rede entre múltiplos *ASs*, considerou-se empiricamente necessário a adoção de um método compatível com o domínio de roteamento: o roteamento por segmentos.

2.8 Trabalhos Relacionados

As abordagens do estado da arte que discutem fatiamento de rede atacam problemas específicos dos domínios que estão inseridas. Algumas abordagens concentram-se em prover compartilhamento de recursos no nível de acesso, outras no nível do núcleo da rede e outras na borda. Essa divisão em três camadas é

comum em redes móveis, e a proposta de fatiamento de rede desta tese vai além, considerando as redes de transporte da Internet.

Por haver entendimento plural de como se deve realizar o fatiamento de rede, é relevante descrever abordagens do estado-da-arte que trouxeram propostas e mecanismos com esse objetivo. Assim, esta seção descreve alguns trabalhos que lidaram diretamente com o problema do fatiamento de rede em multi-domínios. Esta tese adota intercambiavelmente os conceitos multi-domínio e múltiplos *AS* para descrever o conceito de fatias de rede que perpassam múltiplos domínios administrativos. Ao final desta seção, está proposto uma tabela que sistematiza as abordagens conforme características relevantes. Nas próximas subseções, está descrito e agrupado alguns trabalhos relacionados como Orquestradores de Código Aberto, Projetos de Pesquisa e Contribuições de Pesquisa.

2.8.1 Orquestradores de Código Aberto

Algumas iniciativas, predominantemente de código aberto, direcionam-se para a orquestração de Serviços de Rede e são essenciais para habilitar um ecossistema de fatiamento de rede. Nesta subseção estão apresentados alguns orquestradores de código aberto que objetivaram realizar o fatiamento de rede.

2.8.1.1 *ESCAPE*

O orquestrador *Extensible Service ChAin Prototyping Environment (ESCAPE)* de 2013 materializa os objetivos do projeto *UNIFY* CsÁszÁr et al. (2013) ao usar tecnologias habilitadores como: *Mininet* Lantz, Heller e McKeown (2010), *Click* Kohler et al. (2000), *NETCONF* e *POX* para prover orquestração de serviços de rede na perspectiva de multidomínios. Sua arquitetura consiste em três camadas que se comunicam mutualmente, as camadas são: *Service Layer*, *Orchestrator Layer* e *Infrastructure Layer*.

A Camada de Serviço é a interface que recebe a requisição de serviço e a encaminha para a Camada de Orquestração, que interpreta a requisição e aloca recursos para o serviço. A Camada de Infraestrutura contém os mecanismos de gerenciamento de recursos que compreendem: rede, armazenamento e computação. Essas camadas organizadas provêm as funcionalidades previstas para a arquitetura

do *ESCAPE*, além disso garante o gerenciamento dos serviços que rodam sobre ela (SONKOLY et al., 2015).

O *ESCAPE* viabiliza a característica multidomínios devido sua habilidade de descobrir, detectar e gerenciar infraestruturas de rede e computação com diferentes tecnologias. Além disso, ele torna possível lidar com diferentes *Virtualized Infrastructure Managers (VIMs)* os quais estão ligados diretamente a ele por meio de um domínio *SDN* baseado em *Mininet*. A característica multi-domínio da solução lida apenas com o plano de controle, em domínios diferentes do experimental, como no roteamento hierárquico da Internet, um plano de controle externo pode não ser capaz de modificar o comportamento do plano de dados de outros domínios.

Assim, o *ESCAPE* possui a limitação de não alterar o plano de dados em domínios vizinhos. Politicamente e tecnicamente, não é trivial garantir que acordos firmados no plano de dados de um domínio singular sejam honrados em outros domínios, que podem eventualmente possuir suas próprias políticas. O plano de controle do *ESCAPE* é capaz de alcançar *VIMs* em diferentes domínios tecnológicos, no entanto, o plano de dados é limitado a conectividade direta que os *switches SDN* provêm.

2.8.1.2 *OpenSource MANO (OSM)*

Lançado em 2016, o projeto *Open Source MANO (OSM)* é um ecossistema alinhado com a especificação *ETSI* para o gerenciamento do ciclo de vida de *VNFs*. A arquitetura do *OSM* contém três componentes, o *NFVI* que é responsável para hospedar máquinas virtuais e conectá-los por meio de *links* virtuais. Segundo, o *MANO* mantém a configuração e gerencia o ciclo de vida das VNFs, NS as Fatias de Rede.

O terceiro bloco é a coleção das *VNFs*, serviços de rede, fatias de rede que são combinadas e interconectadas no *NFVI* para materializar a instância do serviço. O *OSM* possui uma arquitetura de serviço baseada em *containers* o que traz modularidade. Além disso, o *OSM* provê a conectividade das *VNFs* em um plano de dados baseados em *SDN* (ETSI, 2016).

O *OSM* fornece um plano de controle para vários domínios, uma vez que os *VIMs* sob seu gerenciamento operam e implantam serviços em recursos computaci-

onais administrativamente distintos. No entanto, o plano de dados multidomínios não é claramente estabelecido uma vez que o *OSM* não aplica e nem garante políticas diferenciadas para as *VNFs* de uma rede lógica.

O *OSM* provê o fatiamento de rede em instâncias lógicas, que são construídas com tecnologias baseada em *VLAN* e *VxLAN*. Para lidar com esses desafios, a presente proposta endereça o provimento de plano de dados limitados ao domínio do *data center*, tornando-o inter-domínios.

2.8.1.3 *ONAP*

A proposta *Open Network Automation Platform (ONAP)* trouxe aos desenvolvedores e provedores de serviço uma plataforma para gerenciamento de ciclo de vida de novos serviços (Linux Foundation, 2017). Os princípios básicos do *ONAP* são: orquestração em tempo real, orientada à política e automação dos recursos físicos e virtuais. A plataforma é a união de duas abordagens *MANO* a *OPEN-O* e *OpenECOMP* (SOUSA et al., 2019) que permitiu desacoplar detalhes dos serviços da tecnologia por meio de modelos de padronização de informação, gerenciamento genérico e plataforma de orquestração central.

Além disso, o *ONAP* provê a implantação e gerenciamento de serviços de rede valendo-se de *big data* e inteligência artificial como tecnologias habilitadoras. A implantação dos serviços por meio do *ONAP* pode ocorrer na rede do operador ou em nuvens privada.

O *ONAP* compõe rol de orquestradores para estabelecimento de serviços de rede em formato de fatias de recursos (TALEB et al., 2017b; BARAKABITZE et al., 2020; RODRIGUEZ; GUILLEMIN; BOUBENDIR, 2020). Conforme alguns de seus pares, o *ONAP* falha ao garantir um plano de dados multi-domínios, uma vez que ele instancia serviços como fatias de rede e os encadeiam em redes típicas de *data centers*, baseadas em *VLAN*, *VxLAN*, ou acopladas a domínios *SDN*. O aspecto multidomínios, sobretudo a troca de dados entre *ASs*, quando geograficamente distribuído se dá por meio da Internet.

Tecnologias de *Virtual Private Network (VPN)* e *SDWAN* desempenham o plano de dados para os serviços encadeados entre os domínios administrativos e tecnológicos, mas a rede de propósito geral – modelo *IP* da Internet–, é inerte às

tecnologias de fatiamento de rede. Além disso, não se observa a possibilidade de fatiamento recursivo, de forma que, um uma vez alocado um recurso a um usuário, ele consiga estabelecer seu plano de controle e gerenciar os recursos da fatia de rede conforme sua conveniência.

2.8.2 Projetos de Pesquisa

Muitas entidades padronizadoras, indústria, academia e associações realizaram iniciativas para fornecer modelos e descobertas científicas para compor *frameworks* de futuras arquiteturas de rede, especialmente na realização de fatiamento de rede. Nesta subseção estão descritas alguns trabalhos de pesquisa, destacando seus *frameworks* arquitetural e suas implementações.

2.8.2.1 *SONATA*

O projeto *SONATA*, de 2015, propôs uma arquitetura de serviço e estrutura de orquestração para o desenvolvimento de serviços virtualizados que são concebidos como fatias de recursos. A construção *SONATA* contêm dois componentes, o *Software Development Kit (SDK)*, no qual o modelo e as ferramentas de programação implementam os serviços usando a abordagem do *DevOps*. No *SDK*, há um catálogo público, onde ele armazena artefatos, como arquivos manifesto que descrevem funções e serviços.

A *Service Platform (SP)* integra o *SONATA* com interfaces para operadores de plataforma e desenvolvedores de serviços, além de permitir a implantação de serviços em diferentes infraestruturas. Essa interação foi mantida com o componente e a infraestrutura do *SDK* por meio do gerenciamento de *VIMs*, que coloca em execução os serviços (DRÄXLER et al., 2017).

2.8.2.2 *5G-Crosshaul*

No contexto do projeto *5G-Crosshaul* de 2015, foi proposta uma arquitetura de transporte para redes móveis calcada nos requisitos de flexibilidade e alocação eficiente de recursos baseada na especificação *NFV*. Os objetivos da solução foram permitir que uma fatia de rede fosse implantada na modalidade *"as-a-service"*, isso

se materializou valendo-se dos princípios de programabilidade suportadas por *SDN*. Assim, o mecanismo de controle da solução encadeia funções de rede inerentes ao ecossistema das redes móveis.

Sobre o mecanismo de controle, instanciação e colocação de recursos os autores exploraram o conceito de infraestrutura virtual de serviços, que permitiu que fossem explorados casos de uso como o de redes móveis virtuais. Os autores sustentam o ineditismo da solução na capacidade da solução de prover recursão de serviço para multi-inquilinos, uma vez que uma instância de serviço pode instanciar outra (LI et al., 2017a).

2.8.2.3 *5G!Pagoda*

O projeto *5G!Pagoda* de 2016 trouxe avanços para fatiamento de recurso no âmbito das redes móveis, especificamente para verticais de cidades inteligentes. A solução se realiza com uma arquitetura e um *framework* para provimento de fatias de rede com características de programabilidade e gerenciamento escalável. No âmbito do projeto foram construídos *testbeds* com as características supracitadas onde se permitiu a experimentação de aplicações de clima e sociedade segura.

A arquitetura da solução é representada por um modelo hierárquico de orquestração, onde cada domínio tecnológico possui seu próprio gerenciador de recursos que estão associados a um orquestrador geral, o qual desempenha o gerenciamento multidomínios. A arquitetura da solução é inspirada no *framework* da *ETSI*, no entanto, o projeto estendeu as características basilares para cumprir os objetivos do projeto. Assim, a arquitetura da solução possui três blocos interconectados: Orquestrador de *Slice* Multidomínios, Orquestrador Comercial de Fatia de Serviço, Orquestrador de Fatia Específico do Domínio (KOTULSKI et al., 2018).

2.8.2.4 *5GEx*

O projeto *5GEx* de 2016, inspirou-se no conceito de *IPExchange* e trouxe a orquestração de serviços pela perspectiva multidomínios. Além disso, proveu um gerenciamento híbrido de tecnologias de virtualização. Em seu *design* contem um Orquestrador, de alto nível, que lida com o aspecto multidomínios e fornece interface para gerenciamento do serviço.

O Orquestrador proposto lida simultaneamente com o gerenciamento de redes, e seu fatiamento, e máquinas virtuais em diferentes domínios por meio da troca de mensagens entre os componentes de Orquestração. A arquitetura da solução prevê que os Orquestradores estejam localizados em cada domínio para permitir a o fatiamento dos recursos em redes lógicas e permitir o gerenciamento centralizado de cada domínio sobre seus recursos (BERNARDOS et al., 2016).

Pela perspectiva de multi-domínios, o *5GEx* não provê um plano de dados no topo dos roteadores da Internet. Considera-seque o plano de dados é parcialmente orientado a característica multidomínios, pois a solução não encaixa plenamente aos múltiplos *ASs* da Internet, sobretudo ao modelo de roteamento *IP*. A separação geográfica do controle dos domínios, centrada em Orquestradores, suporta os requisitos tecnológicos e funcionais da interoperabilidade, no entanto, somente no nível de controle, pois o plano de dados considera a conectividade direta entre os *VIMs*.

2.8.2.5 *5GTransformer*

De forma análoga, o projeto *5GTransformer* de 2017 construiu um *testbed* para experimentação de verticais de experimentação para redes móveis para aplicações automotiva, entretenimento, *eHealth*, Indústria 4.0 e Redes Móveis Virtuais. O modelo de orquestração multidomínios do projeto é baseado em federação, onde cada infraestrutura, geograficamente distribuída, se conecta para provimento dos serviços requisitados.

Os principais componentes do modelo arquitetural do *testbed* é o fatiador de verticais, orquestrador de verticais e a plataforma de gerenciamento do transporte e recurso de computação das redes móveis. Esse último, em termos concretos muito se assemelha ao *VIM* do *framework NFV* da *ETSI*.

Além desses três componentes principais, a arquitetura da solução contém um mecanismo de monitoramento que perpassa os três e coleta métricas para garantia de acordos de serviço e qualidade de serviço. Também, a arquitetura prevê interfaces para suporte de operação e para recebimento das verticais de experimentação. No modelo arquitetural, o plano de dados inter domínios previu tecnologias de encapsulamento como: *VLAN*, *VxLAN*, *MPLS*, *VPN* e outros para troca de da-

dos entre os domínios administrativos e tecnológicos: *data center* (OLIVA et al., 2018).

2.8.2.6 *X-MANO*

X-MANO propõe um *framework* para implantação de serviços de redes virtualizados para diferentes domínios tecnológicos e administrativos. A proposta estabelece o princípio de confidencialidade, onde a troca de informações interdomínios ocorre somente por entidades autorizadas e cada domínio deve possuir interfaces de dados para troca de tráfego. A estrutura do *framework* compreende três interfaces lógicas, a saber: Gerenciador de Federação, Agente de Federação e o Orquestrador de Domínio.

O Gerenciador de Federação recebe do portal do usuário a descrição do serviço, organizado conforme sintaxe específica, e então distribui a especificação do serviço para cada domínio conforme descrito no arquivo *manifest*. O Agente de Federação, entidade de cada domínio, recebe a requisição e a converte a especificação de acordo com a especificidade de cada domínio. A especificação da implantação do serviço é recebida pelo Orquestrador de Domínio que proverá a implantação do serviço (FRANCESCON et al., 2017).

Além disso, o *X-MANO* utiliza pontos de troca de tráfego entre os domínios para trocar informação como o mecanismo de tunelamento entre os domínios, por isso, assume-se que seu plano de dados não é multidomínio. Questões de escalabilidade e políticas são qualitativamente deficientes e são observadas no mecanismo de controle, uma vez que sua estrutura hierarquicamente centralizada exige definir em qual domínio a entidade central do plano de controle permanecerá.

A abordagem dos autores se mostra inadequada considerando performance e concentração de um único ponto de falhas em uma entidade central. Para lidar com o desafio posto, a proposta desta tese realiza o conceito de controle distribuído de fatiamento de redes sobre múltiplos *ASs*.

2.8.2.7 *Katana*

O *framework Katana* foi desenvolvido dentro do escopo do projeto *5G Genesis* (KOUMARAS et al., 2018), que objetivou realizar fatiamento de rede na

borda (KOURTIS et al., 2020). Os componentes estruturais do *Katana* são entidades e papéis como *API* norte, mapeador de fatias de rede, provisionador de fatias de rede, monitoramento de fatias de rede e uma camada de adaptação. A *API* norte provê o gerenciamento do ciclo de vida das fatias de rede pelo experimentador ou pelo gerenciador da fatia de rede. O mapeador de fatias de rede provê um mecanismo de seleção ótimo para alocar recursos para novas fatias de rede.

O provisionador de fatias de rede é responsável por manter caminhos ao longo de uma conectividade *WAN* ou baseada em rádio. A entidade de monitoramento da fatia de rede lida com o monitoramento das fatias de rede implantada. A camada de adaptação provê uma abstração para os domínios tecnológicos que traduzem mensagens em formatos suportados pelos componentes. Experimentos conduzidos pelos autores considerando a *Key Performance Indicator (KPI)* implantação do serviço constrastou a aplicabilidade do *Katana* com soluções baseadas no *OSM* para implantação de fatias de rede na borda.

2.8.2.8 NECOS

O projeto *Novel Enablers for Cloud Slicing (NECOS)* é baseado no conceito de nuvem definida por fatia leve (LSDC) e materializa uma abordagem de fatia como serviço que abrange várias infra-estruturas de computação em nuvem. O *NECOS* visou enfrentar os desafios da implantação de aplicações e serviços por operadores de rede e provedores de serviços. Os desafios de implantação incluem eficiência energética, versatilidade, segurança e disponibilidade de recursos.

Além disso, os desafios da computação em nuvem tradicionais alavancaram o modelo de implantação do *NECOS* para enfrentar os desafios de estabilidade. O lançamento e a manutenção de serviços exigem um esforço considerável do operador e oportunidades limitadas para novos aplicativos e negócios. Os recursos de *design* incluíram gerenciamento de nuvem e rede, orquestração de serviços e monitoramento de recursos distribuídos (SILVA et al., 2018).

A abordagem *NECOS* propôs abordar os desafios inerentes ao fatiamento de recursos de computação para materializar o conceito *Slice as a Service*. Desafios como fatias de rede de vários domínios permaneceram abertas para investigação.

Nesse sentido, a proposta desta tese lida com essa problemática, trazendo uma abordagem para o estabelecimento de fatia de rede entre vários domínios.

2.8.2.9 *5GinFIRE*

A proposta *5GinFIRE*, desenvolvida em 2018 no âmbito do projeto *H2020*, propôs um ecossistema de experimentação baseado no modelo de referência *ETSI NFV* para a implantação de aplicativos na esteira do *5G*. O projeto abordou os desafios de uma visão holística e unificada de um *testbed* de experimentação vertical, bem como um ecossistema de testes para hospedagem e integração de aplicativos para atender aos requisitos aplicáveis a cada vertical.

A arquitetura da solução consiste em recursos computacionais geograficamente distribuídos e interconectados em um formato *hub-and-spoke* para a entidade de controle central. O projeto definiu o conceito de portal de verticais para usuários construírem, implantar e experimentar *VNFs* e *NSs*. O *5GinFIRE* avançou o estado-da-arte como um plataforma de testes de última geração, incorporando a escalabilidade da estrutura *ETSI MANO* para definir e descrever descrições de experimentos e propor o conceito de experimentação de formato vertical no modelo do *5G* (SILVA et al., 2019).

O *5GinFIRE* não fornece um mecanismo de plano de dados para vários domínios, pois a conectividade entre as *VNFs* de uma experimento ocorre no mesmo domínio da rede, de acordo com um plano preliminar de endereçamento. Portanto, a característica Plano de Dados Multidomínio não pode ser estabelecida, consequentemente uma fatia de rede de fim-a-fim não pode ser estabelecida. A arquitetura de controle é centralizada na entidade *MANO* que lida com aspectos de gerenciamento, controle e implantação de verticais sobre recursos distribuídos.

A proposta *5GinFIRE* insere seu conceito de fatia de recurso exclusivamente no *data center* e seu formato de comunicação baseado em *VPN*. As tecnologias habilitadoras, de acordo com seus pares contemporâneos, são *cloud computing* e *SDN*. Quanto ao Local de Implantação, considerando a perspectiva do usuário, é externo. O *5GinFIRE* é totalmente compatível com a estrutura *ETSI MANO*, tornando-o flexível, escalável e programável.

2.8.2.10 POWDER

Em 2018, o projeto *Platform for Open Wireless Data-drive Experimental Research* (*POWDER*) de Breen et al. (2020) foi proposto nos Estados Unidos. O *POWDER* endereçou o compartilhamento de recursos e desenvolveu um novo ecossistema de experimentação de aplicações. O *POWDER* é uma plataforma em escala urbana para experimentação sem fio avançada, como novas tecnologias de comunicação e banda larga sem fio avançadas. Os recursos altamente programáveis e flexíveis impulsionaram a plataforma a oferecer múltiplas entradas e múltiplas saídas, um capacitador chave para redes *5G* e além.

O *POWDER* tem três componentes principais: infraestrutura física onde as instalações são construídas, a funcionalidade que combina *hardware* e *software* para realizar a funcionalidade da infraestrutura e a estrutura de controle que gerencia recursos e fornece serviços de experimentação aos usuários.

A camada de *hardware* do *POWDER* possui equipamentos de rádios, estações base e computação interconectados por fibra dedicada em áreas centrais, residenciais e em um ambiente de *campus* universitário acidentado. Além disso, a camada de *software* inclui virtualização de rede de uso geral bem conhecida e pilhas de computação em nuvem e várias pilhas de rádio definido por *software*.

2.8.3 Contribuições de Pesquisa

Existem inúmeros esforços e provas de conceito que objetivaram oferecer fatiamento de rede entre múltiplos domínios. Esta subseção considera trabalhos que variam desde fatiamento de redes móveis até abordagens mais genéricas. Além disso, traz trabalhos fora da esteira dos projetos ou consórcios *5G*. Nesta subseção serão apresentadas algumas dessas abordagens, destacando seus papeis tecnológicos e seus conceitos fundamentais.

2.8.3.1 *5G Cross-Domain*

Taleb et al. (2019) propôs um modelo arquitetural para prover fatiamento de redes entre diferentes domínios administrativos conforme a combinação de quatro componentes. O primeiro, o *Service Broker*, trata as requisições de criação de

fatias entre os múltiplos domínios e mantém um repositório global que suporta a implantação dos serviços. O componente *Service Conductor* provê o gerenciamento entre domínios federados, uma vez que a requisição de criação da fatia é submetida e validada pelo *Service Broker*. O terceiro componente, cujo comportamento é similar a orquestrador, aloca recursos internos de cada domínio no processo de estabelecimento de fatiamento e lida com o ciclo de vida das fatias.

Além disso, é possível dividir o terceiro componente em funcionalidades menores, quais sejam: gerenciamento do serviço, gerenciamento do ciclo de vida da fatia, gerenciamento e orquestração para subdomínios e controlador *SDN* para os subdomínios. O quarto componente representa a infraestrutura dos subdomínios, que envolvem: *VNFs*, recursos virtuais, camada de virtualização, infraestrutura física.

Em contraste com a presente proposta, a solução dos autores não provê um plano de dados multi-domínios. Além disso, o mecanismo de controle baseado em federação concentra em um único orquestrador, de forma monolítica, as funcionalidades de ciclo de vida das fatias de rede. Não se discute a possiblidade de gerenciamento e provimento recursivo fatias de rede e nem a possibilidade de ser implantada sobre as redes legadas, notadamente a Internet.

2.8.3.2 *5G Framework*

Em Alhuseini e Olama (2019) está proposto um *framework* para projetar e operar fatias de rede com flexibilidade, automação e colaboração. A proposta subdivide a implementação da solução em componentes como Definição, Solucionamento, Escopo, Instanciação e Atualização do Serviço. A Definição, diz respeito à fase onde as características dos serviços, isto é, "o que" são definidas por meio de um documento contendo a descrição do serviço. O Solucionamento do Serviço, é a fase onde são definidas as ligações, compatibilização de *hardware* e *software*, visualização da cadeia dos serviços.

Além disso, contém o "como fazer" que é expresso em formato estruturado, que inclui as interfaces, características, recursos e ativos que descrevem o serviço. O Escopo do Serviço compreende a tradução das especificações de *design* para entregáveis que contêm o escopo do serviço. Esses entregáveis são: oferta, preço,

lista de materiais, plano de entrega e demanda de time. O fim da fase de escopo provê o documento de Projeto de Alto Nível que alimenta as fases subsequentes.

A etapa de Instanciação do Serviço trata da implementação por meio do gerenciamento do ciclo de vida que aciona mecanismos de automação. A definição do serviço está descrita no documento: Projeto de Baixo nível que está formatado conforme as linguagens *TOSCA* e *YANG*. Além do *framework*, os autores contribuem com um método para instanciação do fatiamento de rede baseado em histórias dos usuários. A atualização do serviço utiliza a interface de ciclo de vida para criar, terminar e modificar as fatias de rede, conforme se modifica os requisitos de negócios.

Os autores experimentaram a solução com cenários onde houve fatiamento de rede, a métrica utilizada foi caga de trabalho no processo de registro de assinantes de serviço de telefonia. Diferente desta tese, o *framework* proposto contém uma estrutura monolítica, não viabiliza um plano de dados e controle multi-domínios, além de não prover o fatiamento fim-a-fim. O lugar da rede que a solução dos autores consegue atuar é no *data center*, a tecnologia de separação da fatias de rede é baseada em *SDN*, assim não se compatibiliza com redes legadas e não provê suporte para fatiamento recursivo.

2.8.3.3 Fatiamento Recursivo 1

HÉno, Boubendir e Simoni (2019) descreveram um modelo conceitual de fatiamento recursivo de rede inspirado na definição de requisitos de virtualização de recursos computacionais de Popek e Goldberg (1974). O modelo conceitual de fatiamento de rede recursivo prevê a capacidade de uma fatia de rede ser capaz de suportar fatias de rede aninhadas. O principal componente do modelo é o nó de rede que é uma máquina genérica que suporta virtualização e encaminhamento de pacotes.

O modelo proposto divide o provedor de infraestrutura em dois papéis: o provedor e operador da infraestrutura física que é o gerenciador do *AS* e o provedor e operador da infraestrutura virtualizada que oferecem seus serviços de forma independente da camada subjacente.

Diferente desta tese, a solução de HÉno, Boubendir e Simoni (2019) não é

compatível com o *framework* gerenciamento e orquestração da *ETSI* (ETSI, 2013a). Nossa proposta apresenta um modelo hierárquico para fatiamento de rede que permite o estabelecimento de fatias de rede recursiva ao longo de múltiplos *ASs*. Diferente de HÉno, Boubendir e Simoni (2019) a presente proposta se apresenta mais flexível por permitir que uma fatia de rede seja implantada por caminhos alternativos ao construído pelos algoritmos de roteamento.

2.8.3.4 Fatiamento Recursivo 2

Em Yousaf et al. (2019) está apresentado uma abordagem de fatiamento de recursos de rede e computação de forma recursiva no formato de *MANO* como serviço (*MANO-as-a-Service*). A proposta dos autores, ao estender o *framework NFV*, viabiliza que sejam criadas para os usuários fatias de recursos gerenciadas exclusivamente por um *MANO* privativo. Além disso, são destacadas as contribuições dos autores, a saber: uma arquitetura *MANOaaS* e o processo distribuído de negociação do gerenciamento dos níveis de acordo entre a infraestrutura e os inquilinos.

Por meio de simulações na ferramenta *MATLAB*, os autores demonstraram que a taxa de sucesso na resposta de implantação de uma fatia de recursos, quando existem fatiamento prévio sobre a infraestrutura, é percentualmente maior quando utilizado a abordagem deles frente a abordagem parcial e completamente sem gerenciamento.

Em contraste com a esta tese, a abordagem de Yousaf et al. (2019) não prove um plano de dados orientado a multidomínios, pois a tecnologia de conectividade entre os domínios é inerte ao estabelecimento de uma fatia de rede. Os autores não deixam claro qual a tecnologia de troca de tráfego entre os domínios, intuitivamente, por se tratar de uma atuação dentro do domínio do *data center*, é razoável presumir que a abordagem considera uma comunicação baseada em *SDN*. No que se refere ao plano de controle, pode-se classificá-lo como orientado a multidomínios. A arquitetura da solução dos autores é monolítica, mas o serviços que ele propõe, nesse caso a implantação dos *MANOs*, são distribuídas ao longo dos domínios. Além disso, na perspectiva do plano de dados da Internet, a abordagem não provê um fatiamento fim-a-fim.

Assim, classifica-se essa abordagem como não orientada a rede legadas. Por outro lado, a abordagem dos autores é compatível com o *framework MANO* da *ETSI*, de fato, os autores avançam o estado da arte com um *framework* aprimorado para esse fim. Dois aspectos importantes são: a interface para implantação de serviços e a interface para serviços terceiros, por ser uma abordagem compatível com o *framework MANO*, assume-se que diretamente a proposta satisfaz esses requisitos.

Em referência ao fatiamento recursivo, conforme a taxonomia proposta, a abordagem dos autores não realiza plenamente a característica. Nesta tese, entende-se como fatiamento recursivo a possibilidade de uma fatia de rede poder ser redimensionada indefinidamente, com gerenciamento único e independente para cada fatia, até o limite de recursos da fatia original.

2.8.3.5 *PERMIT*

A abordagem *PERMIT* torna possível o fatiamento de recursos em redes móveis com objetivo de prover diferentes níveis de granularidade, como: por rede, aplicação, grupo de usuários, usuários individuais e por dados dos usuários. Similar a seus pares, a abordagem dos autores combina conceitos de programabilidade de rede com computação em nuvem como tecnologias habilitadoras do *framework*.

São previstos dois orquestradores na solução, o *Mobile Network Personalization Service Orchestrator (MNP-SO)* e o *Mobile Service Personalization Service Orchestrator (MSP-SO)*, que possui mecanismos para a personalização do serviço e da rede móvel para os usuários. A arquitetura da solução é inspirada no *framework NFV* e avançou o estado da arte com um orquestrador de funcionalidades voltadas a criação de redes móveis virtuais.

Além disso, no *framework* está previsto que entre os orquestradores de redes móvies virtuais a entidades *Slice Border Control (SBC)* são responsáveis pela troca de dados, classificação das aplicações e atua no estabelecimento de encadeamento de funções de rede. No que diz respeito ao plano de dados multidomínios, a solução dos autores inviabiliza que recursos reservados para uma fatia em um domínio seja, ao longo do caminho da Internet, garantidos em outro domínio.

Além disso, o plano de controle não exerce influência em outros domínios ad-

ministrativos e tecnológicos. É possível observar na abordagem dos autores um orquestrador capaz de fatiar recursos nas entidades de redes móveis de forma que o grau de personalização da conexão do usuário é de granularidade elevada devido a programabilidade (TALEB et al., 2017a).

Em contraste com esta tese, em Li et al. (2017a) o fatiamento de recursos se dá somente na perspectiva da aplicação e de controle. O plano de dados não é instanciado indefinidamente até o limite da fatia original conforme esta tese propõe. Além disso, o mecanismo dos autores prevê uma escolha fixa do caminho ao longo da fatia de serviço. Devido a solução dos autores não ser orientado a multidomínios, não é considerado a possibilidade de a escolha do caminho ser baseada na escolha do usuário/dono da fatia de rede.

Nesse sentido, considerando a característica de orientação a política aberta, a solução dos autores não adiciona dinamismo no processo de estabelecimento e manutenção de fatias de rede. Isso abre espaço para contribuições de um mecanismo de fatiamento de rede recursivo que engloba os espectros de aplicações, planos de controle e dados. Além disso, permite argumentar a aplicabilidade de uma solução com interfaces abertas para especificação de caminhos de fatias de rede.

2.8.3.6 Otimizador Multidomínio

A abordagem de Sun et al. (2019) refere-se a mecanismo orientado a otimização para o gerenciamento de funções de rede ao longo de múltiplos *ASs*. Nesse sentido, os autores formularam e experimentaram um algoritmo de otimização que dado um conjunto de restrições matemáticas deve-se prover o melhor encadeamento que observe a maximização a eficiência energética.

A definição do problema de otimização descreve que na medida que as requisições de implantação de serviços computacionais chegam, sob uma probabilidade definida, a solução deve ser capaz de prover o melhor mapeamento da requisição do usuário para os recursos computacionais ao longo de um cenário multidomínio. Adicionalmente, no modelo de otimização proposto foram observadas a conectividade entre as entidades virtualizadas e seu melhor encadeamento ao longo de vários domínios.

Em contraste com a proposta desta tese, os autores propuseram um mecanismo

de gerenciamento de encadeamento de funções de rede. Entretanto, é importante pontuar as diferenças conceituais entre a abordagem dos autores com a descrita nessa tese. A solução de conectividade descrita em Sun et al. (2019) é orientada a encadeamento de funções de rede, ao passo que a desta tese propõe um mecanismo de conectividade lógica que transcende o encadeamento de funções para os espectros de separação de aplicação, plano de controle e de dados em cenário de múltiplos *ASs*. Portanto, a solução dos autores não provê a implementação de um plano de dados multidomínios, ainda que seja possível caracterizar seu plano de controle como orientado a multidomínios.

A arquitetura da solução é hierárquica e o lugar da rede onde a solução configura os parâmetros de encadeamento de funções é no núcleo da rede. No entanto, cabe destacar que a abordagem dos autores permanece no campo da simulação, portanto não é apresentado um experimento conciso em um cenário próximo do real. Sob a perspectiva de plano de dados da Internet a solução não provê o fatiamento fim-a-fim, uma vez que se dedica a prover conectividade de entidades virtualizadas ao longo de múltiplos *ASs*.

Não é discutido na abordagem dos autores um mecanismo de troca de dados entre os domínios, no entanto é possível depreender da descrição que a solução permite que sejam implantadas entidades virtualizadas no núcleo da rede, logo pela perspectiva do usuário doméstico é uma abordagem externa.

A compatibilidade com redes legadas é outro ponto questionável, uma vez que a solução não apresenta a implementação de um mecanismo para lidar com o fatiamento de rede nesse ecossistema. Além disso, não é possível constatar que solução dos autores seja compatível com o *framework NFV* da *ETSI*. Por haver uma interface para receber a requisição de fatiamento de rede é rezoável sustentar a característica de interface para serviços terceiros.

No entanto, no que toca o fatiamento recursivo, ou eventualmente encadeamento de funções recursivas, a solução dos autores não satisfaz essa característica. Por fim, no que diz respeito ao dinamismo na implantação do serviço a solução dos autores, similar a proposta desta tese, por abstrair a topologia como uma estrutura de dados Grafo, é capaz de adicionar dinamismo na escolha dos caminhos com algoritmos de otimização.

2.9 Discussão

A Tabela 1 sumariza algumas abordagens do estado da arte como Orquestradores de Código Aberto, Projetos de Pesquisa e Contribuições de Pesquisa. Essas abordagens objetivaram realizar de alguma forma a implantação de fatias de rede sobre recursos computacionais e de rede.

Multi-domain Data Plane. Define se o mecanismo de construção de caminhos lida com parâmetros de configuração entre entidades que estão em domínios diferentes. Esse mecanismos deve prover configurações lógicas e acordos de serviços honrando suas métricas, mesmo quando o caminho construído perpassa outros *ASs*. Algumas abordagens provêm um plano de dados multi-domínios como em Oliva et al. (2018), Bernardos et al. (2016), mas a maioria deles não provêm (DRÄXLER et al., 2017; LI et al., 2017a).

Multi-domain Control Plane. Refere-se às rotinas de controle e configuração no processo de estabelecimento de fatia de rede que perpassam múltiços *ASs*. Esta coluna refere-se à habilidade de lidar com orquestração da fatia de rede e gerenciamento distribuído entre múltiplos domínios. Algumas soluções oferecem essa característica em domínios epecíficos como *data center* ou nas redes móveis (TALEB et al., 2017a; TALEB et al., 2019), outras na rede de transporte (BERNARDOS et al., 2016; SUN et al., 2019).

Control Architecture. Essa característica diz respeito ao bloco estrutural da arquitetura. Muitas soluções são monolíticas (DRÄXLER et al., 2017), implicando em um único ponto de falha. Outras abordagens são hierárquicas (SILVA et al., 2018), lançando dúvidas sobre a independência lógica e funcional dos domínios que a fatia de rede passa. Por fim, arquiteturas de fatiamento de rede distribuídas são conhecidas (BERNARDOS et al., 2016), mas a despeito de serem distribuídas não oferecem um plano de dados distribuído.

Network Placing. Caracteriza cada solução com relação a sua efetividade e habilidade de influenciar entidades fim-a-fim na rede de transporte. Algumas soluções lidam com o fatiamento de rede dentro do *data center* (KOTULSKI et al., 2018; Linux Foundation, 2017), implicando em uma influência limitada, pois atua somente nesse domínio tecnológico. Outas provêm fatiamento de rede no núcleo da rede ou na rede de acesso e suas entidades (TALEB et al., 2019; TALEB

et al., 2017a).

Slicing on top of the Internet. Caracteriza o mecanismo fatiador com relação a sua habilidade de pavimentar um plano de dados no topo das entidades da Internet, isto é, os roteadores. Muitas propostas de fatiamento de rede usam tecnologias para a pavimentação de caminho de dados como *VPN* (KOTULSKI et al., 2018; SILVA et al., 2019), encapsulação (HUANG et al., 2017), ou restritas a um único domínio como as abordagens baseadas em *SDN* (YOUSAF et al., 2019).

Inter-Domain Data Exchange. Preocupa se as soluções garantem o fatiamento de rede entre múltiplos *ASs*. Algumas soluções endereçam isso separadamente, habilitando o fatiamento de rede no acesso (TALEB et al., 2019), no núcleo (SUN et al., 2019), outras no *back-haul* (ZHANG, 2019) deixando de lado o fatiamento de rede entre múltiplos *ASs*.

Enabling Technologies. Se preocupa em descrever o *framework* tecnológico ou a tecnologia prevalecente que a abordagem incluiu em sua arquitetura. Muitas soluções utilizam as capacidades *SDN* (KOTULSKI et al., 2018; SONKOLY et al., 2015). Outras utilizam *NFV* (YOUSAF et al., 2019) ou *frameworks* específicos para lidar com o fatiamento de rede (Alhuseini; Olama, 2019).

Legacy Network Compatibility. Caracteriza as soluções conforme suas incorporação em um *Internet Service Provider (ISP)* ou *AS* sem requerer mudanças significativas no formato de implementação e operação dos recursos. Algumas soluções, da perspectiva de implantação, a abordagem de fatiamento requer mudanças consideráveis na arquitetura de redes e seus níveis ou em aquisição de equipamentos. Algumas soluções requerem modificações no núcleo, na *WAN* e no acesso (YE et al., 2018).

ETSI MANO Compatibility. Define se a solução de fatiamento de rede é compatível como o bem estabelecido *ETSI MANO*. Algumas soluções provêm mecanismos de gerenciamento e orquestração com compatibilidade limitada ao domínio que atuam (SILVA et al., 2018; Linux Foundation, 2017). A visão de fatiamento de rede desta tese preocupa-se em um orquestrador aberto e capaz de atuar em cooperação com soluções padronizadas e amplamente aceitas como o *OSM*.

Service Chaining. Descreve se a solução de fatiamento de rede provê serviços de rede e computação interligados logicamente dentro de uma fatia de rede

implantada entre múltiplos domínios.

Third-party Interface. Caracteriza se o mecanismo de fatiamento de rede pode criar, modificar ou deletar as especifidades e propriedades da fatia de rede sem requerer o gerenciador ou orquestrador do *ISP* ou *AS*. Muitas soluções de fatiamento de rede ainda não dão aos usuários o controle sobre suas fatias de rede (CSOMA et al., 2014; LI et al., 2017a; SILVA et al., 2019).

Recursive Slicing Establishment. Descreve se a solução é rapas de redimensionar uma fatia de rede já implantada, diminuindo-a ou criando novas sub-fatias até alcançar o limite físico e lógico da fatia de rede principal. No redimensionamento é esperado entregas sub-fatias aos usuários com plano de gerenciamento, controle e de dados separados. Apesar de os *frameworks* de especificação definir o fatiamento recursivo de rede (SABOORIAN; THIEBAUT; XIANG, 2017), muitas soluções não realizam o fatiamento de rede entre *ASs*, especialmente de maneira recursiva (DRÄXLER et al., 2017; TALEB et al., 2017a; OLIVA et al., 2018).

Dynamic Service Deployment. Diz respeito a possibilidade do usuário ou dono da fatia de rede ser capaz de definir o caminho de dados da sua fatia de rede e seus atributos para o *AS*. Isso dá ao usuário a possibilidade de se definir os parâmetros da fatia de rede nas entidades intermediárias de um *ISP* ou *ASs*, pavimentando um caminho ponta a ponta e perpassando vários domínios. A maioria das soluções não dão aos usuários ou donos das fatias de rede a liberdade de especificar os parâmetros e o caminho para suas fatias de rede.

No levantamento realizado, além das características supracitadas da tabela, destacam-se os marcadores que aludem a pertinência da solução quanto a descrição da coluna. Assim, os marcadores definem três dimensões da característica em análise, e são denotados conforme: (○) representa a ausência da característica para a solução; (●) sugere que a solução do estado-da-arte satisfaz plenamente a característica; (◑) representa o cumprimento parcial da característica em análise. Uma observação que se faz da Tabela 1 é que não há abordagem que cumpre todos os itens que as qualificam. Isso abre caminho para posicionar a presente contribuição frente a seus pares e sistematizar o detalhamento qualitativo de cada solução.

Tabela 1 – Discussão dos Trabalhos Relacionados.

Approach	SOTA Category	Multi-domain Data Plane	Multi-domain Control Plane	Control Architecture	Network Placing	Slicing on top of the Internet	Inter-Domain Data Exchange	Enabling Technologies	Legacy Network Compatibility	ETSI MANO Compatibility	Service Chaining	Third-party Interface	Recursive Slice Establishment	Dynamic Service Deployment
ESCAPE (CSOMA et al., 2014)	Orquestrador Open-source	○	●	Monolithic	N/A	○	SDN Domain	SDN and Cloud	○	○	●	○	○	○
X-MANO (FRANCESCON et al., 2017)	Research Project	○	●	Hierarchically Centralized	DC	○	IP Tunnel	SDN and Cloud	○	●	●	◑	○	○
OSM (ETSI, 2016)	Orquestrador Open-source	○	●	Monolithic	DC	○	SDN Domain and VPN	SDN and Cloud	○	●	●	●	○	○
5GEx (BERNARDOS et al., 2016)	Projeto de Pesquisa	◑	●	Distributed	Core	○	IP Exchange	SDN and Cloud	●	●	●	●	○	○
5G Framework (Alhuseini; Olama, 2019)	Contribuição de Pesquisa	○	○	Monolithic	DC	○	SDN Domain	SDN, Cloud, and BPMN	○	●	●	●	○	○
5G Cross-Domain (LI et al., 2017a)	Contribuição de Pesquisa	○	●	Monolithic	DC, and Mobile Network Access	○	SDN Domain	SDN, and Cloud	○	●	●	○	○	○
SONATA (DRÄXLER et al., 2017)	Projeto de Pesquisa	○	●	Monolithic	DC	○	SDN Domain	SDN, and Cloud	○	●	●	●	○	○
5GinFIRE (SILVA et al., 2019)	Projeto de Pesquisa	○	○	Monolithic	DC	○	SDN Domain, and VPN	SDN, and Cloud	○	●	○	●	○	○
5G!Pagoda (KOTULSKI et al., 2018)	Projeto de Pesquisa	○	●	Monolithic	DC	○	Internet, and VPN	SDN, and Cloud	○	●	○	○	○	○
5GTransformer (OLIVA et al., 2018)	Projeto de Pesquisa	●	●	Distributed	DC	○	VPN Encapsulation	SDN, and Cloud	○	●	○	●	○	○
ONAP (Linux Foundation, 2017)	Orquestrador Open-source	○	●	Monolithic	DC	○	Internet, VPN, and SDN	SDN, and Cloud	○	○	●	●	○	○
PERMIT (TALEB et al., 2017a)	Contribuição de Pesquisa	○	○	Monolithic	Mobile Network Access	○	Internet	SDN, Cloud, and RAN	○	●	●	●	○	○
NECOS (SILVA et al., 2018)	Projeto de Pesquisa	○	●	Hierarchical	Core, and Edge	○	VPN, and SDN	SDN, Cloud, and Containers	○	○	●	◑	○	○
POWDER (BREEN et al., 2020)	Projeto de Pesquisa	○	○	Monolithic	DC, Mobile Network Access, Edge	○	L2 and L3 Domain	SDN, NFV, Cloud, SDR, MEC	○	○	○	●	○	○
5G Crosshaul (LI et al., 2017a)	Projeto de Pesquisa	○	○	Hierarchical	Core	○	PBB-TE	SDN, and NFV	○	○	●	○	◑	○
Recursive Slicing 1 (HÉNO; BOUBENDIR; SIMONI, 2019)	Contribuição de Pesquisa	●	○	Monolithic	DC, and Core	○	VLAN over Segment Routing	SDN, NFV, and Cloud	○	○	●	○	●	○
Recursive Slicing 2 (YOUSAF et al., 2019)	Contribuições de Pesquisa	○	○	Monolithic	Core	○	L2 Domain	SDN, and Cloud	○	●	○	●	◑	○
Multidomain Optimizer (SUN et al., 2019)	Contribuições de Pesquisa	○	○	Hierarchical	Core	○	N/A (Simulation)	SDN, and Cloud	○	○	●	◑	○	◑
Katana (KOURTIS et al., 2020)	Projeto de Pesquisa	○	●	Monolithic	Mobile Network Access and Edge	○	Internet VPN or L2 Domain	SDN, SDR, Cloud, MEC	○	●	●	◑	○	○
NASOR	Orquestrador Open-source	●	●	Hierarchically Distributed	Core	●	Internet	SDN, NFV, Cloud, and SR	●	●	●	●	●	●

2.10 Considerações Finais

Pelas tecnologias que a comunidade científica gradativamente tem aprimorado, torna-se possível construir serviços e mecanismos de gerenciamento sofisticados para lidar com os requisitos dos usuários. O vetor da evolução que se assiste, em inúmeras delas, está associado a demandas dos usuários, e ao formato de requisitos que tecnologias se moldam para atender novas funcionalidades e requisitos. Nesse sentido, *cloud computing*, *SDN*, virtualização de rede, fatiamento de rede e roteamento por segmento, podem ser agrupados, de forma que, o mais apropriado de cada tecnologia, combina-se para construir um novo formato de fatiamento de rede.

Inúmeras propostas do estado-da-arte, conforme apresentadas nesse capítulo, consideram essas tecnologias para suas propostas de fatiamento de recursos. Além disso, os desafios inerentes ao fatiamento de redes receberam respostas da comunidade de maneira multiforme. Algumas, se deram em projetos voltados para as redes móveis como *5G* que exaustivamente buscou incorporar programabilidade em seus elementos e estabelecer um ecossistema de experimentação. Por outro lado, encontram-se também abordagens mais agnósticas a tecnologia de rede, as quais buscaram contribuir com o fatiamento de rede em *data centers*, redes de acesso, *WAN* e *Core*.

Capítulo 3

NASOR

3.1 Considerações Iniciais

Neste capítulo é apresentada uma abordagem para fatiamento de recursos inter domínios administrativos. O fatiamento de recursos, concebido como *network slicing* conforme Capítulo 2, se dá sobre o plano de dados construído pelos algoritmos de roteamento, o que torna a proposta desta tese compatível com as redes legadas e apropriada para prover um fatiamento fim-a-fim na perspectiva de *ASs*. A abordagem desta tese, estende o *framework* de gerenciamento e orquestração de rede proposto pela *ETSI* ao materializar uma solução distribuída e capaz de estabelecer um fatiamento de rede recursivo ecoando sobre múltiplos *ASs*.

3.2 Visão Conceitual do Fatiamento de Rede

A perspectiva de compartilhamento de recursos, consolidada no processamento de tempo compartilhado e remanescentemente na virtualização, direcionou a comunidade científica na padronização e construção de mecanismos capazes de prover separação lógica de recursos com garantias e isolamento. Em especial, no ecossistema das redes móveis, o fatiamento de recursos tem sido explorado com vistas a suportar um modelo de serviços orientado a verticais, onde se buscou abordagens de conectividade com métricas de rede definidas pelas aplicações. Realizar a separação lógica de recursos, com vistas customização da conectividade, tornou-se

possível após os *frameworks* arquiteturais das redes designarem entidades, papéis e a separação de planos de gerenciamento.

A separação dos planos de dados e controle percebida no paradigma *SDN* pode ser estendida para outros contextos. Nesta tese, é proposta a separação dos planos de gerenciamento e serviços nos *frameworks* arquiteturais dos gerenciadores de serviços de rede. Como exemplo, o *OSM* possui o plano de gerenciamento e serviços desassociados, especificamente sob uma perspectiva de virtualização de funções de rede e suas conectividades. O *OSM* foi construído à luz do *framework* da *ETSI*, por isso incorporou em seus módulos os papéis que realizam o gerenciamento e orquestração dos serviços de rede.

Conforme o *framework* da *ETSI*, é possível atribuir ao *MANO* papéis e classificá-lo como uma entidade do plano de gerenciamento das *VNFs*. Incluem o rol de entidades gerenciáveis: os recursos de *hardware*, *software* e serviços. Os recursos de *hardware* representam a infraestrutura computacional, rede e armazenamento subjacente. Sobre esses recursos, os *softwares* gerenciados são referidos como *VNFs* que se organizam em cadeia e materializam o serviço implantado ou uma fatia de rede.

O plano de gerenciamento especificado no *framework* da *ETSI* sugere que a organização desses recursos, implantados sobre tecnologias de virtualização, entregam o conceito de fatiamento de recursos. Ao introduzir mecanismos como *WAN Infrastructure Manager (WIM)*, conforme especificação técnica da versão 5 do *OSM* (ETSI, 2019), objetivou-se realizar o conceito de estabelecimento fatias de rede sobre múltiplos *ASs*. Por isso, o plano de gerenciamento incluindo os mecanismos *VIM* e *WIN*, que são entidades do *framework MANO*, intencionaram realizar em conjunto o fatiamento de rede em cenários de múltiplos *ASs*. Existem abordagens realizando o gerenciamento de recursos ópticos intencionando prover o fatiamento de rede inter domínios (CASELLAS et al., 2018a).

É certo que as abordagens de conectividade inter-site existem variando conceitualmente e tecnicamente, como a proposta de Casellas et al. (2018b). Essa proposta foi oficialmente incluída no *framework* de gerenciamento do *MANO* como um mecanismo de fatiamento de rede entre múltiplos sites por meio de soluções ópticas. Outras abordagens como o estabelecimento de *VPN* entre entidades do seviço em cadeia são conhecidas – conforme descrito no Capítulo 2, e sustentam

argumentos de realizar o fatiamento de rede entre múltiplos domínios.

Nesse sentido, a concepção de fatiamento de redes nesses *frameworks*, sobretudo a entrega de serviços, realiza-se por meio da entidade *MANO*. As capacidades de implantar fatias de rede entre múltiplos *ASs* dessas entidades do plano de gerenciamento, sobretudo sua capacidade de gerenciamento e orquestração, são limitadas ao domínio e dependentes de tecnologias de conectividade cruzada como *VPN* ou *MPLS*. Logo, nesta tese entidades *MANO* como o *OSM* são classificadas como *MANO* Local.

Além disso, nesta tese está proposto um mapeamento do papel de fatiamento de rede que o *NASOR* realiza em contraste com dois *frameworks* padrão. Para tornar esse mapeamento claro, o *NASOR* será descrito a partir de dois pontos de vista: o primeiro referindo-se ao fatiamento de rede no contexto de serviços de rede para sistemas não *3GPP*, que é o caso das entidades *MANO* tradicionais como o *OSM*. O segundo ponto de vista refere-se ao fatiamento de rede para sistemas *3GPP*, onde o *NASOR* assume e desempenha papéis para a realização de fatiamento de rede em entre múltiplos domínios. O *NASOR* compila essas duas perspectivas propondo um *framework* para implantação de fatias de rede sobre entre múltiplos *ASs*.

De acordo com a *ETSI*, um Serviço de Rede é a composição e organização de funções de rede interconectadas. Um serviço de rede pode eventualmente conter uma ou mais instâncias de fatias de rede ou funções de rede interconectadas com substrato físico ou virtualizado. O *NASOR* assume e realiza papéis de gerenciamento rede no lugar das entidades *MANO OSM*, dado que essa entidade não endereça satisfatoriamente o fatiamento de rede entre múltiplos *ASs*.

Por outro lado, o *3GPP* compreende que uma fatia de rede suporta Serviços de Comunicação. Assim, conforme a perspectiva do *3GPP*, uma Fatia de Rede pode conter Instâncias de Sub-rede de Fatias de Rede. Tanto o *3GPP* quanto o *ETSI* possuem papéis e relacionamentos intercambiáveis nas suas visões de implantação de fatias de rede. No entanto, essas entidades e seus papéis carecem de serem distinguidos de acordo com a visão de fatiamento de rede da *ETSI* e *3GPP*.

3.2.1 *ETSI*

No contexto das soluções de gerenciamento e orquestração, é possível atribuir papéis ao *MANO* e classificá-los como uma entidade do plano de gerenciamento. Sem perda de rigor, um exemplo exemplo de fatiamento de rede sem o *NASOR* é conforme ilustrado na parte esquerda da Figura 8. Note que a conectividade entre *VNFs*, estando ou não entre múltiplos *ASs* é de responsabilidade do *WIM*. Mas as tecnologias *WIM* conhecidas no estado da arte comumente implantam as fatias de rede com tecnologias acopladas ao domínio, como *SDN*, *MPLS* ou eventualmente de domínios cruzados como *VPN*. Por isso, esta tese lança olhar para a parte direita da Figura 8 propondo um novo método que é orientado ao plano de dados construído pelos algoritmos de roteamento da Internet.

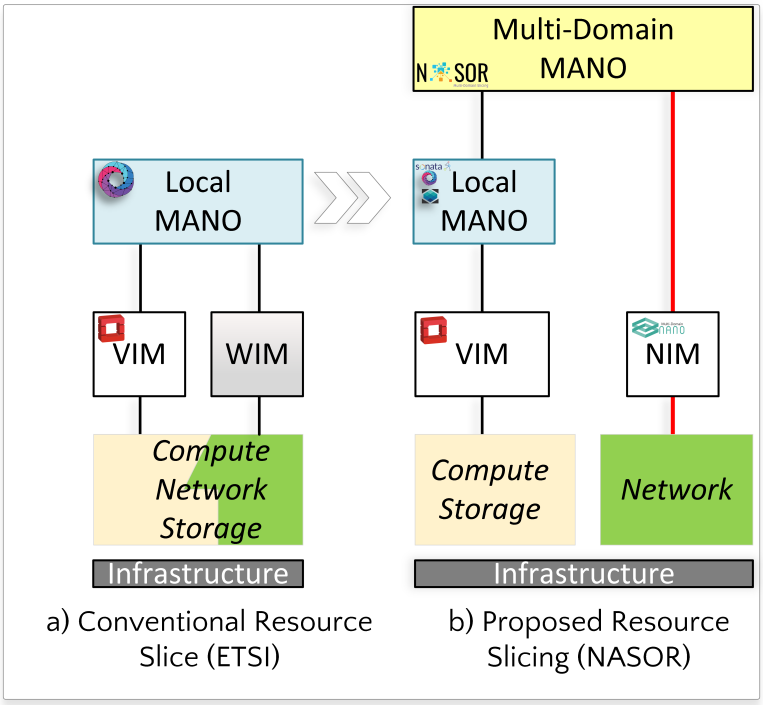

Figura 8 – Visão do Fatiamento de Rede estendido da *ETSI*.

Assim, a visão de fatiamento de rede desta tese é orientada por um plano de gerenciamento entre múltiplos domínios. Na Figura 8 estão ilustrados dois mecanismos de implantação e gerenciamento de fatias de rede. A parte *a* da

figura representa a estrutura convencional de implantação de serviços, conforme previsto no *framework* da *ETSI*. No topo está o *MANO* local que além dos demais papéis interage diretamente com o *VIM*. Quando a implantação de fatias de rede perpassam múltiplos domínios, quem realiza esse papel é o mecanismo *WIM* para prover e configurar os parâmetros da conectividade. O plano de dados que o *WIM* estabelece, tradicionalmente, considera tecnologias como *VPN*.

A visão de fatiamento de redes desta tese é orientada por um plano de gerenciamento de múltiplos domínios, conforme a parte *b* da Figura 8. Considerando uma leitura topo para baixo, esse plano de gerenciamento vale-se da capacidade de implantação de serviços de computação existente nos *MANOs* Local. No entanto, o papel da configuração dos parâmetros de conectividade passam a ser desempenhados pela entidade *Network Infrastructure Manager (NIM)*. A entidade *NIM* é parte da proposta desta tese para realização de implantação de fatias de rede entre múltiplos *ASs*. Com essa abordagem, o *MANO* Local lida somente com aspectos de serviços de computação, ao passo o fatiamento de rede entre múltiplos domínios necessita de uma entidade global. Essa entidade global desempenha o papel de *MANO* Multidomínio e interage com outras entidades de mesmo nível em outros domínios para pavimentar um plano de dados que perpassa diversos domínios.

A entidade *NIM* substitui o *WIM* tradicional incorporando um gerenciamento de entidades da Internet, como os roteadores. Com essa estratégia, remove-se a responsabilidade da parte de rede do *MANO* Local delegando-a para a entidade *NIM*. Assim, a implantação de fatia de rede torna-se livre das restrições de atuação em diferentes domínios políticos e tecnológicos, uma vez que o *MANO* Local limita-se a lidar com recursos internos ao *data center*. Além disso, o *MANO* Local é incapaz de aplicar a especificação de parâmetros em roteadores da Internet. Essa estratégia não suplanta o papel do *MANO* tradicional, antes introduz uma nova abstração nos planos de gerenciamento, dividindo-os em Local e de Múltiplos-Domínio. O *framework NASOR* especifica e realiza o papel do *MANO* Multidomínio.

O conceito de fatia de rede explorado nesta tese traz forma de uma conectividade lógica estabelecida salto por salto considerando múltiplos roteadores da Internet, onde cada fatia de rede contem um plano de gerenciamento, controle e de dados separados.

3.2.2 *3GPP*

Na Figura 9 está mapeado as entidades do *3GPP* e seus papéis contrastados com o *framework NASOR*. O *3GPP* especifica entidades e papéis para gerenciar fatias de rede como o *Communication Service Management Function (CSMF)*, que é responsável por traduzir uma especificação de comunicação de alto nível para uma especificação de instância de fatia de rede.

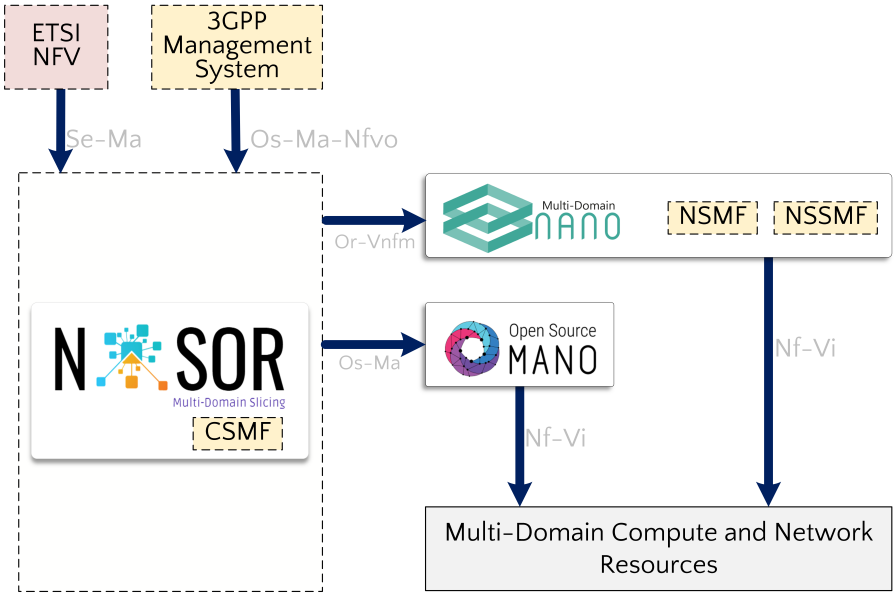

Figura 9 – Visão de Fatiamento de Rede estendido da *3GPP*.

Não há definição consensual de implementação dos papéis do *CSMF*, especialmente para fatias de rede implantadas sobre Redes de Transporte, onde a fatia de rede perpassa entre múltiplos *ASs*. Logo, *NASOR* implementa e realiza esse papel ao receber um descritor do serviço de comunicação por meio da interface *Os-Ma-Nfvo*. O descritor detalha tecnicamente os parâmetros da fatia de rede entre múltiplos domínios e provê uma implementação da fatia de rede sobre uma rede de transporte. Assim, pela perspectiva do *3GPP* o *NASOR* implementa e estende o papel do *CSMF* propondo um *CSMF* distribuído capaz de implantar fatias de rede entre múltiplos *ASs*.

O *NASOR* é a entidade que recebe a descrição do serviço de comunicação e interpreta-o para implantar fatias de rede entre múltiplos *ASs*. A especificação do

3GPP reconhece a entidade *Network Slice Management Function (NSMF)* para lidar com o ciclo de vida de uma fatia de rede. Além disso, o *3GPP* prevê a entidade *Network Slice Subnet Management Function (NSSMF)*, que desempenha papéis de gerenciamento de sub-instâncias de fatias de rede.

Essas duas entidades encontram correspondentes no *framework NASOR* por meio da interface *Or-Vnfm*. O *Network and Orchestration (NANO)* habilita a implantação e gerenciamento de fatias de rede entre múltiplos domínios. Além disso, o *NANO* habilita e realiza o conceito de fatiamento de rede recursivo. Fatiamento recursivo é uma fatia de rede dentro de outra fatia de rede maior. Todas as fatias de rede nesse contexto são gerenciadas independentemente. O *framework NASOR* provê um plano de dados, gerenciamento e controle independentes para cada instância de fatia de rede implantada entre múltiplos *ASs*.

3.3 Fatiamento de Rede Recursivo

Nesta tese é proposto um método de fatiamento de rede recursivo que consiste em subdividir uma fatia de rede em fatias de redes menores. Essa característica ainda está em estágios iniciais no estado da arte e é relevante de tratativa por habilitar a hierarquia de serviços, novas aplicações e modelos de negócio como B2B2C (HÉNO; BOUBENDIR; SIMONI, 2019; WIJETHILAKA; LIYANAGE, 2021). O método proposto objetiva realizar o fatiamento de rede recursivo inspirado no *framework* de orquestração da *ETSI*.

Implantar fatias de rede recursivamente significa dividir uma fatia de rede em fatias de redes filhas que possuem recursos limitados à capacidade dos seus pais (GENG et al., 2017; HÉNO; BOUBENDIR; SIMONI, 2019). Cada fatia de rede filha é gerenciada independentemente, mas possui dependência de recursos da fatia de rede pai, e pode subdividir-se gerando novas filhas de uma maneira recursiva.

Não se encontra no estado da arte abordagens compatíveis com o *framework* de orquestração da *ETSI* que objetivamente estabeleceram uma arquitetura que realiza o fatiamento de rede recursivo perpassando múltiplos *ASs*. Nesse intuito, a Figura 10 ilustra uma proposta conceitual do fatiamento de rede recursivo considerando as interfaces dos roteadores ao longo dos múltiplos *ASs* que uma fatia

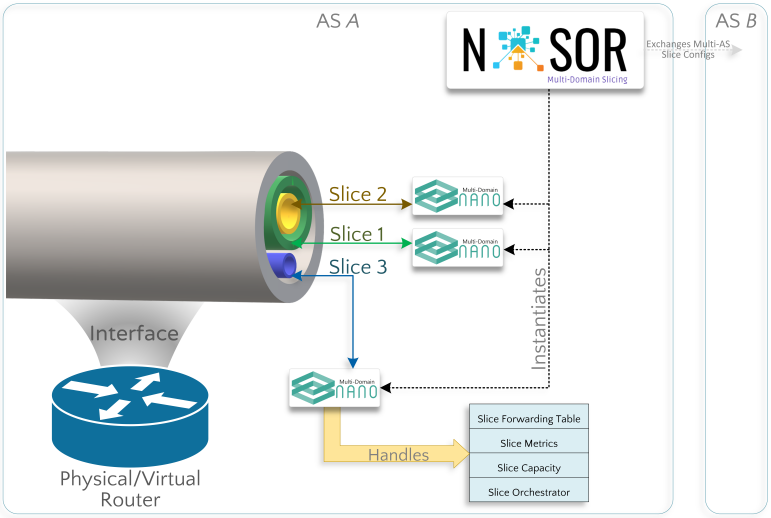

Figura 10 – Fatiamento de Rede Recursivo.

de rede perpassa. Na Figura 10 estão ilustradas três fatias de redes, onde a Fatia de Rede 2 está instanciada dentro da Fatia de Rede 1; a Fatia de Rede 3 é independente e não possui fatias de rede filhas.

O roteador pode ser físico ou virtual, para as duas modalidades, prevê-se o gerenciamento da fatia de rede que perpassa múltiplos *ASs* feito por orquestradores específicos de cada fatia de rede. Esses orquestradores específicos das fatias e sub-fatias de rede lidam com as métricas de rede, capacidade, interface com usuário e tabela de encaminhamento. Os orquestradores específicos das fatias de rede são instanciados por uma entidade global de cada *AS* por onde a fatia de rede transitar. Como esse método, é possível prover isolamento de orquestração, permitindo aos usuários gerenciarem suas fatias de rede livremente, podendo subdividi-las em fatias menores provendo planos de gerenciamento, dados e controle distintos para cada instância.

Cada fatia de rede possui uma tabela de encaminhamento virtual que habilita o caminho de dados entre múltiplos *ASs* pavimentando uma conectividade fim-a-fim para as entidades. Cada fatia de rede deve possuir também seu próprio orquestrador, que dá aos usuários uma interface de gerenciamento para lidar com o ciclo de vida de uma fatia de rede. Ao utilizar esse orquestrador, o usuário poderá criar novas sub-fatias que serão limitadas aos recursos da fatia pai. Uma

nova fatia de rede terá sua própria tabela de encaminhamento virtual, orquestrador de recursos e planos de gerenciamento, controle e dados independentes. Utilizando o orquestrador da sub-fatia de rede, um usuário poderá recursivamente criar novas sub-fatias. Esse processo para quando a soma dos recursos das sub-fatias de rede é igual a capacidade total da fatia de rede pai.

Após criar uma nova fatia de rede, seu orquestrador de recursos irá atualizar a tabela virtual de encaminhamento até que as novas rotas da fatia de rede permitam a alcançabilidade da entidade no *AS* final. A construção da tabela de encaminhamento virtual é feita a partir do domínio de origem da fatia de rede, que procede inserindo entradas correspondentes para cada fatia de rede contendo os identificadores dos segmentos ao longo dos roteadores e suas tabelas de encaminhamento virtual. Ao encontrar um novo *AS* a requisição de implantação de fatia de rede é repassada a entidade *NASOR* global daquele domínio que procederá com a implantação da fatia de rede partindo do roteador de *peering* até encontrar a entidade destino ou o próximo roteador *peering*. O *framework* arquitetural da *ETSI* não contempla o comportamento de fatiamento de rede recursivo.

3.4 *Network and Slice Orchestrator (NASOR)*

Para corroborar com a hipótese levantada no Capítulo 1 é proposto um mecanismo para estabelecimento e gerenciamento de fatias de rede multidomínios denominada *Network and Slice Orchestrator (NASOR)*. Nesse sentido, esta seção propõe descrever em alto nível os componentes que suportam suas funcionalidades, responsabilidades e interfaces. Inicialmente, é razoável mapear a estrutura do *NASOR* frente a descrição conceitual do *framework NFV* de gerenciamento e orquestração de *VNFs*, de forma a destacar os avanços que a presente proposta materializa.

O primeiro aspecto do mapeamento, conforme ilustrado na Figura 11, é a equiparação da proposta desta tese com a tríade estrutural do *framework NFV* que viabiliza o gerenciamento de funções de rede virtualizadas. Assim, o *NASOR* estabelece uma relação direta com os componentes, entidades e papéis do *framework*, quais sejam: o *MANO*, *NFVI* e *VNFM*. Inicialmente, o papel central do *NASOR* associa-se diretamente ao descrito para o *MANO*, pois suas finalidades constituem:

gerenciar a camada de instância do serviço, isto é, o plano de dados e controle de uma fatia de rede estabelecida entre domínios; gerenciar o substrato físico dos domínios e as fatias de recursos.

No que diz respeito ao gerenciamento das fatias de rede, sobretudo seu plano de controle, a arquitetura *NASOR* provê tal funcionalidade por meio da entidade *Network and Orchestration (NANO)*. O *NANO* é um orquestrador leve de gerenciamento de fatias de rede, instanciado pelo administrador do domínio para efetivar o estabelecimento de uma fatia de rede por meio de rotinas e configurações de caminhos lógicos entre entidades de roteamento da Internet. Além disso, o *NASOR* aprimora o *framework NFV* ao propor um comportamento de fatiamento recursivo conforme intenciona a hipótese.

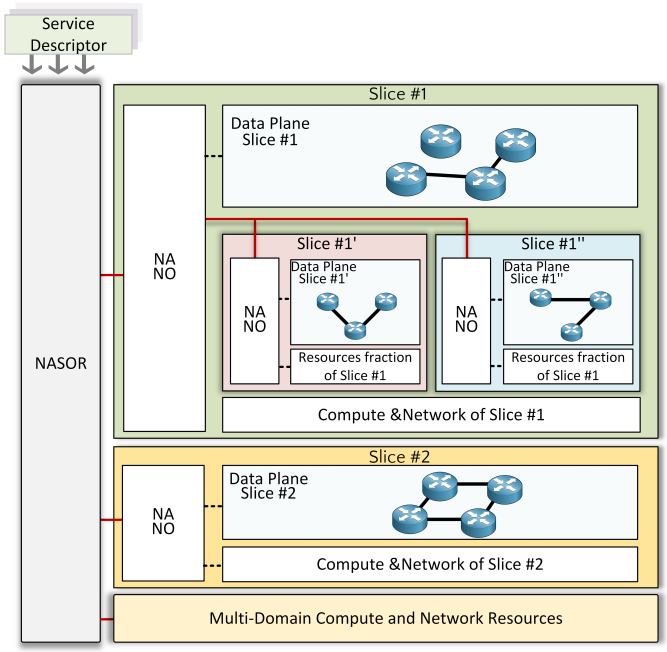

Figura 11 – Uma extensão recursiva do *framework NFV* da *ETSI*.

O segundo ponto, assim como no *framework* da *ETSI*, o *NASOR* possui uma interface bem definida com o substrato físico dos domínios. Em termos concretos, essa é a interface entre as rotinas de gerenciamento e a tecnologia de virtualização. No mapeamento proposto, a interface de virtualização dos domínios é a *NFVI* e seus componentes – roteadores da Internet, recursos computacionais, tecnologias

de virtualização, armazenamento e funções de rede. Portanto, é possível equiparar a camada *NFVI* quanto aos papéis que suas entidades desempenham com o *framework* estrutural. Assim, a interface *NFVI* do *NASOR* permite gerenciar recursos de rede e computação, e que esses sejam tratados de maneira uniforme com vistas a suportar a implantação da fatia de serviço — um termo genérico para fatiamento de rede.

O terceiro aspecto permite equiparar o papel do *NASOR* com a entidade *Virtualized Infraestructure Manager (VIM)*, previsto no *framework* arquitetural. Essa entidade controla e gerencia a interação entre o serviço virtualizado e o substrato computacional e de rede. Assim, o gerenciamento de recursos, alocação e visibilidade, inventário de serviços de rede: fatias de serviço e gerenciamento de capacidade, são tratados pela entidade *VIM* do *NASOR*. A presente proposta avança o estado-da-arte, sobretudo o *framework NFV*, na abordagem de implantação do serviço.

No modelo tradicional, os *VIMs* proveem entrega de funções de rede encadeadas no mesmo domínio tecnológico e com a conectividade padrão desse domínio, conforme o modelo de descrição do serviço. Por outro lado, o *NASOR* trata o *VIM* como um implantador tradicional de serviços de computação, isto é, aproveita-se o aspecto de multidomínios tecnológicos consolidados do *VIM* para demandas de computacionais. Por outro lado, para a conectividade do serviço faz-se uso do estabelecimento de fatias de rede multidomínios do *NASOR*, que se baseia no plano de dados da Internet.

O quarto aspecto de equiparação da proposta desta tese com o *framework NFV*, recai sobre o fato do *NASOR* possuir uma interface para criação e gerenciamento das fatias, similar à interface norte. Essa interface, concebida em formato de interface gráfica do usuário – *Graphical User Interface (GUI)*–, garante-lhes o gerenciamento do ciclo de vida dos seus serviços. Nesse ponto, os usuários submetem ao *NASOR* do seu domínio a requisição de criação de fatia do serviço: que envolvem características e detalhamento dos recursos computacionais, conectividade entre as entidades e o descritor do estabelecimento de fatias interdomínios. O método adotado para definir a sintaxe dos arquivos descritores considera o padrão do estado-da-arte: *TOSCA* ou *YANG*. Uma vez submetida a requisição de criação do serviço, o *NASOR* provê rotinas para alocar um orquestrador *NANO* para a

fatia de rede. Paralelamente, a parte de computação é tratada pelas rotinas que interagem com o *VIM* do domínio tecnológico.

Além dos aspectos do *NASOR* que equiparam aos papéis do *framework NFV*, são propostas funcionalidades adicionais para a entidade para lidar com os desafios científicos e tecnológicos observados no Capítulo 1. Depreende esses desafios: a construção arquitetural distribuída de um orquestrador, que dentre outros benefícios, mostra-se fundamental para a independência do plano de controle e de dados dos domínios. Com esse modelo arquitetural, a instanciação do serviço leva em conta as singularidades de segurança, política e tecnologia de cada domínio. Também, um componente de gerenciamento de ciclo de vida de *NANOs*, criados para prover uma fatia de rede.

Adicionalmente, o *NASOR* propõe que cada *NANO* crie e gerencie o ciclo de vida de micro-orquestradores, que são responsáveis pelo plano de controle do fatiamento recursivo de rede. A arquitetura do micro-orquestrestrador compreende um repositório de informações das fatias de rede: dono, tempo de criação, especificação da separação lógica de redes, capacidades de rede e computacionais, uma interface gráfica para o usuário. Cada domínio administrativo e tecnológico possui um *NASOR*, que por sua vez instancia, gerencia e mantêm *NANOs*, os quais criam e gerenciam micro-orquestradores de fatias ou subfatias de rede.

3.5 Detalhamento da Arquitetura e Interfaces do *NASOR*

A arquitetura *NASOR* possui interfaces e relacionamentos com inúmeras entidades tecnológicas para prover o fatiamento de rede fim-a-afim. Conforme se observa na Figura 12, a interface de gerenciamento do *NASOR* permite que o administrador do domínio instancie uma fatia de rede, conforme sua demanda, de maneira prática e intuitiva, isso se dá por meio de uma interface gráfica do usuário, *GUI*. A razão dessa interface, é garantir que a requisição de estabelecimento de uma fatia de rede possa ser recebida em cada domínio administrativo. Nesse sentido, para um domínio compor rol dos compatíveis com o fatiamento de recursos de rede ele deverá possuir a entidade *NASOR* bem como seus sub com-

ponentes e configurações para prover o serviço de fatiamento. Não é escopo desta tese apresentar um método de ingresso automático e seguro para novas entidades *NASOR*.

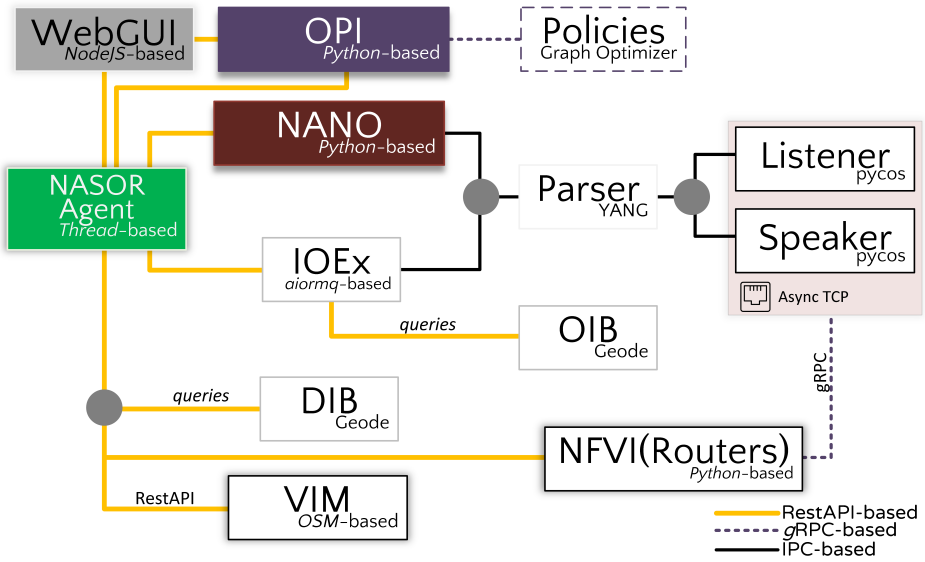

Figura 12 – Detalhamento das Entidades e Interfaces do *NASOR*.

Por se tratar de uma solução distribuída, o *NASOR* faz uso de repositórios persistente de dados. Inicialmente, o *Orchestrator Information Base (OIB)* armazena informações dos orquestradores de outros domínios: como endereço *IP* do seu *NASOR*, *ASN*, inventário de capacidades de rede e computacionais — limitadas à política de independência e segurança de cada domínio. Nesse ponto, cada *NASOR* é responsável por atualizar as informações de alcançabilidade e capacidades de rede e computação do seu domínio no repositório assíncrono, de forma que uma modificação de alguma entrada no *OIB* local é persistida nos demais domínios. Assim, quando o *NASOR* é requisitado para criação de uma fatia de rede multidomínios, o *NASOR* do domínio corrente, após prover as configurações compatíveis no seu domínio, encaminha a requisição para a *NASOR* do outro domínio. Esse ponto de troca de dados entre os *NASORs* é implementado por meio das rotinas *Listener* e *Speaker* que são padronizadas conforme um módulo *Parser*.

Dentre as informações relacionadas aos múltiplos domínios que são compatíveis com a abordagem de fatiamento, o repositório *OIB* é a entidade que armazena

informações dos roteadores e interfaces de borda dos domínios. Sempre que a requisição de criação de uma fatia de rede é do tipo multidomínios, o *NASOR* consulta seu *OIB* local para determinar o endereço *NASOR* do domínio vizinho. Na hipótese de haver mais de dois domínios em uma requisição de fatiamento de rede, haverá consequentemente domínios de trânsito. Portanto, a abordagem iterativa, que considera o caminho de menor custo da Internet: calculado pelo *BGP*, é adotada.

Nesse ponto, cada *NASOR* proverá a configuração da fatia de rede em seu domínio privativo até sua borda, onde ocorre a troca de tráfego. Nesse sentido, a proposta permite a independência política e tecnológica de cada domínio, de forma que a configuração do fatiamento de recursos é estabelecida até a interface borda dos domínios envolvidos no fatiamento. Assim, a presente proposta garante que a mesma configuração do nível de acordo do serviço, submetida no domínio origem, perpasse domínios intermediários até o ponto final do fatiamento de rede. Além disso, garante que os acordos de serviços implementados em um domínio sejam implantados e honrados nos demais.

A arquitetura proposta nesta tese, implementa um repositório adicional com informações privativas dos domínios. Previamente, discutiu-se o *OIB* cuja interação com o *NASOR* se dá para o estabelecimento de fatias de rede multidomínios. No entanto, o *DIB* se presta a armazenar informações privativas dos domínios onde o *NASOR* está inserido. Dentre essas informações, é possível conhecer os roteadores intermediários do domínio pelos quais a fatia de rede é iterativamente configurada. Além do inventário de roteadores, repositório *DIB* contêm o endereçamento que garante a alcançabilidade dos *VIMs* e as capacidades computacionais de cada domínio.

A sincronização das bases de orquestradores dos múltiplos *ASs* por onde a fatia de rede passará se atualiza de maneira assíncrona. O tecnologia *Apache GEODE*[1] é bem estabelecida na indústria oferecendo baixa latência, elasticidade, performance e consistência de replicas armazenadas em diferentes locais. Na construção e avaliação do método desta tese, os dois repositórios baseados no *Apache GEODE*, o *OIB* e o *DIB*, foram construídos.

[1] Repositório de dados distribuído baseado em chave-valor. Disponível em <https://geode.apache.org/>.

O Agente *NASOR* possui interface direta com a *API* do *VIM* que provê a instanciação dos serviços de computação sobre os múltiplos domínios tecnológicos. A interface pela qual ocorre essa interação é baseada em *RestAPI*. Como prova de conceito, esta tese implementa a interoperabilidade por meio dessa interface com o *MANO OSM*. A escolha do *OSM*, em detrimento de seus pares, se dá pela maturidade atual de suas versões e por sua compatibilidade com o *framework* de gerenciamento e orquestração da *ETSI*.

O *NASOR* possui interface com a camada de virtualização, sobretudo com o substrato de *hardware* de seu domínio. A entidade *NFVI* mantém o conjunto de roteadores do domínio, sua capacidade, configuração e localização. O objetivo de manter essa interface diretamente ligada ao *NASOR* é permitir o gerenciamento dos roteadores e do serviço de fatiamento de redes sobre o plano de dados da Internet. A entidade *NFVI* faz uso do modelo de comunicação assíncrona proposto nesta tese, onde comandos de consulta e gerenciamento —baseado em *gRPC*–, sobre os roteadores utilizam a interface *Listener* e *Speaker* implementadas para tratar mensagens que que chegam aperiodicamente.

Optou-se por utilizar o *gRPC* por ele ser de alta performance, amplamente utilizado pela indústria e compatível com várias linguagens de programação, o que tornou-o mais adequado para compor a prova do conceito do *NASOR*.

Adicionalmente, no detalhamento das entidades e interfaces do *NASOR* é previsto uma comunicação direta com o a entidade *NANO*. Esse relacionamento, conforme ilustrado na Figura 12, é baseado na instanciação da funcionalidade no formato de *thread* com execução assíncrona. Dessa forma, o serviço torna-se independente para prover as funcionalidades de fatiamento que se espera da solução proposta. Portanto, essa interface é responsável por efetuar as rotinas que implementam o plano de dados e controle interdomínios conforme o modelo fatiamento de rede e uma interface de gerenciamento delas para os usuários ou terceiros.

Na arquitetura do *NASOR* está previsto o relacionamento e com a entidade *NANO* que provê o gerenciamento e manutenção de um repositório de micro-orquestradores. Cada micro-orquestrador possui rotinas para gerenciamento de uma fatia de rede e oferece interface gráfica para os administradores. Assim, o administrador do domínio gerencia e mantém as informações dos *NANOs*, e cada *NANO* gerencia e mantém micro-orquestradores que se prestam a executar as

rotinas de configuração de cada fatia de rede implementadas no domínio. Essa abstração permite a recursividade de estabelecimento de fatias de rede.

O primeiro nível de fatiamento de rede é entregue pelo *NANO*. O segundo, ao instanciar uma fatia de rede e delegar um micro-orquestrador para gerenciá-la, o micro-orquestrador possui autonomia para redimensionar a fatia de rede, criar um micro-orquestrador adicional para cuidar da outra fatia de rede oriunda do redimensionamento original. A exclusão de uma fatia de rede no nível *NANO* desencadeia a exclusão de subfatias de rede que eventualmente foram criadas.

3.6 *Network and Orchestration (NANO)*: Interfaces e Componentes

A arquitetura *NANO* possui duas abstrações de interfaces, uma baseada no plano de configuração da fatia de rede, que é diretamente associada às rotinas do *NASOR*. Essa interface lida com o primeiro nível da fatia de rede interdomínios. Por outro lado, a segunda interface é baseada no plano de gerenciamento da fatia de rede, projetada para o administrador do domínio. Ambas as interfaces convergem para uma entidade central, de forma que, cada *NANO* possui um gerenciador de *containers*, sobre os quais estão implementadas a funcionalidade micro-orquestradores.

Também, o administrador do domínio possui acesso ao gerenciamento dos micro-orquestradores. A entidade *Micro-Orchestrator Manager (MOM)* é um submódulo do *NANO*; a construção da presente proposta considera que eles se comunicam por meio de interfaces com vistas ao desacoplamento. A funcionalidade básica do *MOM* é gerenciar micro-orquestradores que executam em formato de *containers*. Os micro-orquestradores são responsáveis por prover as rotinas de estabelecimento da fatia de rede, isto é, o plano de dados e controle e oferecer de uma interface de gerenciamento para terceiros. Além disso, o *NANO* desempenha o papel do *NIM* para o estabelecimento de fatias de rede por meio da configuração de parâmetros nos roteadores.

Nesse sentido, o gerenciamento e implementação do fatiamento de rede é baseado na interação do *NANO* com o *MOM*. O lado esquerdo da Figura 13 apresenta o

detalhamento da operação do *NANO* por meio de suas duas interfaces. A primeira interface baseada em *RestAPI*, por onde o *NASOR* estabelece a chamada remota de mecanismos onde as rotinas de estabelecimento de fatias de rede são invocadas, coordenadas e executadas. Essa interface é baseada em invocação remota de procedimentos com vistas a compatibilidade, fraco acoplamento e escalabilidade da proposta. A segunda interface, provê um plano de gerenciamento gráfico para o administrador do domínio, que permite gerenciar o conjunto de micro-orquestradores relacionados a uma fatia-de-rede, modificar parâmetros e consultar o estado dos recursos.

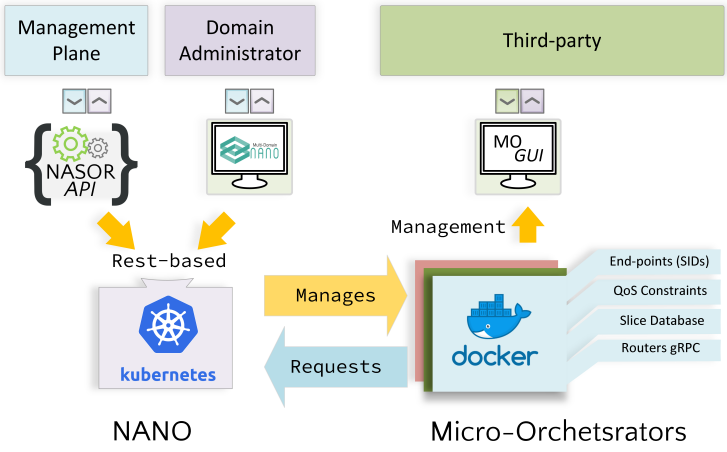

Figura 13 – Detalhamento das entidades e interfaces do *NANO*.

Os micro-orquestradores implementados em formatos de *containers* são gerenciados pelo *MOM*, que provê dois níveis de gerenciamento. A funcionalidade de gerenciar fatias de rede interdomínios é garantida pelas rotinas implementadas nos micro-orquestradores. Um *Micro-Orchestrator (MO)* é responsável por orquestrar uma fatia de rede, atribuí-la a um usuário, e persistir as informações relacionadas a essa fatia de rede que são: End-point *Segment Identifiers (SIDs)*, as restrições de *QoS* da fatia de rede, o banco de dados da fatia de rede — informações de persistência e manutenção dos dados da fatia de rede e seu dono — e a lista de roteadores que são gerenciados pelo *MO*. Cada *MO* provê uma interface de gerenciamento da fatia de rede, assim o usuário/dono pode modificar parâmetros da fatia de rede e conhecer estatísticas com vistas ao monitoramento.

Para suportar o cenário recursivo, uma interface de volta entre a instância do *MO* e o *MOM* garante que um *MO* instancie outro *MO*. Limitações são impostas a um *MO* para que o processo de criação de fatias recursivas considere as capacidades disponíveis da fatia de rede a qual ele faz parte. As característica e informações de uma fatia de rede administrada por um *MO* são persistidas, as informações são: capacidade original da fatia de rede, o dono da fatia de rede e os recursos de rede e computacionais alocados a essa fatia de rede. A interface entre o *MO* e o *MOM* se limita a permitir a criação de um novo *MO* condicionada as capacidades do *MO* criador, não é permitida a exclusão ou modificação de um *MO* associado a outra fatia de rede, apenas ao *NANO* do administrador é permitida exclusão de *Micro-Orchestrators (MOs)*.

A arquitetura da presente solução prevê uma interface para o usuário/terceiros gerenciarem suas fatias. Conforme ilustrado no lado direito da Figura 13 um *MO* oferece uma interface gráfica que permite o dono da fatia de rede consultar e gerenciar parâmetros do serviço contratado. A modificação nesse nível é condicionada aos recursos previamente reservados no processo inicial de fatiamento de rede, provido pelo *NANO*. Ocorre que, essa interface foi concebida para oferecer ao dono de uma fatia de rede, possibilidade de administrar o serviço contratado.

A dinâmica dessa interface é garantir que: uma vez que o serviço de fatia de rede foi contratado e implantado pelo *NANO* o usuário possui acesso ao nível gerencial a fatia de rede via *MO*, de sorte que ele pode redimensionar ou modificar os parâmetros do serviço até o limite da fatia original contratada. Além disso, a concepção dessa interface permite que eventualmente o dono de uma fatia de rede instanciada pelo *NANO* monetize sua fatia por meio do redimensionamento e criação de outros micro-orquestradores. Um micro-orquestrador deve ser exclusivo e não pode ser global porque ele é a entidade que gerencia o serviço de um usuário específico.

Propostas do estado-da-arte emergiram como provedoras candidatas para fatiamento fim-a-fim multidomínio. Conforme observado, houve empenhos bem estabelecidos no plano de controle, ao passo que o plano de dados se valeu de técnicas conhecidas e limitadas para domínios singulares (*SDN*, *VLAN* e outras). Nesse sentido, a solução *NASOR* alinha-se a proposta da *ETSI*, conforme ilustrado na Figura 11, uma vez que interage diretamente com *MANOs* construídos à luz da

arquitetura referencial da *ETSI* (*OSM, SONATA* e outros).

Nesse sentido, o *NASOR* recebe a requisição de implantação de um serviço de rede baseado em fatiamento de recursos, que contém os arquivos manifestos de descrição de serviço (*YANG*) que definem a fatia de serviço: *Virtual Network Function Descriptor (VNFD)*, a composição do serviço *Network Service Descriptor (NSD)*, para encadeamento dentro do *datacenter* e o *Network Slice Template Descriptor (NSTD)*.

De forma a subsidiar a compreensão da interação dos componentes do *NASOR*, é ilustrado na Figura 14 o processo sequencial do estabelecimento de fatias de rede inter domínios. Inicialmente, o processo sequencial diz respeito ao estabelecimento do fatiamento de rede em um único domínio, na modalidade plana, isto é, não há implementação de fatias recursivas.

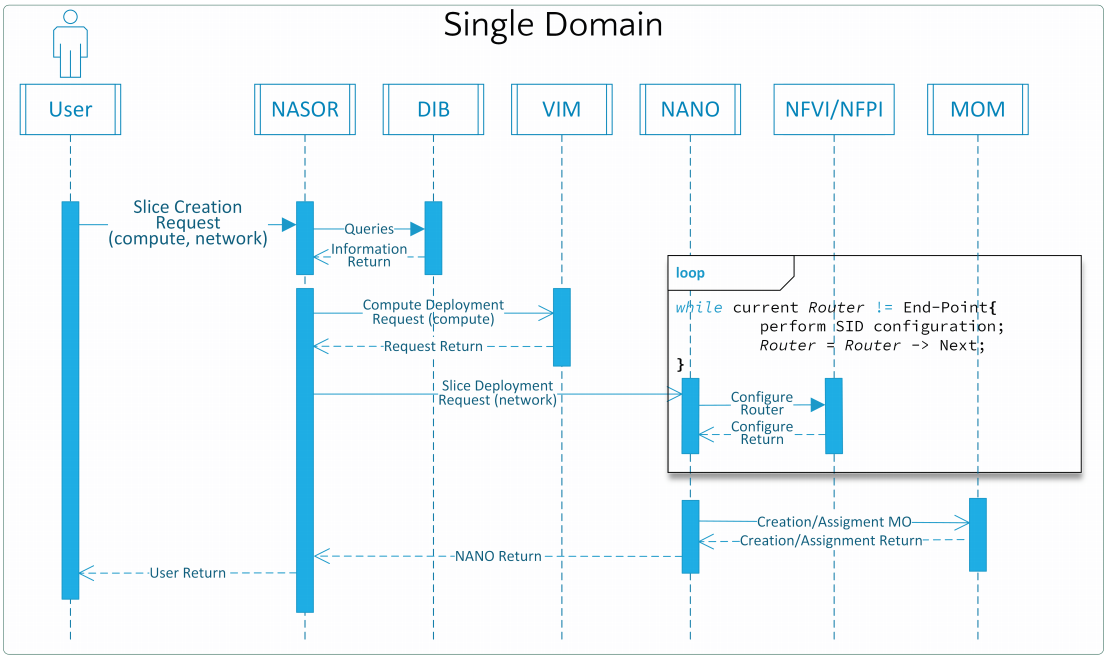

Figura 14 – Sequência: estabelecimento de uma fatia de rede em um domínio singular.

O processo consiste nos seguintes passos:

1. O Usuário encaminha o arquivo manifesto para o *NASOR* do seu domínio. O arquivo possui as especificidades de computação e de rede;

2. O *NASOR* recebe a requisição, consulta o repositório *DIB* para recuperar informações do *NANO* e *VIM* local;

3. O *NASOR* ao dividir descrição de rede e computação no arquivo manifesto, encaminha a parte de computação para o *VIM* de seu domínio;

4. Assincronamente, a descrição de rede é encaminhada para o *NANO* criado para lidar com a fatia de rede;

5. Em um laço, o *NANO* provê a configuração da fatia de rede, que envolve configuração dos parâmetros do roteamento por segmento. O laço termina quando todos os roteadores do caminho de menor custo até o ponto final foi percorrido;

6. Após configurada os parâmetros da fatia de rede, nos equipamentos virtuais ou físicos, o *NANO* cria e atribui um *MO* para gerenciar a fatia de rede, e retorna a especificação do *MO* ao usuário requisitante.

Adicionalmente, a Figura 15 ilustra a sequência lógica no estabelecimento de fatia de rede em um cenário de múltiplos domínios. Cabe nesse ponto descrever duas abordagens de múltiplos domínios: a primeira refere-se a uma fatia de rede que tem origem em um domínio A e o ponto final da fatia de rede está em um domínio B, em que este está em uma distância máxima de um salto, assim intercomunicação entre domínios é direta via troca de tráfego. A segunda, refere-se a um estabelecimento de fatia de rede em que o domínio B não é vizinho do domínio de origem A.

A presente tese considera que os domínios intermediários — concebidos como domínios de trânsito–, proporcionarão as configurações adequadas em seus elementos para o estabelecimento de fatias de redes entre os dois domínios para o qual ele se presta a ser trânsito.

Esta sequência de passos, relacionada com o diagrama ilustrado na Figura 15, sistematiza o processo de estabelecimento de fatia de rede entre múltiplos domínios:

1. Um Usuário requisita o *NASOR* de seu domínio o estabelecimento de uma fatia de rede. Na solicitação são encaminhadas as especificações de computação e rede em formato *YANG*;

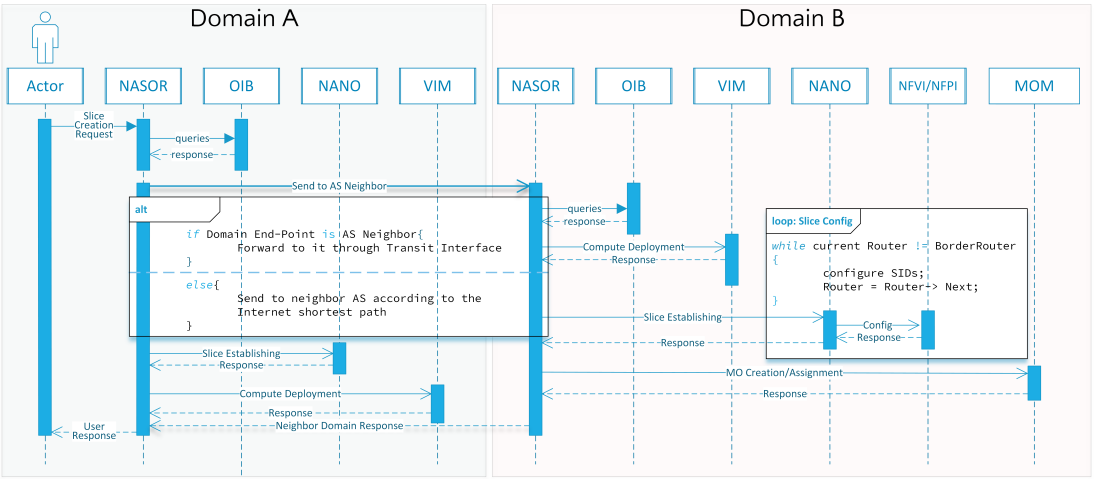

Figura 15 – Sequência: estabelecimento de uma fatia de rede em domínios múltiplos.

2. O *NASOR*, após tratar a admissibilidade da requisição e ao identificar o envolvimento de múltiplos domínios consulta o repositório *OIB* para recuperar o endereço do *NASOR* vizinho/ponto final;

3. Posteriormente, ele encaminha assincronamente a requisição estabelecimento de fatia de rede para o domínio B, o qual está descrito no *ASN* no arquivo manifesto inicial. Para o encaminhamento são feitas duas ponderações:

   ❏ Se o ponto final é o *AS* vizinho, então encaminhe para ele por meio da interface de trânsito;

   ❏ Se o ponto final não é o *AS* vizinho, encaminhe para o *NASOR* do próximo *AS* considerando o caminho de menor custo;

4. A requisição de estabelecimento de fatia de rede é encaminhada para o *NANO* local que procederá com a criação e configuração de todos os elementos virtuais e físicos;

5. Paralelamente, a parte de computação da requisição inicial é recebida pelo *VIM* dos domínios envolvidos, que procederão com a implantação conforme descrição no arquivo manifesto;

6. O *NASOR* do *AS* vizinho — o domínio B –, por ter recebido uma requisição de implantação da fatia de rede desencadeia a configuração do serviço em seu domínio ao encaminhar a parte de computação para seu *VIM* local;

7. Ao mesmo tempo, a parte de rede é encaminhada para o *NANO* criado para lidar com a fatia de rede requisitada;

8. Iterativamente o *NANO* do domínio B configura nos elementos físicos e virtuais os parâmetros da fatia de rede;

9. Ao fim da configuração, um *MO* será criado e associado à fatia de rede criada.

10. O endereço de gerência do *MO* será devolvido ao usuário por meio de mensagens de retorno entre *NASORs*.

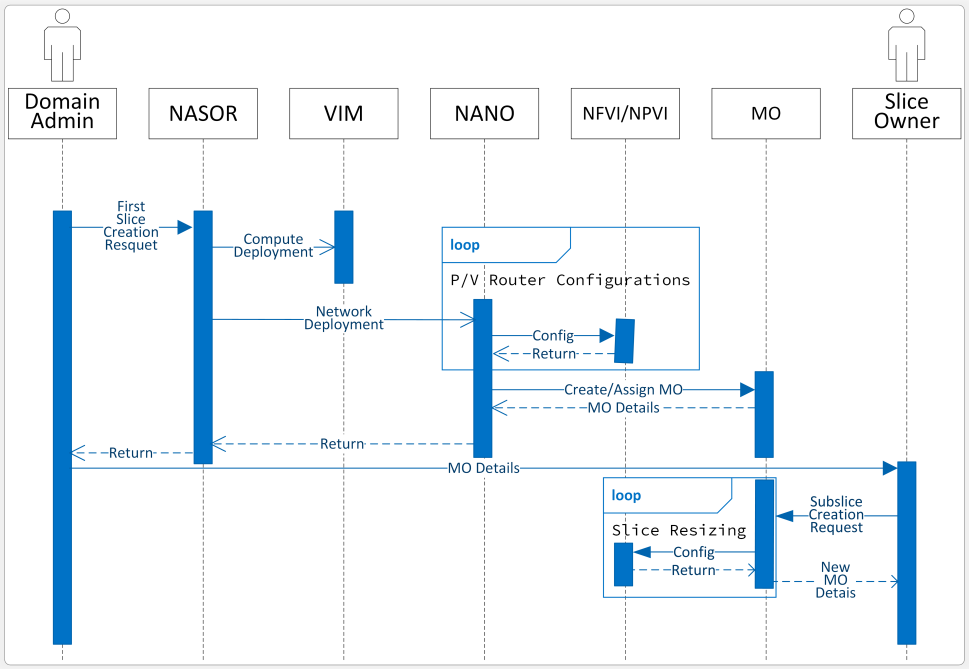

Figura 16 – Sequência: estabelecimento de uma fatia de rede recursiva em um domínio singular.

O processo de estabelecimento de fatia de rede recursiva é ilustrado na Figura 16. Esse processo é referente a um determinado Usuário que solicita a criação de

uma fatia de rede, e posteriormente ele redimensiona o serviço, isto é, cria uma subfatia conforme sua especificidade.

Os passos que compõe o fatiamento recursivo são estes:

1. O administrador do domínio encaminha para seu *NASOR* local a requisição de criação de uma fatia de rede;

2. O *NASOR* a separa do aspecto de rede e computacional, e encaminha, após consulta no repositório do domínio (*DIB*), a requisição de implantação para o *VIM* local;

3. Simultaneamente, a nova instância do *NANO*, criada para lidar com a requisição de fatiamento, provê iterativamente a configuração da fatia de rede nos roteadores do domínio;

4. Ao término da configuração, o *NANO* cria e atribui um *MO* para gerenciar a nova fatia de rede e retorna seu detalhamento para o administrador do domínio;

5. O administrador do domínio encaminha os detalhes do *MO* ao dono da fatia de rede — aquele que contratou o serviço;

6. O doto da fatia de rede, por meio do MO, é capaz de modificar ou criar uma nova fatia de rede dentro dos limites da fatia original;

7. A criação recursiva de fatias de rede implica na criação de um novo *MO* que proverá o gerenciamento dela, assim o novo administrador da fatia, até o limite dos recursos que lhe fora garantidos, pode criar novas fatias de rede e novos *MOs*;

Adicionalmente, estão presentes no Apêndice A uma série de ilustrações que representam as interações sequenciais das entidades da presente proposta para lidar com o ciclo de vida de fatias de rede. As ilustrações exemplificam tanto o cenário de domínio singular quanto multidomínios, esse leva em conta o plano de dados da Internet.

3.6.1 A Interface de Política Aberta

A Interface de Política Aberta, *Open Policy Interface* (*OPI*), é uma funcionalidade do *NASOR* que permite que mediante um arquivo de configuração o *NANO* seja instruído a estabelecer um caminho multidomínios conforme parâmetros definidos pelos usuários. Um exemplo de um arquivo de configuração *YANG* para estabelecimento de fatias de rede entre múltiplos domínios está disponível no Apêndice G. É possível observar nas linhas 9 . . . 15 a extensão que a presente tese propõe na sintaxe e mecanismo padrão de fatiamento de rede padronizado pelo *MANO* da *ETSI*. Nessas linhas são descritos em formato de lista os *ASs* que estão envolvidos no fatiamento de rede, ao longo do caminho entre a origem e o destino. Os *ASs* trânsito podem ser omitidos pois a política de caminhos baseado no plano de dados dos algoritmos de roteamento é capaz de estabelecer a configuração fim-a-fim neles.

Adicionalmente, pode ser observado as linhas 13 . . . 15 e especificação do mecanismo de escolha de caminhos que o *NANO* deverá considerar no processo de configuração das entidades de roteamento de cada domínio. Especificamente, a referência daquele arquivo de configuração propõe que o *NANO* utilize uma abordagem de configuração nos elementos do caminho que satisfaz a condição performance de rede. A entrada *"name: network_performance-aware"* faz referência à política de definição de caminhos disponível no catálogo do *NASOR*. Essa política instrui o *NANO* a considerar e testar a qualidade instantânea dos enlaces para configurar os parâmetros da fatia de rede. No mesmo sentido, a entrada convencional *"name: default"* instrui ao *NANO* que a política de definição de caminhos a ser utilizada é a baseada no plano de dados dos algoritmos de roteamento.

A implementação dessa interface é baseada na estrutura de dados Grafo que o *NASOR* abstrai da topologia de cada domínio o qual está inserido. Assim, essa estrutura de dados considera os nós como roteadores e as arestas como *links* e permite que sejam atribuídas propriedades a cada um. Além disso, permite que sejam implementados mecanismos de busca de caminhos, observância e atribuição de pesos nas arestas, definição de características dos nós e dos *links*. Nesse sentido, o administrador do domínio é responsável por gerenciar o catálogo de políticas de definição de caminhos para as fatias de rede *OPI*.

O serviço de topologia é baseado em *Link Layer Discovery Protocol (LLDP)* que executa em cada *AS* que o *NASOR* influencia. Esse serviço executa como *daemon* nos roteadores e responde a requisições externas devolvendo a topologia momentânea do domínio.

Os requisitos para adicionar um novo mecanismo de escolhas de caminhos é implementar essa interface no *NASOR*. O esquema ilustrado na Figura 17 propõe exemplificar extensão e adição do mecanismo de definição de caminhos do *NASOR*. Primeiro o *NASOR* recebe do usuário a solicitação de estabelecimento de fatias de rede, nessa requisição está o arquivo *NSTD* e a política a ser utilizada. Quando o *NASOR* constata que a política não é a padrão, conforme lado esquerdo da ilustração, ele requisita a topologia do seu domínio, e encaminha-a para a implementação do *OPI*.

O mecanismo de escolhas de caminhos alternativos recebe um objeto com a estrutura de dados Grafo do seu domínio, que semanticamente representa a topologia, e opera conforme sua implementação de escolha do caminho. Assim, o agente *NANO* ao receber a estrutura de dados Grafo processada pelo mecanismo é capaz de configurar os parâmetros da fatia de rede por meio de chamada remota de procedimentos.

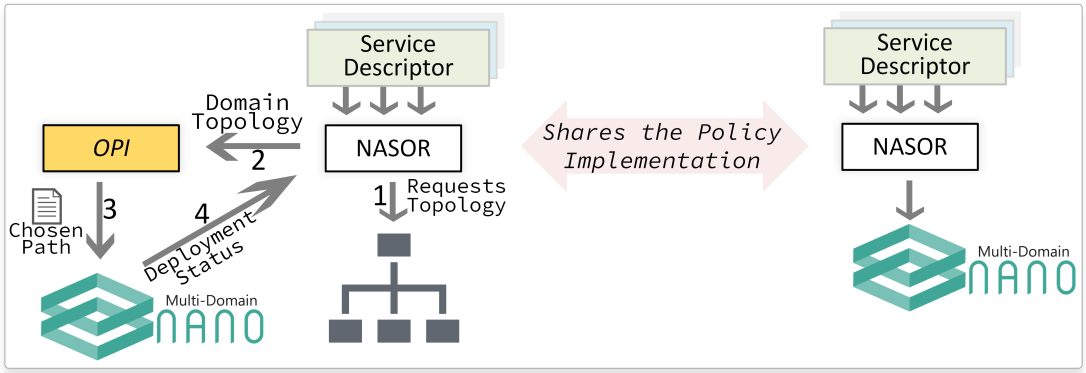

Figura 17 – Esquema da Interface de Política Aberta.

A parte direita da Figura 17 diz respeito ao mecanismo padrão de estabelecimento de fatias de rede, aquele que considera o plano de dados construído pelo algoritmos de roteamento como o caminho a ser configurado pelo agente *NANO*.

3.7 O Plano de Dados Multidomínios

Previamente, foi detalhado o processo de estabelecimento de fatias de rede em cenários multidomínios com foco nos mecanismos de controle e nas interações das entidades que provêm esse serviço. Nesse ponto, é importante parametrizar as referências "domínio" e *"AS"* como intercambiáveis. Adicionalmente, fica homogenizada a referência "fatia de rede" e "caminho lógico". Portanto, esta seção detalha o mecanismo de estabelecimento de um plano de dados entre domínios (conectividade lógica).

Para isso, é importante descrever o mecanismo de roteamento por segmentos e como essa tecnologia pode, valendo-se de orquestradores distribuídos, elevar o grau de granularidade e customização de parâmetros de rede e traduzir essa customização em fatiamento de rede. A tecnologia de roteamento por segmentos é uma premissa tecnológica adotada no desenvolvimento da solução que esta tese apresenta. Portanto, os roteadores físicos e virtualizados localizados nos domínios devem ser compatíveis com a tecnologia. A proposta desta tese avança o estado da arte no aspecto de gerenciamento, por meio de interfaces de controle baseadas em *gRPC*, o *NASOR* é capaz de instalar novos parâmetros no plano de encaminhamento que materializa a criação de uma conectividade lógica fim-a-fim.

Pela ótica de roteamento, os domínios administrativos onde se situam as entidades do plano de dados, são referenciados como *AS* e organizam entre si uma conectividade interdomínios por meio dos roteadores de borda. O ponto de troca de dados entre os *ASs* é baseado em *peering*, de forma que um *AS* faz fronteira com muitos *ASs* e as sub-redes de cada *AS* precisam ser conhecidas e divulgas pelos roteadores de borda. Nesse ponto, os algoritmos de roteamento, especificamente os *inter-AS*, são responsáveis por atuar no *backbone* que interliga os *ASs*. Na área de roteamento da Internet, o *backbone*, além da troca de tráfego *Internet Exchange (IX)*, ocorrem anúncios de rotas internas dos *ASs*. Esses anúncios são subsídios para o funcionamento dos algoritmos de roteamento, os quais provêm uma lista com os melhores caminhos que são baseados na métrica de menor custo, os quais consideram a quantidade de saltos em um caminho *BGP*.

O algoritmo de roteamento *inter-AS BGP* estabelece um mecanismo de controle que atua sobre duas estruturas dos roteadores. A troca de rotas que ocorrem

sistematicamente, conforme implementação do *BGP*, permitem que os roteadores que compõe a sessão *BGP* construam uma estrutura denominada *Routing Information Base (RIB)*. A *RIB* contém a informação de rotas dos inúmeros roteadores que compõe uma sessão *BGP*. Adicionalmente, plano de controle do algoritmo de roteamento *BGP* constrói outra estrutura denominada *Forwarding Information Base (FIB)*, que armazena informação atual e de melhor custo para o encaminhamento de um pacote para uma determinada rota. Nesse sentido, estrutura *FIB* dos roteadores possuem um conjunto de instruções para cada pacote, de forma que este é encaminhado para uma interface em detrimento de outra conforme cálculo de melhor rota computada pelo algoritmo de roteamento.

Para lidar com os desafios de escalabilidade da Internet, a saber: a numerosidade de sub-redes que eram propagadas e que implicaram o crescimento exponencial da tabela de roteamento, foi necessário que o *BGP* implementasse o modelo *Classless Inter-Domain Routing (CIDR)* como forma de mitigar esse desafio. Genericamente, o modelo *CIDR* permitiu cadenciar o crescimento da tabela com agregação de rodas pelo característica de prefixo das sub-redes. Paralelo ao desafio de escalabilidade, o modelo de roteamento para o qual protocolo *BGP* originalmente se prestou (*IPv4*) experimentou o desafio de escassez de *IPs*. Por haver uma quantidade limitada para endereçamento e identificação das entidades na Internet (2^{32}), o *IPv6* foi proposto. Assim, os algoritmos de roteamento, em especial o *BGP* foi aprimorado para lidar com o novo modelo de endereçamento. A solução descrita nesta tese considera o plano de dados baseado em *IPv6* e o algoritmo de roteamento *BGP* compatível com essa versão.

Adicionalmente, a solução descrita nesta tese, é baseada em roteadores virtuais e vale-se do algoritmo de roteamento *BGP* para o cálculo de rotas entre os *ASs*. Nesse sentido, conforme intenciona a hipótese, a solução de fatiamento de rede baseada em *BGP* permite prevalecer um caminho lógico entre domínios. O roteamento por segmentos permite que sejam definidas na origem um conjunto de entidades ou interfaces pelo qual um segmento deverá percorrer. A arquitetura *NASOR*, por conhecer o plano de encaminhamento dos roteadores e por possuir uma interface de controle com estes, pode instalar e aplicar políticas para lidar com os os caminhos lógicos (fatias de rede) que são orientadas pelos *SIDs*. As interações entre a entidade *NFVI* do *NASOR* com os roteadores por meio da in-

terface de gerenciamento permite instruir o roteador de ingresso, ponto de entrada do pacote na sua fatia de rede, a adicionar uma lista de entidades pelo qual o pacote deverá percorrer e adicionalmente os comportamentos aplicáveis em cada localização do pacote.

A Figura 18 ilustra o posicionamento e a interação das entidades de controle e de encaminhamento no processo de criação de uma fatia de rede. O plano de dados dos roteadores abarca um mecanismo de instruções de encaminhamento adicional, que são baseadas na tabela *SID* do roteamento por segmentos. Conforme ilustrado, o *NANO* de cada domínio via chamada remota de procedimentos, implementada sobre o *gRPC*, instala e altera a tabela de *SIDs* nos roteadores de cada domínio.

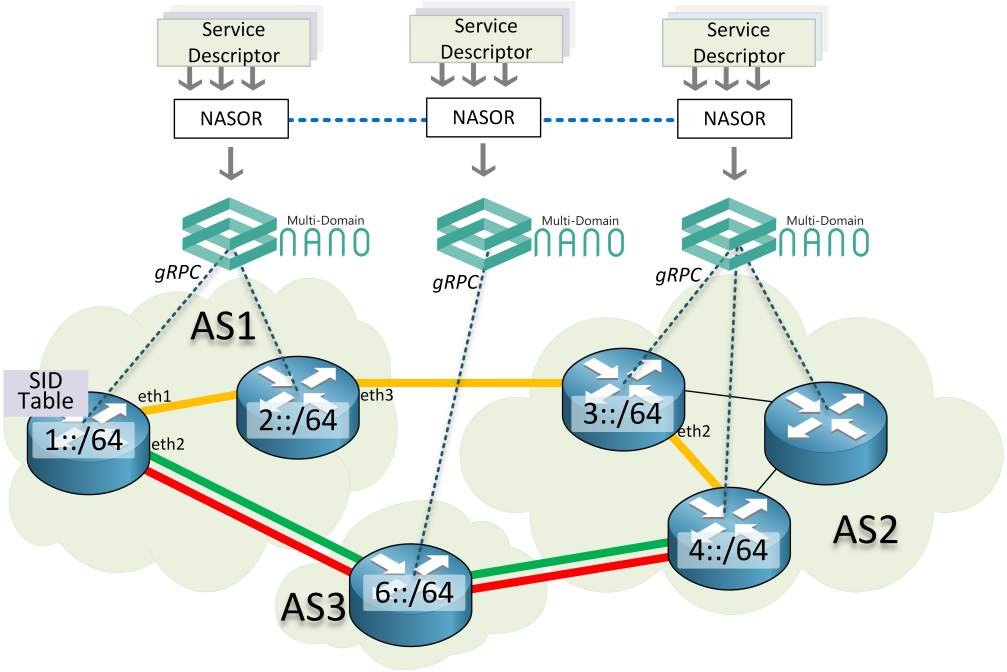

Figura 18 – Detalhamento do Plano de Dados proposto pelo *NASOR*.

Esse controle na tabela *SID* permite instalar e alterar uma fatia de rede, de forma que o *AS* que originou a alteração deve, via *NASOR*, propagar a alteração bem como seus parâmetros para o *NASOR* de cada domínio. Esse mecanismo de controle proposto pelo *NASOR* provê a criação e manutenção de um caminho lógico *inter-AS* para entidades eventualmente em domínios diferentes. Nesse aspecto, sustenta-se que uma fatia de rede é o plano dados e controle de cada caminho lógico,

implementado com roteamento por segmentos, sobre os roteadores da Internet. Adicionalmente, a Figura 18 ilustra um cenário multidomínios com três fatias de rede instaladas sobre três *ASs* distintos. O fatiamento de rede para caminhos lógicos similares, utiliza a abordagem de coloração para diferenciar os caminhos. Portanto, um mesmo caminho lógico pode, mediante o parâmetro de coloração receber uma política diferente de seus pares.

Por ter acesso direto ao plano de gerenciamento dos roteadores, a solução *NASOR* possui acesso a rotas alternativas a de menor custo. Dessa forma, é possível estabelecer uma fatia de rede que utiliza um caminho alternativo ao de menor salto entre os *ASs*. A consulta na *RIB* dos roteadores, que se dá por chamada remota de procedimentos, traz ao *NASOR* interfaces alternativas de encaminhamento que poderão ser consideradas para implantação de caminhos lógicos, implementados por meio da tabela de *SIDs*.

O Algoritmo 1 descreve os passos iterativos que são adotados desde o roteador origem até o roteador destino no âmbito do estabelecimento de uma fatia de rede. A saída do algoritmo é: para todo roteador pertencente ao caminho mais curto entre duas entidades A e B, eventualmente em domínios diferentes, calculado pelo *BGP*, proveja a instalação de entradas na tabela *SID* dos roteadores. A instrução inicial do algoritmo é: recupere do repositório *DIB* e do *OIB* do domínio corrente, originário da requisição de criação de uma fatia de rede, as especificações dos roteadores e dos *NASORs* dos *ASs* envolvidos.

Algoritmo 1: Multi-Domain Network Slicing Establishing (MD-NSE).

Result: $\forall$ $R \in$ {BGP Shortest-Path entre os Roteadores} $\rightarrow$ Estabeleça um Plano de Dados Multidomínios
Input: *NSTD* em formato *YANG*
1 Separe a parte de **Rede** e **Computação** do arquivo manifesto;
2 Recupere do *OIB* e *DIB* as especificações dos Roteadores, *VIMs* e *NASORs*;
3 Encaminhe a parte de computação para o *VIM* local;
4 Encaminhe o arquivo manifesto para o *NASOR* do *AS* imediato;
5 **while** *R is not end-point target* **do**
6 **if** *R* == *Roteador de Borda* **then**
7 Configure os *SIDs* e as Políticas na Interface *peering*;
8 *break*;
9 **end**
10 Configure a tabela *SID* e Políticas conforme encaminhamento *BGP*;
11 $R = R \rightarrow Neighbor$;
12 **end**

Então, de forma iterativa e até que seja encontrado último roteador, isto é,

a última milha para o ponto final, o algoritmo prevê: configure as tabelas *SID* baseado no caminho mais curto nos roteadores, configure as políticas definidas no modelo do serviço. Verifique, se o roteador atual é o de borda: configure o *SID* e as políticas na interface de troca de tráfego. Se o roteador atual não é o de borda: caminhe pelo plano de dados do roteador, calculado pelo *BGP*, de forma que a configuração prossiga no próximo roteador de menor salto. Ao fim da visita em todos os roteadores de um domínio, até encontrar o roteador de borda, a fatia de rede terá sido configurada conforme especificação do serviço.

3.8 Considerações Finais

Esse capítulo apresentou o detalhamento técnico da solução concebida e implementada como um prova de conceito que corrobora com a hipótese suscitada. Para tanto, foi proposto um mecanismo de fatiamento de rede multidomínios sobre o plano de dados da Internet que provê aos usuários recursos de rede em formato de conectividade lógica; e para os provedores de aplicações um novo ecossistema de implantação de aplicações. A hipótese de utilizar o plano de dados da Internet sugere tornar a solução proposta compatível com o cenário operacional das redes legadas.

Além disso, o mecanismo de controle distribuído torna possível que cada domínio mantenha resguardadas suas políticas e níveis de segurança. O mecanismo de criação de subfatias de rede, no formato recursivo, garante que terceiros gerencie com alto grau de granularidade a fatia de rede adquirida. A hipótese de fatiamento de recursos fim-a-fim sustenta-se no fato da solução proposta oferecer conectividade lógica *inter-AS*, o que garante que um acordo ou política de qualidade de serviço seja honrada entre os domínios envolvidos no fatiamento de rede. Este capítulo apresentou um sequenciamento lógico adotado para a construção e validação dos elementos desta tese.

Capítulo 4

Experimentos e Análise dos Resultados

Este capítulo apresenta experimentos que objetivam avaliar a aplicabilidade do *NASOR* para lidar com o fatiamento de rede entre múltiplos *ASs* em contraste com seus pares do estado da arte. Os experimentos versam sobre a aplicabilidade, performance e abrangência da solução para lidar com os requisitos que se impõe sobre a rede na perspectiva dos usuário e do *ISP*. Um diferencial dos experimentos propostos é sua proximidade com o ambiente operacional das redes dos *ISPs*, isto é, as entidades de roteamento, sua configuração, a troca de dados via *IX* e o algoritmo de roteamento.

Simulações de elementos do ambiente de produção provê proximidade acentuada com o cenário real. Isso garante sustentar um importante aspecto da hipótese, especialmente quanto a aplicabilidade da solução como um implantador de fatias de rede para redes legadas. Inicialmente é apresentado uma proposta de implementação e suas tecnologias. Após, apresenta-se casos de uso que foram explorados para validar a hipótese.

4.1 Considerações Iniciais

Avaliar qualitativamente um mecanismo de fatiamento de rede entre múltiplos *ASs* exige propor e construir cenários que requeiram da solução comportamentos

que possam ser mensurados. Além disso, medir quantitativamente a qualidade operacional e de serviço do mecanismo de implementação de fatias de rede entre múltiplos *ASs* requer métodos específicos para cada cenário. Por isso, esta tese organiza os experimentos em cenários onde cada cenário possui seus habilitadores tecnológicos, o serviço ofertado e uma metodologia de avaliação. A construção dos cenários e seus critérios de avaliação considerou a proximidade com o cenário real e sua pertinência científica. Os cenários propostos variaram em finalidade e contexto, mas têm como componente principal o *NASOR*.

O Cenário Experimental 1 avalia qualitativamente o *NASOR* quanto qualidade funcional de implantar fatias de rede entre múltiplos domínios. Além disso, avalia quantitativamente a qualidade do serviço que uma aplicação específica experimenta sobre uma fatia de rede implantada pelo *NASOR*. Ao passo que o Cenário Experimental 2 avalia qualitativamente o *NASOR* quanto sua capacidade de implantar fatias de rede com políticas que suportem requisitos específicos dos usuários. A avaliação quantitativa desse experimento objetiva medir a escalabilidade e performance de tempo do *NASOR* para implantar fatias de rede entre múltiplos *ASs*.

Já no Cenário Experimental 3 avalia qualitativamente o *framework NASOR* quanto seu atendimento a requisitos específicos para verticais de aplicações. Avaliou-se também quantitativamente, a escalabilidade e performance do mecanismo *NASOR* para implantação de fatias de rede. Foram avaliados quais fatores e níveis mais impactam em variáveis nas variáveis de resposta latência e tempo de processamento. Além disso, observou-se a escalabilidade do método e no desempenho de aplicações que rodam dentro de fatias de rede implantadas pelo *NASOR*.

No Cenário Experimental 4, avalia-se qualitativamente a funcionalidade do *framework NASOR* de admitir componentes terceiros definirem o caminho e a política para se implantar uma fatia de rede entre múltiplos *ASs*. Nesse experimento, observou-se quantitativamente a aplicabilidade de incorporar mecanismos baseados em inteligência artificial para determinar um caminho para a fatia de rede considerando a classe de tráfego predominante. O experimento propõe e avalia um método para classificar tráfego de rede baseado em aprendizado de máquina supervisionado. Além disso, o experimento busca constatar sob quais circunstâncias um classificador pode ser incorporado ao *framework NASOR* para aprimorar

o dinamismo no estabelecimento de fatias de rede.

A Tabela 2 resume os principais pontos explorados nas avaliações experimentais descritas nesta tese. A coluna "Escalabilidade" refere-se ao aspecto de o *NASOR* implantar exaustivamente fatias de rede sobre um *hardware* subjacente. A coluna "Implantação de Fatias de Rede" refere-se à capacidade do *NASOR* de implantar sobre múltiplos roteadores da Internet uma fatia de rede. Já a coluna "Orquestração multi-AS" representa a capacidade do *NASOR* influenciar e proceder com a implantação de fatias de rede ao longo de múltiplos *ASs*. "Inteligência Artificial" refere-se ao habilitador tecnológico utilizado na avaliação experimental como uma extensão das capacidades do *NASOR*. A "Customização da Implantação" relaciona-se com a possibilidade de customizar a implantação da fatia de rede considerando aplicações terceiras.

Cenário	**Escalabilidade**	**Implantação de Fatias de Rede**	**Orquestração multi-AS**	**Inteligência Artificial**	**Customização de Implantação**
#1		✓	✓		
#2	✓	✓	✓		✓
#3	✓	✓	✓		
#4				✓	✓

Tabela 2 – Visão Geral dos Cenários Experimentais

Todos esses cenários experimentais exploram o *framework NASOR* e sua prova de conceito para verificar e corroborar com as questões de pesquisa levantadas. As premissas e tecnologias habilitadores dos experimentos estão apresentadas no contexto de cada caso de uso.

4.2 Proposta de Implementação

Para validar um mecanismo de controle distribuído, considerando no plano de dados construído pelos algoritmos de roteamento, é proposto e implementado uma prova de conceito do *framework NASOR*. O modelo operacional de fatiamento de rede oferecido pelo *NASOR* baseia-se em uma conectividade lógica construída salto a salto. O código da implementação do *NASOR* e seus componentes (*daemons*) estão disponíveis sob licença pública em <https://doi.org/10.5281/zenodo.3899171>.

Essa implementação consiste em máquinas virtuais com sistema operacional *Linux Debian Squeeze* de 64 bits com um *Kernel 4.1*. Sobre essas máquinas virtuais instalou-se um simulador de protocolos de roteamento integrado com as pilhas nativas de rede do *Unix/Linux*. Com isso, qualquer alteração específica em uma interface de rede refletem no comportamento e conteúdo dos algoritmos de roteamento, que conforme sua implementação divulgam as atualizações de rotas.

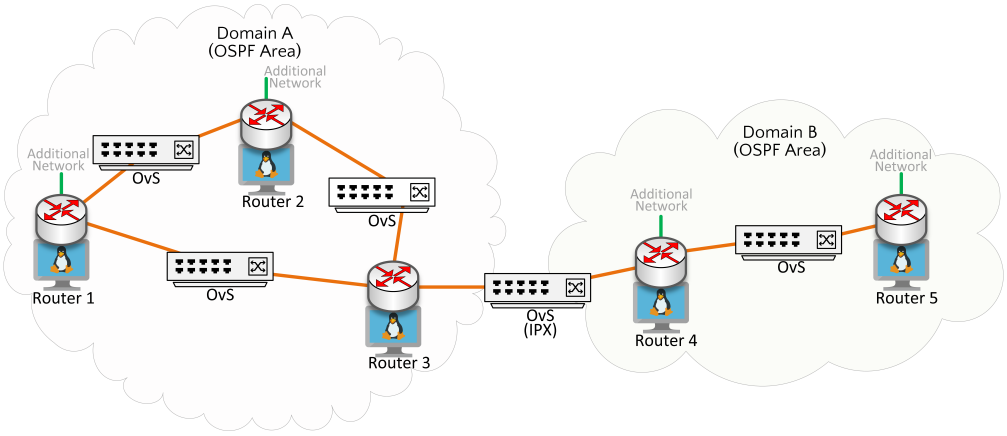

Figura 19 – *Testbed* de Implementação e Experimentação.

A Figura 19 ilustra a proposta de implementação e experimentação do mecanismo de fatiamento proposto, e destaca as principais tecnologias habilitadoras. A versão do protocolo *OSPF* e *BGP* referem-se a implementação compatível com a versão 6 do protocolo *IP*.

Adicionalmente, sobre as máquinas virtuais, que simulam o comportamento de roteadores, foi configurado domínios de roteamento com redes adicionais que são divulgadas pelos protocolos de roteamento. Além disso, foram instalados módulos de *Kernel* para habilitação e identificação de roteamento por segmentos (*srext*[1]). A conexão e ponto de troca de tráfego entre os roteadores ocorre sobre *switches* virtuais (*Open vSwitch*) configurados para atuarem proativamente, isto é, encaminham os pacotes sem a necessidade de consultar um controlador *SDN*.

Dentro dos roteadores virtuais estão os processos clientes (*daemons*) que suportam chamadas remota de procedimentos. Sobre tais recursos são instanciadas

[1] Módulo do *Kernel Linux* que habilita o reconhecimento dos cabeçalhos de roteamento por segmentos (CISCO, 2017).

máquinas virtuais e *containers* para atuarem como hospedeiros cliente e servidor. As rotinas do mecanismo de estabelecimento de fatias de rede são invocadas pelo *NASOR* do domínio local, e quando distribuído encaminha-se a requisição – conforme fluxo de estabelecimento de fatias de rede – na banda (*in-band*), ou seja, as mensagens assíncronas do plano de gerenciamento são transportadas sobre o plano de dados construído pelos algoritmos de roteamento da Internet.

Dentre as limitações dessa proposta de implementação, é imperativo pontuar que os repositórios de dados distribuído, que suportam a separação de domínio e tecnológica, devem estar povoados em tempo prévio ao experimento. Isso inclui endereços *IP* de gerência dos roteadores, dos *MANO* locais e especificidades do *AS* como seu número e nome. Além disso, *scripts* de configuração de parâmetros preliminares, como configuração de interfaces dos roteadores e suas sub-redes, estão disponíveis no repositório desta tese[2] e precisam ser observados antes da experimentação. A escalabilidade experimental do mecanismo *NASOR* é condicionada a observância desses aspectos.

4.2.1 Ferramentas

Esta seção detalha os habilitadores tecnológicos que compuseram a estrutura da solução de fatiamento de rede proposta nesta tese. São descritos o papel de cada ferramenta e sua interação com os demais componentes da solução. Por fim, é proposto uma tabela que sistematiza as ferramentas escolhidas conforme nome, versão e descrição do seu papel.

Na especificação de roteamento por segmento estão previstos dois modelos de planos de dados, baseado em *MPLS* ou *Internet Protocol version 6 (IPv6)*. O desenvolvimento do *NASOR* considerou, por questões de escalabilidade e compatibilidade com o plano de dados construído pelos algoritmos de roteamento da Internet, a utilização do *IPv6*. As funcionalidades do roteamento por segmentos foram estendidas do *Kernel Linux* compatível com os cabeçalhos *IPv6* e *SRH* e que implementam as tabelas *SIDs* para correspondência dos pacotes. O serviço de roteamento, virtualmente implementado sobre as instâncias Linux, é compatível com

[2] Repositório do *NASOR*: <https://doi.org/10.5281/zenodo.3899171>

o protocolo de roteamento *BGPv6* para anúncio de rotas *inter-AS*. Um exemplo de configuração de rotas entre três domínios está disponível no Apêndice B.

A ferramenta utilizada para simular o protocolo de roteamento *BGP* foi a *FRRouting*, nela são configuradas as interfaces, redes e os parâmetros da sessão *BGP* das entidades. Adicionalmente, a implementação do *NASOR* possui uma interface com a tabela *FIB* de cada roteador, essa interface foi implementada sobre a tecnologia *gRPC*. O *gRPC* é um modelo de comunicação baseado em *buffers* de protocolo implementam uma Linguagem de Definição de Interface que permite que os agentes do *NASOR* interajam com os roteadores sobre os quais uma fatia de rede é estabelecida. Um conjunto de comandos válidos do roteamento por segmentos, no que se refere a criação e gerenciamento de entradas nas tabelas *SID*, são exemplificados no Apêndice C.

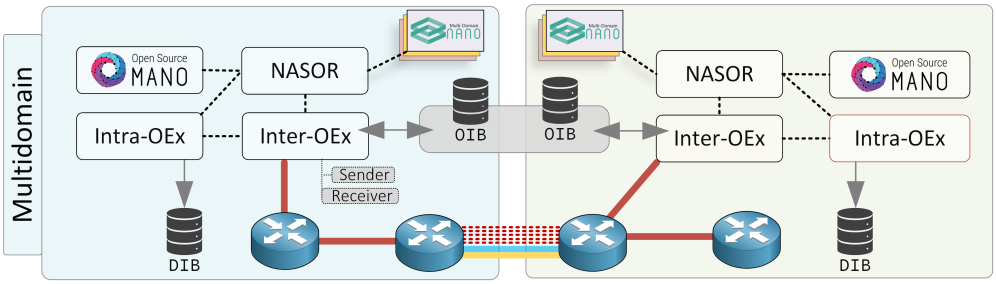

Figura 20 – Estrutura de Conectividade dos Repositórios *OIB* e *DIB*.

Uma modificação operacional no plano de dados entre os roteadores, provocada pelo *BGP*, dispara um gatilho para o micro-orquestrador gerenciador da fatia de rede, alertando-o sobre possíveis modificações de rotas. O micro-orquestrador deverá reconfigurar os *SIDs* levando em conta o novo caminho proposto pelo plano de controle do *BGP*. A implementação desse *watcher* de rotas nas tabelas *FIB* dos roteadores é baseado em uma *API Python* com o *FFRouting* e com a biblioteca *pyroute2*. De forma análoga ao gerenciamento da tabela *SID*, a presente solução implementa um modelo de comunicação baseado em *buffers* para consulta do estado da tabela de rotas.

Implementação do repositório de dados que suportam a persistência das informações hierárquicas dos domínios considerou o habilitador tecnológico *Apache Geode*. Conforme descrito na arquitetura da solução *NASOR*, cada *AS* possui dois

repositórios estruturados conforme ilustrado na Figura 20. Esses repositórios possuem conectividade direta entre o *NASOR*, e no caso dos alocados sobre múltiplos *ASs* sua comunicação ocorre por meio de *IPs* públicos.

O habilitador tecnológico *Apache Geode*, baseado no armazenamento de chave-valor, possibilita que as informações dos domínios sejam armazenadas e recuperadas em formato *JSON*. O modelo distribuído desse repositório de dados permite que eventuais falhas em um repositório de um domínio não comprometa a consistência geral das informações. Nesse sentido, todos os *ASs* são coordenadores do repositório *OIB*, por outro lado, no repositório privativo do domínio há somente um coordenador e com nível de replicação definida pelo administrador. Uma interface, baseada em *RestAPI*, permite que o *NASOR* gerencie as entradas chave-valor armazenadas nos repositórios.

Na Tabela 3 estão sistematizadas as ferramentas utilizadas para prover a solução de fatiamento de rede multidomínios sobre o plano de dados construído pelos algoritmos de roteamento da Internet. A tabela está organizada em três colunas, quais sejam: o nome da ferramenta, versão — aspecto importante porque determinadas características estão associadas a versões específicas –, também há uma descrição textual de como a tecnologia contribui e se comporta na solução proposta.

Ferramenta	**Versão**	**Descrição**
FFRouting	7.2	Suíte de Protocolos de Roteamentos baseados no *Kernel Linux*
Kubernetes	1.17	Gerenciador de *Containers* Docker
Docker	19.03.8	Plataforma dos Micro-Orchestradores
Apache Geode	1.11.0	Repositório de Dados (*OIB* e *DIB*)
OpenSource MANO (OSM)	5	*Local MANO*
gRPC	1.27.3	*RPC Framework*
gemfire-rest	1.0.2	*API* do *NASOR* com os Repositórios (*DIB* e *OIB*)
Kernel Linux	4.1	Compatível com *Match* de cabeçalhos de *Segment Routing*
pyroute2	0.5.10	*Watcher* da Tabela de Rotas
pycos	4.8.15	Comunicação assíncrona dos *NASORs*
NetworkX	2.4	Apresentação das Topologias das Fatias de Rede
Flask RestAPI	0.3.8	*API* para interação baseada em *Web-services* entre os Módulos
Open vSwtich	2.5.0	Provimento de Conectividade entre os MOs e para os Roteadores
PyYAML	0.2.2	*Parser* de Descritores de Serviço (*NSTD*, *VNFD*, *NSD*)
OpenStack (VIM)	*Pike*	Gerenciador de Máquinas Virtuais (*VNFs*)

Tabela 3 – Habilitadores tecnológicos da solução *NASOR*.

4.3 Avaliação Experimental

O primeiro experimento propõe medir o impacto da inserção do *NASOR* em um ambiente em operação puro, isto é, sem quaisquer abordagens de fatiamento de recursos. As próximas subseções discutirão métricas e sua pertinência para mensurar a sobrecarga de sinalização e gerenciamento para o plano de dados que o modelo de fatiamento de rede eventualmente implica. Adicionalmente, discute-se o nível de sobrecarga em contraste com outras abordagens. Para isso é proposto um cenário de múltiplos domínios, onde são executados algoritmos de roteamento que garantem a conectividade *inter-AS*. Cada infraestrutura de *AS* possui recursos computacionais que são ofertados aos usuários na modalidade *IaaS*. O caso de uso desta avaliação experimental consiste em estabelecer uma fatia de rede para as aplicações e mensurar eventuais atrasos que são impostos na entrega de pacotes transportados sobre a fatia de rede.

O segundo experimento valida um caso de uso que exemplifica a aplicabilidade e abrangência da solução para prover o fatiamento de rede e a implantação de serviços de rede sobre múltiplos *ASs*. O cenário consiste em fatiar logicamente uma rede para duas entidades: um cliente e um servidor que se comunicam no contexto de uma aplicação *DNS*. Nesse experimento é mensurado a performance de uma fatia de recursos — aquela que envolve recursos de rede e de computação –, quanto ao atraso de requisições sob um lapso temporal.

O terceiro experimento valida a prova de conceito do *NASOR* no quesito escalabilidade de implantação de fatias de rede sobre um único recurso subjacente. Além disso, esse experimento avaliou a capacidade do *NASOR* implantar fatias de rede para aplicações que requerem latências muito baixas. O experimento por meio da avaliação de influência de fatores vista conhecer quais fatores mais influências na qualidade de serviço percebida pelos serviços que rodam sobre as fatias de rede implantadas pelo *NASOR*.

O quarto experimento avaliou a aplicabilidade do *NASOR* de valer-se de mecanismos terceiros para decidir pelo caminho da fatia de rede ao longo dos múltiplos enlaces e *ASs*. Esse experimento, vale-se de tecnologias de Inteligência Artificial, conforme apresentado na Seção 2.5, especialmente redes neurais convolucionais para instruir o *NASOR* sobre o tipo de classe de tráfego predominante em um

canal. Essa instrução permite o *NASOR* agrupar as fatias de rede implantadas por classe de tráfego. O aspecto mais importante desse experimento, é avaliar a pertinência de haver mecanismos terceiros capazes de influenciar o processo de estabelecimento de fatias de rede entre múltiplos *ASs*.

Além disso, uma descrição pragmática dos habilitadores tecnológicos e dos cenários experimentais bem como suas métricas e resultados são apresentadas.

4.4 Cenário Experimental 1: Implantação de Fatias de Rede

Conforme ilustrado na Figura 21, o cenário explora a capacidade do *NASOR* de prover a conectividade lógica para *VxF* entre *ASs*, e valida a aplicabilidade desse mecanismo de fatiamento de rede como uma tecnologia habilitadora para diminuir a latência de aplicações. O *NASOR* recebe via *Northbound Interface (NBI)* os arquivos descritores do serviço (*VNFD*, *NSD* e *NSTD*) contendo as especificações da *VxF* e o descritor da fatia de rede multi domínios contendo os *ASs* envolvidos na fatia de rede.

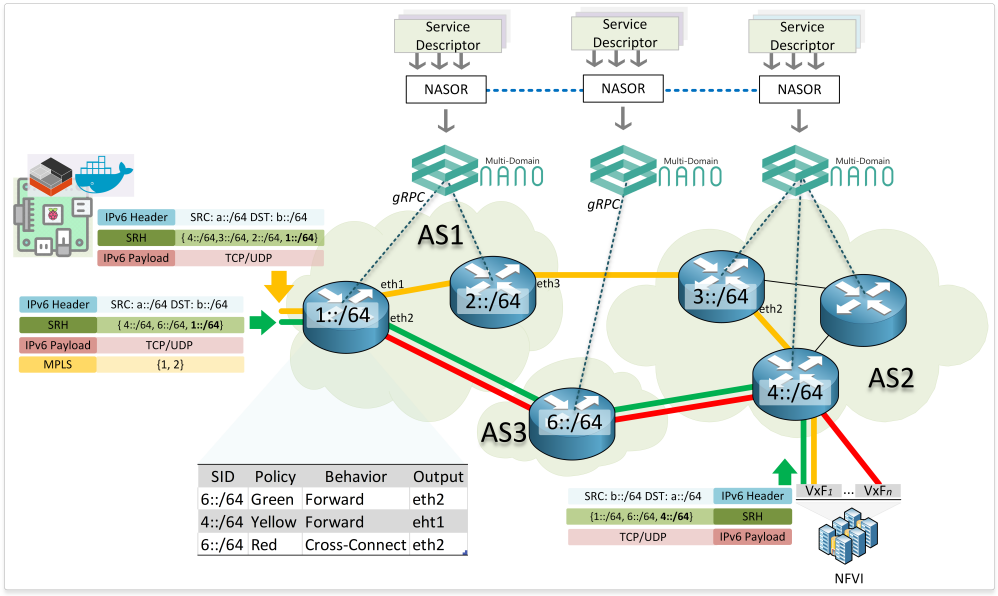

Figura 21 – Cenário Experimental 1 – Fatiamento de Rede *Inter-AS*.

Para o cenário experimental foi utilizado o *OSM* conforme apresentado na Tabela 3 que gerencia um *VIM* baseado em *OpenStack* com imagens previamente configuradas. A *VxF LW-DNS* foi construída sobre uma imagem *Ubuntu* 16 com a aplicação *Bind9*, além disso, para explorar a aplicabilidade da tecnologia de fatiamento de rede foi construído um *container* do tipo *Docker* com a aplicação *Bind9*. As imagens compatíveis com a virtualização foram adicionadas ao catálogo do *VIM* e as baseadas em *container* no gerenciador *Docker*.

Adicionalmente, na Figura 21 está ilustrado três domínios em que o plano de dados é estabelecido pelo algoritmo de roteamento *BGPv6*, os *ASs* 1, 2 e 3. Cada roteador contém a implementação *SR* que viabiliza o *NANO* local associar *SIDs* às interfaces que compõe o caminho mais curto e aos serviços (tráfego). No topo da ilustração, a entidade *NASOR* recebe os arquivos manifesto contendo a descrição do serviço e a encaminha ao *NANO* e ao *MANO* Local, após encaminha para o *NASOR* do *AS* vizinho. O estabelecimento de uma fatia de rede é iterativo e *top-down* entre os roteadores dos domínios, assim o *NANO* consulta a *FIB* dos roteadores locais para determinar e instalar os *SIDs* nas interfaces que representam o caminho fim-a-fim entre *VxF*.

Para esse experimento, o caminho que o algoritmo de estabelecimento de fatias de rede toma considerou o caminho mais curto computado no plano de controle do *BGP*. No entanto, o *NASOR* possui acesso a rotas alternativas que podem ser consideradas pelo mecanismo de fatiamento de rede.

Antes das *VxF*s injetarem pacotes na rede com alvo ao seu destino, pelo transporte de sua fatia de rede, a entidade de entrada adiciona um cabeçalho ao pacote, o *SRH*. O *SRH* contém o caminho pré-estabelecido por onde o pacote deverá passar, esse caminho pode transladar os *ASs*. Um exemplo desse cabeçalho é ilustrado na parte inferior esquerda da Figura 21, nela estão apresentadas três pilhas de protocolos dos pacotes.

É possível notar que o cabeçalho *SRH* está entre a carga útil do *TCP/UDP* e o cabeçalho *IPv6*, isso implica que, como o processamento do pacote se dá de baixo para cima, antes de encontrar o cabeçalho original de endereçamento, haverá sido encontrado a lista de roteadores que o pacote deverá ser encaminhado primeiramente. Esse modelo de processamento da pilha de protocolos permite que a cada nó contido na lista de segmentos seja visitado, estabelecendo o caminho

lógico entre domínios.

O cabeçalho *SRH* contém um campo para definir a política e a coloração, conforme ilustrado, todas as fatias de rede possuem a mesma origem e destino. A entrada coloração na tabela *SID* é preenchida conforme ilustrado na Figura 21 para o *AS3*. Nessa tabela, existe a entrada *SID* que representa o endereço do próximo nó da lista de segmentos, a política e o comportamento destinado ao pacote e a interface pela qual o pacote poderá sair.

A coloração permite que sejam atribuídos ao pacote uma política de encaminhamento específica. Quanto ao comportamento, é previsto na arquitetura *SR* executar operações sobre pacotes *SRH* como adicionar, alterar ou retirar elementos da lista de segmentos. No processo interativo de estabelecimento de uma fatia, ao encontrar a fronteira do domínio, é considerado o término da jurisdição do *NASOR* que configura a interface de *peering* e aguarda a configuração do *NASOR* do domínio vizinho.

4.4.1 Método de Avaliação: Cenário Experimental 1

É proposto validar o *NASOR* através de dois experimentos, o primeiro constata a aplicabilidade do fatiamento multidomínios para aprimoramento da percepção dos usuários finais quanto ao tempo de resposta de uma consulta *DNS*. Para isso, foi mensurada a latência percebida pelos usuários finais nas consultas *DNS* em fatiamento multi domínio por meio da ferramenta *Namebench* (GOOGLE, 2019).

Segundo, foi experimentado a sobrecarga de tempo em termos de atraso fim-a-fim causado pelo processamento dos pacotes em uma fatia multidomínio por meio da ferramenta *Flent* (HØILAND-JØRGENSEN et al., 2017). Adicionalmente, foi verificado o desempenho do fatiamento de rede em múltiplos domínios providos pelo *NASOR* com o cenário não orientado ao fatiamento de rede — notadamente um *baseline* – e com a abordagem baseada em *VPN*. Todos os experimentos consideraram uma rede sem congestionamentos e sem transporte pacotes de outa natureza, como os provenientes de protocolos de controle que eventualmente poderiam estar executando em segundo plano.

Similar à metodologia de Ager et al. (2010), foi avaliado o desempenho da *VxF LW-DNS* descrita e implantada conforme a especificação *ETSI* sobre a infraestru-

tura de múltiplos domínios. A *VxF LW-DNS* foi hospedada em um *Raspberry Pi 3 (RPi3)* que recebeu consultas *DNS* pela *VxF* cliente (sobre virtualização *bare-metal*) através da fatia de rede entre múltiplos domínios previamente estabelecida. Majoritariamente, usuários domésticos utilizam o serviço de *DNS* provido pelo seu *ISP* (AGER et al., 2010), alternativamente, com a *VxF LW-DNS* é proposto novos entendimentos relacionados à colocação de servidores *DNS* próximo do usuário, sobretudo em fatias de rede exclusivas dos usuários.

Estudos realizados por Mao et al. (2002) e Shaikh, Tewari e Agrawal (2001) introduziram e avaliaram os benefícios da proximidade *DNS* com os usuário finais. Os experimentos propostos trazem à luz o conceito de *DNS* da Fatia de Rede como um caso de uso experimental do *NASOR*. Para evitar *bias* na métrica responsividade do serviço de *DNS*, foi considerada da base Alexa (2019) as 2.000 consultas mais populares e as 2.000 menos populares.

O objetivo principal é responder as seguintes questões:

- ❑ É estatisticamente melhor utilizar infraestruturas de *DNS* de alta performance ao invés de abordagem de funções de rede virtualizada dentro de uma fatia de rede?

- ❑ Quanto é a sobrecarga de tempo que os pacotes experimentam com a abordagem *NASOR* para fatiamento de rede multi domínios?

4.4.2 Avaliação dos Resultados: Cenário Experimental 1

A partir do gráfico ilustrado na Figura 22 é possível inferir que o tempo de resposta para consultas *DNS* não é sempre estatisticamente melhor para servidores de alto desempenho. A análise permite inferir que quando se utiliza infraestruturas de alto desempenho como *Google DNS* ou *OpenDNS* comparado com uma abordagem baseada em função de rede implantada dentro de uma fatia de rede podem ocorrer com desempenho similar.

Nesse sentido, o tempo de consulta *DNS*, desconsiderando a possibilidade de armazenamento prévio em *cache*, da *VxF LW-DNS* comparada com o *Google DNS* são aparentemente com intervalo de confiança de 95% equivalentes. O desempenho

da infraestrutura *OpenDNS* foi inferior, portanto, não se discutirá quantitativamente seus resultados com a abordagem proposta.

Para verificar com rigor se uma consulta *DNS* dentro de fatia de rede implantada pelo *NASOR* experimenta uma latência maior do que fora de uma fatia de rede, foi conduzido um teste de hipóteses Z. Por isso, as seguintes hipóteses foram conjecturadas: $H_0 : \mu \leq 241.4ms$ e $H_a : \mu > 241.4ms$, onde μ refere-se a amostra das latências dentro de uma fatia de rede implantada pelo *NASOR*. Admitindo-se um erro de 5%, constatou-se um $Z_{observado} = 3.57$ e um $Z_{crítico} = 1.64$, sugerindo a rejeição da hipótese H_0 porque o $Z_{observado}$ está dentro da região crítica da distribuição normal. Logo, a latência experimentada dentro de uma fatia de rede implantada pelo *NASOR* não é menor ou igual a experimentada fora da fatia de rede.

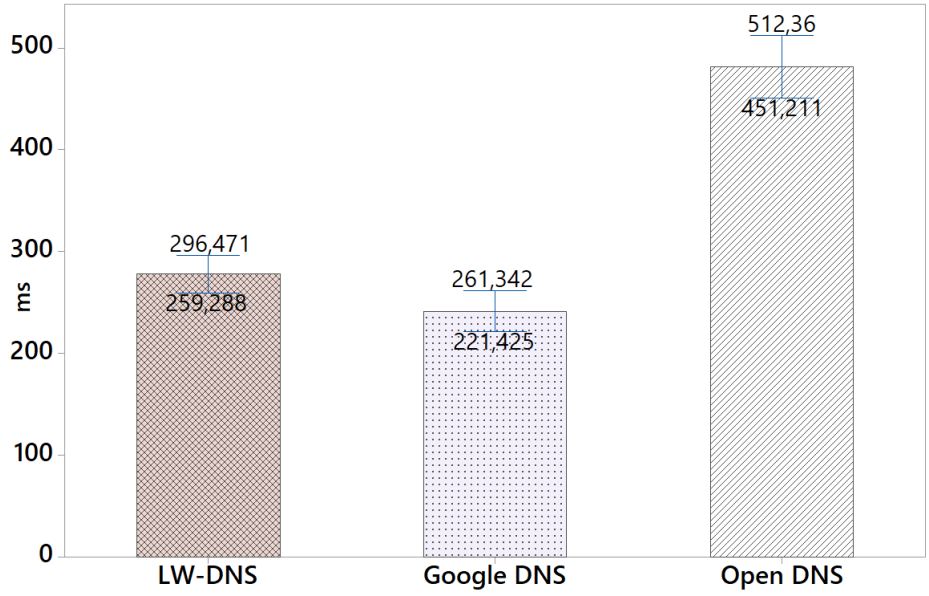

Figura 22 – Latência de uma consulta *DNS* dentro da Fatia de Rede *inter-AS*.

Em termos quantitativos a latência de uma consulta *DNS* dentro de uma fatia de rede estabelecida pelo mecanismo *NASOR* é de 277.88 *ms* para a *LW-DNS* e 241.4 *ms* para o *Google DNS*. No entanto, para evitar *bias* estatístico foram executadas 30 vezes o *benchmark* o que permitiu validar o tempo de resposta de uma consulta com 95% de confiança. Assim, esse resultado permite conjecturar

que o desempenho de aplicações dentro de uma fatia de rede estabelecida pelo *NASOR* não padecem com a queda de desempenho significativas comparadas com aplicações que executam sobre plataformas de alto desempenho.

Constatou-se que a latência de uma consulta *DNS* dentro de uma fatia de rede implantada pelo *NASOR* é cerca de 15.07% maior do que fora da fatia de rede. Em contrapartida à sobrecarga, na fatia de rede implantada pelo *NASOR* são garantidos aos usuários acesso a recursos exclusivos da fatia de rede que o acordo do nível de serviço prevê. Uma tabela contendo a estatística descritiva desse experimento está disponível no Apêndice D.

Além disso, conforme Figura 23 é possível constatar que a média da latência percebida dentro da fatia de rede é de 277.8ms, esse valor ainda está abaixo do terceiro quartil (302.8ms) das latências registradas fora da fatia de rede. Isso permite concluir que a média de latência dentro da fatia de rede ainda é menor que 75% das latências experimentadas fora da fatia de rede.

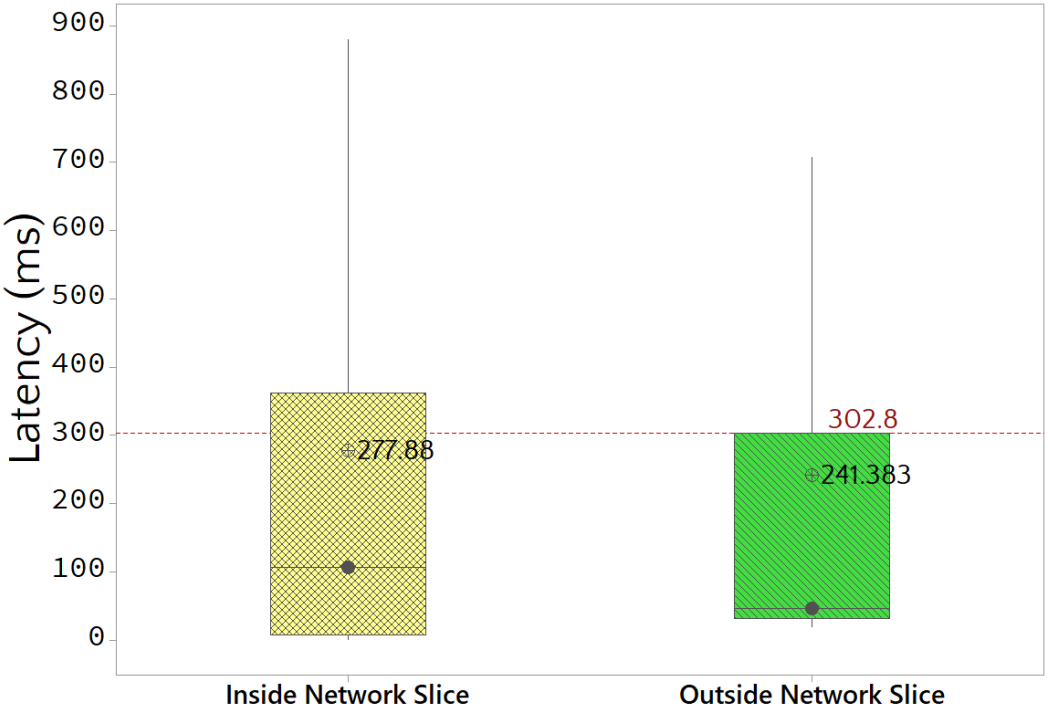

Figura 23 – Latência de uma consulta *DNS* dentro e fora da fatia de rede *inter-AS*.

Adicionalmente, na Figura 24 está ilustrada a compilação estatística que consi-

derou a inclusão da característica de *cache* no *benchmark* da aplicação *DNS* sobre a fatia de rede estabelecida pelo *NASOR*. Assim, pela ótica da probabilidade acumulada, em 95% dos casos, a latência de uma consulta experimentada pela aplicação na fatia de rede estabelecida será de 12 *ms* para o *LW-DNS* e 43 *ms* para o *Google DNS*.

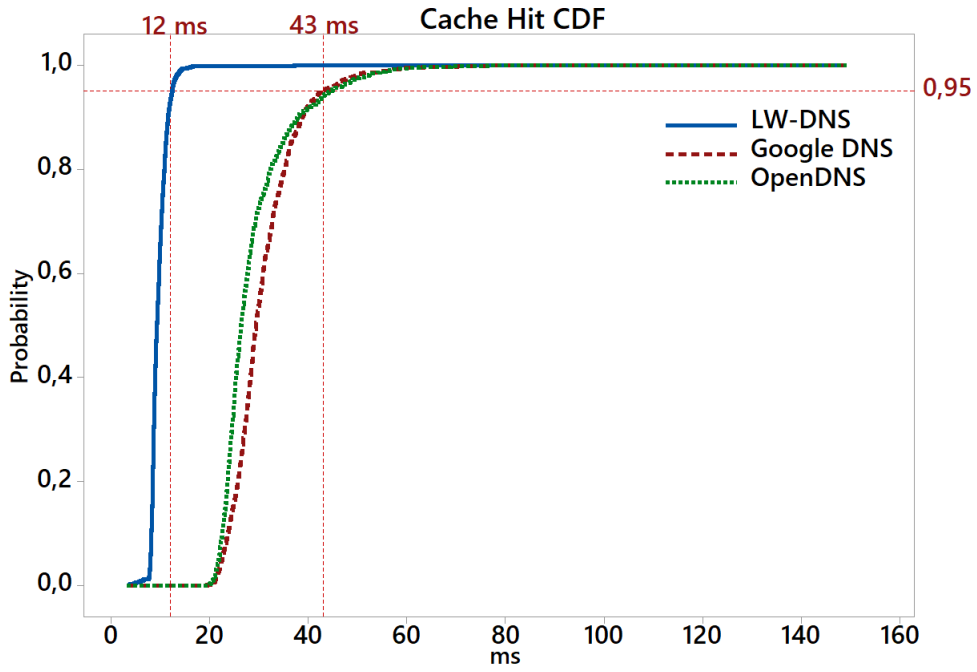

Figura 24 – Probabilidade Acumulada – Consulta *DNS* com cache na Fatia de Rede *inter-AS*.

A infraestrutura *OpenDNS* apresentou desempenho próximo do *Google DNS*. A superioridade de performance da aplicação na fatia de rede estabelecida com a característica de *cache* habilitada é ≈ 3 vezes superior a seus pares, *Google DNS* e *OpenDNS*.

De acordo com Qian, Rabinovich e Al-Qudah (2011) a maioria das consultas *DNS* são conhecidas, portanto a percepção da performance da aplicação pelo usuário pode ser aprimorada com abordagens de *LW-DNS* colocados na fatia de rede. Além disso, é razoável questionar questões de segurança e privacidade nos servidores *DNS* de alta performance como o *Google DNS* e *OpenDNS*.

O estado da arte discute alternativas para lidar com esses desafios (ZHU et al., 2014; SHULMAN, 2014; NAKATSUKA; PAVERD; TSUDIK, 2019). Desse modo, a abordagem *NASOR* de fatiamento de rede permite um ecossistema para experimentar novas abordagens de aplicações, tradicionalmente de propósito geral, para servir exclusivamente a usuários.

A outra vertente de experimentos propostos visaram mensurar a sobrecarga de desempenho que a abordagem de fatiamento de rede *NASOR* impõe no processamento dos pacotes ao longo da rede lógica *inter-AS*. Para essa validação, foi mensurado o desempenho da rede pela ferramenta *Flent* para uma conectividade entre duas entidades colocadas em dois domínios diferentes. Conforme o cenário ilustrado na Figura 21, a fatia de rede criada pelo *NASOR* para validar o *overhead* de processamento dos pacotes é a de coloração vermelha.

Os resultados estatísticos compilados e ilustrados na Figura 25, passíveis de aferição nesse experimento, sugerem que a abordagem *NASOR* para fatiamento de rede impõe um atraso fim-a-fim para aplicações que se comunicam entre *AS* distintos. Assim, com 95% de certeza, a latência de uma comunicação que envolve múltiplos domínios e que utiliza abordagem do *NASOR* proverá um desempenho entre 1.59 *ms* e 1.70 *ms* de latência para uma comunicação entre *AS* diferentes. O atraso mensurado diz respeito ao tempo de processamento do *SRH* nos pacotes que transpassam as entidades de roteamento.

Foi proposta uma *baseline* para contrastar a performance do fatiamento de rede proposto pelo *NASOR*. A *baseline* é a comunicação entre *ASs* sem quaisquer mecanismos de fatiamento, portanto foi nomeada como *flat*. Nessa abordagem o *overhead* de processamento que existe está relacionado a tarefa de roteamento padrão entre os domínios, provido pelo *BGP*. Baseado no cenário experimental ilustrado na Figura 21, a comunicação *baseline* é de coloração verde, disso decorre que o caminho de menor custo entre duas entidades colocadas nos *ASs* 1 e 2 será transportado por esses elementos de roteamento.

Por outro lado, outra comparação que se estabelece a abordagem de fatiamento inter domínios do *NASOR* com as propostas do estado-da-arte predominantemente baseadas em *VPN*. Portanto, considerando a mesma topologia de experimentos da Figura 21, foi estabelecida uma *VPN* entre as entidades cliente e servidor considerando uma abordagem de comunicação *flat*, isto é, a comunicação *IP* pura entre

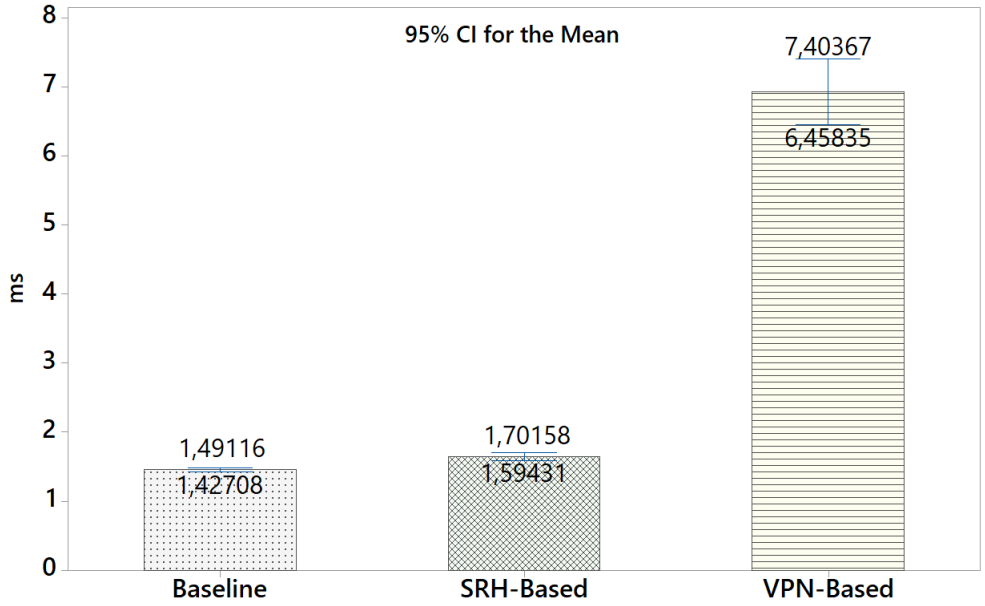

Figura 25 – *Overhead* da abordagem *NASOR* para Fatiamento de Rede *inter-AS*.

domínios. Assim, o desempenho da abordagem baseada em *VPN* é trivialmente inferior a abordagem de fatiamento *NASOR*.

A abordagem baseada em *VPN* logra vantagens de segurança e privacidade, uma vez que a comunicação é baseada em troca de chaves que provê a confidencialidade dos dados trocados. Isso justifica o acréscimo de tempo de processamento dos pacotes que são trocados sobre essa rede lógica inter domínios. Além disso, a aplicação de políticas para cada rede lógica, baseada em *VPN* não é direta, comparada com a abordagem *NASOR*. Nesse sentido, a abordagem *VPN* foi posta a prova porque majoritariamente as soluções de fatiamento e implantação de serviços sobre múltiplos domínios a possuem como habilitador tecnológico. No entanto, a rigor, não é admissível, conforme levantamento bibliográfico, assumir que uma *VPN* é qualitativamente compatível com uma fatia de rede.

As respostas dos questionamentos que motivaram os experimentos podem ser conjecturadas baseadas nas estatísticas dos experimentos conduzidos. Portanto no que diz respeito a eventual vantagem de se utilizar infraestruturas de *DNS* de alta performance ao invés de abordagens de funções de rede virtualizadas dentro de uma fatia de rede pode ser justificada pela privacidade que um *DNS* exclusivo

pode oferecer. No entanto, a abordagem de alto desempenho oferece um desempego ligeiramente melhor. A abordagem de fatiamento de rede do *NASOR* proveu desempenho próximo às de alto desempenho, exigiu-se, portanto, que outros aspectos qualitativos como segurança e privacidade fossem introduzidos observados.

No mesmo sentido, uma variável permitiu mensurar qual foi o nível de sobrecarga de processamento que recaiu sobre os pacotes na abordagem de fatiamento de rede, provida pelo *NASOR*. Assim, o questionamento se satisfaz estatisticamente com a asserção de que em 95% dos casos a sobrecarga de processamento do cabeçalho identificador das fatias de rede baseada no *NASOR* foi de 13.10%. Em termos numéricos, uma média de 1.45 *ms* para a abordagem *flat*, isto é, sem quaisquer mecanismos de fatiamento e 1.64 *ms* para a abordagem de fatiamento *inter-AS* baseada no *NASOR*. A estatística descritiva desse experimento está disponível no Apêndice D.

4.5 Cenário Experimental 2: Orquestração Multidomínios e Customização da Implantação

Com objetivo de validar as funcionalidades do *NASOR* foi explorado uma vertente de experimentos adicionais. No cenário da Figura 26 está ilustrado dois domínios (*ASs*) que se comunicam por uma interface de *peering* e divulgam rotas de redes alcançáveis conforme política interna de cada domínio. O domínio 1 possui *VNFs* clientes conectadas diretamente ao roteador $R1$, no roteador $R5$ está conectado outra *VNF*.

Ao ingressar no domínio *IP* do roteador os pacotes são tratados conforme a política a área comum do *OSPF*. Isso significa que para cada *AS* há uma área de roteamento interno específico que implica que os roteadores de borda precisam divulgar rotas internas na sessão *BGP*. Além disso, o agente *NANO* leva em conta o caminho de dados estabelecido pelo *OSPF* na política tradicional de escolha de caminhos para as fatias de rede.

A Figura 26 propõe ilustrar uma interface de gerenciamento onde o *NASOR* de cada domínio exerce influência sobre o plano de dados de seus roteadores para estabelecer o fatiamento de rede entre múltiplos domínios. Assim, pode ser ob-

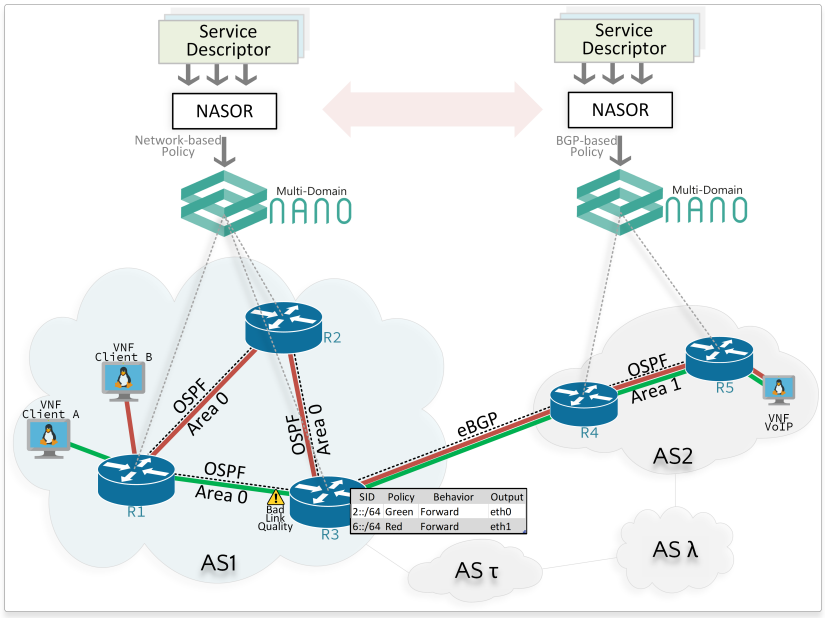

Figura 26 – Cenário Experimental 2: Políticas de Definição de Caminhos para Fatias de Rede.

servado três listras que conectam os roteadores ao longo de um caminho, elas são: preta tracejada, vermelha e verde; a tracejada se refere à conectividade entre os roteadores para divulgação de rotas e transporte de dados. As demais listras, verde e vermelha e eventualmente outas, referem-se à conectividade lógica, sobre o plano de dados *IPv6*, criada para a comunicação das *VNFs*.

Nesse cenário, os roteadores *R3* e *R4* são roteadores de borda e se comunicam mutuamente por uma interface de *peering*, sobre a qual são transportados anúncios de rotas *BGP*. Há de se notar uma tabela *SID* no roteador *R3*, nela estão representadas instruções instaladas para as duas fatias de rede instanciadas para o experimento proposto. Cada entrada na tabela *SID* contém um comportamento, configurado pelo agente *NANO* da fatia de rede, que instrui os pacotes que satisfazem a combinação do *SID* a percorrer um determinado caminho ou ter determinado comportamento em um nó específico.

Todos os roteadores, inclusive os do domínio τ e λ, são gerenciados por seus *Network and Slice Orchestrators (NASORs)* domésticos, eles se comunicam entre si na troca de informações no processo de criação da conectividade lógica. O arquivo

de configuração dos roteadores desse experimento está disponível no Apêndice F.

A dinâmica de experimentos consiste em um *NASOR* do Domínio 1 (*AS1*) receber o arquivo descritor do serviço baseado em *YANG* contendo as especificações de computação e da fatia de rede para tratá-los de maneira apropriada. Em especial, a especificação da fatia de rede, sob a ótica de conectividade, contém a informação do tipo de política que o mecanismo de fatiamento de rede, isto é, o caminho que o agente configurador de *SIDs* — o *NANO* – deverá estabelecer entre múltiplos domínios.

Um exemplo de arquivo de descrição do serviço baseado em *YANG* está disponível no Apêndice G. O serviço de computação para esse experimento é uma aplicação *VoIP* com duas entidades se comunicando em voz, cliente A e B. O transporte dos pacotes da aplicação *VoIP* translada os *ASs* e experimentam condições diferentes de rede.

No cenário da Figura 26 foram exploradas duas políticas para estabelecimento de fatias de rede entre os múltiplos domínios. Com isso, é possível explorar a funcionalidade de Política de Fatiamento Aberta que o *NASOR* possui. As políticas podem ser definidas externamente, isso se materializa pela interface *OPI* implementada no *NASOR*. Assim, caminhos diferentes entre múltiplos domínios que consideram aspectos de qualidade de rede, como métricas granulares de cada roteador ou política de bilhetagem entre os domínios podem ser construídas.

Nesse ponto, a proposta da interface *OPI* permite que sejam exploradas abordagens diversas para construção de caminhos. Os caminhos são construídos por implementações terceiras que operam seu mecanismo sobre um tipo abstrato de dados: um Grafo duplamente conectado com características de rede nos *links*.

Nesse sentido, o experimento proposto considera duas políticas para estabelecimento de conectividade lógica entre múltiplos domínios. Uma baseada na rede e a outra baseada no caminho de dados fornecido pelo *BGP* e *OSPF*. Nesse experimento, a discussão que se propõe é: avaliar a viabilidade de uma interface aberta, implementada como uma funcionalidade do *NASOR*, para especificação de caminhos para fatias de rede ao longo de vários *ASs*. Para endossar a discussão, é proposto falhas nos *links* com vistas a avaliar a aplicabilidade de uma política em detrimento de outra. Essa interface aberta permite que uma política de estabelecimento do caminho ao longo dos *ASs* seja definida pelo criador da fatia de rede.

Com isso, a proposta de oferecer dinamismo para o estabelecimento de fatias de rede.

Para explorar a *OPI* é proposto um algoritimo para escolha de caminhos para estabelecimento de fatias de rede sobre múltiplos domínios. O Algoritmo 2 é uma abordagem construída para subsidiar a discussão da aplicabilidade da Interface de Política Aberta para estabelecimento de fatias de redes. O resultado do algoritmo é: a escolha de um caminho que considera, conforme uma abordagem gulosa, o caminho que melhor satisfaz a política de rede dentro do *AS* até o roteador de borda. Como saída é retornado: uma estrutura de dados Grafo que alimenta o *NANO* para orientá-lo sobre os roteadores e interfaces que devem ser configurados para prover a separação lógica da fatia de rede.

Quanto aos processos do algoritmo têm-se: 1) considere o próximo *Autonomous System Number (ASN)* contido no *AS PATH*, que representa o caminho inter-domínios que a fatia de rede será estabelecida. Avalie a qualidade da conectividade dos roteadores de borda entre os *ASs*; 2) o roteador de borda que melhor cumprir os parâmetros da política de rede, no caso deste experimento: a latência, é escolhido como o ponto final, ou seja, o *gateway* do domínio para a fatia de rede; 3) considere como ponto final do domínio corrente o roteador de borda escolhido, aquele que melhor cumpri a política na interface de *peering*/transito, calcule o caminho interno do domínio considerando a política de qualidade de rede; 4) retorne ao *NANO* doméstico do *AS* corrente o grafo que representa os roteadores e as interfaces que serão configuradas para prover a conectividade lógica; os passos 5-12 representam o processo interativo de configuração de *SIDs* ao longo do caminho interno do *AS*; 13) encaminhe ao *NASOR* do próximo *AS PATH* a requisição, para estabelecimento de fatias de rede, que considera como ponto inicial o roteador de borda vizinho do *AS* predecessor. O *NASOR* provê o cálculo, conforme estratégia gulosa do caminho a ser configurado pelo *NANO* para a fatia de rede.

Além disso, este cenário de experimentos permite avaliar o tempo de implantação de fatias de rede, e propõe verificar os aspectos teóricos, tecnológicos e de implementação que influenciam diretamente nele. Conforme ilustrado, há duas fatias de rede de coloração verde e vermelha que foram implantadas considerando premissas diferentes, sobre um cenário de múltiplos domínios. Além disso, o cenário permite avaliar a aplicabilidade da Interface de Política Aberta para estabe-

Algoritmo 2: Multidomain Network Performance-based Path (MDPP).

Result: $\forall\ R \in \{$Caminhos entre uma Origem s e um Destino $d\} \rightarrow$ Escolha um caminho que considera o menor *overhead* local das interfaces corforme a política descrita no template *NSTD*.
Input: Um Grafo Direcionado que representa a topologia Interna do Domínio do *NASOR* recebeu a requisção de estabelcimento da fatia de rede.

```
Considere o AS-PATH do BGP para uma fatia de rede fim-a-fim e avalie a interface de peering ou trânsito;
Escolha o roteador de borda que possui a interface de peering/trânsito com o melhor desempenho conforme a métrica;
Considere o Grafo que possui caminhos entre s e d: compute um caminho conforme a estratégia gulosa que melhor satisfaz a métrica até o roteador de borda;
Retorne ao NANO a árvore que representa o caminho escolhido para a fatia de rede;
while R is not end-point target do
    if R == Roteador de Borda then
        Configure os SIDs e as Políticas na Interface peering/trânsito;
        break;
    end
    Configure a tabela SID e Políticas conforme o caminho baseado na métrica de rede;
    R = R → Neighbor;
end
Encaminhe ao próximo NASOR do AS-PATH a requisição de estabelecimento de fatia de rede bem como a política a ser utilizada.
```

lecimento de fatiamento de rede levando em conta dois caminhos distintos entre vários *ASs*. Para um caminho, especificamente o verde entre os roteadores $R1$ e $R3$, foram adicionados fatores adversos na comunicação, similar ao que ocorre em uma conexão *WAN* como perdas de pacotes e atrasos conforme um percentual a ser detalhado posteriormente.

A fatia de rede verde, estabelecida pelo *NASOR*, leva em conta o caminho de dados entre a *VNF* cliente A e B escolhido pelos algoritmos de roteamento *OSPF* e *BGP*. Nesse ponto, há de se assumir que é o caminho de menor custo/salto entre a *VNF* cliente A e B. Adicionalmente, a fatia de rede vermelha, cujo caminho foi definido por uma implementação externa ao *NASOR*, considerou para a implantação da conectividade lógica as métricas de rede, especificamente: quantidade de tráfego e latência entre as adjacências ao longo do caminho que interliga as *VNFs*.

Assim, este cenário de experimentos propõe quatro avaliações:

- ❏ medir e avaliar o tempo de implantação de uma fatia de rede considerando o caminho de dados construído pelos algoritmos de roteamento;

- ❏ medir e avaliar o tempo de implantação de uma fatia de rede considerando uma abordagem alternativa ao caminho de dados dos algoritmos de rotea-

mento (*OSPF* e *BGP*), notadamente uma política que considera condições de rede;

- ❏ avaliar quantitativamente e qualitativamente a pertinência da adoção de caminho para uma fatia de rede em detrimento de outro em um cenário com condições adversas na rede;

- ❏ discutir a pertinência de uma interface aberta para especificação de caminhos que fatias de rede entre múltiplos *ASs* podem adotar.

4.5.1 Método de Avaliação: Cenário Experimental 2

Para esse cenário de experimentos foi considerado o tempo gasto pelos *NASORs* para configuração dos parâmetros de uma fatia de rede ao longo de múltiplos domínios. Para isso, foi registrado o instante que o *NASOR* recebeu uma requisição de estabelecimento de fatia de rede até a conclusão da configuração dos parâmetros da fatia pelo último *NASOR* do domínio. Nesse ponto, é possível medir o tempo que o *NASOR* demandou para configurar os parâmetros da fatia de rede por meio da operação: $TempodeImplantação = TempoFinal - TempoInicial$. A partir dessa grandeza de tempo é possível compreender a influência do comportamento assintótico do algoritmo de definição de caminhos no tempo de configuração da fatia de rede ao longo dos domínios.

A avaliação do tempo gasto no processo de implantação de uma fatia de rede consiste em comparar a proposta que considera o caminho de dados oferecido pelos algoritmos de roteamento, concebidos como uma *baseline*, com os demais algoritmos construídos por terceiros, que consideram como pesos nas arestas condições diversas da *link* de rede. Com isso, abre-se a possibilidade de discussão da qualidade de uma abordagem em detrimento de outra, além disso, traz outra percepção no que diz respeito ao dinamismo de estabelecimento de fatias de rede em cenários multidomínios. Portanto, a equação de tempo se aplica tanto para novas abordagens de estabelecimento de caminhos quanto para o processo *baseline* de implantação de fatias de rede, a saber: o que considera o caminho de dados escolhido pelos algoritmos de roteamento.

Um ponto importante do Algoritmo 2 que subsidia a discussão sobre a aplicabilidade da *OPI*, funcionalidade do *NASOR*, é que os *ASs* que hospedam o caminho de uma fatia de rede fim-a-fim leva em conta a lista que o plano de controle do *BGP* possui, isto é, o *AS PATH*. Em outras palavras, o próximo *AS* que os parâmetros do fatiamento serão configurados em seus roteadores é o que está previsto na lista *AS PATH*, e não o que poderia oferecer uma melhor qualidade em uma perspectiva global. A discussão que essa abordagem suscita é que eventualmente pode haver um caminho, por outros *ASs* que estão fora do *AS PATH*, que ofereça um melhor custo/benefício em termos de qualidade nas métricas da política, que não necessariamente é o caminho do *AS PATH*.

A abordagem de escolha do melhor caminho que ocorre internamente nos domínios, considerando os parâmetros da política de rede: latência entre os enlaces, permite sem perda de generalidade descrever o algoritmo proposto como um *Dijkstra* orientado a qualidade da rede. Os pesos que são atribuídos às arestas respeitam esta função: $y = f(x)$, onde $f(x) = \overline{X}$ que é a latência média aferida pelo processo *probe* entre pares de roteadores pertencentes ao caminho candidato.

A funcionalidade *OPI* do *NASOR* permite que no processo *probe* o tempo de aferição da latência, ou outra métrica como utilização do enlaçe, seja adaptável conforme o arquivo de descrição da fatia de rede. Para o experimento proposto a média da latência entre os pares de roteadores é do intervalo de 2 segundos. Nesse ponto, as duas abordagens em comparação: o algoritmo baseado na qualidade do enlace (o Algoritmo 2) e o caminho de dados do *OSPF* e *BGP* (o Algoritmo 1).

No que toca a aferição da experiência das aplicações sobre as fatias de rede, a métrica *jitter* para chamadas *VoIP* pôde ser observada à luz da Figura 26. Uma fatia de rede foi implantada considerando o caminho de dados proposto pelo *OSPF* e *BGP* e a outra considerou a abordagem de orientação à política de rede. Voluntariamente, no caminho de dados do *OSPF*, ainda que o de menor custo pela perspectiva de roteamento, foi inserido um mecanismo defletor de qualidade no *link* com vistas a simular a ocorrência de perdas de pacotes, latência e *jitter*.

Conforme apontado em Jurgelionis et al. (2011), a ferramenta *NetEm*[3] permite adicionar condições adversas em um *link* de rede, sua escolha se deu pela

[3] Ferramenta que emula a funcionalidade da rede com diferentes parâmetros (HEMMINGER et al., 2005).

boa acurácia dado parâmetros de entrada e a resposta comportamental da rede (VELÁSQUEZ; GAMESS, 2012). Portanto, os parâmetros de configuração arbitrariamente utilizados para degradar a qualidade do enlace entre os roteadores $R1$ e $R3$ foram: 30 ms de *jitter* e 20 ms de latência.

De acordo com a *RFC* 4689, *jitter* é a diferença entre o atraso de encaminhamento entre dois pacotes enviados consecutivamente pertencentes a um mesmo fluxo de transmissão (PORETSKY et al., 2006). Ele pode ser expresso pela equação: $|D(i) - D(i-1)|$, onde D é o atraso de encaminhamento e i é a ordem que são recebidos. A escolha da métrica *jitter* como parâmetro para qualidade de chamadas *VoIP* se dá pelo concesso na literatura da relação direta entre *jitter* e a qualidade da chamada (PAULSEN; UHL; NOWICKI, 2011; LI et al., 2003; ZHENG; ZHANG; XU, 2001).

Existem abordagens que dado uma amostra de *jitter*, elas conseguem classificar a qualidade da chamada, além disso, é muito discutido técnicas de aumento dos *buffers* para mitigar o efeito causado pelo *jitter* (TORAL-CRUZ; PATHAN; PACHECO, 2013; BARATVAND et al., 2008). Esse experimento não se ateve a explorar a influência do *jitter* no *Quality of Service (QoS)* ou *Quality of Experience (QoE)* das chamadas que ocorreram sobre as fatias de rede implantadas sobre os múltiplos domínios.

O objetivo deste experimento é responder as seguintes questões:

- Quanto é o tempo de implantação de uma fatia de rede utilizando a abordagem convencional, aquela que segue o caminho do plano de dados dos algoritmos de roteamento, comparado com uma abordagem que explora a característica *OPI* para definição de um caminho para fatia de rede sobre o cenário multidomínio da Figura 26?

- Existe um cenário que o caminho de dados da fatia de rede baseada algoritmos de roteamento não oferece desempenho satisfatório comparado com uma fatia de rede que implantada sobre um caminho alternativo, aquele que considera a qualidade dos enlaces?

4.5.2 Avaliação dos Resultados: Cenário Experimental 2

Para responder os questionamentos norteadores do experimento foram executadas requisições de implantações consecutivas que perfizeram 50 amostras. Ou seja, 50 requisições de implantação de uma fatia de rede foram disparadas contra o *NASOR* do Domínio B considerando a abordagem do caminho de dados dos algoritmos de roteamento; a outra considerando a *OPI*, disparada contra o *NASOR* do Domínio A.

O gráfico ilustrado na Figura 27 permite inferir que o tempo de implantação de uma fatia de rede, isto é, partindo do domínio B para o domínio A considerando a análise em tempo real das qualidade dos enlaces, foi em média de 12.27 ms sob uma margem de erro de 5%. Em contraste, a implantação *baseline*, que leva em conta o caminho de dados dos algoritmos de roteamento, isto é, partindo do domínio A para o domínio B não fazendo nenhuma análise em tempo real da qualidade dos enlaces foi de 1.07 ms, também sujeitas ao intervalo de confiança de 95%. O tempo de implantação considerado neste experimento refere-se exclusivamente ao processo de configuração dos parâmetros de conectividade para estabelecimento da fatia de rede entre múltiplos *ASs*, não se considera o tempo de implantação de instâncias virtualizadas no escopo de computação.

Quantitativamente é possível discutir a disparidade do tempo de implantação de uma abordagem em detrimento de outra. Ainda que o intuito do experimento seja avaliar a aplicabilidade da *OPI* como um mecanismo de estabelecer fatias de rede de forma mais dinâmica e próxima aos requisitos dos usuários, é importante discutir alguns aspectos matemáticos que influenciaram o desempenho das duas abordagens em discussão. O primeiro ponto de destaque é: a abordagem que considera uma política baseada na qualidade da rede, apresenta-se como mais próxima aos requisitos dos usuários e possui dois componentes que influenciam no tempo final de implantação. O primeiro componente é o tempo da avaliação que o mecanismo *probe* faz nos *links* entre os caminhos candidatos.

No caso desse experimento, que utiliza uma abordagem de orientação a qualidade de rede, ocorre aferição da qualidade do enlace para cada par de roteadores em um caminho candidato, especificamente é aferido a latência. Por se tratar de uma análise em tempo real, é necessário que haja um intervalo mínimo de tempo,

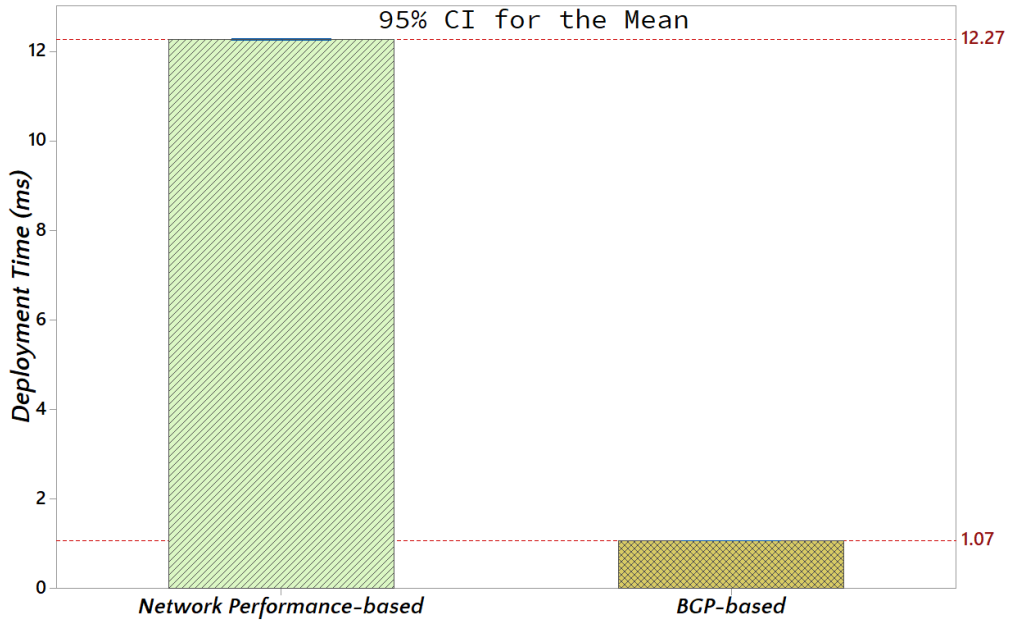

Figura 27 – Tempo de Implantação de fatias de rede considerando políticas diferentes.

ou seja, a mensuração da latência se dá quando há um instante inicial e final. Portanto, o arquivo de descrição da fatia de rede estabeleceu que o *NASOR* deverá avaliar no intervalo de tempo de 2 segundos.

Tempos variados poderiam ser definidos nos arquivos de descrição da fatia de rede, no entanto um tempo maior de avaliação do *link* representaria um tempo adicional constante $K \times n$, onde K é o lapso temporal da avaliação e n é o número de enlaces a serem avaliados. Outras métricas de rede também requerem análise de lapso temporal para possuírem valor semântico, como a vazão. Não há de se falar em quantidade de tráfego passante em um *link* sem que haja delimitadores temporais que especificam o período da amostra. O tamanho do tempo pode suscitar discussões a cerca da acurácia da medição, apontamentos no sentido de haver possibilidade de momentaneamente um *link* estar com rajadas de tráfego ou desproporcionalidade da operação média.

O fenômeno da desproporcionalidade ocorre quando há valores extremos que alteram o valor da média. Como exemplo, uma série que representa a utilização de um canal de dados ao longo de um intervalo de tempo, assumindo que o tráfego

cursado no período noturno, notadamente na madrugada, pode ser baixo baixo. Portanto, a operação média de utilização, ainda que contenha uma boa amostra, não representa a assertivamente a situação do enlace. Pode haver períodos do dia que a utilização da interface é próxima a sua totalidade, e isso implica em degradação na comunicação mesmo que isso não seja apresentado na média. Portanto, a análise de qualidade das métricas de desempenho de rede devem ser observadas pontualmente.

Alternativamente, para endossar o dinamismo no estabelecimento de fatias de rede que consideram métricas de rede podem valer de modelos matemáticos de previsão ou soluções de inteligência artificial para aprimorar o desempenho do serviço contratado. Neste experimento, foi observado o tempo *baseline*, que é o mecanismo basilar desta tese, qual seja: um mecanismo de estabelecimento de fatias de rede em cenários multidomínios sobre o plano de dados *BGP*. Ao explorar a *OPI* e comparar os mecanismos de estabelecimento de caminhos suportadas por ela com o *baseline*, nativo do *NASOR*, o fim que se pretende é adicionar e discutir o dinamismo nas operações de fatiamento de rede. Deve ser assumido que mecanismos de estabelecimento de fatias que melhores observam aspectos de desempenho de tempo ou espaço de memória podem ser considerados para o *NASOR*.

O segundo aspecto que impacta o tempo de implantação de uma fatia de rede é a natureza do algoritmo de escolhas de caminhos, isto é, uma função que dado o tamanho da entrada implica em uma quantidade de tempo para produzir a saída. Para este experimento, que explora a *OPI*, ocorrem duas operações matemáticas: (1) encontrar todos os caminhos dado uma origem s e um destino d; (2) ponderar a utilização de cada enlace ao longo de cada caminho candidato. No que toca o formalismo matemático, a implementação do Algoritmo 2 para retornar todos os possíveis caminhos é uma implementação modificada do algoritmo de busca em profundidade, disponível na ferramenta *NetworkX*[4]. Portanto, conforme a notação de complexidade de algoritmos, o pior caso para retornar os possíveis caminhos (no caso deste experimento: caminhos candidatos) é da ordem $O(n!)$ (SEDGEWICK, 2001). Além disso, cabe pontuar que é um algoritmo pouco eficiente para amostras

[4] "Pacote *Python* para criação, manipulação e estudo da estrutura e funções de redes complexas" (HAGBERG; SCHULT; SWART, 2005). No caso do *NASOR*, a estrutura de dados Grafo é manipulada pelo pacote por meio de funções implementadas.

n suficientemente grandes.

Em contraste, o estabelecimento de fatias de rede que consideram o caminho de dados construído pelos algoritmos de roteamento não experimentam essa sobrecarga de tempo adicional. O processo de estabelecimento da fatia de rede, quando não orientado à política de rede, caminha pelos elementos e interfaces do plano de controle dos roteadores e configura os parâmetros da fatia de rede. O tempo de computação de um caminho (redes alcançáveis) é uma tarefa operacional nativa dos algoritmos de roteamento, isto é, eles já fazem isso autonomamente.

Logo, o cálculo de caminhos bem como o plano de dados sobre os algoritmos de roteamento não influenciam no tempo do estabelecimento da fatia de rede pelo *NASOR*, dado que esses caminhos já estão calculados e sumarizados nas tabelas dos roteadores. Portanto, a disparidade aparente reportada pelo gráfico do experimento se fundamente nessa premissa, isto é, em uma política não é necessário computar o caminho ao longo dos *ASs*, na outra se faz necessário computar o caminho, dado que essa política considera o desempenho corrente de cada enlace.

Adicionalmente, o tempo de implantação nessa abordagem é influenciado pela operação de busca sequencial na tabela *FIB* dos roteadores ao longo do caminho de dados. Essa busca é necessária para o agente *NANO* conheça a interface de saída pela qual deverá configurar os parâmetros da fatia de rede e operar as demais configurações nos próximos roteadores. Em outras palavras, a abordagem convencional de estabelecimento de fatias do *NASOR* caminha sobre o plano de dados construído pelos algoritmos de roteamento.

No que diz respeito ao formalismo matemático, o processo do *NASOR* de realizar uma busca sequencial na tabela *FIB* do roteador é de ordem polinomial n_{FIB_Search}. No entanto, essa operação é realizada para cada roteador n_{Router}, portanto $n_{Router} \times n_{FIB\_Search} \equiv n^2$, cujo limite superior é $\mathcal{O}n^2$. No mesmo sentido, é importante pontuar a complexidade de tempo do algoritmo *OSPF* dos *ASs*, tem-se portanto $E \times \log V$, onde E são os *links* e V são os roteadores (DOSS, 2016; DASGUPTA, 2008). Além disso, o tempo do algoritmo depende da implementação das filas de prioridade do algoritmo, que é uma decisão de projeto do fabricante (CORMEN, 2009).

Então, mesmo que a quantidade de roteadores ao longo dos *ASs* aumente significativamente, a complexidade do Algoritmo 2 estará sujeita ao limite superior

deduzido, que é $\mathcal{O}n^2$. O intuito desse experimento não é propor um algoritmo ótimo para definição de caminhos para configuração dos parâmetros de uma fatia de rede. Antes, se fundamenta na discussão da aplicabilidade de uma interface aberta para o usuário/dono da fatia de rede definir caminhos da fatias de rede e com isso adicionar dinamismo no fatiamento de rede, uma vez que se observa na literatura ausência de soluções com essa abordagem.

Adicionalmente, o Algoritmo 2 caracteriza-se como guloso, logo uma abordagem mais eficiente é admissível, no entanto, conforme Gong et al. (2014), é um problema de classe *NP-hard* cujo mérito transcende o objetivo desta tese. A estatística descritiva do experimento que avalia o tempo de implantação está disponível no Apêndice E especificamente na Tabela 15.

Outro experimento, que avalia o impacto da escolha de um caminho para uma fatia de rede em detrimento outro na qualidade das aplicações, é reportado no gráfico da Figura 28. Para este experimento, existem somente dois caminhos possíveis para o estabelecimento da fatia de rede, o que percorre os roteadores $R1$, $R3$ e $R4$ e o caminho $R1$, $R2$, $R3$ e$R4$. Adicionalmente, pelo gráfico é possível compreender que existem duas grandezas de *jitter* a serem analisadas, uma média de 23.60 ms e outra de 1.1 ms.

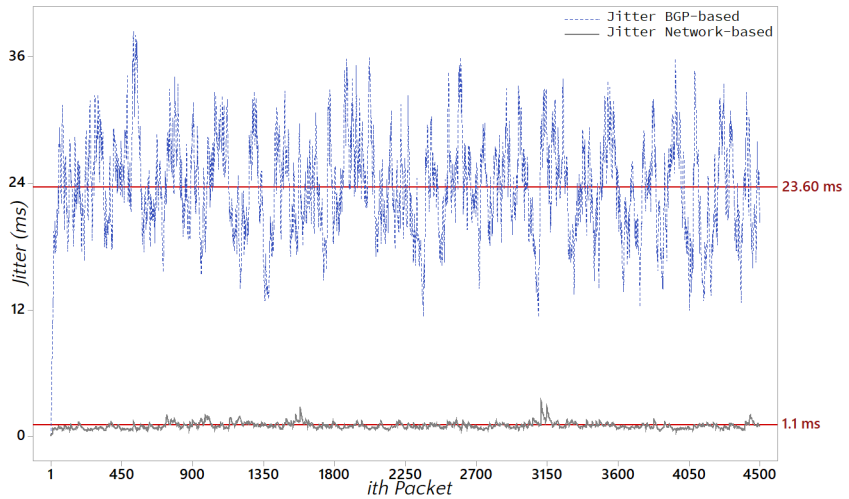

Figura 28 – *Jitter* experimentado pela aplicação *VoIP* sobre fatias de rede implementadas sobre duas modalidades de escolhas de caminhos.

Foram transportados 4500 pacotes *UDPs* contendo *chunks* de voz processados pelo *codec G711a* (HIWASAKI; OHMURO, 2009) . O *jitter* de 23.60 ms foi observado na fatia de rede de coloração verde, conforme cenário experimental ilustrado na Figura 26. Essa fatia de rede foi implantada pelo *NASOR* considerando sua política padrão, isto é, a baseada no caminho de dados dos algoritmos de roteamento. Entre os roteadores $R1$ e $R3$, foi voluntariamente adicionada uma degradação na qualidade do enlace, de forma que, para um dado volume de tráfego a latência percebida é em média de 20 ms, o *jitter* de 30 ms ocorrendo conforme distribuição normal e com uma correlação de 20% do pacote atual com o anterior.

A grandeza de *jitter* aferida na fatia de rede vermelha foi na média 1.1 ms. Essa fatia de rede, foi implantada conforme o mecanismo que utilizou a *OPI*, consequentemente observou a qualidade dos enlaces antes da fatia de rede ser implantada. Após aferir a qualidade dos enlaces nos caminhos candidatos e, conforme planejado, o melhor caminho aferido pelo mecanismo *OPI* foi o que percorreu os roteadores $R1$, $R2$, $R3$ e $R4$ consecutivamente. Então, a aplicação *VoIP* sobre essa fatia de rede experimentou um *jitter* de 1.1 ms. Na literatura são sugeridos limites de *jitter* aceitáveis para que uma chamada de voz seja além de audível e compreensível (SKORIN-KAPOV et al., 2004; FLUCKIGER, 1995; DASH; DURRESI; JAIN, 2003).

É importante pontuar que essas fatias de rede, sobre as quais a aplicação *VoIP* transportou seus dados, seguiu o mesmo rito de implantação do experimento anterior. Assim, a fatia de rede vermelha é escolhia após a avaliação exaustiva da qualidade de todos os enlaces dos dois caminhos candidatos. Além disso, a fatia de rede vermelha conforme visto anteriormente demandou mais tempo, pois foram avaliadas a qualidade de todos os enlaces da topologia, quais sejam: $R1 \leftrightarrow R2$, $R2 \leftrightarrow R3$, $R3 \leftrightarrow R4$, $R1 \leftrightarrow R3$ e $R3 \leftrightarrow R4$ respectivamente.

Assim, a avaliação quantitativa e qualitativa da funcionalidade *OPI* permite fazer algumas asserções. Primeiro, que adicionar dinamismo no estabelecimento de fatias de rede é estabelecer uma contribuição no estado da arte. Uma vez que as soluções disponíveis não provêm esse dinamismo, sobretudo na modalidade multidomínios. Segundo, porque dependendo do objetivo da fatia de rede, o caminho de dados baseado nos algoritmos de roteamento pode eventualmente não ser a melhor decisão. É certo que existem abordagens para o *OSPF* que fazem balanceamento

de tráfego, multicaminhos e outras que lidam com cenários de pesos iguais para vários caminhos possíveis. Mas o âmago da discussão é a materialização de um mecanismo para fatiamento de rede que abre ao usuário/administrador do domínio a possibilidade de se definir, previamente, o caminho da fatia de rede conforme uma política descrita no arquivo *NSTD*.

A estatística descritiva do Experimento de Latência de Aplicações sobre as fatias de rede sobre múltiplos domínios pode ser consultada no Apêndice E.

4.6 Cenário Experimental 3: Escalabilidade e Especialização de Requisitos em Fatias de Rede

Dado que o *NASOR* implanta fatias de rede entre múltiplos *ASs*, esta seção descreve um cenário adicional, que explora avanços no processo de implantação de fatias de rede. Neste cenário, o mecanismo original do *NASOR* é aprimorado para suportar a implantação de fatias de rede que ofereçam baixa latência. No contexto de fatiamento de rede, aplicações que requerem baixa latência são nomeadas na literatura como *Ultra-reliable low-latency communication (URLLC)* (POPOVSKI et al., 2018). Ocorre que ultra-confiabilidade e baixa latência são conflitantes, porque satisfazer uma taxa de transmissão com sucesso requer alocar mais recursos, que eventualmente pode comprometer a latência (SORET et al., 2014; ELBAMBY et al., 2018). Assim, este cenário dedica-se a explorar as capacidades do *NASOR* para implantar fatias de rede que cumprem requisitos de baixa latência. O requisito de ultra-confiabilidade é melhor abordado por soluções de redes auto-organizáveis e redes auto-sustentáveis (SAAD; BENNIS; CHEN, 2020).

Para este cenário experimental, foi adicionado o habilitador tecnológico *BSD Packet Filter (BPF)* ao *framework NASOR*. O *BPF* é um método, proposto em 1993, para filtragem de pacotes que podem ser capturados nos registradores *buffer* da placa de rede (MCCANNE; JACOBSON, 1993). Essa estratégia de captura permite que aplicações ouvintes recebam uma cópia do pacote entregue ao *driver* do dispositivo de rede, permitindo uma performance superior a abordagem que utiliza a pilha de protocolos implantada no *kernel*, uma vez que o processamento do pacote ocorre precocemente. Posteriormente, é possível inserir instruções para

o processamento do pacote expressas em linguagem de alto nível.

O *BPF* evoluiu para suportar processamento de pacotes em vários níveis do *kernel* como na camada *Traffic Control (TC)*, *Netfilter* e na camada *eXpress Data Path (XDP)*. O *XDP* é uma abordagem programável que permite inserção de códigos de alto nível como linguagem *C* para instruir o processamento precoce de pacotes (HØILAND-JØRGENSEN et al., 2018). Essa técnica possui restrições relacionadas a estrutura da codificação, que não pode conter *loops*, limite de pilha a 512 *bytes* e variáveis globais não estáticas. No tempo de compilação é verificada a corretude do código e a adequabilidade com padrão da máquina *BPF*.

Para tornar o *NASOR* capaz de implantar fatias de rede que suportam baixa latência entre múltiplos domínios foi proposto o mecanismo *XCP*. O *XCP* é um mecanismo proativo que instala regras de encaminhamento em mapas *XDP*. Um mapa *XDP* é uma estrutura de dados em memória onde as entradas armazenam regras que processam os pacotes conforme seus cabeçalhos. Na Figura 29 está ilustrado a arquitetura conceitual do *XCP* onde são evidenciadas as interações e os componentes do mecanismo. Na parte inferior está o *hardware* de rede, que contém o *driver* em paralelo os *buffers* de transmissão e recepção.

Quando um pacote ingressa na placa de rede, logo após admitido, são alocados *buffers*. Na abordagem *XDP* o pacote é processado conforme regras especificadas por um programa escrito em linguagem de alto nível. Esse programa pode especificar encaminhamento, alteração de campos do cabeçalho, promover descarte de pacotes dentre outras possibilidades. Caso o pacote recebido não possa ser processado pelo programa *XDP*, ele é encaminhado para a pilha de protocolos de rede que o *kernel* do sistema operacional implementa. Nesse cenário, o pacote será processado pelos mecanismos tradicionais do sistema operacional e será entregue à aplicação por demultiplexação no nível do *socket*.

Caso o programa *XDP* seja capaz de processar o pacote, ele será imediatamente encaminhado para o *buffer* de transmissão e seguirá seu destino para a interface correspondente. A decisão de encaminhamento do pacote baseia-se em regras que são armazenadas em um mapa. O mapa pode ser atualizado pelo nível do usuário, desta forma havendo um mecanismo com visão global que povoe o mapa com regras adequadas de encaminhamento é possível estabelecer um plano de dados para uma fatia de rede oferecendo baixa latência.

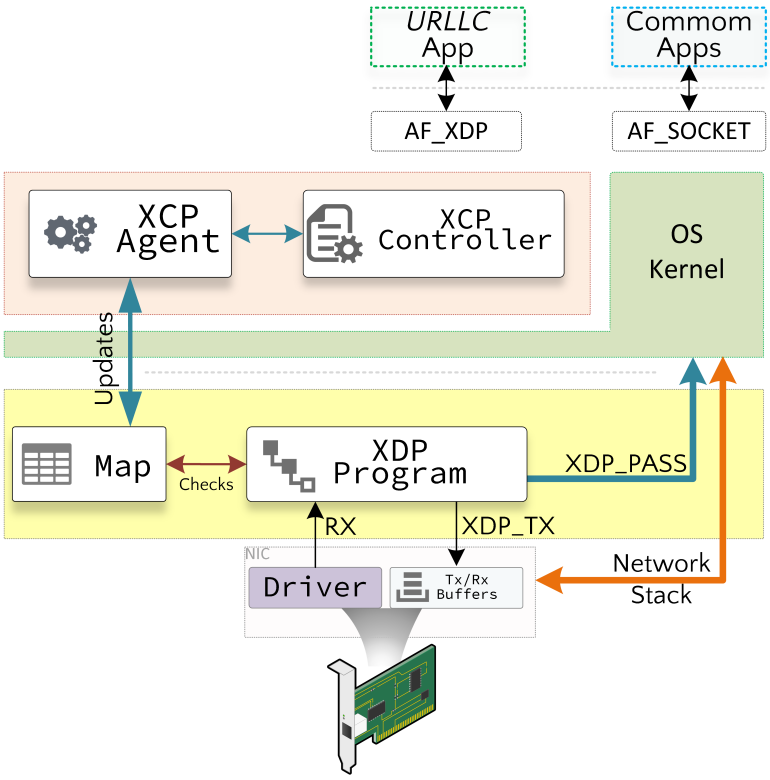

Figura 29 – Arquitetura do *XCP*.

Fonte: Moreira et al. (2021b)

Neste cenário experimental, o *NASOR* interage com o *XCP* para povoar mapas *XDP* e realizar a conectividade lógica para uma fatia de rede. Na Figura 30 está esquematizado o processo global de estabelecimento de fatiamento de rede *inter-AS* e o relacionamento do *NASOR* com o *XCP*. A descrição técnica da fatia de rede é recebida pelo *NASOR* de um domínio específico, a origem da fatia de rede, que divide a especificação em recursos computacionais e de rede. A parte computacional é direcionada para um *MANO* Local que procederá com a criação e operacionalização dos serviços. A configuração e os parâmetros da fatia de rede são especificadas no descritor e é encaminhada para o *NANO* designado para a implantação da fatia de rede. Assincronamente, se o descritor do serviço constar uma fatia de rede entre múltiplos *ASs*, o *NASOR* encaminhará para seu par situado em outro *ASs*.

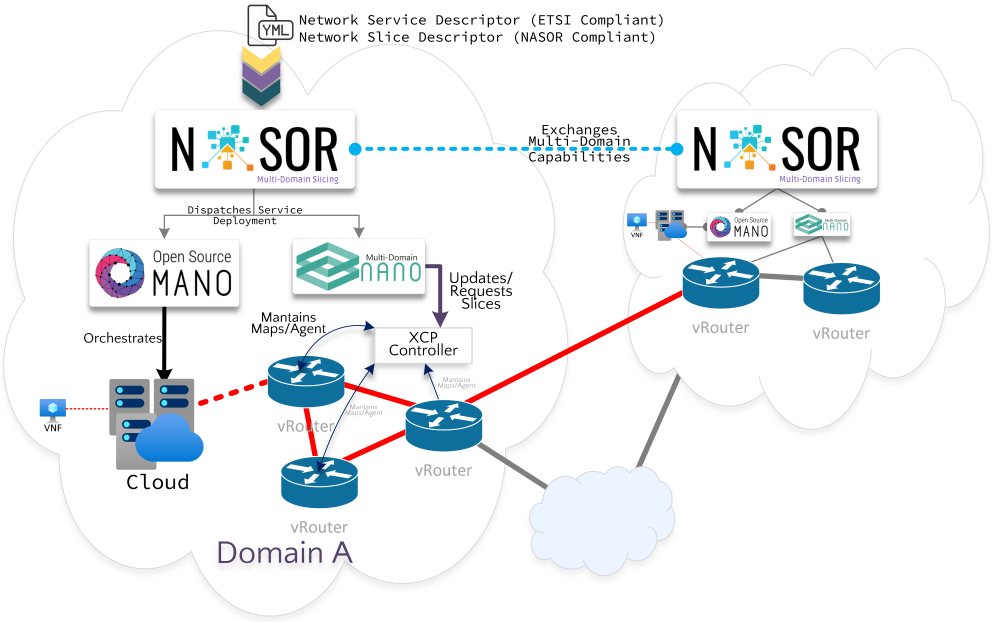

Figura 30 – Combinando o *NASOR* com o Controlador *XCP*.

Fonte: Moreira et al. (2021b)

O *NANO* designado para implantar uma fatia de rede interage com o *XCP Controller* entregando-lhe via *API* as especificação das interfaces que povoarão o mapa *XDP*. Dado que o *NASOR* atua como um *MANO* Multidomínio, ele possui a topologia do seu domínio, assim é possível pavimentar um plano de dados até o roteador de borda. Essa pavimentação é materializada pela configuração das regras de encaminhamento dentro dos mapas *XDP* para habilitar o processamento precoce dos pacotes e, em hipótese, oferecer uma fatia de rede com baixa latência.

4.6.1 Método de Avaliação: Cenário Experimental 3

O método de avaliação para o cenário experimental três considerou *baselines* de performance para verticais de aplicações e *Key Performance Indicator* (*KPI*) no fatiamento de rede (KUKLINSKI; TOMASZEWSKI, 2019). De acordo com a *Ericsson*, algumas *baselines* de performance são desejáveis para algumas aplicações, como de controle de veículos autônomos que requerem latência máxima

de $5ms$, automação de fábricas requerendo latência abaixo de $1ms$, media sob demanda com $200ms$, trens de alta velocidade $10ms$ e outras (ERICSSON, 2017).

Considerando as métricas de *KPI* para fatiamento de rede, são reconhecidas: integridade, vazão, consumo de energia e segurança (CUNHA et al., 2020). Neste cenário experimental, foi considerado a *KPI* de integridade que versa sobre latência fim a fim (3GPP, 2019). Avaliar a latência fim a fim é significativo porque neste cenário experimental as entidades dispostas em *ASs* distintos são conectadas por uma fatia de rede que perpassa *ASs* intermediários, ralizando uma conectvidade fim a fim.

O *testbed* ilustrado na Figura 31 refere-se à conectividade entre dois sistemas finais que é configurada pelo *NASOR* em conjunto com o *XCP*. Logo, o plano de dados que permite que pacotes fluam de uma origem a um destino é implementado sobre a tecnologia *XDP*. Este *testbed* foi configurado em um ambiente de máquinas virtuais que desempenharam o papel de roteador da Internet. Cada máquina virtual denominada *vRouter* executou o *Quagga Software Routing Suite* (JAKMA; LAMPARTER, 2014). O conjunto de *vRouters* foram gerenciados pelo monitor de máquina virtual *VirtualBox* e os *vRouters* foram executados sobre uma máquina hospedeira Intel *Core* I7-7700HQ com $24GB$ de *RAM*.

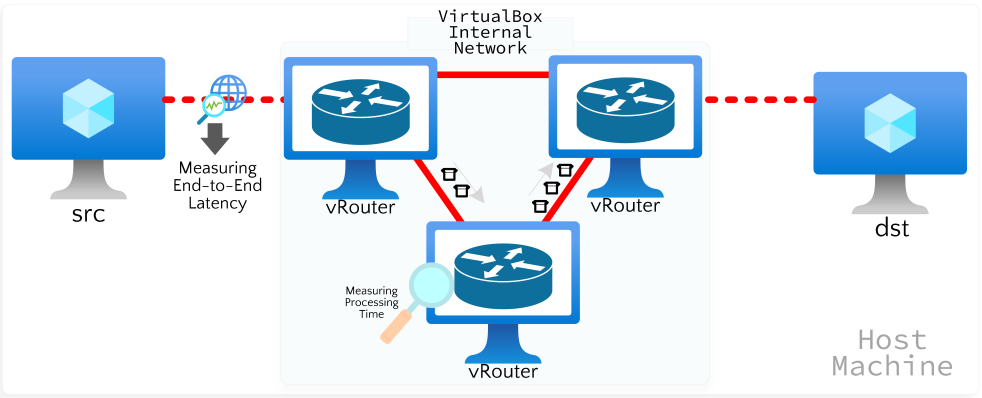

Figura 31 – Topologia do *testbed* experimental.

Fonte: Moreira et al. (2021b)

Sobre esse *testbed* ilustrado na Figura 31, foi proposto uma avaliação experimental sob o método fatorial parcial. Esse método permite análises através de

um modelo de regressão. Esse método de avaliação permitiu compreender a performance, estabilidade e escalabilidade do fatiamento de rede implantado pelo *NASOR* sobre tecnologia *XDP*. O método fatorial parcial considera k fatores com n_i níveis para cada fator i, e avalia a interação e a influência desses fatores em variáveis de resposta.

Foram designadas duas variáveis de resposta: tempo de processamento do pacote (PT) e latência (L). Para avaliar o PT em cada *vRouter*, foi capturado utilizando a ferramenta *Wireshark* a marca de tempo dos pacotes na fila de entrada e na fila de saída. Com as marcas de tempo de cada pacote nas filas de ingresso e egresso, foi possível mensurar o tempo de processamento de um pacote, uma vez que a diferença de tempo S é alcançada subtraindo $PacketOutputTime_{\beta} - PacketInputTime_{\alpha}$. Essa técnica de mensuração de tempo de processamento de pacote, baseada em filas de ingresso e egresso, já é aceita pela comunidade (SAAD; BENNIS; CHEN, 2020).

Nesta avaliação experimental, foi garantido que os *vRouters* não transmitissem dados relacionados a quaisquer aplicações. O *Wireshark* utilizou como referência o tempo a chegada do primeiro pacote nas filas de entrada e saída de uma mesma placa de rede. Com vistas a acurácia, a escala de tempo utilizada neste experimento foi a nanosegundos (10^{-9}).

A configuração da avaliação de performance, incluindo os fatores e seus níveis, estão descritos na Tabela 4. Depreende-se desta tabela dois fatores: o *Flavor* do *vRouter* e o tamanho do mapa da fatia de rede. Esses fatores foram variados em dois níveis respectivamente: 2 *vCPU* com 2 *GB RAM* e 4 *vCPU* com 4 *GB RAM*.

Tabela 4 – Configuração da Avaliação Experimental.

Experimento	**vRouter Flavor**	**Slice Map Size**	**vRouter Flavor Factor**	**Slice Map Size Factor**
#1	2 vCPU 2 GB	Pequeno (1 Fatia de Rede)	-1	-1
#2	4 vCPU 4 GB	Pequeno (1 Fatia de Rede)	1	-1
#3	2 vCPU 2 GB	Grande (100 k Fatias de Rede)	-1	1
#4	4 vCPU 4 GB	Grande (100 k Fatias de Rede)	1	1

Usando o modelo de regressão, foram conduzidos quatro experimentos organizados conforme a Tabela 4. O modelo de projeto experimental 2^2 é dado por uma equação $y = q_0 + q_A X_A + q_B X_B + q_{AB} X_{AB}$. Substituindo as quatro observações no modelo, os valores q_0, q_A, q_B e q_{AQ} são obtidos de acordo com as seguintes equações:

$q_0 = \frac{1}{4} \times (y_1 + y_2 + y_3 + y_4)$, $q_A = \frac{1}{4} \times (-y_1 + y_2 - y_3 + y_4)$, $q_B = \frac{1}{4} \times (-y_1 - y_2 + y_3 + y_4)$, $q_{AB} = \frac{1}{4} \times (y_1 - y_2 - y_3 + y_4)$. Com essas substituições garante-se que todos os fatores e níveis experimentados mutuamente pelo menos uma vez.

Assim, para os valores q_0, q_A, q_B e q_{AB} é possível determinar a Soma dos Quadrados (SS). A soma dos quadrados dará a variação total das variáveis de resposta e as variações devido a influência de algum fator específico A ou B, e da interação entre A e B. Assim, ao calcular a variância total de y ou Soma dos Quadrados Total (SST), onde $SST = \sum_{i=1}^{2^2} (y - \bar{y})^2$ ou $SST = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$ será possível mensurar proporcionalmente o peso de cada variável.

Para isso, toma-se a soma dos quadrados de A, pela equação $SSA = 2^2 q_A^2$ dividindo-a por SST como $\frac{SSA}{SST}$ e se obtêm a influência isolada do Fator A na variável de reposta. De forma análoga, toma-se a soma dos quadrados de B, pela equação $SSB = 2^2 q_B^2$ como $\frac{SSB}{SST}$ e se obtêm a influência do Fator B. Por fim, toma-se a soma dos quadrados de AB, pela equação $SSAB = 2^2 q_{AB}^2$ como $\frac{SSAB}{SST}$ e se mensura a interação conjunta dos dois fatores A e B na variável de resposta.

O conjunto de experimentos executados, conforme organização da Tabela 4 e seguindo a organização fatorial parcial, objetivaram responder as seguintes questões:

- ❏ Considerando o roteamento por seguimentos baseado na tecnologia *SREXT* (ABDELSALAM et al., 2017) comparado com a abordagem baseada em *XDP* usando redirecionamento *REDIRECT* e *REDIRECT_MAP*, quanto é o *overhead* no tempo de processamento dos pacotes?

- ❏ Qual tecnologia de fatiamento de rede do *framework NASOR* oferece maior estabilidade?

- ❏ Considerando a técnica de encaminhamento baseada em mapas, qual fator é preponderante na latência experimentada (L) e no tempo de processamento PT de um pacote em uma fatia de rede?

4.6.2 Avaliação dos Resultados: Cenário Experimental 3

Para responder as perguntas motivadoras do cenário experimental 3, foram implantadas fatias de rede por meio do *framework NASOR* conforme ilustrado no

testbed da Figura 31. Após iniciar o utilitário *ping*, foram coletadas 30 médias, onde cada média continha um espaço amostral de *pings* do protocolo *Internet Control Message Protocol Version 6 (ICMPv6)*. Os *pings* estavam encapsulados dentro de um pacote *SRH* entre uma origem (*src*) e um destino (*dst*) participantes da fatia de rede criada pelo *NASOR*. Na Tabela 5 está descrito a estatística descritiva do experimento.

Tabela 5 – Estatística descritiva do experimento Tempo de Processamento *PT*.

Variável	Média (μs)	Erro Médio Padrão	Desvio Padrão	Variância	Mediana (μs)
REDIRECT_MAP	471.12	5.10	54.64	2985.80	469.48
SREXT	482.52	4.87	73.81	5448.57	490.13
REDIRECT	490.42	5.07	64.27	4131.14	483.43

A variável *REDIRECT_MAP* refere-se ao método de encaminhamento de um pacote na camada *XDP*. Essa abordagem permite que um pacote seja encaminhado, conforme regras programadas em alto nível, para uma interface específica. A variável *REDIRECT* referes-se ao encaminhamento padrão na camada *XDP*, que considera somente uma interface fixa. Esse modelo foi considerado como *baseline* da tecnologia *XDP*. Para essas três variáveis, consolidou-se a soma das médias dos tempos de processamento dos pacotes considerando duas abordagens. A primeira, é baseada no processamento dos pacotes *SRH* que ocorre no *kernel*. A segunda, baseada no processamento precoce de pacotes *SRH* que ocorrem na camada *XDP*.

Como se vê no gráfico ilustrado na Figura 32, o *PT* médio dos pacotes utilizando *SREXT* foi 490.133 µs. Esse *PT* é maior para as fatias de rede implantadas sobre *SREXT* do que nas fatias implantadas sobre a tecnologia *XDP*, estritamente considerando o encaminhamento do tipo *REDIRECT_MAP*. Assim, pode-se perceber um *overhead* de processamento menor na abordagem baseada em *XDP*. Tempos de processamento menores são mais adequados para fatias de rede que requerem baixa latência. Portanto, fatiamento de rede sobre tecnologias de processamento precoce de pacotes mostram-se vantajosas na realização de requisitos de baixa latência.

Ao utilizar o encaminhamento de pacotes com a tecnologia *REDIRECT*, o tempo de processamento foi 490.418 µs em contraste com a tecnologia de processamento de pacotes baseada em *SREXT* que foi de 482.525 µs.

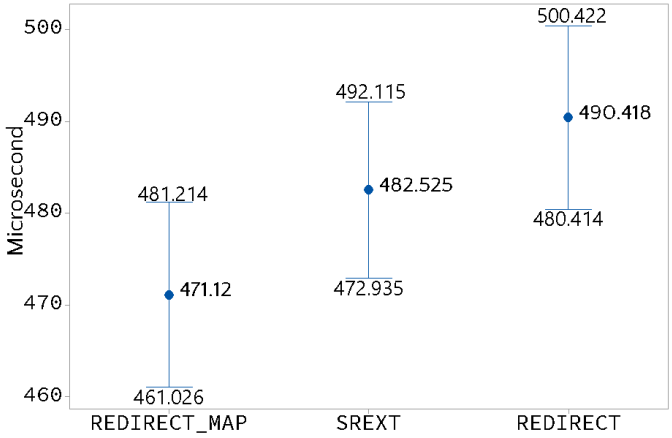

Figura 32 – Tempo de processamento (PT) médio.

Considerando um intervalo de confiança de 95% é sugestivo perceber que os métodos de processamento de pacotes *REDIRECT_MAP* e *SREXT* podem ser estatisticamente equivalentes.

Para validar essa equivalência foi conduzido um teste de hipóteses Z a um nível de confiança de 95%. As hipóteses conjecturadas são H_0 : a média dos tempos de processamento com fatias de rede baseadas em *REDIRECT_MAP* é igual (=) a $482.525\mu s$ e H_a : a média dos tempos de processamento com fatias de rede baseadas em *SREXT* é menor (<) que $482.525\mu s$. Constatou-se um $Z_{observado} = -1.66$ e um $Z_{crítico} = -1,64$, sugerindo que $Z_{observado}$ está dentro da região crítica levando a rejeição da hipótese H_0.

Logo, com rigor estatístico, é possível admitir que os tempos de processamento de fatias de rede baseadas em *REDIRECT_MAP* são inferiores a abordagem baseada em *SREXT*. Assim, para fatias de rede que requerem baixa latência, o processamento precoce baseado em mapas mostra-se adequado.

A Figura 33 permite verificar a variabilidade nos tempos de processamento considerando os três métodos de processamento nodal dos pacotes em uma fatia de rede. Nota-se que a variabilidade no PT é menor considerando a tecnologia *REDIRECT_MAP* em contraste com a *SREXT* e *REDIRECT*. Assim, é possível perceber que 75% dos *PTs* experimentados pelos pacotes da fatia de rede sob a tecnologia *REDIRECT_MAP*, que é baseada em *XDP*, correspondem a tempos

inferiores a 492.121 µs enquanto que os pacotes na fatia de rede baseada no *SREXT* foram de 516.278 µs.

Figura 33 – Comparação da variabilidade no tempo de processamento (PT).

Quanto a variabilidade e conforme a Figura 33, o PT que os pacotes transportados por uma fatia de rede baseada em *XDP* obteve foi de 21.211% menos variabilidade do que a abordagem de fatiamento tradicional do *NASOR*, que é *SREXT*. Portanto, fatias de rede baseadas em *XDP*, especialmente com tecnologia *REDIRECT_MAP*, experimentaram maior estabilidade no transporte e processamento dos pacotes conforme a diferença entre os quatis $Q3$ e $Q2$ das duas abordagens.

Esse comportamento era empiricamente esperado porque o processamento de pacotes do tipo *SRH* na abordagem *SREXT* ocorre dentro do espaço do *kernel* do sistema operacional. Os pacotes atravessam a pilha de rede do sistema operacional, resultando em um *overhead* maior. Por outro lado, considerando a abordagem de processamento precoce de pacotes, nota-se um *overhead* de processamento menor.

Dentre as razões que justificam a superioridade de performance do encaminhamento nodal baseado em *REDIRECT_MAP* em contraste com *REDIRECT* incluem os melhoramentos na implementação e na modificação do método de encaminhamento dos pacotes. Conforme descrito na Tabela 5, que apresenta a estatística descritiva deste experimento, o PT médio nas abordagens *XDP* são 471.12 µs para a baseada em *REDIREC_MAP* e 482.52 µs para a baseada em *SREXT*, ambas sujeitas a um erro máximo de 5%.

Os experimentos realizados permitem constatar que o fatiamento de rede considerando o método de encaminhamento *REDIRECT_MAP* supera o método de encaminhamento *REDIRECT*. Nesse sentido, a abordagem do *REDIRECT_MAP* demanda 3.93% menos PT do que a abordagem *REDIRECT*. Além de prover uma performance melhor, a abordagem de encaminhamento baseada em mapas permite instalar regras para alcançar comportamentos sofisticados no processamento precoce do pacote implicando na realização de uma fatia de rede de baixa latência. Portanto, incorporar o *XCP Controller* no *framework NASOR* para implantação de fatias de rede estende o estado da arte como um aprimoramento no processo de implantação e como um direcionador para futuras contribuições em técnicas de fatiamento de rede.

Para verificar a escalabilidade e estabilidade da abordagem *NASOR* para implantação de fatias de rede entre múltiplos *ASs* foi conduzido uma avaliação de performance do tipo fatorial parcial. Essa avaliação de performance visou mensurar quais aspectos mais influenciam no PT de um pacote e na latência L experimentada por uma aplicação que roda sobre uma fatia de rede baseada em *XDP*. Nessa avaliação foi considerada a abordagem tradicional de processamento dos pacotes, aquela baseada no *SREXT*, e a abordagem de implantação de fatias de rede sobre a tecnologia *XDP*.

A execução e replicação dos experimentos considerou o *testbed*, metodologia e configuração apresentada na subseção 4.6.1. Foram consideradas duas variáveis de resposta, PT e L e a quantidade padrão de recursos dos roteadores (*vRouter Flavor*) e número de fatias de rede configuradas no mapa (*Slice Map Size*) como os fatores. Esses fatores foram variados e combinados um a um conforme o método de avaliação de performance fatorial parcial (subseção 4.6.1). Após conduzir 30 experimentos, compilou-se as médias alcançadas por cada experimento conforme apresentado na Tabela 6.

De acordo com a Tabela 7, é possível inferir que o fator que empiricamente mais influenciou L foi *Slice Map Size* que suporta o encaminhamento de pacotes para cada fatia de rede, com uma influência de 36.24% na variável de resposta. O segundo fator que mais influenciou L foi *vRouter Flavor*, com uma influência de 32.76% na variável de resposta L. A variação percentual dos dois fatores é significativa, cerca de 10.64%. Além disso, a combinação dos dois fatores, isto é, A

Tabela 6 – Avaliação de Performance: médias gerais dos experimentos.

	Fatores		**Variável Dependente**	
Experimento	**vRouter Flavor**	**Slice Map Size**	**Latência (*L*) (ms)**	***PT* (segundos)**
#1	-1	-1	0.5135	0.000471120
#2	1	-1	0.5013	0.000496615
#3	-1	1	1.4141	0.000469149
#4	1	1	0.5365	0.000162345

e B simultaneamente representam também um percentual significativo na variável de resposta L com 30.99%.

Tabela 7 – Avaliação de Performance: Latência (L) e Tempo de Processamento (PT).

	Parâmetros				**Soma dos Quadrados**			
	q_0	q_A	q_B	q_{AB}	SS_A	SS_B	SS_{AB}	SST
Latência (*L*)	0.7413375	-0.2224425	0.2339625	-0.2163575	0.197922663	0.218953806	0.187242271	0.60411874
				Influência (%)	**32.76%**	**36.24%**	**30.99%**	
Tempo de Processamento (*PT*)	0.000399807	-0.000070327	-0.000084060	-0.000083075	0.000000020	0.000000028	0.000000028	0.000000076
				Influência (%)	**26.15%**	**37.36%**	**36.49%**	

Considerando o PT, quando se analisa os resultados da avaliação de performance, é possível constatar empiricamente que o fator que mais influenciou PT é também o *Slice Map Size*. A interação entre os fatores *vRouter Flavor* e *Slice Map Size* representou o segundo maior percentual de influência na variável de reposta, representando 36.49% de influência. Observando isoladamente o fator *vRouter Flavor*, é possível constatar que a variação dos seus níveis exerceu 26.15% de influência em PT.

É possível inferir que o fator *Slice Map Size* isolado em contraste com as interações simultâneas dos outros fatores é próxima de equidistante na influência de PT, com uma diferença de 2.38%. Além disso, é possível perceber a estabilidade do mecanismo de fatiamento de rede sobre a tecnologia de processamento precoce de pacotes *XDP* uma vez que há homogeneidade na influência percentual dos fatores nas variáveis de reposta.

Ao considerar um dos níveis do fator *Slice Map Size*, Grande – com cerca de 100 mil fatias de rede, PT médio e a latência L experimentada sobre um único canal, é possível admitir que o método de fatiamento de rede proposto pelo *framework NASOR* provê escalabilidade e estabilidade na conectividade. Além disso,

o Cenário Experimental 3 demonstra a aplicabilidade do *NASOR* na realização de fatiamento de rede para aplicações com requisitos de baixa latência.

4.7 Cenário Experimental 4: Integrando o *NASOR* com Técnicas de IA através da *OPI*

Discussões sobre a aplicabilidade da *OPI* foram descritas na Seção 4.4, nesta seção será explorado um caso de uso que combina técnicas de inteligência artificial com o *framework NASOR* para aprimorar o estabelecimento de fatias de rede entre múltiplos *ASs*.

A *OPI* é um conjunto de métodos que permitem que implementações terceiras sejam acopladas ao *NASOR* para prover funcionalidades específicas. A Figura 34 ilustra combinação de componentes terceiros com o *framework NASOR*. O relacionamento e interação ocorre por meio de uma interface *RestAPI*, que habilita a trocas assíncronas de mensagens contendo a topologia de um domínio e os parâmetros de rede relacionados aos *links*. Por meio da *OPI* é possível que implementações ou métodos terceiros operem sobre uma estrutura de dados grafo e defina um caminho onde a fatia de rede será implantada.

Aumenta-se o dinamismo no processo de estabelecimento de fatias de rede entre múltiplos *AS* ao combinar o *framework NASOR* com técnicas de inteligência artificial. O modo de operação tradicional do *NASOR* implanta fatias de rede sobre o plano de dados criados pelos algoritmos de roteamento, mas neste cenário a escolha do caminho para uma fatia de rede será computada baseada no tipo de aplicação que predominantemente transporta pacotes em um canal. Assim, por meio da *OPI* foi acoplado o mecanismo *Packet Vision* (MOREIRA et al., 2020), que é baseado em *Convolutional Neural Network* (*CNN*), Rede Neural Convolucional.

As redes neurais convolucionais são um método de aprendizado profundo e utilizam redes neurais multicamadas para aprender características e classificar em diferentes estágios. Essa classificação ocorre em tempo de execução e não requer extração manual de características (GOODFELLOW; BENGIO; COURVILLE, 2016). Todas as redes neurais profundas permitem extrair representações de dados de alto nível por meio do processamento de imagens multi-estágio (PONTI et al.,

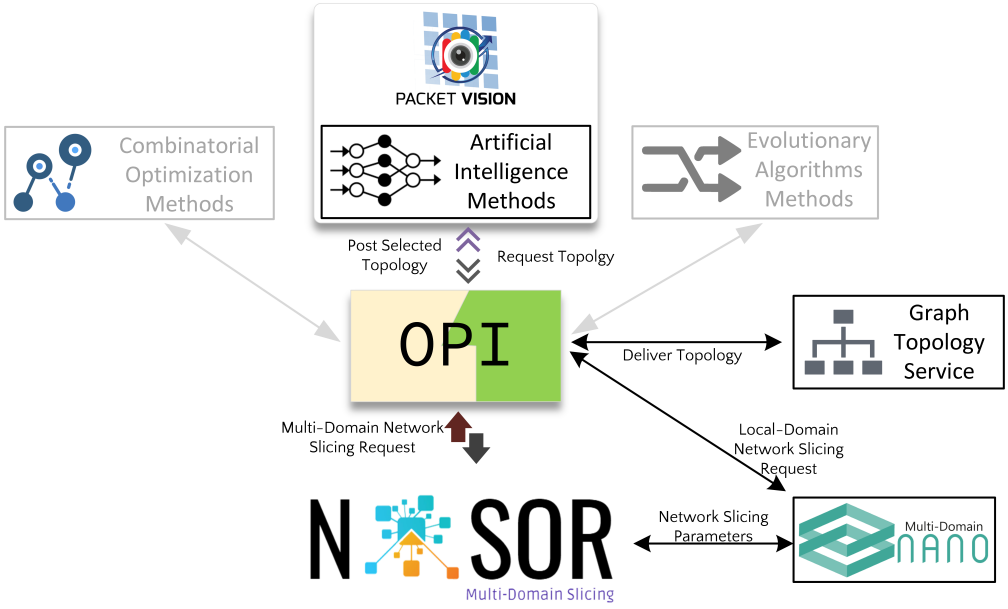

Figura 34 – Habilitação de aplicações terceiras no *NASOR* através da *OPI*. Moreira et al. (2021a).

2017).

Antes de o mecanismo de classificação de tráfego estar apto a atuar como componente auxiliar do processo estabelecimento de fatias de rede do *NASOR*, são requeridas algumas premissas. A primeira é possuir um método de classificação de tráfego capaz de receber cópia dos pacotes correntes na rede ou uma amostra para proceder com a classificação. A segunda é possuir um método de classificação adequadamente treinado dentro de um sistema cliente-servidor que receba uma estrutura de dados do tipo grafo, que representa toda a topologia do domínio, e retorne um caminho escolhido baseado na classe de tráfego predominante.

Para endereçar a primeira premissa, escolheu-se o método de aprendizado de máquina supervisionado: redes neurais convolucionais. Esse tipo de classificador é especializado em classificação de imagens, sendo assim amplamente utilizado em contextos de classificação de imagens médicas (ZHOU et al., 2021; SHEN; WU; SUK, 2017). Todavia, para se apropriar das capacidades de acurácia e desempenho que essa tecnologia oferece, foi proposto o método *Packet Vision* que transforma

um pacote de rede em uma imagem.

O método *Packet Vision* é composto de seis passos conforme representado na Figura 35. O primeiro refere-se a coleta dos pacotes de redes que são transportados em uma interface de rede. A aplicação *Wireshark* e suas bibliotecas de extensão permitem a coleta do pacote de rede sem afetar o seu processamento e trânsito nas pilhas do sistema operacional. O segundo passo consiste em transformar o pacote que é representado por um *array* de *bytes* contendo valores hexadecimais em uma matriz de tamanho $n \times 8$.

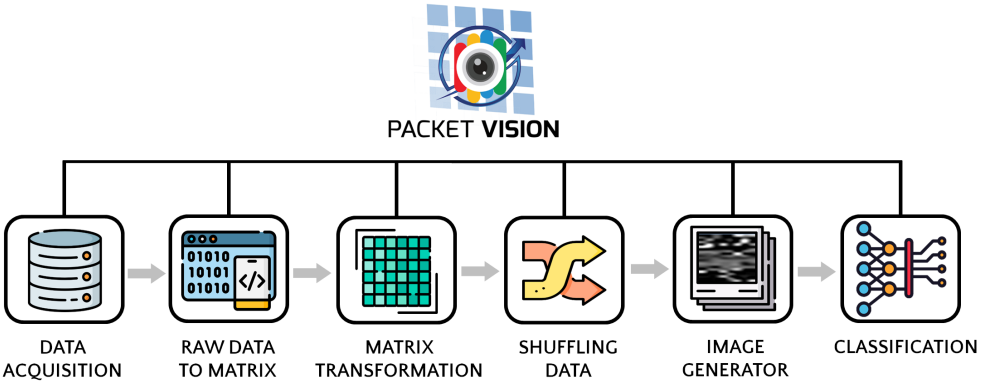

Figura 35 – Método *Packet Vision*.

Moreira et al. (2020).

A matriz possui essa dimensão porque a biblioteca do *Wireshark* entrega as informações contidas no pacote, incluindo o cabeçalho e carga útil, em um formado compreensível para humanos, logo, há pacotes que são maiores que outros implicando na necessidade de uma compatibilização de diferenças. Quando o *array* de *bytes* que o *Wireshark* entregar não permitir a construção direta de uma matriz, são adicionados *bytes* nulos (*0xFF*) até alcançar a configuração de uma matriz.

O terceiro passo do método *Packet Vision* é transformar a matriz $n \times 8$ contendo valores hexadecimais em uma matriz decimal. Ao final desse passo, haverá uma matriz $n \times 8$ contendo valores decimais. Logo no quarto passo, todos números decimais matriz, até o limite do tamanho do cabeçalho, têm suas posições embaralhadas aleatoriamente conforme uma distribuição de probabilidade. É importante fazer esse embaralhamento para garantir que informações sensíveis como

endereço e porta, por serem comuns a todos os pacotes, não influenciem no processo de treinamento e que resguarde em certo grau a privacidade do remetente e destinatário.

O quinto passo contém o procedimento de transformar cada número decimal da matriz em um canal de cores *RGB*. Nesse passo, serão geradas figuras no formato *Portable Graphics Format (PNG)* que contêm as informações e estrutura do pacote expressas em colocação e intensidade de *pixels* em uma figura. A Figura 36 traz exemplos de pacotes que foram transformados em figuras.

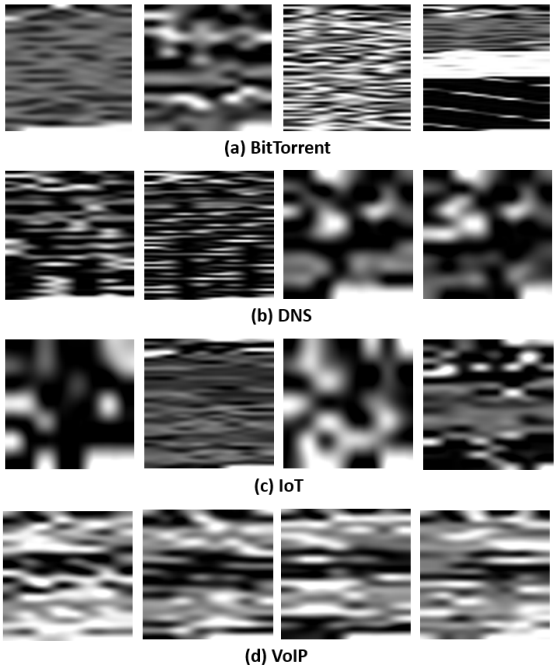

Figura 36 – Exemplos de pacotes gerados pelo *Packet Vision*.

Moreira et al. (2020).

O sexto passo contém etapas de treinamento e validação estatística da acurácia do mecanismo de aprendizado profundo. Várias arquiteturas de redes neurais convolucionais são conhecidas, por isso, foi necessário avaliar a performance das mais relevantes da literatura para identificar a que é mais adequada, em critérios de acurácia, para a classificação de pacotes de rede. Após essa etapa de treinamento e validação do modelo, cada pacote passante na rede pode ser submetido

ao classificador que retornará, probabilisticamente, sua classe de tráfego.

A distribuição do conjunto de imagens criado é sumarizado na Tabela 8:

Tabela 8 – Distribuição de Imagens por Classes.

Classe	**Amostras**
Bit Torrent	1217
DNS	1412
VoIP	1320
IoT	1848
Total	5797

A primeira classe de tráfego refere-se às aplicações *IoT*, essa base de dados contém traços de comunicação de pacotes, no formato *pcap*. Essa classe de tráfego foi construída por 31 dispositivos *IoT* para casas inteligentes de 27 tipos, incluindo sensores e atuadores (MIETTINEN et al., 2017).

A segunda fonte de traços de comunicação de pacotes, no formato *pcap*, agrupou-se aos traços de comunicação anteriores mais duas classes: *DNS* e *BitTorrent*. Essas duas classes de tráfego foram apresentadas por Carela-Español, Bujlow e Barlet-Ros (2014) que descreveram a classificação de tráfego na Internet considerando essas classes de aplicações.

Ao construir um conjunto de imagens referentes às várias classes de tráfego e treinar um classificador, consolida-se a segunda premissa, que é criar subsistema cliente-servidor acoplado ao *NASOR* capaz de retornar o caminho escolhido em um domínio para a fatia de rede que será implantada. Conforme a Figura 34, o subsistema cliente-servidor solicita via *RestAPI* uma estrutura de dados grafo ao *NASOR* por meio da *OPI*. Internamente, a *OPI* solicita ao serviço de topologia o estado atual do grafo do domínio de rede e uma amostra de pacotes passantes para cada aresta do grafo que representa a topologia do domínio. Após, esses dados são devolvidas como resposta à requisição ao subsistema que possui o classificador treinado que computará para cada aresta do grafo a classe de tráfego passante predominante.

Após aferidas as classes de tráfego predominantes de cada aresta, o mecanismo de escolha de caminho mínimo em um grafo construirá o grafo e devolverá ao *NASOR*. Quando o *NASOR* receber a estrutura de dados grafo contendo um ca-

minho mínimo candidato para a fatia de rede ele procederá com o estabelecimento e configuração da fatia de rede ao longo dos roteadores e *ASs*.

4.7.1 Método de Avaliação: Cenário Experimental 4

Para comparar as arquiteturas de *Convolutional Neural Networks (CNNs)* para decidir qual poderia ser conectada ao subsistema que interage com a *OPI*, três arquiteturas foram testadas. Esse teste considerou o método de validação cruzada com *k-fold* estratificado, dado que esse método é mais robusto para lidar com valores discrepantes e sobreajuste eventual. O sobreajuste ocorre quando o modelo probabilístico do classificador se ajusta ao conjunto de dados de treinamento e em predições futuras ele se mostra ineficaz. Além disso, as *CNNs* foram treinadas utilizando a estratégia de *fine-tunning* (GOODFELLOW; BENGIO; COURVILLE, 2016) em modelos pré-treinados com a base *ImageNet* (DENG et al., 2009).

Escolheu-se empiricamente três arquiteturas de *CNN* que foram mutuamente avaliadas para constatar suas adequabilidades para atuarem como uma aplicação terceira no processo de fatiamento de rede. Essa definição inicial baseou-se na performance prévia que essas arquiteturas desempenharam em processos de classificação de imagens registradas na literatura. Assim, a arquitetura de *CNN* que melhor performar neste experimento, considerando o critério de acurácia, deverá ser utilizada como o mecanismo de classificação de tráfego para atuar como uma aplicação terceira ao *NASOR*. As arquiteturas estabelecidas para comparação mútua foram: *ResNet50* (HE et al., 2016), *SqueezeNet* (IANDOLA et al., 2016a) e *VGG-16* (SIMONYAN; ZISSERMAN, 2014).

Neste cenário experimental, foram criadas 5797 imagens que compuseram um conjunto de dados que no processo de treinamento e validação foi particionado em cinco *folds*. O *dataset* construído no contexto desta tese está disponibilizado publicamente[5], ele é a composição de outros *datasets* que são disponíveis na literatura. As classes do conjunto de dados deste experimento são: *BitTorrent*, *DNS*, *VoIP* e *Internet of Things (IoT)*. Em cada iteração da validação cruzada, um dos *folds* foi escolhido para testar o modelo treinado e $k-1$ *folds* foram utilizados para treinamento. Esse procedimento foi repetido k vezes, alternando-se os *folds* de teste.

[5] *Packet Vision* disponível em <https://romoreira.github.io/packetvision/>.

Com isso, garantiu-se que cada imagens participou do processo de treinamento também fez parte do processo de teste do modelo.

Além disso, foi mensurado a média de performance considerando quatro índices da matriz de confusão. A matriz de confusão registra as frequências de classificação das classes do modelo. O índice verdadeiro-positivo (TP) registra quando o modelo classificou corretamente uma imagem. O verdadeiro-negativo (TN) registra quando o modelo classificou uma imagem como não pertencente à classe que não se buscava prever. O índice falso-positivo (FP) ocorre quando a classe que se buscou prever foi incorretamente prevista. O índice falso-negativo (FN) refere-se ao cenário que a classe em que não se buscava prever foi incorretamente prevista.

A mensuração da qualidade do aprendizado do classificador baseou-se nas métricas que são construídas a partir da matriz de confusão. A acurácia refere-se a performance-se de classificação geral do modelo, sendo representada por: $A = \frac{TP+TN}{TP+TN+FP+FN}$. Outra métrica é a precisão, que se refere as classificações corretas da classe positivo feitas pelo modelo.

A precisão pode ser representada por $P = \frac{TP}{TP+FP}$. A métrica sensibilidade (*recall*) representa a quantidade de classificações corretas considerando todas as situações da classe positivas. A sensibilidade é representada pela equação $R = \frac{TP}{TP+FN}$. A última métrica é a média harmônica (*F1-score*) entre a precisão e sensibilidade que é representada pela equação $F = 2 \times \frac{P \times R}{P+R}$.

Este cenário experimental foi executado sobre serviço de computação em nuvem *Google Colaboratory* (BISONG, 2019) com *hardware* Intel Xeon 2.20 GHz, 12 *GB* de *RAM* e *GPU NVIDIA Tesla T4*. Os experimentos foram programados utilizando *Python* verão 3.6 e o *framework* de aprendizado profundo *PyTorch* verão 1.7.

Os experimentos executados objetivaram responder as seguintes questões:

- ❏ Qual método baseado em redes neurais convolucionais é mais apropriado para classificar tráfego de rede considerando o conjunto de dados construído, o cenário proposto e as métricas de acurácia, precisão, *recall* e *f1-score*?

- ❏ Em termos de custo computacional, quanto tempo e qual método de classificação despende mais tempo no processo de treinamento do modelo?

- ❏ Incorporar mecanismos baseados em inteligência artificial com o *NASOR* por meio da *OPI* é adequado para aumentar o dinamismo nos métodos de implantação de fatias de rede entre múltiplos *ASs*?

4.7.2 Avaliação dos Resultados: Cenário Experimental 4

Após particionar o conjunto de dados em cinco *folds* usando *k-fold* estratificado e o método de validação cruzada, as imagens, que em essência são pacotes de rede, foram redimensionadas para 224×224 *pixels* considerando interpolação bilinear (GONZALEZ; WOODS, 2006). O redimensionamento das imagens não implicou em aumento ou remoção de informação nas imagens. O redimensionamento aplicado objetivou adaptar as imagens para torná-las entradas adequadas nas arquiteturas de *CNN* que foram avaliadas neste cenário experimental.

As arquiteturas de *CNN* foram treinadas utilizando o otimizador Adam de Kingma e Ba (2014). Além disso foram definidos os seguintes parâmetros: a taxa de aprendizado em 0.001, tamanho do lote (*batch*) em 32 e 30 épocas de treinamento. Adicionalmente, foi mensurado o desvio padrão do percentual de acurácia obtida pelas três arquiteturas de *CNN*, sendo: *VGG-16* 0.042, *ResNet-50* 0.018 e *SqueezeNet* 0.016. Esses valores sugerem que o processo de treinamento é confiável uma vez que a dispersão das amostras é pequena levando a uma homogeneidade nas acurácias alcançadas por cada experimento.

Com relação a performance de classificação, a Tabela 9 apresenta as métricas e os valores alcançados ao fim do experimento. Para cada métrica: acurácia, precisão, sensibilidade e *F1-score* foram computadas as médias dos *5-folds* no processo de treinamento do classificador.

Tabela 9 – Média das métricas de performance dos *5-folds*.

CNN	**Acurácia (%)**	**Precisão (%)**	**Sensibilidade (%)**	***F1-Score* (%)**
ResNet-50	95.00	95.50	94.75	95.00
SqueezeNet	**96.80**	**98.00**	**98.00**	**98.00**
VGG-16	91.75	92.60	92.00	92.00

A acurácia refere-se a taxa entre o número de classificações corretas e o total de amostras. Logo, conforme apresentado na Tabela 9, a arquitetura *SqueezeNet*

performou melhor do que seus pares, considerando o conjunto de dados e o cenário proposto. Essa performance relaciona-se com a robusteza da sua arquitetura que há um módulo denominado *fire* que é empilhado e dividido na compressão e expansão dos filtros convolucionais, permitindo um grande conjunto de mapas de característica realizando uma classificação dos pacotes de rede com uma acurácia de 96.80%.

Além disso, a *SqueezeNet* foi projetada para executar sobre computadores de baixo custo, sugerindo que essa arquitetura de *CNN* é um mecanismo eficiente e escalável para realizar um mecanismo de classificação de tráfego na borda. Logo, torna-se adequado acoplar essa tecnologia ao subsistema que interage com o *NASOR*, via *OPI*, para instruir o processo de estabelecimento de fatias de rede, que considere no processo de configuração de um caminho a classe de tráfego que possui predominância estatística. Segmentar o tráfego de rede pelo seu padrão de tráfego e aplicação pode levar a um gerenciamento da rede mais previsível.

Neste cenário experimental, onde se buscou validar a aplicabilidade de incorporar mecanismo de classificação de tráfego para instruir o *NASOR* no processo de estabelecimento de fatias de rede, avaliou-se o comportamento do aprendizado de cada arquitetura *CNN*. A análise da perda (*loss*) e acurácia durante a fase de de treino e média de todas as interações do processo *k-fold* é conforme a Figura 37.

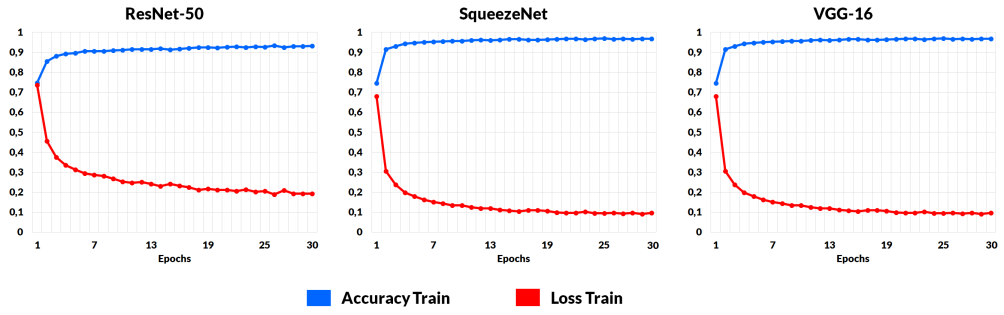

Figura 37 – Gráficos que mostram a evolução dos valores de precisão e perda para cada *CNN* considerando o conjunto de treinamento 5 vezes médio.

Ao longo de todo o treino, o comportamento da função de perda (*loss*) sugere que todas as arquiteturas de *CNN* gerou valores baixos. Esse comportamento ilustrado nos gráficos sugere que, na fase de treinamento, não ocorreu sobre-

ajustamento do modelo ao conjunto de dados, implicando que os três classificadores mantiveram suas propriedade de generalização. Quando um classificador possui a propriedade de generalização, quando lhe for apresentado uma nova imagem, diferente daquela do conjunto de dados, ele será capaz de classificá-la adequadamente.

Quanto a qualidade da classificação, foram construídas matrizes de confusão para cada arquitetura de *CNN* avaliada neste experimento. Matrizes de confusão descrevem vários aspectos do problema de classificação investigado neste experimento. Em especial, representa o número de classificações de pacotes corretos e incorretos no conjunto de dados.

Conforme apresentado na Tabela 10, a arquitetura *CNN ResNet-50* (a) confundiu predominantemente as imagens IoT, classificando-as como se fossem *DNS*. No entanto, a *SqueezeNet* (b) errou balanceadamente as imagens da classe IoT, classificando-as como pertencentes as classes *Bit Torrent* e *DNS*. A arquitetura *VGG-16* (c) classificou erroneamente $\approx 14\%$ dos pacotes IoT, uma vez que o total de pacotes da classe IoT era de 1.848 e a *CNN* classificou erroneamente 210 imagens.

As classificações errôneas variaram entre as arquiteturas *CNNs* e estão descritas na matriz de confusão de cada classificador.

Tabela 10 – Matriz de confusão das classificações dos *5-folds* para cada arquitetura de *CNN*.

(a) ResNet-50

	Bit Torrent	**DNS**	**IoT**	**VoIP**
Bit Torrent	**1082**	6	123	6
DNS	1	**1353**	58	0
IoT	21	96	**1728**	3
VoIP	2	0	9	**1309**

(b) SqueezeNet

	Bit Torrent	**DNS**	**IoT**	**VoIP**
Bit Torrent	**1132**	2	82	1
DNS	1	**1393**	18	0
IoT	39	36	**1768**	2
VoIP	1	0	2	**1317**

(c) VGG-16

	Bit Torrent	**DNS**	**IoT**	**VoIP**
Bit Torrent	**1081**	27	106	3
DNS	27	**1351**	34	0
IoT	50	194	**1588**	16
VoIP	1	1	21	**1297**

Conforme apontado pela matriz de confusão, a maior incidência de erros recaiu sobre a classe IoT. Ao observar visualmente as imagens que são geradas dos pacotes do tipo IoT, notou-se que as imagens possuíam padrões texturais consideravelmente diferentes entre si, ao passo que imagens das outras classes (*Bit Torrent*, *VoIP* e *DNS*) mantiveram certo padrão textural.

Isso sugere que pacotes IoT geraram imagens com texturas heterogêneas, ou seja, com muita variação no cabeçalho ou carga útil do pacote. Isso foi corroborado ao analisar o conjunto de dados da classe IoT que produziu as imagens para treinamento e teste das arquiteturas de *CNN*. Constatou-se que a classe IoT

foi gerada por meio da captura de pacotes de 27 dispositivos heterogêneos como sensores e atuadores, além de aplicações distintas (MIETTINEN et al., 2017).

Neste cenário experimental foi avaliado o tempo de treinamento de cada arquitetura *CNN*. O tempo de treinamento é um fator importante porque é operacionalmente desejável que uma data arquitetura de *CNN* gaste a menor quantidade de tempo possível no processo de treinamento. Além disso, por se tratar de um aprendizado supervisionado, sempre que houver necessidade de aprimorar o classificador, uma nova etapa de treinamento com outas classes de pacotes no conjunto de dados será necessária.

Para avaliação do tempo de treinamento foi levado em consideração o tempo de submissão da carga de trabalho até o fim do seu processamento. A arquitetura *SqueezeNet*, que além de ter performado bem conforme a métrica acurácia, exigiu $\approx 2.7\times$ menos tempo computacional do que a arquitetura *VGG-16* e $\approx 58.1\%$ menos que a *ResNet-50*. Na etapa de treinamento do modelo, a *VGG-16* consumiu em média 27:24 min, *ResNet-50* 17:24 min e a *SqueezeNet* 10:07 min.

Quando a performance de tempo na fase de treinamento for essencial, a arquitetura *SqueezeNet* apresenta-se como a melhor escolha. Em cenários distribuídos, a *SqueezeNet* exige menos comunicação e largura de banda entre os servidores, e é um modelo adequado para implantação em *hardware* com limitação de memória (IANDOLA et al., 2016b). Logo, dado a natureza distribuída do *framework NASOR* e a performance dos classificadores de tráfego de rede, é possível assumir que a *SqueezeNet* mostra-se adequada para ser utilizada como uma aplicação terceira que utiliza a *OPI* para influenciar no processo de fatiamento de redes do *NASOR*.

Arquiteturas de Internet do Futuro, incluindo redes móveis, podem valer-se de técnicas de classificação de tráfego de rede para prover um melhor gerenciamento e qualidade no oferecimento de serviços de conectividade personalizados. A partir dos experimentos realizados e baseado na acurácia dos modelos de classificação, é adequado que abordagens de fatiamento de rede tenham interfaces padronizadas e disponíveis para aprimorar os comportamentos de seus mecanismos internos, especialmente no fataiemnto de rede, onde cada usuário ou aplicação requerem da rede especifidades.

Por isso, a interface *OPI* do *framework NASOR* avança o estado da arte ao

aprimorar o dinamismo no gerenciamento e estabelecimento de fatias de rede entre múltiplos *ASs* influenciando a escolha de um caminho para a fatia de rede considerando o tipo de tráfego predominante.

4.8 Considerações Finais

Ao avaliar funcionalmente, qualitativamente e quantitativamente o *NASOR* ao longo de cenários experimentais relevantes é possível admitir que a hipótese e as questões de pesquisa acessórias foram exploradas e respondidas. O ponto mais significativo é que ao explorar qualitativamente o *NASOR*, constata-se que é viável e funcionalmente aceitável concebê-lo como um *framework* de controle hierárquico-distribuído, compatível com o modelo de gerenciamento e orquestração da *ETSI*, e que implanta fatias de rede sobre múltiplos *ASs*, considerando o plano de dados construído pelos algoritmos de roteamento da Internet.

Além disso, mostra ser possível implantar fatias de rede sobre múltiplos *ASs*, e que tecnologias como roteamento por segmentos e repositórios de dados persistentes, hierárquicos e distribuídos mostram-se essenciais para lidar com a natureza distribuída dos *ASs*. Também, o *NASOR* e seu componente funcional *OPI* habilita aos usuários/donos da fatia de rede especificar parâmetros estritos para suas fatias de rede, implicando em um processo de implantação de fatias de rede mais dinâmico.

Por fim, no contexto do *framework NASOR* foi apresentado um arcabouço conceitual e tecnológico para habilitar o fatiamento recursivo de rede entre múltiplos domínios. O componente basilar desse processo é a concepção de micro-orquestradores que desempenham papel de gerenciamento e orquestração de recursos computacionais e de rede obedecendo os limites físicos da fatia de rede pai. O principal papel do micro-orquestrador é configurar parâmetros de rede baseado em tabelas de roteamento virtuais que são construídas a partir da tabela de roteamento primária do roteador, físico ou virtual.

Capítulo 5

Conclusão

Esta tese reposiciona o conceito de fatiamento de redes ao prover um método de implantação fatia de rede que perpassa múltiplos *ASs* sobre os roteadores da Internet. O fatiamento de recursos, amplamente discutido na esteira das novas redes móveis, não é um conceito remanescente, já se discutia inspirado na virtualização mecanismos para prover a separação lógica, de controle e de dados de recursos de rede de propósito geral.

Nesta tese, após uma apreciação do estado da arte, foi constatado que predominantemente as soluções se prestaram a oferecer fatiamento de recursos de rede eram centradas no âmbito das redes móveis, e nas entidades dessa arquitetura de rede. Por outro lado, valendo-se das tecnologias habilitadores em voga *SDN*, *NFV*, computação em nuvem, roteamento por segmentos e demais. Conjecturou-se nesta tese hipóteses sobre premissas como: o plano de dados construído pelos algoritmos de roteamento da Internet pode ser um candidato a prover conectividade para fatias de rede em alternativa as abordagens que usam outros tipos tecnologias como a de rede sobreposta.

Direcionado por alguns pontos que as soluções do estado da arte não lidavam plenamente, o objetivo de prover um mecanismo de fatiamento de rede multi-domínios orientou-se de padrões bem estabelecidos da academia e indústria e se materializou no mecanismo *NASOR*. Além disso, os componentes e interfaces da abordagem desta tese foi amplamente apresentado considerando seu comportamento e tecnologias habilitadoras. Uma sistematização dos trabalhos relacionados

permitiu materializar uma revisão da literatura que considerou e classificou as abordagens do estado da arte que argumentam realizar fatiamento de redes. A revisão da literatura permitiu que cada trabalho fosse analisado quanto às características comuns e essenciais para o cenário distribuído da Internet.

Adicionalmente, essa avaliação da literatura permitiu estruturar, construir e avaliar o mecanismo de fatiamento de redes multidomínios. A observância dos trabalhos relacionados tornou possível ponderar tecnologias e procedimentos adequados para adequar um método de desenvolvimento. Esse método de desenvolvimento é de natureza incremental, uma vez que atividades de etapas iniciais são sistematicamente revisitadas.

Além disso, a pertinência da proposta frente a literatura permitiu propor casos de usos relevantes para o cenário de aplicações entre múltiplos domínios. Aferir a qualidade da solução para suprir os requisitos das aplicações por fatias de redes com recursos personalizados exigiu fosse planejado e avaliado em cenários experimentais.

A hipótese levantada e seus desdobramentos foi corroborada ao longo dos experimentos relatados e discutidos nesta tese. Destaca-se no Cenário Experimental 1 a capacidade do método proposto de implantar serviços de rede personalizados como *DNS* sobre fatias de redes dos usuário, evidenciando um novo formado de implantação e entrega de serviços. O Cenário Experimental 2 avaliou o método proposto quando a escalabilidade e tempo de implantação de fatias de rede entre múltiplos *ASs*. Nesse cenário, constatou-se que o método proposto nesta tese suporta dinamismo na implantação de fatias de rede ao construir um caminho entre múltiplos *ASs* considerando a performance momentânea dos enlaces.

No Cenário Experimental 3 foi constatado que o processamento precoce de pacotes viabiliza a implantação de fatias de rede com requisitos de latência extremamente baixas. Além disso, foi avaliado a escalabilidade do método proposto nesta tese em implantar inúmeras fatias de rede ao longo de múltiplos *ASs*. O Cenário Experimental 4 permitiu aferir que soluções para fatiamento de rede quando oferecem interfaces abertas que influenciam o estabelecimento de fatias de rede entre múltiplos *ASs* aprimoram o dinamismo da solução. Nessa avaliação, constatou-se adequado o método de construção de um plano de dados para fatia de rede considerando a classe de tráfego corrente nas interfaces.

5.1 Contribuições

As contribuições desta tese são bidimensionais, a saber: tecnológica e científica. A contribuição tecnológica diz respeito a entrega de um *framework* de fatiamento de rede e gerenciamento de recursos para os administradores de rede[1]. A contribuição científica reflete na prova, aceitação e validação por pares do método de implantação do fatias de rede entre múltiplos *AS*. Além disso, a proposta de fatiamento de rede recursivo inaugurou de forma consistente a compreensão do fatiamento de rede recursivo sobre múltiplos *ASs*.

Dentre as funcionalidades que classificam o *NASOR* frente a seus pares, destaca-se a oferta de fatiamento de rede recursivo. Ela redesenha o formato de concepção, entrega, operação e monetização de conectividade na forma como se conhece, baseada na entrega *link* com métricas de qualidade e *Service-Level Agreement (SLA)* definidos entre as partes. Com a abordagem *NASOR*, o dono de uma fatia de rede tem dinamismo para redimensioná-la e monetizá-la e conforme seus critérios.

5.2 Contribuições em Produções Bibliográficas

Esta seção apresenta as publicações originadas da construção, experimentação e validação do método apresentado nesta tese. As produções bibliográficas estão relacionadas indireta ou diretamente com a esta tese, quais sejam:

Indiretamente:

- Silva, A. P., Tranoris, C., Denazis, S., Sargento, S., Pereira, J., Luís, M., ... Simeonidou, D. (2019). ***5GinFIRE: An end-to-end open5G vertical network function ecosystem***. Ad Hoc Networks, 93, 101895. <https://doi.org/10.1016/j.adhoc.2019.101895> - Qualis **A2**

- Moreira, R., Silva, F. D. O., Rosa, P. F., & Aguiar, R. L. (2020). ***A smart network and compute-aware Orchestrator to enhance QoS on cloud-based multimedia services***. International Journal of Grid and Utility Computing, 11(1), 49. <https://doi.org/10.1504/ijguc.2020.103969> - Qualis **B2**

[1] *Framework NASOR* disponível em <https://romoreira.github.io/nasor>

- Richards, V.; Moreira, R. and Silva, F. (2020). ***Enabling the Management and Orchestration of Virtual Networking Functions on the Edge***. In Proceedings of the 10th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER, ISBN 978-989-758-424-4, pages 338-346. DOI: <10.5220/0009398203380346> - Qualis **A2**

Diretamente:

- Moreira, R., Rosa, P. F., Aguiar, R. L. A., & de Oliveira Silva, F. (2020). ***Enabling Multi-domain and End-to-End Slice Orchestration for Virtualization Everything Functions (VxFs)***. In Advanced Information Networking and Applications (pp. 830–844). Springer International Publishing. <https://doi.org/10.1007/978-3-030-44041-1_73> - Qualis **A2**

- R. Moreira, L. F. Rodrigues, P. F. Rosa, R. L. Aguiar and F. d. O. Silva, ***Enhancing dynamism in management and network slice establishment through deep learning***, 2021 International Conference on Information Networking (ICOIN), 2021, pp. 321-326, doi: <10.1109/ICOIN50884.2021.9333872>. - Qualis **B1**

- R. Moreira, L. F. Rodrigues, P. F. Rosa, R. L. Aguiar and F. d. O. Silva, ***Packet Vision: a convolutional neural network approach for network traffic classification***, 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2020, pp. 256-263, doi: <10.1109/SIBGRAPI51738.2020.00042>. - Qualis **B1**

- MOREIRA, Rodrigo; RODRIGUES, Larissa; ROSA, Pedro; SILVA, Flávio. ***Improving the network traffic classification using the Packet Vision approach***. In: WORKSHOP DE VISÃO COMPUTACIONAL (WVC), 16. , 2020, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 146-151. DOI: <https://doi.org/10.5753/wvc.2020.13496> - Qualis **B5**

- Moreira R., Rosa P.F., Aguiar R.L.A., de Oliveira Silva F. (2021) **Deploying Scalable and Stable XDP-Based Network Slices Through NASOR**

Framework for Low-Latency Applications. In: Barolli L., Woungang I., Enokido T. (eds) Advanced Information Networking and Applications. AINA 2021. Lecture Notes in Networks and Systems, vol 226. Springer, Cham. <https://doi.org/10.1007/978-3-030-75075-6_59> - Qualis **A2**

- da Cunha H.G.V.O., Moreira R., de Oliveira Silva F. (2021) ***A Comparative Study Between Containerization and Full-Virtualization of Virtualized Everything Functions in Edge Computing***. In: Barolli L., Woungang I., Enokido T. (eds) Advanced Information Networking and Applications. AINA 2021. Lecture Notes in Networks and Systems, vol 226. Springer, Cham.
<https://doi.org/10.1007/978-3-030-75075-6_63> - Qualis **A2**

- Moreira, R., Rosa, P. F., Aguiar, R. L. A., & de Oliveira Silva, F. (2021). ***NASOR: A Network Slicing Approach for Multiple Autonomous Systems***. In Computer Communications ISSN: 0140-3664 - <https://doi.org/10.1016/j.comcom.2021.07.028> - Qualis **A1**

- Moreira, R., & de Oliveira Silva, F. (2021). ***Towards 6G Network Slicing***. ANAIS DO WORKSHOP DE PESQUISA EXPERIMENTAL DA INTERNET DO FUTURO (WPEIF) - Qualis **B4**

Outras Produções:

- Pedido de registro de patente do método ***Packet Vision*** no Instituto Nacional da Propriedade Industrial (INPI).

5.3 Trabalhos Futuros

Nesta tese foi proposto e avaliado o *framework NASOR* para implantação de fatias de rede entre múltiplos *ASs*. O fatiamento de rede ainda é um problema que requer avanços especialmente no núcleo da rede. Desafios inerentes a novos mecanismos de criação e manutenção de planos de dados mais flexíveis e programáveis em equipamentos da rede de transporte merecem investigações à parte.

O *framework NASOR* pode ser aprimorado para suportar mecanismos de encapsulamento e identificação de fatias de rede para além do roteamento por segmentos. A demonstração da adequação do *NASOR* teve como premissa que seus componentes distribuídos não falhariam e que as mensagens de sincronização e operação entre as entidades colocadas nos múltiplos *ASs* não se perderiam. Essa premissa é razoável na validação da hipótese, mas incorporar mecanismos de disponibilidade e confiabilidade tornaria o *NASOR* robusto.

O fatiamento recursivo de rede e seus desdobramentos emerge como um desafio necessário de ser endereçado, especialmente para que os usuários tenham uma experiência satisfatória e personalizada de conectividade. As aplicações possuem requisitos cada vez mais sofisticados e peculiares, e a rede precisa ofertar tais recursos para satisfazer esses requisitos. Nesse sentido, compreende-se que o fatiamento recursivo de rede pode ser um o ponto de partida para rupturas no modelo de implantação e operação de novos serviços e aplicações de rede.

Referências

3GPP. Management and Orchestration: 5G end to end Key Performance Indicators (KPI). 2019. Disponível em: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3415>.

ABDELSALAM, A. et al. Implementation of virtual network function chaining through segment routing in a linux-based NFV infrastructure. In: **2017 IEEE Conference on Network Softwarization (NetSoft)**. Bologna, Italy: IEEE, 2017. p. 1–5.

_____. Performance of IPv6 segment routing in linux kernel. In: **2018 14th International Conference on Network and Service Management (CNSM)**. Rome, Italy: IEEE, 2018. p. 414–419. ISSN 2165-9605.

ABDULLAH, Z. N.; AHMAD, I.; HUSSAIN, I. Segment Routing in Software Defined Networks: A Survey. **IEEE Communications Surveys Tutorials**, v. 21, n. 1, p. 464–486, Firstquarter 2019. ISSN 2373-745X.

ABELS, T.; DHAWAN, P.; CHANDRASEKARAN, B. An overview of XEN virtualization. **Dell Power Solutions**, v. 8, p. 109–111, 2005.

Afolabi, I. et al. Network Slicing and Softwarization: A survey on principles, enabling technologies, and solutions. **IEEE Communications Surveys Tutorials**, v. 20, n. 3, p. 2429–2453, 2018.

AGER, B. et al. Comparing DNS resolvers in the wild. In: **Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement**. New York, NY, USA: ACM, 2010. (IMC '10), p. 15–21. ISBN 978-1-4503-0483-2. Disponível em: <http://doi.acm.org/10.1145/1879141.1879144>.

AHN, J.; PARK, C. H.; HUH, J. Micro-sliced virtual processors to hide the effect of discontinuous CPU availability for consolidated systems. In: **2014 47th Annual IEEE/ACM International Symposium on Microarchitecture**. Cambridge, UK: IEEE, 2014. p. 394–405.

AKYILDIZ, I. F.; WANG, P.; LIN, S.-C. Softair: A software defined networking architecture for 5G wireless systems. **Computer Networks**, v. 85, p. 1 – 18, 2015. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128615001632>.

ALEXA. Top global sites (2019). **URL http://www.alexa.com/topsites**, 2019.

Alhuseini, M. U.; Olama, M. M. 5G service value chain and network slicing framework using ecosystem modeling, agile delivery, and user-story automation. **IEEE Access**, v. 7, p. 110856–110873, 2019. ISSN 2169-3536.

ANDERSON, T. et al. Overcoming the internet impasse through virtualization. **Computer**, v. 38, n. 4, p. 34–41, 2005.

ANSAH, F. et al. Network slicing : An industry perspective. In: **2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)**. Zaragoza, Spain: IEEE, 2019. p. 1367–1370.

AZODOLMOLKY, S.; WIEDER, P.; YAHYAPOUR, R. Cloud computing networking: challenges and opportunities for innovations. **IEEE Communications Magazine**, v. 51, n. 7, p. 54–62, July 2013. ISSN 0163-6804.

BANGERA, P.; HASAN, S.; GORINSKY, S. An advertising revenue model for access ISPs. In: **2017 IEEE Symposium on Computers and Communications (ISCC)**. Heraklion, Greece: IEEE, 2017. p. 582–589.

BARAKABITZE, A. A. et al. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. **Computer Networks**, v. 167, p. 106984, 2020. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128619304773>.

BARATVAND, M. et al. Jitter-buffer management for VoIP over wireless LAN in a limited resource device. In: **Fourth International Conference on Networking and Services (ICNS 2008)**. Gosier, France: IEEE, 2008. p. 90–95.

BARI, M. F. et al. Data center network virtualization: A survey. **IEEE Communications Surveys Tutorials**, v. 15, n. 2, p. 909–928, Second 2013. ISSN 2373-745X.

BASIT, A. et al. Interconnecting networks with optimized service provisioning. **Telecommunication Systems**, v. 73, n. 2, p. 223–239, 2020. ISSN 1572-9451. Disponível em: <https://doi.org/10.1007/s11235-019-00606-3>.

BASTIN, N. et al. The instageni initiative: An architecture for distributed systems and advanced programmable networks. **Computer Networks**, v. 61, p. 24 – 38, 2014. ISSN 1389-1286. Special issue on Future Internet Testbeds – Part I. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613004477>.

BEN-DAVID, S.; KUSHILEVITZ, E.; MANSOUR, Y. Online Learning versus Offline Learning. **Machine Learning**, v. 29, n. 1, p. 45–63, Oct 1997. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1007465907571>.

BERDE, P. et al. ONOS: Towards an Open, Distributed SDN OS. In: **Proceedings of the Third Workshop on Hot Topics in Software Defined Networking**. New York, NY, USA: ACM, 2014. (HotSDN '14), p. 1–6. ISBN 978-1-4503-2989-7. Disponível em: <http://doi.acm.org/10.1145/2620728.2620744>.

BERMAN, M. et al. Geni: A federated testbed for innovative network experiments. **Computer Networks**, v. 61, p. 5 – 23, 2014. ISSN 1389-1286. Special issue on Future Internet Testbeds – Part I. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613004507>.

BERNARDOS, C. J. et al. 5GEx: realising a Europe-wide multi-domain framework for software-defined infrastructures. **Transactions on Emerging Telecommunications Technologies**, v. 27, n. 9, p. 1271–1280, 2016. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3085>.

BILAL, K. et al. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. **Computer Networks**, v. 130, p. 94 – 120, 2018. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128617303778>.

BISONG, E. Google colaboratory. In: _____. **Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners**. Berkeley, CA: Apress, 2019. p. 59–64. ISBN 978-1-4842-4470-8. Disponível em: <https://doi.org/10.1007/978-1-4842-4470-8_7>.

BREEN, J. et al. Powder: Platform for open wireless data-driven experimental research. In: **Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization**. New York, NY, USA: Association for Computing Machinery, 2020. (WiNTECH'20), p.

17–24. ISBN 9781450380829. Disponível em: <https://doi.org/10.1145/3411276.3412204>.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In: **2008 10th IEEE International Conference on High Performance Computing and Communications**. Dalian, China: IEEE, 2008. p. 5–13.

CABALLERO, P. et al. Network slicing games: Enabling customization in multi-tenant mobile networks. **IEEE/ACM Transactions on Networking**, v. 27, n. 2, p. 662–675, 2019.

CAFARO, M.; ALOISIO, G. Grids, clouds, and virtualization. In: _____. **Grids, Clouds and Virtualization**. London: Springer London, 2011. p. 1–21. ISBN 978-0-85729-049-6. Disponível em: <https://doi.org/10.1007/978-0-85729-049-6_1>.

CALVERT, K. L. et al. Directions in active networks. **IEEE Communications Magazine**, v. 36, n. 10, p. 72–78, Oct 1998. ISSN 1558-1896.

CAMPBELL, A. T. et al. Open signaling for ATM, internet and mobile networks (opensig'98). **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 29, n. 1, p. 97–108, jan. 1999. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/505754.505762>.

CARELA-ESPAÑOL, V.; BUJLOW, T.; BARLET-ROS, P. Is our ground-truth for traffic classification reliable? In: FALOUTSOS, M.; KUZMANOVIC, A. (Ed.). **Passive and Active Measurement**. Cham: Springer International Publishing, 2014. p. 98–108. ISBN 978-3-319-04918-2.

CASADO, M. et al. Fabric: A retrospective on evolving SDN. In: **Proceedings of the First Workshop on Hot Topics in Software Defined Networks**. New York, NY, USA: Association for Computing Machinery, 2012. (HotSDN '12), p. 85–90. ISBN 9781450314770. Disponível em: <https://doi.org/10.1145/2342441.2342459>.

CASELLAS, R. et al. Metro-haul: Supporting autonomic nfv services over disaggregated optical networks. **EuCNC**, 2018.

_____. Metro-haul: SDN control and orchestration of disaggregated optical networks with model-driven development. In: **2018 20th International Conference on Transparent Optical Networks (ICTON)**. Bucharest, Romania: IEEE, 2018. p. 1–4.

CHOWDHURY, N. M. M. K.; BOUTABA, R. Network virtualization: state of the art and research challenges. **IEEE Communications Magazine**, v. 47, n. 7, p. 20–26, July 2009. ISSN 1558-1896.

CHUN, B. et al. Planetlab: An overlay testbed for broad-coverage services. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 33, n. 3, p. 3–12, jul. 2003. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/956993.956995>.

CISCO. **Linux Kernel**. 2017. Disponível em: <https://www.segment-routing.net/open-software/linux/>.

_____. **Internet adoption and Network Performance, 2018–2023**. 2018. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf>. Online; accessed 27 March 2020.

CORMEN, T. **Introduction to algorithms**. Cambridge, Mass: MIT Press, 2009. ISBN 9780262033848.

COSTA-PEREZ, X. et al. Radio access network virtualization for future mobile carrier networks. **IEEE Communications Magazine**, v. 51, n. 7, p. 27–35, July 2013. ISSN 1558-1896.

CREASY, R. J. The origin of the VM/370 time-sharing system. **IBM Journal of Research and Development**, v. 25, n. 5, p. 483–490, 1981.

CSOMA, A. et al. ESCAPE: Extensible service chain prototyping environment using mininet, click, netconf and pox. In: **Proceedings of the 2014 ACM Conference on SIGCOMM**. New York, NY, USA: Association for Computing Machinery, 2014. (SIGCOMM '14), p. 125–126. ISBN 9781450328364. Disponível em: <https://doi.org/10.1145/2619239.2631448>.

CSÁSZÁR, A. et al. Unifying cloud and carrier network: EU FP7 Project UNIFY. In: **2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing**. Dresden, Germany: IEEE, 2013. p. 452–457.

CUI, C. et al. Network functions virtualisation: Network operator perspectives on industry progress. white paper no. 3, issue 1. In: **SDN and OpenFlow World Congress, Dusseldorf-Germany**. Darmstadt-Germany: SDN and OpenFlow World Congress, 2014.

CUNHA, V. A. et al. MTD to set network slice security as a KPI. **Internet Technology Letters**, v. 3, n. 6, p. e190, 2020. E190 ITL-20-0040.R1. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.190>.

CZIVA, R.; PEZAROS, D. P. Container network functions: Bringing NFV to the network edge. **IEEE Communications Magazine**, v. 55, n. 6, p. 24–31, 2017.

DASGUPTA, S. **Algorithms**. Boston: McGraw-Hill Higher Education, 2008. ISBN 9780073523408.

DASH, D. S.; DURRESI, A.; JAIN, R. Routing of VoIP traffic in multilayered satellite networks. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Performance and Control of Next-Generation Communications Networks**. Orlando, Florida, United States, 2003. v. 5244, p. 65–75.

DEAN, J. Mapreduce and other building blocks for large-scale distributed systems at google. In: . Santa Clara, CA: USENIX Association, 2007.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. Miami, FL, USA: IEEE, 2009. p. 248–255.

DOLUI, K.; DATTA, S. K. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In: **2017 Global Internet of Things Summit (GIoTS)**. Geneva, Switzerland: IEEE, 2017. p. 1–6.

DORIA, A. et al. **RFC3292: General Switch Management Protocol (GSMP) V3**. USA: RFC Editor, 2002.

DOSS, R. **Future network systems and security : second International Conference, FNSS 2016, Paris, France, November 23-25, 2016, Proceedings**. Cham, Switzerland: Springer, 2016. ISBN 978-3-319-48020-6.

DRÄXLER, S. et al. SONATA: Service programming and orchestration for virtualized software networks. In: **2017 IEEE International Conference on Communications Workshops (ICC Workshops)**. Paris, France: IEEE, 2017. p. 973–978.

ELBAMBY, M. S. et al. Toward low-latency and ultra-reliable virtual reality. **IEEE Network**, v. 32, n. 2, p. 78–84, 2018.

ENNS, R.; BJORKLUND, M.; SCHOENWAELDER, J. **RFC4741: NETCONF configuration protocol**. USA: RFC Editor, 2006.

ERICSSON. **Ericsson white paper uen 284 23-3251 rev b**. 2017. Disponível em: <https://www.ericsson.com/49daeb/assets/local/reports-papers/white-papers/wp-5g-systems.pdf>.

ETSI. Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action. In: TECHNICAL REPORT, SDN AND OPENFLOW WORLD CONGRESS. Darmstadt-Germany: ETSI, 2012. Disponível em: <https://portal.etsi.org/NFV/NFV_White_Paper.pdf>.

_____. Network functions virtualisation–network operator perspectives on industry progress. **Updated White Paper**, ETSI, Frankfurt-Germany, 2013. Disponível em: <https://portal.etsi.org/NFV/NFV_White_Paper2.pdf>.

_____. Network functions virtualisation (NFV): Architectural framework. **ETSI Gs NFV**, v. 2, n. 2, p. V1, 2013. Disponível em: <https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf>.

_____. OSM release five technical overview. jan 2019. Disponível em: <https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseFIVE-FINAL.pdf>.

ETSI, O. Open source mano. **OSM home page**, 2016. Disponível em: <https://osm.etsi.org/>.

FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to SDN: An intellectual history of programmable networks. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 44, n. 2, p. 87–98, abr. 2014. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/2602204.2602219>.

FERNANDO, N.; LOKE, S. W.; RAHAYU, W. Mobile cloud computing: A survey. **Future Generation Computer Systems**, v. 29, n. 1, p. 84 – 106, 2013. ISSN 0167-739X. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X12001318>.

FILSFILS, C.; FRANCOIS, P. Previdi. **S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC**, v. 8402, 2016.

FILSFILS, C. et al. The segment routing architecture. In: **2015 IEEE Global Communications Conference (GLOBECOM)**. San Diego, CA, USA: IEEE, 2015. p. 1–6.

_____. Segment routing with MPLS data plane. **IETF draft-ietf-spring-segment-routing-mpls-00**, 2014.

FIORE, U. et al. Traffic matrix estimation with software-defined NFV: Challenges and opportunities. **Journal of Computational Science**, v. 22, p. 162 – 170,

2017. ISSN 1877-7503. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877750317302405>.

FLUCKIGER, F. **Understanding Networked Multimedia: Applications and Technology**. GBR: Prentice Hall International (UK) Ltd., 1995. ISBN 0131909924.

FOSTER, I. et al. Cloud computing and grid computing 360-degree compared. In: **2008 Grid Computing Environments Workshop**. Austin, TX, USA: IEEE, 2008. p. 1–10.

FOUKAS, X. et al. Network slicing in 5G: Survey and challenges. **IEEE Communications Magazine**, v. 55, n. 5, p. 94–100, May 2017. ISSN 1558-1896.

FOY, X. de; RAHMAN, A. Network Slicing-3GPP use case. **Internet-Draft, IETF Trust.**, 2017.

FRANCESCON, A. et al. X-mano: An open-source platform for cross–domain management and orchestration. In: **2017 IEEE Conference on Network Softwarization (NetSoft)**. Bologna, Italy: IEEE, 2017. p. 1–6.

GALIS, A. et al. **Programmable Networks for IP Service Deployment**. USA: Artech House, Inc., 2004. ISBN 1580537456.

GARRICH, M. et al. Network optimization as a service with net2plan. In: **2019 European Conference on Networks and Communications (EuCNC)**. Valencia, Spain: IEEE, 2019. p. 443–447.

GENG, L. et al. **Network Slicing Architecture**. 2017. <https://tools.ietf.org/id/draft-geng-netslices-architecture-01.html#rfc.section.3.3.1>. Online; accessed 06 January 2020.

GIURGIU, I. et al. Calling the cloud: Enabling mobile phones as interfaces to cloud applications. In: BACON, J. M.; COOPER, B. F. (Ed.). **Middleware 2009**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 83–102. ISBN 978-3-642-10445-9.

GONG, L. et al. Toward profit-seeking virtual network embedding algorithm via global resource capacity. In: **IEEE INFOCOM 2014 - IEEE Conference on Computer Communications**. Toronto, ON, Canada: IEEE, 2014. p. 1–9.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (3rd Edition)**. USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. USA: MIT Press, 2016. <http://www.deeplearningbook.org>.

GOOGLE. **Namebench: Open-source DNS Benchmark Utility.** 2019. <https://code.google.com/archive/p/namebench/>. Online; accessed 25 November 2019.

GROUP, G. P. A. W. et al. View on 5G architecture. **White Paper, July**, 2016.

GUSEV, M.; DUSTDAR, S. Going back to the roots—the evolution of edge computing, an iot perspective. **IEEE Internet Computing**, v. 22, n. 2, p. 5–15, 2018.

GUTIERREZ-ESTEVEZ, D. M. et al. 5G-MoNArch use case for ETSI ENI: Elastic Resource Management and Orchestration. In: **2018 IEEE Conference on Standards for Communications and Networking (CSCN)**. Paris, France: IEEE, 2018. p. 1–5.

HABIBI, P. et al. Fog computing: A comprehensive architectural survey. **IEEE Access**, v. 8, p. 69105–69133, 2020.

HAGBERG, A.; SCHULT, D.; SWART, P. Networkx: Python software for the analysis of networks. **Mathematical Modeling and Analysis, Los Alamos National Laboratory**, 2005.

HANAFIZADEH, P.; HATAMI, P.; BOHLIN, E. Business models of internet service providers. **NETNOMICS: Economic Research and Electronic Networking**, v. 20, n. 1, p. 55–99, 2019. ISSN 1573-7071. Disponível em: <https://doi.org/10.1007/s11066-019-09130-7>.

HANDZISKI, V. et al. Twist: A scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In: **Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality**. New York, NY, USA: Association for Computing Machinery, 2006. (REALMAN '06), p. 63–70. ISBN 1595933603. Disponível em: <https://doi.org/10.1145/1132983.1132995>.

HAWILO, H. et al. NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). **IEEE Network**, v. 28, n. 6, p. 18–26, 2014.

HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, NV, USA: IEEE, 2016. p. 770–778.

HEMMINGER, S. et al. Network emulation with NetEm. In: CITESEER. **Linux conf au**. [S.l.], 2005. v. 5, p. 2005.

HERRERA, J. G.; BOTERO, J. F. Resource allocation in NFV: A comprehensive survey. **IEEE Transactions on Network and Service Management**, v. 13, n. 3, p. 518–532, 2016.

HILL, J. F. Internet fragmentation. **Highlighting the Major Technical, Governance and Diplomatic Challenges for US Policy Makers. Cambridge, John F. Kennedy School of Government, Harvard University**, 2012.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. **science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.

HIWASAKI, Y.; OHMURO, H. ITU-T G.711.1: extending G.711 to higher-quality wideband speech. **IEEE Communications Magazine**, v. 47, n. 10, p. 110–116, 2009.

HÉNO, M. L.; BOUBENDIR, A.; SIMONI, N. Telco network slicing models enabling recursive multi-tenancy. In: **2019 10th International Conference on Networks of the Future (NoF)**. Rome, Italy: IEEE, 2019. p. 40–47.

HØILAND-JØRGENSEN, T. et al. The express data path: Fast programmable packet processing in the operating system kernel. In: **Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies**. New York, NY, USA: Association for Computing Machinery, 2018. (CoNEXT '18), p. 54–66. ISBN 9781450360807. Disponível em: <https://doi.org/10.1145/3281411.3281443>.

HØILAND-JØRGENSEN, T. et al. Flent: The flexible network tester. In: **Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools**. New York, NY, USA: ACM, 2017. (VALUETOOLS 2017), p. 120–125. ISBN 978-1-4503-6346-4. Disponível em: <http://doi.acm.org/10.1145/3150928.3150957>.

HU, Y. C. et al. Mobile edge computing—a key technology towards 5G. **ETSI white paper**, v. 11, n. 11, p. 1–16, 2015.

HUANG, T. et al. A survey on large-scale Software Defined Networking (SDN) testbeds: Approaches and challenges. **IEEE Communications Surveys Tutorials**, v. 19, n. 2, p. 891–917, 2017.

HUSAIN, S. et al. Mobile edge computing with network resource slicing for internet-of-things. In: **2018 IEEE 4th World Forum on Internet of Things (WF-IoT)**. Singapore: IEEE, 2018. p. 1–6.

IANDOLA, F. N. et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. **arXiv preprint arXiv:1602.07360**, 2016.

_____. Firecaffe: Near-linear acceleration of deep neural network training on compute clusters. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, NV, USA: IEEE, 2016. p. 2592–2600.

ITU. **3001, Future Networks: Objectives and Design Goals**. Switzerland: ITU Geneva, Switzerland, 2011.

JAIN, R.; PAUL, S. Network virtualization and software defined networking for cloud computing: a survey. **IEEE Communications Magazine**, v. 51, n. 11, p. 24–31, November 2013. ISSN 1558-1896.

JAKMA, P.; LAMPARTER, D. Introduction to the quagga routing suite. **IEEE Network**, v. 28, n. 2, p. 42–48, 2014.

JOHN, W. et al. Research directions in network service chaining. In: **2013 IEEE SDN for Future Networks and Services (SDN4FNS)**. Trento, Italy: IEEE, 2013. p. 1–7.

JURGELIONIS, A. et al. An empirical study of NetEm network emulation functionalities. In: **2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)**. Lahaina, HI, USA: IEEE, 2011. p. 1–6.

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. **Journal of artificial intelligence research**, v. 4, p. 237–285, 1996.

KATO, N. et al. Ten challenges in advancing machine learning technologies toward 6g. **IEEE Wireless Communications**, v. 27, n. 3, p. 96–103, 2020.

KAZMI, S. M. A. et al. **Network Slicing for 5G and Beyond Networks**. 1st. ed. Switzerland: Springer Publishing Company, Incorporated, 2019. ISBN 3030161692.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KOHLER, E. et al. The click modular router. **ACM Trans. Comput. Syst.**, Association for Computing Machinery, New York, NY, USA, v. 18, n. 3, p. 263–297, ago. 2000. ISSN 0734-2071. Disponível em: <https://doi.org/10.1145/354871.354874>.

KOTULSKI, Z. et al. Towards constructive approach to end-to-end slice isolation in 5G networks. **EURASIP Journal on Information Security**, v. 2018, n. 1, p. 2, 2018. ISSN 2510-523X. Disponível em: <https://doi.org/10.1186/s13635-018-0072-0>.

KOUMARAS, H. et al. 5GENESIS: The Genesis of a flexible 5G facility. In: **2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)**. Barcelona, Spain: IEEE, 2018. p. 1–6.

KOUR, H.; GONDHI, N. Machine learning techniques: A survey. In: SPRINGER. **International Conference on Innovative Data Communication Technologies and Application**. [S.l.], 2019. p. 266–275.

KOURETAS, I.; PALIOURAS, V. Simplified hardware implementation of the softmax activation function. In: **2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)**. [S.l.: s.n.], 2019. p. 1–4.

KOURTIS, M.-A. et al. 5g network slicing enabling edge services. In: **2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)**. Leganes, Spain: IEEE, 2020. p. 155–160.

KSENTINI, A.; NIKAEIN, N. Toward enforcing network slicing on RAN: Flexibility and resources abstraction. **IEEE Communications Magazine**, v. 55, n. 6, p. 102–108, 2017.

KUKLINSKI, S.; TOMASZEWSKI, L. Key performance indicators for 5G network slicing. In: **2019 IEEE Conference on Network Softwarization (NetSoft)**. Paris, France: IEEE, 2019. p. 464–471.

LAKATOS, E. M.; MARCONI, M. d. A. Fundamentos de metodologia científica. 5. reimp. **São Paulo: Atlas**, p. 310, 2007.

LANTZ, B.; HELLER, B.; MCKEOWN, N. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In: **Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks**. New York, NY, USA: ACM, 2010. (Hotnets-IX), p. 19:1–19:6. ISBN 978-1-4503-0409-2. Disponível em: <http://doi.acm.org/10.1145/1868447.1868466>.

LEBRUN, D. A linux kernel implementation of segment routing with IPv6. In: **2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. San Francisco, CA, USA: IEEE, 2016. p. 1019–1020.

LEBRUN, D.; BONAVENTURE, O. Implementing IPv6 segment routing in the linux kernel. In: **Proceedings of the Applied Networking Research Workshop**. New York, NY, USA: Association for Computing Machinery, 2017. (ANRW '17), p. 35–41. ISBN 9781450351089. Disponível em: <https://doi.org/10.1145/3106328.3106329>.

LEE, M.-C.; SHEU, J.-P. An efficient routing algorithm based on segment routing in Software-Defined Networking. **Computer Networks**, v. 103, p. 44 – 55, 2016. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128616300871>.

LEWIS, G. et al. Tactical cloudlets: Moving cloud computing to the Edge. In: **2014 IEEE Military Communications Conference**. Baltimore, MD, USA: IEEE, 2014. p. 1440–1446.

LI, H. et al. Mobile edge computing: Progress and challenges. In: **2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)**. Oxford, UK: IEEE, 2016. p. 83–84.

LI, X. et al. 5G-Crosshaul network slicing: Enabling multi-tenancy in mobile transport networks. **IEEE Communications Magazine**, v. 55, n. 8, p. 128–137, 2017.

_____. Network slicing for 5G: Challenges and opportunities. **IEEE Internet Computing**, v. 21, n. 5, p. 20–27, 2017. ISSN 1941-0131.

LI, Y.; CHEN, M. Software-defined network function virtualization: A survey. **IEEE Access**, v. 3, p. 2542–2553, 2015. ISSN 2169-3536.

LI, Z. et al. Perceived speech quality driven retransmission mechanism for wireless voip. IET, 2003.

Linux Foundation. Open network automation platform (ONAP). 2017. Disponível em: <https://www.onap.org/>.

LOVE, B. C. Comparing supervised and unsupervised category learning. **Psychonomic Bulletin & Review**, v. 9, n. 4, p. 829–835, Dec 2002. ISSN 1531-5320. Disponível em: <https://doi.org/10.3758/BF03196342>.

LUNSFORD, D. L. Virtualization technologies in information systems education. **Journal of Information Systems Education**, v. 20, n. 3, p. 339, 2009.

MACH, P.; BECVAR, Z. Mobile edge computing: A survey on architecture and computation offloading. **IEEE Communications Surveys Tutorials**, v. 19, n. 3, p. 1628–1656, 2017.

MAHINDRA, R. et al. Radio access network sharing in cellular networks. In: **2013 21st IEEE International Conference on Network Protocols (ICNP)**. Goettingen, Germany: IEEE, 2013. p. 1–10.

MAO, U. B. Z. et al. A precise and efficient evaluation of the proximity between web clients and their local DNS servers. In: **2002 USENIX Annual Technical Conference (USENIX ATC 02)**. Monterey, CA: USENIX Association, 2002. Disponível em: <https://www.usenix.org/conference/2002-usenix-annual-technical-conference/precise-and-efficient-evaluation-proximity>.

MATIAS, J. et al. Toward an SDN-enabled NFV architecture. **IEEE Communications Magazine**, v. 53, n. 4, p. 187–193, April 2015. ISSN 0163-6804.

MAYORAL, A. et al. Multi-layer service provisioning over resilient software-defined partially disaggregated networks. **J. Lightwave Technol.**, OSA, v. 38, n. 2, p. 546–552, Jan 2020. Disponível em: <http://jlt.osa.org/abstract.cfm?URI=jlt-38-2-546>.

MCCANNE, S.; JACOBSON, V. The BSD packet filter: A new architecture for User-Level Packet Capture. In: **Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings**. USA: USENIX Association, 1993. (USENIX'93), p. 2.

MCKEOWN, N. et al. OpenFlow: Enabling Innovation in Campus Networks. **SIGCOMM Comput. Commun. Rev.**, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1355734.1355746>.

MECHTRI, M. et al. NFV orchestration framework addressing SFC challenges. **IEEE Communications Magazine**, v. 55, n. 6, p. 16–23, 2017.

MEDVED, J. et al. OpenDaylight: Towards a model-driven SDN controller architecture. In: **Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014**. Sydney, NSW, Australia: IEEE, 2014. p. 1–6.

MERGEN, M. F. et al. Virtualization for High-Performance computing. **SIGOPS Oper. Syst. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 40, n. 2, p. 8–11, abr. 2006. ISSN 0163-5980. Disponível em: <https://doi.org/10.1145/1131322.1131328>.

MIETTINEN, M. et al. IoT SENTINEL: Automated device-type identification for security enforcement in IoT. In: **2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)**. Atlanta, GA, USA: IEEE, 2017. p. 2177–2184.

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

MILLER, K.; PEGAH, M. Virtualization: Virtually at the desktop. In: **Proceedings of the 35th Annual ACM SIGUCCS Fall Conference**. New York, NY, USA: Association for Computing Machinery, 2007. (SIGUCCS '07), p. 255–260. ISBN 9781595936349. Disponível em: <https://doi.org/10.1145/1294046.1294107>.

MITCHELL, T. M. et al. Machine learning. McGraw-hill New York, 1997.

MOREIRA, R. et al. Packet Vision: a convolutional neural network approach for network traffic classification. In: **2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. Porto de Galinhas, Brazil: IEEE, 2020. p. 256–263.

_____. Enhancing dynamism in management and network slice establishment through deep learning. In: **2021 International Conference on Information Networking (ICOIN)**. Jeju Island, Korea (South): IEEE, 2021. p. 321–326.

_____. Deploying scalable and stable XDP-Based network slices through NASOR framework for low-latency applications. In: BAROLLI, L.; WOUNGANG, I.; ENOKIDO, T. (Ed.). **Advanced Information Networking and Applications**. Cham: Springer International Publishing, 2021. p. 715–726. ISBN 978-3-030-75075-6.

_____. A flexible network and compute-aware orchestrator to enhance QoS in NFV-Based multimedia services. In: **2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)**. Krakow, Poland: IEEE, 2018. p. 512–519.

_____. A smart network and compute-aware orchestrator to enhance QoS on cloud-based multimedia services. **International Journal of Grid and Utility Computing**, Inderscience Publishers (IEL), v. 11, n. 1, p. 49–61, 2020.

MÜLLER, V. C.; BOSTROM, N. Future progress in artificial intelligence: A survey of expert opinion. Springer International Publishing, Cham, p. 555–572, 2016. Disponível em: <https://doi.org/10.1007/978-3-319-26485-1_33>.

NAKATSUKA, Y.; PAVERD, A.; TSUDIK, G. Pdot. **Proceedings of the 35th Annual Computer Security Applications Conference on - ACSAC '19**, ACM Press, 2019. Disponível em: <http://dx.doi.org/10.1145/3359789.3359793>.

NGMN Alliance. Description of network slicing concept. 2016. Disponível em: <https://www.ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf>.

NIKAEIN, N. et al. Network store: Exploring slicing in future 5G networks. In: **Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture**. New York, NY, USA: Association for Computing Machinery, 2015. (MobiArch '15), p. 8–13. ISBN 9781450336956. Disponível em: <https://doi.org/10.1145/2795381.2795390>.

NUNES, B. A. A. et al. A survey of Software-Defined Networking: Past, present, and future of programmable networks. **IEEE Communications Surveys Tutorials**, v. 16, n. 3, p. 1617–1634, Third 2014. ISSN 2373-745X.

OLIVA, A. de la et al. 5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals. **IEEE Communications Magazine**, v. 56, n. 8, p. 78–84, 2018.

ORDONEZ-LUCENA, J. et al. Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. **IEEE Communications Magazine**, v. 55, n. 5, p. 80–87, May 2017. ISSN 1558-1896.

PANG, Z. et al. A survey of cloudlet based mobile computing. In: **2015 International Conference on Cloud Computing and Big Data (CCBD)**. Shanghai, China: IEEE, 2015. p. 268–275.

PAUL, M. et al. **Open Network Fundation document "Applying SDN Architecture to 5G Slicing", April 2016**. Palo Alto, CA, USA: ONF, 2016.

PAULSEN, S.; UHL, T.; NOWICKI, K. Influence of the jitter buffer on the quality of service VoIP. In: **2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)**. Budapest, Hungary: IEEE, 2011. p. 1–5.

PÉK, G.; BUTTYÁN, L.; BENCSÁTH, B. A survey of security issues in hardware virtualization. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 45, n. 3, jul. 2013. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/2480741.2480757>.

PETERSON, L.; CULLER, D.; ANDERSON, T. Planetlab: A testbed for developing and deploying network services. **Technical White Paper**, 2002.

PONTI, M. A. et al. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: **2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)**. Niterói, Brazil: IEEE, 2017. p. 17–41.

POPEK, G. J.; GOLDBERG, R. P. Formal requirements for virtualizable third generation architectures. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 17, n. 7, p. 412–421, jul. 1974. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/361011.361073>.

POPOVSKI, P. et al. 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. **IEEE Access**, v. 6, p. 55765–55779, 2018.

PORETSKY, S. et al. Terminology for benchmarking network-layer traffic control mechanisms. **IETF RFC 4689**, 2006.

PREVIDI, S. et al. Source packet routing in networking (spring) problem statement and requirements. **Internet Requests for Comments, RFC Editor, RFC**, v. 7855, 2016.

_____. IPv6 segment routing header (SRH). **Internet Engineering Task Force, Internet-Draft draft-ietf-6man-segment-routing-header-08**, 2017.

QIAN, H.; RABINOVICH, M.; AL-QUDAH, Z. Bringing local DNS servers close to their clients. In: **2011 IEEE Global Telecommunications Conference - GLOBECOM 2011**. Houston, TX, USA: IEEE, 2011. p. 1–6. ISSN 1930-529X.

RAJARAMAN, V. Johnmccarthy—father of artificial intelligence. **Resonance**, Springer, v. 19, n. 3, p. 198–207, 2014.

REFSLUND, B. The outsourcing challenge: organizing workers across fragmented production networks. **European Planning Studies**, Routledge, v. 24, n. 5, p. 1034–1036, 2016. Disponível em: <https://doi.org/10.1080/09654313.2016.1152738>.

RESHETOVA, E. et al. Security of OS-Level virtualization technologies. In: BERNSMED, K.; FISCHER-HÜBNER, S. (Ed.). **Secure IT Systems**. Cham: Springer International Publishing, 2014. p. 77–93. ISBN 978-3-319-11599-3.

RICHART, M. et al. Resource slicing in virtual wireless networks: A survey. **IEEE Transactions on Network and Service Management**, v. 13, n. 3, p. 462–476, 2016.

RIMAL, B. P.; CHOI, E.; LUMB, I. A taxonomy and survey of cloud computing systems. In: **2009 Fifth International Joint Conference on INC, IMS and IDC**. Seoul, Korea (South): IEEE, 2009. p. 44–51.

RITCHIE, D. M. The evolution of the unix time-sharing system. In: TOBIAS, J. M. (Ed.). **Language Design and Programming Methodology**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1980. p. 25–35. ISBN 978-3-540-38579-0.

RITCHIE, D. M.; THOMPSON, K. The unix time-sharing system†. **Bell System Technical Journal**, v. 57, n. 6, p. 1905–1929, 1978. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1978.tb02136.x>.

RODRIGUEZ, V. Q.; GUILLEMIN, F.; BOUBENDIR, A. 5G E2E network slicing management with ONAP. In: **2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)**. Paris, France: IEEE, 2020. p. 87–94.

ROST, P. et al. Mobile network architecture evolution toward 5G. **IEEE Communications Magazine**, v. 54, n. 5, p. 84–91, 2016.

Rost, P. et al. Network slicing to enable scalability and flexibility in 5G mobile networks. **IEEE Communications Magazine**, v. 55, n. 5, p. 72–79, 2017.

RYU. Controller. **URL:"https://osrg. github. io/ryu**, 2014. Disponível em: <https://ryu-sdn.org/>.

SAAD, W.; BENNIS, M.; CHEN, M. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. **IEEE Network**, v. 34, n. 3, p. 134–142, 2020.

SABELLA, D. et al. Mobile-Edge computing architecture: The role of MEC in the Internet of Things. **IEEE Consumer Electronics Magazine**, v. 5, n. 4, p. 84–91, Oct 2016. ISSN 2162-2256.

SABOORIAN, T.; THIEBAUT, L.; XIANG, A. Network slicing and 3GPP service and Systems Aspects (SA) standard. **IEEE Softwarization, Dec**, 2017.

SAHOO, J.; MOHAPATRA, S.; LATH, R. Virtualization: A survey on concepts, taxonomy and associated security issues. In: **2010 Second International Conference on Computer and Network Technology**. Bangkok, Thailand: IEEE, 2010. p. 222–226. ISSN null.

SAHU, M.; DASH, R. A survey on deep learning: Convolution Neural Network (CNN). In: **Intelligent and Cloud Computing**. [S.l.]: Springer, 2021. p. 317–325.

SALIM, J. et al. **RFC3549: Linux Netlink as an IP Services Protocol**. USA: RFC Editor, 2003.

SAMDANIS, K.; COSTA-PEREZ, X.; SCIANCALEPORE, V. From network sharing to multi-tenancy: The 5G network slice broker. **IEEE Communications Magazine**, v. 54, n. 7, p. 32–39, July 2016. ISSN 1558-1896.

SANAEI, Z. et al. Heterogeneity in mobile cloud computing: Taxonomy and open challenges. **IEEE Communications Surveys Tutorials**, v. 16, n. 1, p. 369–392, First 2014. ISSN 2373-745X.

SATYANARAYANAN, M. et al. The case for VM-Based Cloudlets in mobile computing. **IEEE Pervasive Computing**, v. 8, n. 4, p. 14–23, Oct 2009. ISSN 1558-2590.

SEDGEWICK, R. **Algorithms in c, Part 5: Graph Algorithms, Third Edition**. Third. USA: Addison-Wesley Professional, 2001. ISBN 9780768685329.

SHAIKH, A.; TEWARI, R.; AGRAWAL, M. On the effectiveness of DNS-based server selection. In: **Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)**. Anchorage, AK, USA: IEEE, 2001. v. 3, p. 1801–1810 vol.3. ISSN 0743-166X.

SHAN, Z. et al. Facilitating inter-application interactions for OS-Level virtualization. In: **Proceedings of the 8th ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments**. New York, NY, USA: Association for Computing Machinery, 2012. (VEE '12), p. 75–86. ISBN 9781450311762. Disponível em: <https://doi.org/10.1145/2151024.2151036>.

SHEA, R.; LIU, J. Network interface virtualization: challenges and solutions. **IEEE Network**, v. 26, n. 5, p. 28–34, Sep. 2012. ISSN 1558-156X.

SHEN, D.; WU, G.; SUK, H.-I. Deep learning in medical image analysis. **Annual Review of Biomedical Engineering**, v. 19, n. 1, p. 221–248, 2017. PMID: 28301734. Disponível em: <https://doi.org/10.1146/annurev-bioeng-071516-044442>.

SHERRY, J. et al. Making middleboxes someone else's problem: Network processing as a cloud service. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 42, n. 4, p. 13–24, ago. 2012. ISSN 0146-4833. Disponível em: <http://doi-acm-org.ez34.periodicos.capes.gov.br/10.1145/2377677.2377680>.

SHERWOOD, R. et al. Can the production network be the testbed? In: **Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation**. USA: USENIX Association, 2010. (OSDI'10), p. 365–378.

SHIN, M.; NAM, K.; KIM, H. Software-defined networking (SDN): A reference architecture and open APIs. In: **2012 International Conference on ICT Convergence (ICTC)**. Jeju, Korea (South): IEEE, 2012. p. 360–361. ISSN 2162-1241.

SHULMAN, H. Pretty bad privacy: Pitfalls of DNS encryption. In: **Proceedings of the 13th Workshop on Privacy in the Electronic Society**. New York, NY, USA: ACM, 2014. (WPES '14), p. 191–200. ISBN 978-1-4503-3148-7. Disponível em: <http://doi.acm.org/10.1145/2665943.2665959>.

SILVA, A. P. et al. 5GinFIRE: An end-to-end open5G vertical network function ecosystem. **Ad Hoc Networks**, v. 93, p. 101895, 2019. ISSN 1570-8705. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1570870518309387>.

SILVA, F. S. D. et al. NECOS Project: Towards lightweight slicing of cloud federated infrastructures. In: **2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)**. Montreal, QC, Canada: IEEE, 2018. p. 406–414.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SINH, D. et al. SDN/NFV — a new approach of deploying network infrastructure for IoT. In: **2018 27th Wireless and Optical Communication Conference (WOCC)**. Hualien, Taiwan: IEEE, 2018. p. 1–5. ISSN 2379-1276.

SKORIN-KAPOV, L. et al. Analysis of end-to-end QoS for networked virtual reality services in UMTS. **IEEE Communications Magazine**, v. 42, n. 4, p. 49–55, 2004.

SONKOLY, B. et al. Multi-domain service orchestration over networks and clouds: A unified approach. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 45, n. 4, p. 377–378, ago. 2015. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/2829988.2790041>.

SORET, B. et al. Fundamental tradeoffs among reliability, latency and throughput in cellular networks. In: **2014 IEEE Globecom Workshops (GC Wkshps)**. Austin, TX, USA: IEEE, 2014. p. 1391–1396.

SOUSA, N. F. S. de et al. Network service orchestration: A survey. **Computer Communications**, v. 142-143, p. 69 – 94, 2019. ISSN 0140-3664. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366418309502>.

STIEMERLING, M. et al. TORI: User provided future networking testbeds. In: **2009 IEEE International Conference on Communications Workshops**. Dresden, Germany: IEEE, 2009. p. 1–6.

STOLLER, M. H. R. R. L. et al. Large-scale virtualization in the emulab network testbed. In: **USENIX Annual Technical Conference, Boston, MA**. Boston, MA: IEEE, 2008.

SUÑÉ, M. et al. Design and implementation of the OFELIA FP7 facility: The european OpenFlow testbed. **Computer Networks**, v. 61, p. 132 – 150, 2014. ISSN 1389-1286. Special issue on Future Internet Testbeds – Part I. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613004301>.

SUN, G. et al. Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks. **Future Generation Computer Systems**, v. 91, p. 347 – 360, 2019. ISSN 0167-739X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X1831848X>.

TALEB, T. et al. On multi-domain network slicing orchestration architecture and federated resource control. **IEEE Network**, v. 33, n. 5, p. 242–252, Sep. 2019. ISSN 1558-156X.

_____. White paper on 6g networking. 2020.

_____. PERMIT: Network slicing for personalized 5G mobile telecommunications. **IEEE Communications Magazine**, v. 55, n. 5, p. 88–93, May 2017. ISSN 1558-1896.

_____. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. **IEEE Communications Surveys Tutorials**, v. 19, n. 3, p. 1657–1681, thirdquarter 2017. ISSN 2373-745X.

TIANFIELD, H. Cloud computing architectures. In: **2011 IEEE International Conference on Systems, Man, and Cybernetics**. Anchorage, AK, USA: IEEE, 2011. p. 1394–1399. ISSN 1062-922X.

TORAL-CRUZ, H.; PATHAN, A.-S. K.; PACHECO, J. C. R. Accurate modeling of VoIP traffic QoS parameters in current and future networks with multifractal and markov models. **Mathematical and Computer Modelling**, Elsevier, v. 57, n. 11-12, p. 2832–2845, 2013.

TOUCH, J. D. et al. A global X-bone for network experiments. In: **First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities**. Trento, Italy: IEEE, 2005. p. 194–203.

TOWARD, A. Network-wide decision making: Toward a wafer-thin control plane. 2004.

TRAN, T. X. et al. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. **IEEE Communications Magazine**, v. 55, n. 4, p. 54–61, 2017.

TURING, A. M. Computing machinery and intelligence. In: _____. **Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer**. Dordrecht: Springer Netherlands, 2009. p. 23–65. ISBN 978-1-4020-6710-5. Disponível em: <https://doi.org/10.1007/978-1-4020-6710-5_3>.

VELÁSQUEZ, K.; GAMESS, E. A comparative analysis of WAN emulators. In: **Proceedings of the 7th Latin American Networking Conference**. New York, NY, USA: Association for Computing Machinery, 2012. (LANC '12), p. 44–51. ISBN 9781450317504. Disponível em: <https://doi.org/10.1145/2382016.2382024>.

VENTRE, P. L. et al. SDN architecture and southbound APIs for IPv6 segment routing enabled wide area networks. **IEEE Transactions on Network and Service Management**, v. 15, n. 4, p. 1378–1392, Dec 2018. ISSN 2373-7379.

VERBELEN, T. et al. Cloudlets: Bringing the cloud to the mobile user. In: **Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services**. New York, NY, USA: Association for Computing Machinery, 2012. (MCS '12), p. 29–36. ISBN 9781450313193. Disponível em: <https://doi.org/10.1145/2307849.2307858>.

WALTERS, J. P. et al. A comparison of virtualization technologies for HPC. In: **22nd International Conference on Advanced Information Networking and Applications (AINA 2008)**. Gino-wan, Japan: IEEE, 2008. p. 861–868. ISSN 2332-5658.

WANG, L. et al. Cloud computing: a perspective study. **New Generation Computing**, Springer, v. 28, n. 2, p. 137–146, 2010.

WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. Rio de Janeiro - RJ: Elsevier Brasil, 2017. v. 2.

WIJETHILAKA, S.; LIYANAGE, M. Survey on network slicing for internet of things realization in 5G networks. **IEEE Communications Surveys Tutorials**, p. 1–1, 2021.

Wu, X. et al. State of the art and research challenges in the security technologies of network function virtualization. **IEEE Internet Computing**, p. 1–1, 2019. ISSN 1941-0131.

WUHIB, F.; YANGGRATOKE, R.; STADLER, R. Allocating compute and network resources under management objectives in large-scale clouds. **Journal of Network and Systems Management**, v. 23, n. 1, p. 111–136, 2015. ISSN 1573-7705. Disponível em: <https://doi.org/10.1007/s10922-013-9280-6>.

YANG, L. et al. **RFC3746: Forwarding and Control Element Separation (ForCES) Framework**. USA: RFC Editor, 2004.

YE, Q. et al. End-to-End quality of service in 5G networks: Examining the effectiveness of a network slicing framework. **IEEE Vehicular Technology Magazine**, v. 13, n. 2, p. 65–74, 2018.

YOUSAF, F. Z. et al. NFV and SDN—key technology enablers for 5G networks. **IEEE Journal on Selected Areas in Communications**, v. 35, n. 11, p. 2468–2478, Nov 2017. ISSN 1558-0008.

_____. MANOaaS: A multi-tenant nfv mano for 5G network slices. **IEEE Communications Magazine**, v. 57, n. 5, p. 103–109, May 2019. ISSN 1558-1896.

YOUSEFF, L. et al. Paravirtualization for HPC systems. In: MIN, G. et al. (Ed.). **Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 474–486. ISBN 978-3-540-49862-9.

YU, H. et al. Cloud computing and security challenges. In: **Proceedings of the 50th Annual Southeast Regional Conference**. New York, NY, USA: Association for Computing Machinery, 2012. (ACM-SE '12), p. 298–302. ISBN 9781450312035. Disponível em: <https://doi.org/10.1145/2184512.2184581>.

ZHANG, H. et al. Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges. **IEEE Communications Magazine**, v. 55, n. 8, p. 138–145, Aug 2017. ISSN 1558-1896.

ZHANG, J.; SUN, C.; YANG, C. Resource allocation in urllc with online learning for mobile users. In: IEEE. **2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)**. [S.l.], 2021. p. 1–5.

ZHANG, S. An overview of network slicing for 5G. **IEEE Wireless Communications**, v. 26, n. 3, p. 111–117, 2019.

ZHENG, L.; ZHANG, L.; XU, D. Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP). In: **ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)**. Helsinki, Finland: IEEE, 2001. v. 1, p. 122–126 vol.1.

ZHOU, S. K. et al. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. **Proceedings of the IEEE**, v. 109, n. 5, p. 820–838, 2021.

ZHOU, X.; WEI, J.; XU, C.-Z. Quality-of-service differentiation on the internet: A taxonomy. **Journal of Network and Computer Applications**, v. 30, n. 1, p. 354 – 383, 2007. ISSN 1084-8045. Network and Information Security: A Computational Intelligence Approach. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804505000329>.

ZHU, L. et al. T-DNS: Connection-oriented DNS to improve privacy and security (poster abstract). **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 44, n. 4, p. 379–380, ago. 2014. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/2740070.2631442>.

Apêndices

APÊNDICE A

Gerenciamento do Ciclo de Vida das Fatias de Rede

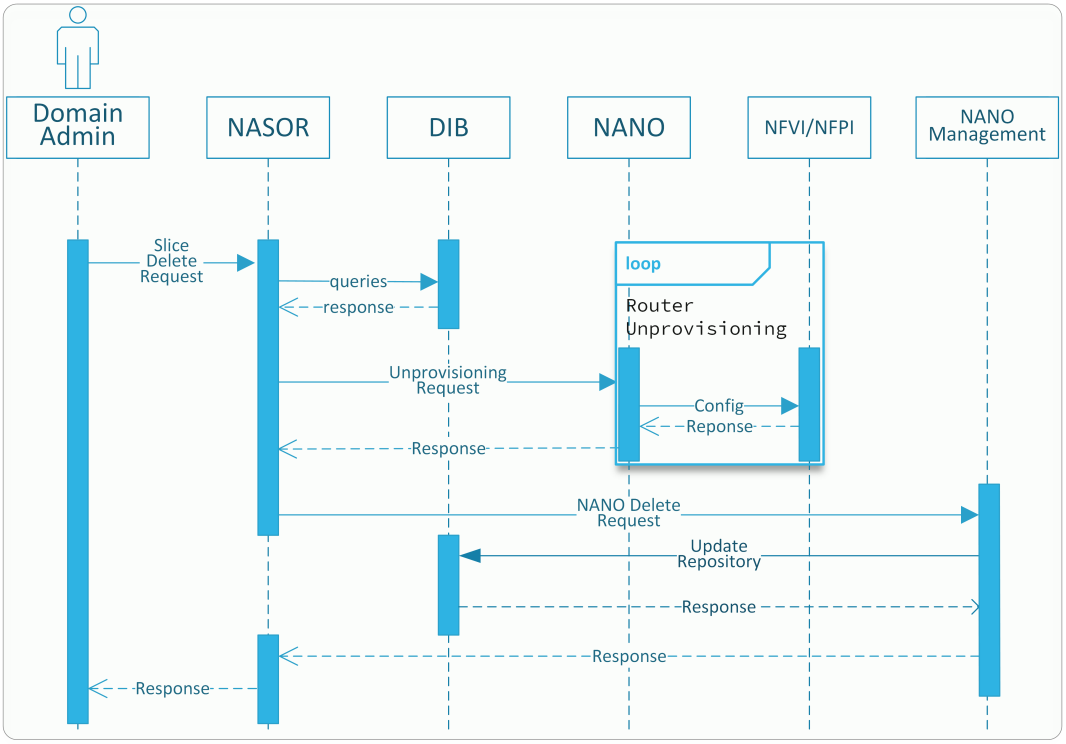

Figura 38 – Sequencia: desaprovisionamento de uma Fatia de Rede em um domínio singular.

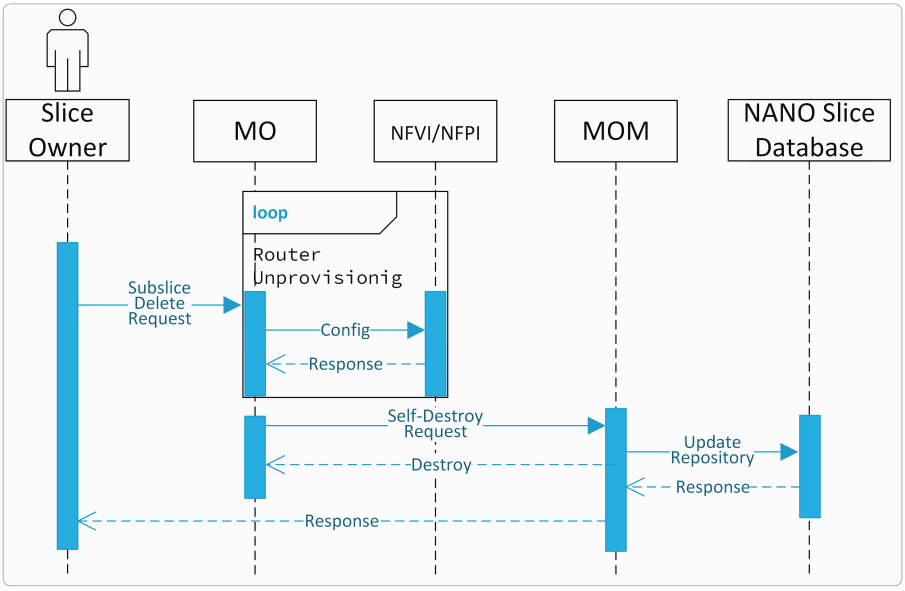

Figura 39 – Sequencia: desaprovisionamento de uma Fatia de Rede recursiva em um domínio singular.

APÊNDICE **B**

Exemplo Rotas do Cenário Multidomínio

```
Codes: K - kernel route, C - connected, S - static, R - RIPng,
O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
> - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
B>* 2000:f0d0:2001:a::/64 [20/0] via fe80::a00:27ff:fe98:7d1a, eth2,
    00:11:33
O   2001:470:28:5a1::/64 [110/4] is directly connected, eth0,
    00:10:58
C>* 2001:470:28:5a1::/64 is directly connected, eth0
O>* 2001:470:28:5a2::/64 [110/19] via fe80::a00:27ff:febf:8601, eth0,
     00:10:57
O   2001:470:28:5a3::/64 [110/21] via fe80::a00:27ff:febf:8601, eth0,
     00:10:52
C>* 2001:470:28:5a3::/64 is directly connected, eth1
C>* 2607:f0d0:2001::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth3
C * fe80::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth0
```

Tabela de Rotas B.1 – Roteador 3.

```
Codes: K - kernel route, C - connected, S - static, R - RIPng,
```

```
2 O - OSPFv6, I - IS-IS, B - BGP, A - Babel,
  > - selected route, * - FIB route
4
  C>* ::1/128 is directly connected, lo
6 C>* 2000:f0d0:2001:a::/64 is directly connected, eth1
  B>* 2001:470:28:5a1::/64 [20/0] via fe80::a00:27ff:fee5:ef9a, eth0,
      00:10:44
8 B>* 2001:470:28:5a2::/64 [20/19] via fe80::a00:27ff:fee5:ef9a, eth0,
      00:09:44
  B>* 2001:470:28:5a3::/64 [20/0] via fe80::a00:27ff:fee5:ef9a, eth0,
      00:10:44
10 B   2607:f0d0:2001::/64 [20/0] via fe80::a00:27ff:fee5:ef9a, eth0,
      00:10:44
  C>* 2607:f0d0:2001::/64 is directly connected, eth0
12 C * fe80::/64 is directly connected, eth2
  C * fe80::/64 is directly connected, eth1
14 C>* fe80::/64 is directly connected, eth0
```

Tabela de Rotas B.2 – Roteador 4.

APÊNDICE C

Exemplo de Comandos para gerenciamento de Roteamento por Segmentos

Comportamento	**Descrição**
End	A função Endpoint (“End”) é a função mais básica. Retira o cabeçalho *SRH*.
End.X	Ponto de extremidade com conexão cruzada a uma matriz de adjacências da camada 3
End.DX2	Ponto de extremidade com decapsulação e conexão cruzada da camada 2 para um OIF (interface de saída)
End.Dx4	Terminal com decapsulação e conexão cruzada com uma adjacência *IPv4*
End.DX6	Terminal com decapsulação e conexão cruzada com uma adjacência do *IPv6*
End.AD4	Ponto de extremidade para o *IPv4 SR* que desconhece o APP por meio de proxy dinâmico
End.AD6	Ponto de extremidade para o *IPv6 SR* sem conhecimento de APP por meio de proxy dinâmico
End.AM	Ponto final para o APP que não conhece o *SR* via mascaramento
End.EAD4	Comportamento End.AD4 estendido que permite que *VNFs* que desconhecem o *SR* sejam o último *Service Function (SF)* no *Service Function Chaining (SFC)*
End.EAD6	Comportamento End.AD6 estendido que permite que *VNFs* que desconhecem o *SR* sejam o último *SF* no *SFC*

Tabela 11 – Comportamentos disponíveis na distribuição *Linux* com suporte a *SR*.

Comando	**Descrição**
srconf localsid show	Mostra ao usuário a lista de *SIDs*, seus comportamentos e a quantidade de pacotes que satisfez a combinação do cabeçalho *SRH*
srconf localsid add 1::D6 END.DX6 ip A::2 veth1_1	Adiciona um *SID* 1::D6 com o comportamento END.DX6 com destino ao *IPv6* a::2 deverá sair pela interface veth1_1
srconf localsid add 2::AD6:F3 END.AD6 ip 2:F3::F3 veth3_2 veth3_1	Adiciona um *SID* 2::AD6:F3 com o comportamento END.AD6 com destino ao *IPv6* 2:F3::F3 que entrou pela Interface veth3_2 deverá saír pela veth3_1
srconf localsid add 2:: END	Adiciona um SID 2:: com comportamento END. Todo pacote com cabecalho *SRH* para o destino 2:: será desencapsulado.
srconf loalsid del 2:: END	Remove o *SID* 2:: da Tabela do *Kernel*

Tabela 12 – Exemplos de Comandos para configuração do *SR*.

APÊNDICE **D**

Estatística Descritiva: Cenário Experimental 1

Variável	N	Média (ms)	Erro Padrão	Desvio Padrão	Mínimos	Percentil: Q1	Mediana	Percentil: Q3	Máximo
LW-DNS	2000	277.88	9.48	423.95	1.48	7.31	107.27	362.35	3429.86
Google DNS	2000	241.4	10.2	455.1	19.7	32.2	46.0	302.8	3500.0
Open DNS	2000	481.8	15.6	697.2	26.2	62.5	273.6	508.2	3500.0

Tabela 13 – Estatística Descritiva – Caso de Uso *LW-DNS*.

Variável	N	Média (ms)	Erro Padrão	Desvio Padrão	Mínimos	Percentil: Q1	Mediana	Percentil: Q3	Máximo
Baseline	208	1.4591	0.0162	0.2344	0.3660	1.3700	1.5000	1.5700	2.9000
SRH	208	1.6479	0.0272	0.3924	0.3950	1.5200	1.5950	1.7000	4.1900
VPN	208	6.931	0.240	3.458	3.290	6.295	6.550	6.917	48.200

Tabela 14 – Estatística Descritiva – *Overhead* Fatiamento de Redes *NASOR*.

APÊNDICE E

Estatística Descritiva: Cenário Experimental 2

Variável	N	Média (ms)	Erro Padrão	Desvio Padrão	Mínimos	Percentil: Q1	Mediana	Percentil: Q3	Máximo
Network Performance-based	50	12.279	0.00747	0.0528	12.206	12.244	12.261	12.290	12.467
BGP-based	50	1.0715	0.000340	0.00241	1.0633	1.0702	1.0717	1.0729	1.0754

Tabela 15 – Tempo de Implantação de Fatias de Rede.

Variável	N	Média (ms)	Erro Padrão	Desvio Padrão	Mínimos	Percentil: Q1	Mediana	Percentil: Q3	Máximo	Coeficiênte de Variação
Network Performance-based	4500	1.0143	0.00447	0.3000	0.00000	0.8042	0.9660	1.1658	3.2523	29.58
BGP-based	4500	23.602	0.0693	4.652	0.0000	20.397	23.613	26.712	38.453	19.71

Tabela 16 – *Jitter* percebido sobre as Fatias de Rede.

APÊNDICE **F**

Configuração dos Roteadores do Experimento 2

Abaixo está disponível o arquivo de configuração do Roteador 3, que é um rodeador de borda do Domínio A. Esse roteador divulga para o Domínio B rotas internas computadas pelo algoritmo *OSPF* que se referem a área **0.0.0.0**.

```
!
interface eth0
no ipv6 nd suppress-ra
ipv6 ospf6 cost 4
ipv6 ospf6 network broadcast
!
interface eth1
no ipv6 nd suppress-ra
ipv6 ospf6 cost 6
ipv6 ospf6 network broadcast
!
interface eth2
no ipv6 nd suppress-ra
ipv6 ospf6 network broadcast
!
interface lo
!
router ospf6
redistribute bgp
router-id 0.0.0.3
```

```
interface eth0 area 0.0.0.0
!
router bgp 16735
bgp router-id 192.168.0.3
neighbor 2607:f0d0:2001::2 remote-as 7675
neighbor 2607:f0d0:2001::2 description "Domain B"
!
address-family ipv6
redistribute connected
redistribute ospf6
neighbor 2607:f0d0:2001::2 activate
exit-address-family
!
```

Configuração F.1 – Roteador 3

Adicionalmente, o Roteador 4 do Domínio B estabelece uma uma sessão *BGP* com o Roteador 3 e, similar a seu par, divulga rotas internas do seu domínio, também calculadas pelo *OSPF*. As rotas que são divulgadas ao Domínio A pelo Roteador 4 referem-se a área **1.1.1.1**. O arquivo contendo essa configuração é conforme abaixo:

```
!
interface eth0
no ipv6 nd suppress-ra
ipv6 ospf6 cost 4
ipv6 ospf6 network broadcast
!
interface eth1
no ipv6 nd suppress-ra
ipv6 ospf6 cost 6
ipv6 ospf6 network broadcast
!
interface eth2
no ipv6 nd suppress-ra
ipv6 ospf6 network broadcast
!
interface lo
!
router ospf6
```

```
redistribute bgp
router-id 0.0.0.3
interface eth0 area 1.1.1.1
!
router bgp 7675
bgp router-id 192.168.0.250
redistribute connected
neighbor 2607:f0d0:2001::1 remote-as 16735
neighbor 2607:f0d0:2001::1 description "Domain A"
!
address-family ipv6
redistribute connected
redistribute ospf6
network 2000:f0d0:2001:a::/64
neighbor 2607:f0d0:2001::1 activate
exit-address-family
!
```

Configuração F.2 – Roteador 4

APÊNDICE **G**

Arquivo *YANG* de Configuração de uma Fatia de Serviço: Rede e Computação

```
--- !Multi-Domain Slice Template Example
nst:
  -   id: slice_voip-example_nstd
    name: slice_voip-example_nstd
    SNSSAI-identifier:
      slice-service-type: IoT
    quality-of-service:
      id: 1
    #AS Number merely exemplary
    asns:
      -   16735 #Algar Telecom
      -   26599 #Telefonica
      -   1916 #Rede Nacional de Ensino e Pesquisa (RNP)
    slice_policy:
      -   id: 1
      name: network_performance-aware
    netslice-subnet:
      -   id: slice_voip-example_nsd_1
        is-shared-nss: "false"
        description: NetSlice Subnet (service) composed by 2 vnf with
    2 cp
```

```
        nsd-ref: voip_2vnf_nsd
    netslice-vld:
      -   id: slice_voip_vld_mgmt
        name: slice_voip_vld_mgmt
        type: ELAN
        mgmt-network: "true"
    nss-connection-point-ref:
      -   nss-ref: slice_voip-example_nsd_1
    nsd-connection-point-ref: nsd_cp_mgmt
      -   id: slice_voip_vld_data1
        name: slice_voip_vld_data1
        type: ELAN
    nss-connection-point-ref:
      -   nss-ref: slice_voip-example_nsd_1
      nsd-connection-point-ref: nsd_cp_data
```

Configuração G.1 – *NSTD*.

```
--- !Multi-Domain VNF Template Example
vnfd:vnfd-catalog:
vnfd:
  -   id: lw-dns
    name: lw-dns
    short-name: lw-dns
    description: Simple VNF example with a Bind9-DNS
    vendor: OSM
    version: '1.0'

  # Place the logo as png in icons directory and provide the name
  here
  logo: lw-dns.png

  # Management interface
  mgmt-interface:
  cp: eth0

  # Atleast one VDU need to be specified
  vdu:
    -   id: lw-dns_vnfd-CN
      name: lw-dns_vnfd-CN
```

```
22          description: lw-dns_vnfd-CN
            count: 1
24
      # Flavour of the VM to be instantiated for the VDU
26    # flavor below can fit into m1.micro
      vm-flavor:
28      vcpu-count: 1
        memory-mb: 256
30      storage-gb: 2

32    # Image/checksum or image including the full path
        image: sameersbn/bind:latest
34      #checksum:

36    interface:
        # Specify the external interfaces
38      # There can be multiple interfaces defined
        -   name: eth0
40        type: EXTERNAL
          virtual-interface:
42        type: VIRTIO
          bandwidth: '0'
44        vpci: 0000:00:0a.0
          external-connection-point-ref: eth0
46
        connection-point:
48        -   name: eth0
            type: VPORT
```

Configuração G.2 – *VNFD*.