

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

Vinícius Fülber Garcia

**UM ARCABOUÇO FERRAMENTAL PARA IMPLANTAÇÃO
PERSONALIZÁVEL DE SERVIÇOS DE REDE
VIRTUALIZADOS**

Santa Maria, RS
2019

Vinicius Fülber Garcia

**UM ARCABOUÇO FERRAMENTAL PARA IMPLANTAÇÃO PERSONALIZÁVEL
DE SERVIÇOS DE REDE VIRTUALIZADOS**

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

Orientador: Prof. Dr. Carlos Raniery Paula dos Santos

Santa Maria, RS

2019

Garcia, Vinícius Fülber

Um Arcabouço Ferramental para Implantação Personalizável de Serviços de Rede Virtualizados / por Vinícius Fülber Garcia. – 2019.
139 f.: il.; 30 cm.

Orientador: Carlos Raniery Paula dos Santos
Dissertação (Mestrado) - Universidade Federal de Santa Maria,
Centro de Tecnologia, Pós-Graduação em Ciência da Computação , RS,
2019.

1. NFV. 2. SFC. 3. RA. 4. Serviço. 5. Implantação. I. dos Santos,
Carlos Raniery Paula. II. Um Arcabouço Ferramental para Implantação
Personalizável de Serviços de Rede Virtualizados.

© 2019

Todos os direitos autorais reservados a Vinícius Fülber Garcia. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

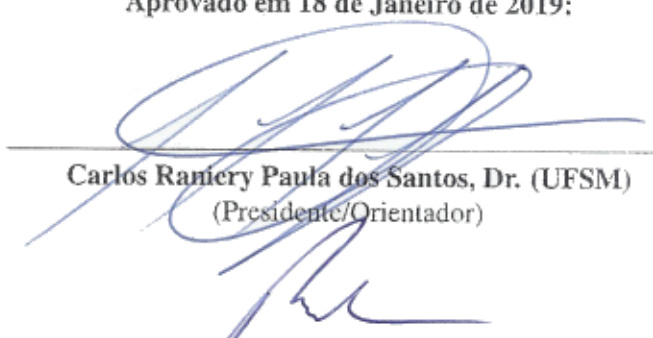
E-mail: vfulber@inf.ufsm.br

Vinícius Fülber Garcia

**UM ARCABOUÇO FERRAMENTAL PARA IMPLANTAÇÃO PERSONALIZÁVEL
DE SERVIÇOS DE REDE VIRTUALIZADOS**

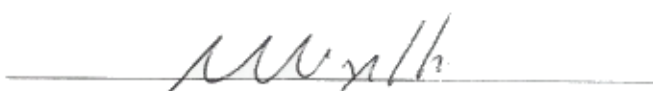
Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC) da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Ciência da Computação**.

Aprovado em 18 de Janeiro de 2019:



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

Raul Ceretta Nunes, Dr. (UFSM)



Marcelo Caggiani Luizelli, Dr. (UNIPAMPA)

Santa Maria, RS

2019

DEDICATÓRIA

Dedico este trabalho aos meus pais Celso Garcia e Nadir Garcia; a minha companheira de todas as horas Luísa Lucca; aos irmãos que a vida me deu Maurício Schneider e Guilherme Gaiardo; aos amigos, familiares e mestres.

AGRADECIMENTOS

*Aos meus pais, **Nadir Garcia** e **Celso Garcia**, pelo amor, carinho e incentivo incondicional. Vocês foram, são e sempre serão meus maiores exemplos de vida.*

*A minha companheira, amor e melhor amiga **Luísa Lucca** por estar ao meu lado durante toda esta etapa, sempre solícita e carinhosa, presente tanto nos momentos difíceis quanto nas comemorações de cada conquista alcançada.*

*Aos amigos **Maurício Schneider** e **Guilherme Gaiardo**, considerados por mim como grandes irmãos, que sempre estiveram próximos nas etapas mais importantes da minha vida, passando boas energias e torcendo pelo meu sucesso.*

*Aos demais **amigos** que, seja em Santo Ângelo, em Santa Maria ou em qualquer outro lugar do mundo, me possibilitaram viver ótimos momentos. Ainda, gostaria de agradecer em especial aos colegas **Matheus Garay**, **Thales Tavares**, **Leonardo Marcuzzo**, **Maurício Donato** e **Brenda Salenave** que, pessoalmente ou virtualmente, estiveram presentes no meu dia a dia nesses últimos dois anos.*

*Ao laboratório **GRECA** que me acolheu de braços abertos durante todo o período do mestrado. Nesse mesmo, tive a oportunidade e o privilégio de trabalhar e compartilhar experiências com excelentes colegas, em especial com meus veteranos e amigos **Nilton Camargo** e **Anderson Monteiro**.*

*Ao grupo de trabalho **FENDE** que participou ativamente da minha formação, me permitindo conhecer e cooperar com pesquisadores de extrema competência e saber.*

*Finalmente, a todos os **professores** que dividiram seus conhecimentos comigo. Principalmente, ao professor **Carlos Raniery** que não apenas me orientou durante a elaboração deste trabalho, mas também na realização de projetos e decisões que o antecederam e que o sucederão. Também, não posso deixar de lembrar do professor **Sérgio Mergen**, cooperativo e incentivador desde meados da minha graduação.*

*Todos os aqui citados ocupam um lugar especial no meu coração, sem vocês nenhuma conquista teria real sentido. Desejo a todos muito sucesso e deixo aqui o meu mais profundo e significativo **ORBRIGADO!!***

*"We can only see a short distance ahead,
but we can see plenty there that needs to
be done."*

(ALAN TURING)

RESUMO

UM ARCABOUÇO FERRAMENTAL PARA IMPLANTAÇÃO PERSONALIZÁVEL DE SERVIÇOS DE REDE VIRTUALIZADOS

AUTOR: VINÍCIUS FÜLBER GARCIA

ORIENTADOR: CARLOS RANIERY PAULA DOS SANTOS

As atuais infraestruturas de telecomunicações são baseadas em diferentes funções de rede implementadas através de equipamentos dedicados capazes de identificar, tratar, modificar e executar os mais variados protocolos e serviços. Entretanto, dificuldades relacionadas à manutenção e atualização desses equipamentos contribuem com o processo conhecido como ossificação da Internet, prejudicando a sua evolução de maneira geral. Nesse contexto, o paradigma *Network Function Virtualization* surge como uma opção para migração dessas funções de um meio físico com equipamentos dedicados a um ambiente virtualizado instanciado sobre máquinas de uso geral, permitindo usufruir da flexibilidade de operações (*e.g.*, monitoramento, escala, compartilhamento) e baixo custo financeiro associados à virtualização. Com isso, novos desafios surgem e são gradualmente explorados pela comunidade científica em vista de viabilizar a adoção prática desse paradigma, entre os principais destaca-se o processo de implantação de serviços de rede virtualizados que, por sua vez, apresenta três etapas principais (composição, integração e agendamento da execução) cujo o conjunto é denominado *Network Function Virtualization Resource Allocation*. Logo, diversas soluções foram desenvolvidas com o objetivo de atender cenários de implantação com requisitos pontuais (*e.g.*, reduzir o atraso, reduzir o custo financeiro, aumentar o tempo de disponibilidade de núcleos de processamento) utilizando, para isso, modelos topológicos e de requisição de serviço particulares e orientados a aplicação. Sendo assim, esta dissertação propõe um arcabouço holístico contendo modelos, metodologias e soluções capazes de descrever e executar o processo de implantação para cenários e serviços genéricos através da definição de multi-requisitos customizados, abstraindo os detalhes de processamento de baixo nível e eliminando quaisquer adaptações de código fonte. As principais contribuições deste trabalho são: um modelo para a especificação de topologias de serviços virtualizados de rede; um modelo genérico destinado a criação de documentos de requisição de serviços virtualizados de rede; uma metodologia baseada em indexadores aplicável a qualquer etapa do *Network Function Virtualization Resource Allocation*; e duas soluções, uma de composição e outra de integração multi-domínio, que utilizam o ferramental construído. Por fim, o arcabouço foi avaliado através da experimentação das soluções utilizando um serviço de segurança, controle de conteúdo e balanceamento de carga para servidores HTTP e HTTPS. Os resultados obtidos demonstraram o correto comportamento operacional das soluções testadas e a representatividade de seus resultados em instanciações reais, sugerindo a viabilidade de utilização do arcabouço desenvolvido.

Palavras-chave: NFV. SFC. RA. Serviço. Implantação.

ABSTRACT

A FRAMEWORK FOR CUSTOMIZABLE DEPLOYMENT OF VIRTUALIZED NETWORK SERVICES

AUTHOR: VINÍCIUS FÜLBER GARCIA

ADVISOR: CARLOS RANIERY PAULA DOS SANTOS

The current telecommunication infrastructures rely on the use of network functions implemented in dedicated hardware which are able to identify, handle, modify, and execute many protocols and services. However, difficulties related to the maintenance and upgrade of these dedicated hardwares leads to a process called Internet ossification, limiting the evolution of the Internet. In this context, the Network Function Virtualization paradigm arises as an option to migrate these functions from a dedicated and physical environment to a virtualized one, completely running on commercial off-the-shelf servers and inheriting the flexibility (*e.g.*, monitoring, scale, sharing) typically found in virtualized environments. Thereby, new challenges emerge and are gradually explored by the scientific community to foster the adoption of NFV. Specifically, the deployment of virtualized network services is composed by three main stages (composition, embedding, and scheduling) whose are named as Network Function Virtualization Resource Allocation. Therefore, diverse solutions were developed to solve specific requirements for single deployment scenarios (*e.g.*, reducing the overall delay, minimizing the financial cost, increase the processing units availability) using, for that, particular and application oriented topological and service request models. This work, on the other side, proposes a holistic framework containing models, methods and solutions capable to describe and execute deployment process for generic scenarios through customized definitions, without the need to lead with low level processing details or source code modification. The main contributions are: a virtualized network service topology specification model; a virtualized network service request generic model; an indexing method applicable to any Network Function Virtualization Resource Allocation stage; and two solutions, one for composition and another for multi-domain embedding, which use the developed tools. Finally, the framework was evaluated through the experimentation of the developed solutions to a security, content control and load balancing service for HTTP and HTTPS servers. The obtained results showed the the tested solutions' correct operational behaviour and their results representativity in real instantiations, suggesting the using feasibility of the developed framework.

Keywords: NFV. SFC. RA. Service. Deployment.

LISTA DE FIGURAS

Figura 1 –	Arquitetura NFV Simplificada	22
Figura 2 –	Arquitetura de Referência para Implementação de SFC.....	24
Figura 3 –	Execução da Alocação de Recurso em NFV	27
Figura 4 –	Categorização de Cadeias de Função de Serviço	33
Figura 5 –	Regras de Produção da SCAG	54
Figura 6 –	Topologias Completamente Validadas	59
Figura 7 –	Bloco de Metadados.....	62
Figura 8 –	Bloco de Serviço	63
Figura 9 –	Bloco de Função Objetivo	64
Figura 10 –	Bloco de Políticas	65
Figura 11 –	Bloco de Implantação	67
Figura 12 –	Bloco de Serviço do Exemplo Padrão	73
Figura 13 –	Bloco de Função Objetivo do Exemplo Padrão.....	73
Figura 14 –	Bloco de Implantação do Exemplo Padrão (<i>Benchmark</i> - Início)	73
Figura 15 –	Bloco de Implantação do Exemplo Padrão (<i>Benchmark</i> - Continuação).....	73
Figura 16 –	Bloco de Implantação do Exemplo Padrão (Ramificações).....	73
Figura 17 –	Topologia Geral do Exemplo Padrão	75
Figura 18 –	Resolução de Ordenamentos Parciais no Exemplo Padrão	76
Figura 19 –	Redução de Ramificações no Exemplo Padrão	78
Figura 20 –	Avaliação de Elementos Operacionais no Exemplo Padrão	85
Figura 21 –	Topologia do Serviço de Teste	88
Figura 22 –	Cenário de <i>Benchmarking</i>	90
Figura 23 –	Bloco de Serviço do Serviço de Teste	93
Figura 24 –	Bloco de Função Objetivo do Serviço de Teste.....	93
Figura 25 –	Bloco de Implantação do Serviço de Teste (<i>Benchmark</i> - Início)	94
Figura 26 –	Bloco de Implantação do Serviço de Teste (<i>Benchmark</i> - Continuação).....	94
Figura 27 –	Cenário de Avaliação do Serviço de Teste	95
Figura 28 –	Taxa de Requisições HTTPS Atendidas	98
Figura 29 –	Tempo Médio de Conexão por Requisição HTTPS Atendida	98
Figura 30 –	Taxa de Requisições HTTP Atendidas.....	99
Figura 31 –	Tempo Médio de Conexão por Requisição HTTP Atendida.....	99
Figura 32 –	Modelo de Documento de Domínios (Domínios e Recursos).....	106
Figura 33 –	Modelo de Documento de Domínios (Métricas Locais e de Transição)	106
Figura 34 –	Documento de Domínios do Exemplo Padrão (Início)	108
Figura 35 –	Documento de Domínios do Exemplo Padrão (Continuação)	108
Figura 36 –	Topologias e Índices de Adequabilidade Topológica do Exemplo Padrão	108
Figura 37 –	Bloco de Políticas do Exemplo Padrão	109
Figura 38 –	Bloco de Implantação do Exemplo Padrão (<i>Flavours</i>)	109
Figura 39 –	Processos Eliminatórios no Exemplo Padrão.....	111
Figura 40 –	Processos de Avaliação no Exemplo Padrão	114
Figura 41 –	Documento de Domínios FENDE (Início)	123
Figura 42 –	Documento de Domínios FENDE (Continuação).....	123
Figura 43 –	Primeira Configuração Bloco de Políticas do Serviço de Teste	124
Figura 44 –	Segunda Configuração Bloco de Políticas do Serviço de Teste	124
Figura 45 –	Terceira Configuração Bloco de Políticas do Serviço de Teste	124

Figura 46 – Topologias Analisadas do Serviço de Teste	125
Figura 47 – Bloco de Implantação do Serviço de Teste (<i>Flavours</i> - Início)	126
Figura 48 – Bloco de Implantação do Serviço de Teste (<i>Flavours</i> - Continuação)	126
Figura 49 – Agregações das Melhores Distribuições por Topologia (Primeira Configuração)	128
Figura 50 – Médias das Melhores Distribuições por Topologia (Segunda Configuração) .	128
Figura 51 – Agregações e Médias das Melhores Distribuições por Topologia (Terceira Configuração).....	129

LISTA DE TABELAS

Tabela 1 –	Comparação de Especificações ETSI e IETF.....	25
Tabela 2 –	Sumarização de Características das Soluções de Composição	40
Tabela 3 –	Sumarização de Características das Soluções de Divisão e Mapeamento.....	44
Tabela 4 –	Notações para a Definição das Premissas do Modelo	49
Tabela 5 –	Símbolos Terminais da SCAG	53
Tabela 6 –	Padronização de Valores de Validação do Modelo	68
Tabela 7 –	Notações Utilizadas na Construção da Função Objetivo	79
Tabela 8 –	Resultados Gerais de Avaliação do Exemplo Padrão	86
Tabela 9 –	Descrição das Funções de Rede do Serviço de Teste	89
Tabela 10 –	<i>Benchmarking</i> HTTP e HTTPS das Funções de Rede do Serviço de Teste ...	91
Tabela 11 –	Composições Candidatas do Serviço de Teste.....	97
Tabela 12 –	Agregado e Taxa de Tráfego do Serviço de Teste.....	101
Tabela 13 –	Recursos Computacionais Utilizados no Serviço de Teste	102
Tabela 14 –	Notações Utilizadas na Avaliação de Distribuições Candidatas.....	117
Tabela 15 –	Processo de Avaliação de Distribuições Candidatas do Exemplo Padrão	119
Tabela 16 –	Indexação e Ranqueamento das Distribuições Candidatas do Exemplo Padrão	120
Tabela 17 –	Matriz de Adyacências das Métricas Coletadas nos Domínios FENDE.....	122
Tabela 18 –	Quantidade de Distribuições do Serviço de Teste.....	126
Tabela 19 –	Indexação das Melhores Distribuições Candidatas por Topologia do Serviço de Teste.....	127

LISTA DE ABREVIATURAS E SIGLAS

ACSR	<i>Algebra of Communicating Shared Resources</i>
ARP	<i>Address Resolution Protocol</i>
BGP	<i>Border Gateway Protocol</i>
CPU	<i>Central Process Unit</i>
DNS	<i>Domain Name System</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FENDE	<i>Federated Ecosystem for Offering, Distribution, and Execution of Virtual Network Functions</i>
GLC	<i>Gramática Livre de Contexto</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IAT	<i>Índice de Adequabilidade Topológica</i>
IAD	<i>Índice de Adequabilidade de Distribuição</i>
IETF	<i>Internet Engineering Task Force</i>
ILP	<i>Integer Linear Programming</i>
InP	<i>Infrastructure Providers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
IU	<i>Índice Unificado</i>
JSON	<i>JavaScript Object Notation</i>
KVM	<i>Kernel-based Virtual Machine</i>
MANO	<i>Management and Orchestration</i>
NF	<i>Network Function</i>
NFV	<i>Network Function Virtualization</i>
NFV-RA	<i>Network Function Virtualization Resource Allocation</i>
NFVI	<i>Network Function Virtualization Infrastructure</i>
NLTK	<i>Natural Language Toolkit</i>
NS	<i>Network Service</i>
pACSR	<i>Packet Algebra of Communicating Shared Resources</i>
PNF	<i>Physical Network Function</i>
PPGCC	<i>Programa de Pós-Graduação em Ciência da Computação</i>
QoS	<i>Quality of Service</i>

RG	<i>Resource Graph</i>
RL	<i>Reinforcement Learning</i>
RNP	Rede Nacional de Pesquisa
RTT	<i>Round Trip Time</i>
SAP	<i>Service Acces Point</i>
SCAG	<i>Service Chain Grammar</i>
SDN	<i>Software Defined Network</i>
SF	<i>Service Function</i>
SFC	<i>Service Function Chain</i>
SFC-C	<i>Service Function Chain Composition</i>
SFC-E	<i>Service Function Chain Embedding</i>
SFC-S	<i>Service Function Chain Scheduling</i>
SFCC	<i>Service Function Chain Controller</i>
SFCR	<i>Service Function Chain Request</i>
SFF	<i>Service Function Forwarder</i>
SFP	<i>Service Function Proxy</i>
SG	<i>Service Graph</i>
SL	<i>Service Link</i>
SLA	<i>Service Level Agreement</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
UFSM	Universidade Federal de Santa Maria
VNF	<i>Virtualized Network Function</i>
VNFD	<i>Virtualized Network Function Descriptor</i>
VNFFG	<i>Virtualized Network Function Forwarding Graph</i>
XML	<i>Extensible Markup Language</i>
YAML	<i>YAML Ain't Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	15
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	VIRTUALIZAÇÃO DE FUNÇÃO DE REDE	20
2.2	CADEIA DE FUNÇÃO DE SERVIÇO	22
2.3	ALOCAÇÃO DE RECURSOS PARA VIRTUALIZAÇÃO DE FUNÇÕES DE REDE	25
2.4	SUMARIZAÇÃO E DISCUSSÃO	28
3	TRABALHOS RELACIONADOS	30
3.1	ESPECIFICAÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO	30
3.2	COMPOSIÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO	36
3.3	DIVISÃO E MAPEAMENTO DE CADEIAS DE FUNÇÃO DE SERVIÇO	41
3.4	SUMARIZAÇÃO E DISCUSSÃO	46
4	MODELOS PARA ALOCAÇÃO DE RECURSOS EM NFV	47
4.1	MODELO DE ESPECIFICAÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO ...	47
4.1.1	Premissas do Modelo	48
4.1.2	Formalização do Modelo	52
4.1.3	Validação do Modelo	56
4.2	MODELO DE REQUISICÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO	59
4.2.1	Modelo de Dados	61
4.2.2	Validação do Modelo	67
4.3	SUMARIZAÇÃO E DISCUSSÃO	70
5	COMPOSIÇÃO ORIENTADA A FUNÇÃO OBJETIVO PERSONALIZADA.	72
5.1	METODOLOGIA DA SOLUÇÃO	72
5.1.1	Expansão de Topologias	74
5.1.2	Construção da Função Objetivo	77
5.1.3	Avaliação de Topologias	83
5.2	AVALIAÇÃO DA SOLUÇÃO	87
5.2.1	Ferramental e Cenário de Teste	87
5.2.2	Resultados	96
5.3	SUMARIZAÇÃO E DISCUSSÃO	103
6	DIVISÃO E MAPEAMENTO ORIENTADA A ANÁLISE PERSONALIZADA DE POLÍTICAS	105
6.1	METODOLOGIA DA SOLUÇÃO	106
6.1.1	Distribuição de Topologia	110
6.1.2	Validação de Distribuições	112
6.1.3	Avaliação de Distribuições Candidatas	115
6.2	AVALIAÇÃO DA SOLUÇÃO	120
6.2.1	Ferramental e Cenário de Teste	121
6.2.2	Resultados	124
6.3	SUMARIZAÇÃO E DISCUSSÃO	130
7	CONCLUSÃO	132
	REFERÊNCIAS	134

1 INTRODUÇÃO

As atuais infraestruturas de telecomunicações dependem de diferentes Funções de Rede (*Network Functions* - NF). Essas NFs são majoritariamente implementadas em equipamentos dedicados – chamados de *physical appliances* – capazes de identificar, tratar e executar tanto protocolos (*e.g.*, OSPF, BGP) quanto oferecer serviços (*e.g.*, VPN, IDS, DNS). Porém, tais equipamentos tipicamente apresentam alto custo financeiro, baixa escalabilidade e processos de manutenção complexos, contribuindo com o problema conhecido como ossificação da Internet (HANDLEY, 2006). Nesse contexto, o paradigma de Virtualização de Função de Rede (*Network Function Virtualization* - NFV) surge com a finalidade de utilizar técnicas de virtualização já existentes para instanciar funções de rede virtuais (*Virtualized Network Functions* - VNF) (NFVISG, 2012) em equipamentos de propósito geral, alcançando benefícios tais como a redução de custos financeiros, flexibilização do monitoramento e da manutenção de NFs, além da simplificação do processo de desenvolvimento e implantação de novos serviços de rede.

Usualmente, serviços complexos em NFV (*i.e.*, que desempenham tarefas que demandam múltiplas funções) são oferecidos através do uso de várias instâncias de VNF. Esses serviços são construídos por meio da interligação e instrumentação das funções de rede para que operem de maneira coordenada, possibilitando que uma tarefa complexa seja dividida em diversas operações simplificadas. A composição e conexão dessas diferentes funções que processam e encaminham pacotes, quadros e/ou fluxos resultam em uma estrutura conhecida como Cadeia de Funções de Serviço (*Service Function Chain* - SFC) (QUINN; NADEAU, 2015). Porém, apesar de SFCs possibilitarem a criação de novos e sofisticados serviços de rede, técnicas de implantação, gerência e orquestração dessas cadeias ainda precisam evoluir através da implementação de soluções de alto nível, adaptáveis e *on-the-fly* para viabilizar sua ampla adoção, trazendo assim novos desafios a comunidade acadêmica (*e.g.*, composição, distribuição, compartilhamento e monitoramento de SFCs).

A criação e manutenção de um serviço de rede em um ambiente virtualizado acontece a partir da execução de uma série de processos inter-relacionados que constituem a implantação de SFC. A implantação, de maneira geral, compreende os processos necessários para o lançamento (*release*) do elemento a ser executado, até sua configuração, instalação e ativação (DEARIE, 2007). Entretanto, a implantação de SFCs tipicamente analisa apenas a topologia da rede e os recursos físicos disponíveis, desconsiderando aspectos das próprias cadeias de serviço

(e.g., heterogeneidade de elementos operacionais, simetria de conexões, compartilhamento de funções virtualizadas). Esse escopo limitado resulta em processos inflexíveis que não são capazes de lidar com a diversidade de métricas e das características de uma SFC (e.g., modificações topológicas, cumprimento de políticas, qualidade de serviço) (QUINN; NADEAU, 2015) (LIU et al., 2017), dificultando a introdução de novos serviços e/ou a readequação de processos já existentes para operarem sobre cenários com necessidades específicas.

Um dos processos que compõem a implantação de SFCs é chamado de Alocação de Recursos NFV (*NFV Resource Allocation* - NFV-RA) (HERRERA; BOTERO, 2016). Em (HERRERA; BOTERO, 2016) este processo é decomposto em três etapas: Composição de SFC (*SFC Composition* - SFC-C), Integração de SFC (*SFC Embedding* - SFC-E) e Agendamento da Execução de SFC (*SFC Scheduling* - SFC-S). Os desafios referentes ao NFV-RA vêm sendo extensivamente explorados pela comunidade científica nos últimos anos através do desenvolvimento de soluções que são empregadas para compor (MEHRAGHDAM; KELLER; KARL, 2014; GIL-HERRERA; BOTERO, 2017; OCAMPO et al., 2017), integrar (MOENS; DE TURCK, 2014; MEDHAT et al., 2015; DIETRICH; ABUJODA; PAPADIMITRIOU, 2015; RIERA et al., 2016; BAEK et al., 2017; ALLYBOKUS et al., 2018) e programar a execução (MIJUMBI et al., 2015; FARACI; LOMBARDO; SCHEMBRA, 2016; KULKARNI et al., 2017) dos serviços virtualizados de rede.

Apesar dos avanços realizados, as soluções do NFV-RA desenvolvidas, em especial aquelas que implementam etapas de composição e integração, são tipicamente orientadas a funções objetivo e análise de políticas rigidamente programadas, levando à uma implantação estática de SFCs, restritiva e sem suporte a personalização. Esse problema é identificado em ferramentas que executam a etapa de composição e adotam funções objetivo determinadas através de um grupo fixo e limitado de métricas (MEHRAGHDAM; KELLER; KARL, 2014; DRÄXLER; KARL, 2017; GIL-HERRERA; BOTERO, 2017; OCAMPO et al., 2017; WANG et al., 2017). Entre as métricas comumente observadas estão a taxa de tráfego, requisitos de banda e recursos computacionais, tamanho da cadeia de serviço e níveis de prioridade definidos para funções de rede. Já durante a etapa de integração, soluções como as apresentadas em (DIETRICH; ABUJODA; PAPADIMITRIOU, 2015; ABUJODA; PAPADIMITRIOU, 2016; RIERA et al., 2016; WANG et al., 2017) realizam somente avaliações pré-programadas sobre um conjunto estático de políticas. Esse conjunto, por sua vez, é tipicamente construído por métricas de avaliação como custos financeiros de alocação, quantidade de provedores requisitados, atraso

inter-domínios e necessidade de recursos computacionais.

Sendo assim, o operador de rede deve adequar as suas necessidades e as características do serviço e do cenário de implantação às soluções disponíveis. Por exemplo, um operador pode necessitar que a implantação de um determinado serviço minimize a taxa de tráfego (vazão média entre funções de rede) e, como objetivo secundário e simultâneo, reduza ao máximo a quantidade de memória consumida. Nesse caso, o operador de rede precisa identificar uma solução de implantação que avalie as duas métricas de interesse e, além disso, seja capaz de ponderar a importância que cada uma delas terá durante a geração do resultado final. Entretanto, caso não haja correspondência entre as funções objetivo das soluções disponíveis e as suas necessidades particulares, é inevitável o desenvolvimento de novas ferramentas ou a adaptação, em código, de uma já existente.

Essa carência de suporte a personalização de soluções de implantação tipicamente apresenta como efeito colateral a adoção de modelos de Requisição de SFC (*SFC Request* - SFCR) pouco flexíveis e com baixa capacidade de reutilização em contextos diversificados. Finalmente, também devido a essa inflexibilidade, as soluções que implementam etapas do NFV-RA acabam por utilizar majoritariamente dois tipos de modelagem topológica de SFCs: extremamente branda, que não define uma topologia de fato, mas sim os possíveis relacionamentos entre VNFs; ou extremamente rígida, onde a topologia é especificada como um elemento imutável através de um modelo suficiente para a execução de determinadas soluções que implementam o NFV-RA, mas que apresentam limitações de análises topológicas que impossibilitam a execução de eventuais otimizações no serviço requisitado e no processo de implantação como um todo (*e.g.*, limitação de tipos de ramificação, impossibilidade de definição de paralelismo, limitações de ordenamento parcial).

Considerando as diferentes necessidades dos operadores de rede (*e.g.*, redução da taxa de tráfego no serviço, do atraso de processamento, do consumo de memória, da quantidade de VNFs), ambientes de virtualização (*e.g.*, centralizado, distribuído, privado, público) e tipos de serviços (*e.g.*, roteamento, segurança, balanceamento de carga) que formam inúmeros cenários de implantação heterogêneos e a baixa capacidade adaptativa a estes cenários por parte das soluções de implantação existentes, a presente dissertação objetiva desenvolver um arcabouço ferramental com modelos, metodologias e soluções necessárias para a execução das etapas do NFV-RA de forma flexível e adaptável. Para isso, uma metodologia dinâmica, elástica e genérica quanto a quantidade, tipo e granularidade de métricas de avaliação é proposta. Essa meto-

dologia é aplicável a qualquer etapa do NFV-RA, além de retornar índices de adequabilidade padronizados como resultado. Ainda, esses índices podem ser analisados conjuntamente em busca das melhores relações custo/benefício entre as etapas de implantação, sendo isto possível mesmo se os índices de adequabilidade forem gerados por diferentes soluções não coordenadas entre si.

Em conjunto a isso, inovações são propostas buscando aumentar a capacidade descritiva dos modelos de especificação que conduzem o processo de implantação como um todo, fornecendo dados fundamentais para a operação das soluções que implementam as suas etapas. Esses modelos oferecem o ferramental necessário à especificação de topologias sofisticadas (*e.g.*, com diferentes tipos de ramificações, dependências e ordenamentos) e de requisições estendidas que definem, além das principais características de um serviço, uma função objetivo e políticas de avaliação. Em seguida, a metodologia de indexação é apresentada no contexto de duas soluções também elaboradas como parte deste trabalho, sendo uma responsável pela etapa de composição e outra pela integração multi-domínio de serviços de rede virtualizados. Finalmente, um ambiente de teste foi construído e experimentações foram realizadas considerando um serviço de segurança, controle de conteúdo e balanceamento de carga para fluxos HTTP/S.

Pontualmente, as principais contribuições deste trabalho são: (1) a construção de um modelo para definição de topologias de SFCs; (2) a introdução de um modelo personalizável e expansível para requisição de SFCs; (3) a apresentação de uma metodologia genérica baseada em índices para a avaliação de múltiplas métricas de implantação; (4) a implementação de soluções que abordam etapas distintas do NFV-RA (*i.e.*, composição e integração multi-domínio através da técnica de divisão e mapeamento) e utilizam os modelos e metodologias definidos; (5) a criação de um serviço de rede virtualizado de segurança, controle de conteúdo e balanceamento de carga para fluxos HTTP/S; e (6) o desenvolvimento e *benchmarking* de funções de rede genéricas e reutilizáveis. Dessa forma, os requisitos, resultados e discussões referentes ao arcabouço ferramental construído e a experimentação efetuada são apresentados no decorrer dos próximos capítulos.

O restante desse trabalho está organizado como se segue: o Capítulo 2 apresenta uma visão geral do paradigma NFV, de SFCs e do processo de implantação com foco no NFV-RA. Em seguida, o Capítulo 3 discute o estado da arte da área abordada nesta dissertação através de seus trabalhos relacionados. No Capítulo 4 os modelos de especificação de SFCs e SFCR propostos são apresentados e em seguida inseridos nas soluções de composição e de divisão

e mapeamento descritas, respectivamente, nos Capítulos 5 e 6. Nesses últimos são detalhadas também a metodologia de avaliação através de indexadores e outras decisões de projeto refletidas na implementação das soluções, além dos testes de validação realizados com as mesmas. Finalmente, o Capítulo 7 conclui a dissertação e apresenta as expectativas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são revisados os principais conceitos e trabalhos da área de Virtualização de Funções de Rede (*Network Function Virtualization* - NFV). Inicialmente objetiva-se apresentar a evolução das pesquisas direcionadas a justificar, organizar e construir arquiteturas para a execução de Funções de Rede Virtualizadas (*Virtualized Network Function* - VNF). Em seguida, são apresentados e discutidos os principais conceitos relacionados a análise, desenvolvimento e execução de Cadeias de Função de Serviço (*Service Function Chain* - SFC). Por fim, um aprofundamento sobre etapas de implantação de SFCs que são englobadas pelo processo de alocação de recursos em NFV é apresentado. O levantamento dessas informações foi parte fundamental para a elaboração das discussões posteriores e desenvolvimento dos modelos e soluções propostas.

2.1 VIRTUALIZAÇÃO DE FUNÇÃO DE REDE

A estrutura tradicional das redes de computadores depende de equipamentos dedicados para a execução de suas funções fundamentais, chamados de *physical appliances*. A decisão pelo uso desses equipamentos decorre principalmente de seu alto desempenho no processamento de pacotes, como *throughput* elevado e latência reduzida. Porém, problemas relacionados a flexibilidade de manutenção, de mobilidade e de escalabilidade resultam na necessidade de soluções complexas para manipular e manter os mesmos (TAYLOR; TURNER, 2004). Esses problemas aumentam os custos operacionais e de capital (CAPEX e OPEX) além de desencadear um processo conhecido como ossificação da rede (SHERRY et al., 2012).

O paradigma NFV surge como uma oportunidade de mudança de modelo para implementação de funções de rede, facilitando os processos de instanciação, provisionamento e gerenciamento das mesmas (MIJUMBI et al., 2016). Para isso, as funções de rede são desenvolvidas em um plano de *software* e tecnologias de virtualização existentes (*e.g.*, máquinas virtuais e contêineres) são utilizadas em servidores de propósito geral para a instanciação das mesmas. Entre as vantagens de se utilizar esse paradigma estão a redução de CAPEX e OPEX, suporte dinâmico a mudanças de requisitos de usuários e operadores de rede e facilidade de escala e migração, além da redução do tempo necessário para a criação, depuração e divulgação de novas funções e protocolos.

A fim de estruturar o paradigma NFV, uma arquitetura foi proposta pela *European Telecommunications Standards Institute* (ETSI) para padronizar as soluções e direcionar os esforços de pesquisa na área (NFVISG, 2013). Essa arquitetura observa características necessárias para o desenvolvimento, implantação e manutenção tanto das funções de rede virtualizadas, quanto do ambiente virtualizado e das tecnologias que as suportam. Entre essas características encontram-se a adaptabilidade da arquitetura com diferentes tipos de recursos e virtualizadores, o interfaceamento entre as funções de rede virtualizadas e os sistemas de gerência, a garantia da segurança dos sistemas virtualizados, os impactos na rede devido a utilização de um plano de software (*e.g.*, menor *throughput*, maior atraso, erros de software) e a compatibilidade com os centros de processamento particulares e/ou provedores de infraestrutura disponíveis.

A organização arquitetural do paradigma NFV, ilustrada de maneira simplificada na Figura 1, é composta por três blocos principais: Infraestrutura NFV (*NFV Infrastructure* - NFVI), Funções Virtualizadas de Rede (*Virtualized Network Functions* - VNF) e Gerência e Orquestração NFV (*NFV Management and Orchestration* - NFV MANO). O NFVI é formado pelos recursos de *hardware* (*i.e.*, computação, armazenamento e rede) juntos a camada de virtualização que torna possível o uso dos mesmos para a instanciação de VNFs. Já as instâncias de funções de rede implementadas e executadas em um plano de *software* constituem o bloco de VNFs, nele ocorre o processamento e o encaminhamento de pacotes, quadros e/ou fluxos. Finalmente, o MANO realiza processos de orquestração e gerência de ciclo de vida tanto dos recursos físicos e virtuais, quanto das instâncias de VNFs, atuando diretamente nos demais blocos.

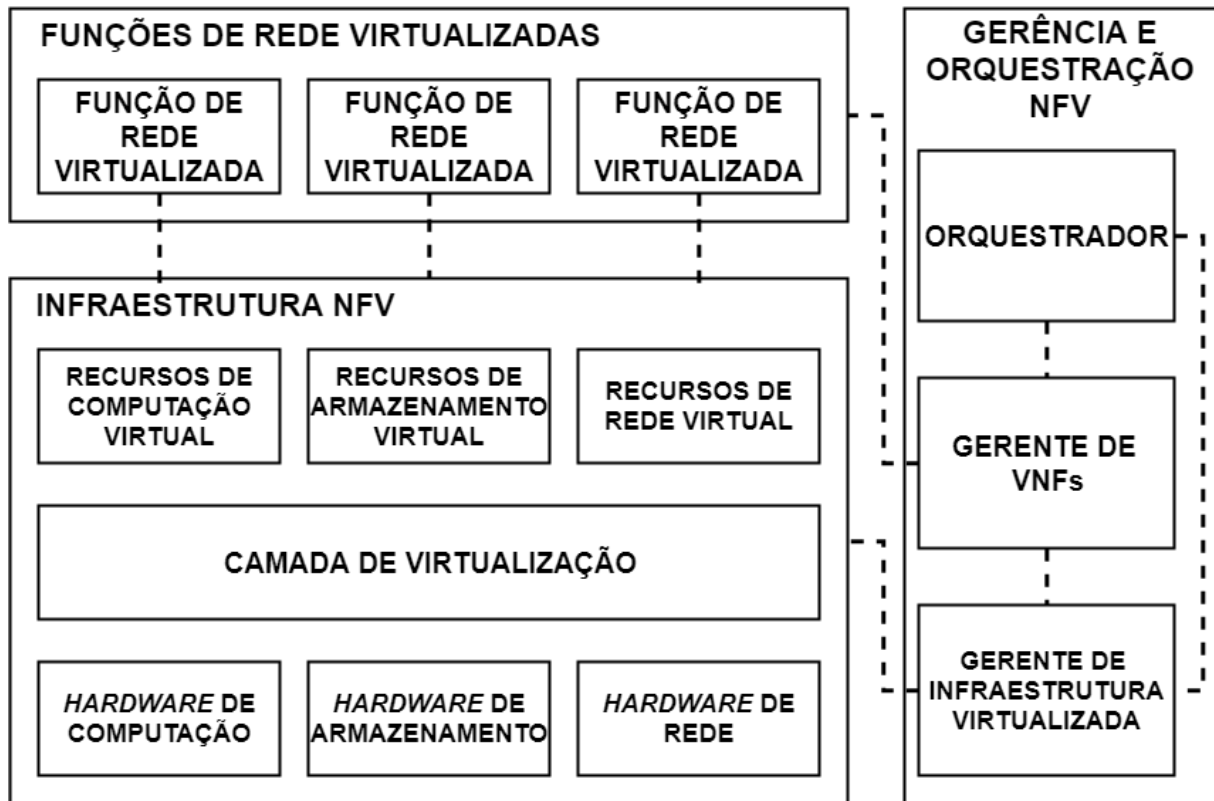
Apesar dos benefícios do paradigma NFV, ainda existem diversos desafios a serem explorados. Entre eles está a degradação do desempenho no processamento de pacotes devido a introdução de uma camada de virtualização. Felizmente, tanto a academia quanto a indústria realizam esforços objetivando o desenvolvimento de soluções que reduzem essa degradação. Dentre essas soluções é possível destacar as plataformas de VNF (*e.g.*, OpenNetVM (ZHANG et al., 2016), ClickOS (MARTINS et al., 2014) e Click-on-OSv (MARCUIZZO et al., 2017)) que empregam tecnologias para aceleração de pacotes (*e.g.*, DPDK¹, NetMap² e XDP³), ademais de técnicas de paralelismo (SUN et al., 2017; ZHANG et al., 2017; ZHENG et al., 2018) e agendamento da execução de VNFs (MIJUMBI et al., 2015; FARACI; LOMBARDO; SCHEMBRA,

¹ <https://www.dpdk.org>

² <http://info.iet.unipi.it/luigi/netmap>

³ <https://www.iovisor.org/technology/xdp>

Figura 1 – Arquitetura NFV Simplificada



2016; KULKARNI et al., 2017).

A passagem do núcleo dos sistemas de telecomunicações para soluções implementadas inteiramente em *software* é uma tendência crescente. Esse fenômeno ocorre para acompanhar a dinamicidade e a grande quantidade de tráfego gerada pelas novas tecnologias, como as redes 5G, *Internet of Things* (IoT) e *big data*. As características dessas tecnologias demandam serviços flexíveis, escaláveis e inteligentes para a realização das comunicações (BLANCO et al., 2017). O paradigma NFV é apontado como um dos pilares dessa nova era das redes, mas outros paradigmas, como Redes Definidas por *Software* (*Software Defined Network* - SDN (KREUTZ et al., 2015)) e computação de borda (*i.e.*, *edge computing* (SHI; DUSTDAR, 2016) e *fog computing* (VAQUERO; RODERO-MERINO, 2014)) também irão prover importantes recursos para desenvolvimento e implantação dessa nova estrutura, atuando de forma conjunta ao NFV.

2.2 CADEIA DE FUNÇÃO DE SERVIÇO

O paradigma NFV permite a criação de serviços de rede através da conexão de múltiplas VNFs. Existem duas principais definições relacionadas a criação de serviços em NFV:

Serviço de Rede (*Network Service - NS*) (NFVISG, 2013) e Cadeia de Funções de Rede (*Service Function Chain - SFC*) (HALPERN; PIGNATARO, 2015). O primeiro é definido pela ETSI e provê uma visão geral da estrutura e operação de um serviço. A visão ETSI define diferentes componentes que, juntos, formam uma especificação completa de um NS (*i.e.*, pontos finais, caminhos, serviços fim-a-fim e grafo de encaminhamento de VNFs). O segundo é definido pela *Internet Engineering Task Force (IETF)* (HALPERN; PIGNATARO, 2015) e também apresenta a estrutura de SFC como um conjunto de componentes que formam um serviço completo (*i.e.*, nós de borda e caminho de função de serviço).

O conceito de NS compreende as funções de rede que provêm um determinado serviço, além de todas as informações relacionadas ao mesmo (*e.g.*, requisitos de desempenho, requisitos de recursos para implantação em diferentes cenários de execução, *scripts* de execução) (NFVISG, 2013) (GSNFV, 2013). A organização do encaminhamento de dados entre as funções de rede que fazem parte de um serviço é descrita através de uma estrutura de Grafo de Encaminhamento entre VNFs (*VNF Forwarding Graph - VNFFG*) (NFVISG, 2014). Em um VNFFG, os vértices presentes no grafo representam as funções de rede (onde pelo menos uma deve ser virtualizada) e as arestas representam as conexões virtuais entre pares de elementos (VNFs, *hardware appliances* e pontos finais), os pontos finais são elementos especiais que determinam limites do grafo e indicam a entrada e saída de dados do serviço. Um NS pode ser definido utilizando descritores de alto nível (*e.g.*, NFV TOSCA⁴) que representam tanto o VNFFG quanto as demais informações requeridas.

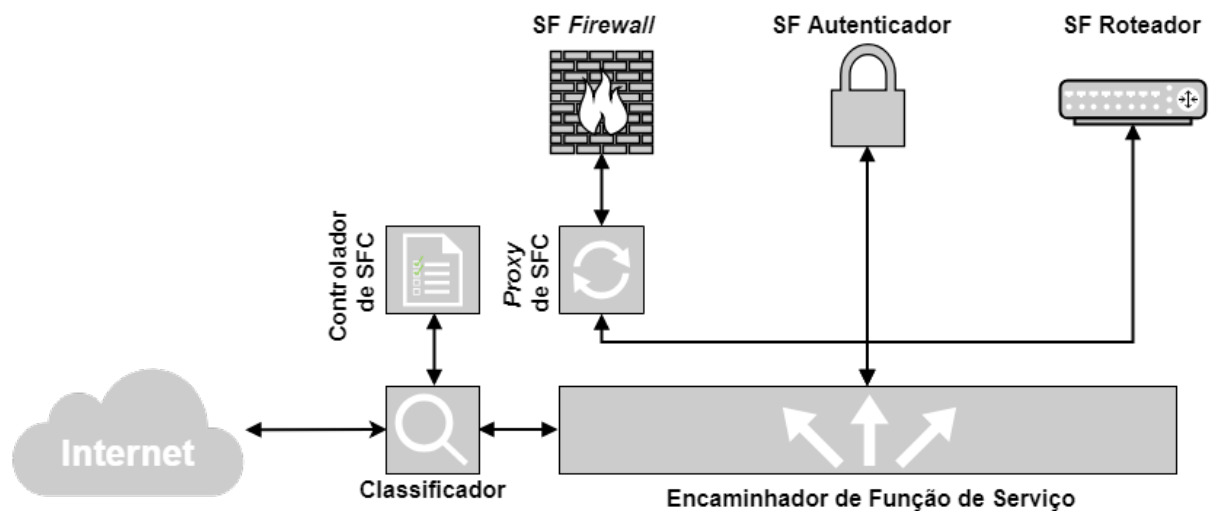
A IETF também aborda a criação de serviços que utilizam o paradigma NFV através do conceito de SFC. Uma SFC indica um grupo de funções de rede, sendo pelo menos uma VNF, conectadas de maneira ordenada de modo a direcionar e processar tráfego entre as mesmas (HALPERN; PIGNATARO, 2015). Essencialmente, uma SFC é criada através da definição das funções de rede que a compõe e de seus pontos de ingresso e egresso de dados, estes últimos chamados de nós de borda. Finalmente, as conexões virtuais entre cada um dos elementos são descritas, gerando uma topologia capaz de ser instanciada em uma infraestrutura virtualizada.

Uma arquitetura para a criação e manutenção de SFCs, exemplificada na Figura 2, também é proposta pela IETF (HALPERN; PIGNATARO, 2015). Essa arquitetura contém tanto componentes de gerência, representados pelo Controlador de SFC (*SFC Controller - SFCC*), Classificador e Encaminhador de Função de Serviço (*Service Function Forwarder - SFF*),

⁴ <http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/tosca-nfv-v1.0.html>

quanto operacionais, através do SFC *Proxy* (SFCP) e das Funções de Serviço (*Service Function* - SF). Uma SF indica um elemento que processa o tráfego de rede, esse é o bloco de construção básico de uma SFC e corresponde a uma função de rede virtualizada ou não. Uma SF pode ser consciente ou agnóstica ao Cabeçalho de Serviço de Rede (*Network Service Header* - NSH) (QUINN; ELZUR; PIGNATARO, 2018). Quando uma função incapaz de computar nativamente este cabeçalho (*i.e.*, agnóstica) é inserida em uma SFC, um SFCP deve operar em conjunto a ela processando o NSH presente no tráfego de entrada (pré-processamento) e de saída (pós-processamento). Sendo assim, o SFCP realiza a remoção, atualização e realocação do NSH, interfaceando a comunicação entre a SF agnóstica e os demais componentes da arquitetura. O SFCC é responsável por manter uma tabela de todos os fluxos ativos e suas respectivas SFCs. O SFCC se comunica diretamente com o Classificador, este último localiza-se logo após ao ponto de entrada de dados da arquitetura, identifica o tráfego e insere o NSH nos quadros/pacotes para indicar o caminho de processamento (*i.e.*, a sequência de SFs) adequado a cada fluxo. O SFCP, por sua vez, analisa o NSH e efetua encaminhamento do tráfego a cada componente indicado.

Figura 2 – Arquitetura de Referência para Implementação de SFC



A comparação dos conceitos definidos pela ETSI e IETF para especificar serviços que utilizam o paradigma NFV evidencia sobreposições conceituais em diversos pontos, havendo apenas diferenciação na terminologia utilizada para cada especificação. A Tabela 1 sumariza a relação entre as especificações de NS (ETSI) e SFC (IETF).

Os pontos finais definidos pela ETSI (GSNFV, 2013) correspondem aos nós de borda da IETF (HALPERN; PIGNATARO, 2015), sendo que ambos definem limites que indicam a entrada e saída de dados tanto do serviço fim-a-fim (ETSI) quanto de uma SFC (IETF). Os

Tabela 1 – Comparação de Especificações ETSI e IETF

		ETSI				
		Ponto Final	Caminho	Serviço Fim-a-Fim	Grafo de Encaminhamento de VNFS	Serviço de Rede
I E T F	Nó de Borda	Correspondência Total				
	Caminho de Função de Serviço	Correspondência Total				
	Cadeia de Função de Serviço			Correspondência Total	Correspondência Total	Correspondência Parcial

Caminhos apresentados pela ETSI (NFVISG, 2014) são similares aos Caminhos de Função de Serviço da IETF (HALPERN; PIGNATARO, 2015), esses indicam os encaminhamentos possíveis em um VNFFG (ETSI) ou em uma SFC (IETF). Já a topologia de uma SFC é comparável ao VNFFG da ETSI (BERNARDOS et al., 2018), ambas especificam VNFS, Funções de Rede Físicas (*Physical Network Function* - PNF) e as conexões lógicas entre elas. Por outro lado, uma SFC considera também os pontos de borda diretamente em sua topologia, estes pontos são abordados pela ETSI como os pontos finais de um serviço fim-a-fim (GSNFV, 2013). Finalmente, a especificação de uma SFC também considera aspectos relacionados a dependências, compartilhamentos e políticas (QUINN; NADEAU, 2015). Esses aspectos também são discutidos no contexto de um NS (NFVISG, 2014), mas este ainda inclui informações operacionais e até mesmo *scripts* de execução (*e.g.*, rotinas de ciclo de vida e indicações de implantação) que não são mencionados no contexto de especificação de uma SFC.

2.3 ALOCAÇÃO DE RECURSOS PARA VIRTUALIZAÇÃO DE FUNÇÕES DE REDE

A implantação de um serviço é composta dos processos compreendidos desde o lançamento até a operacionalização do mesmo (*e.g.*, configuração, instalação e ativação) (DEARIE, 2007). Para o paradigma NFV, o processo de alocação de recursos é parte fundamental para alcançar uma implantação de serviços de rede (HERRERA; BOTERO, 2016). Esse processo é composto de três etapas: composição, integração e agendamento de execução. Essas etapas visam não simplesmente operacionalizar o serviço requisitado, mas também otimizar o mesmo para que apresente maior desempenho (*e.g.*, maior *throughput*, menor latência) e/ou economia de recursos computacionais (*e.g.*, menor consumo de rede, memória, CPU). Além disso, as etapas do NFV-RA podem ocorrer de maneira coordenada ou não coordenada. Uma etapa construída de maneira coordenada objetiva otimizar seus resultados considerando seus próprios objetivos e o comportamento esperado da próxima etapa a ser executada, mantendo consciência

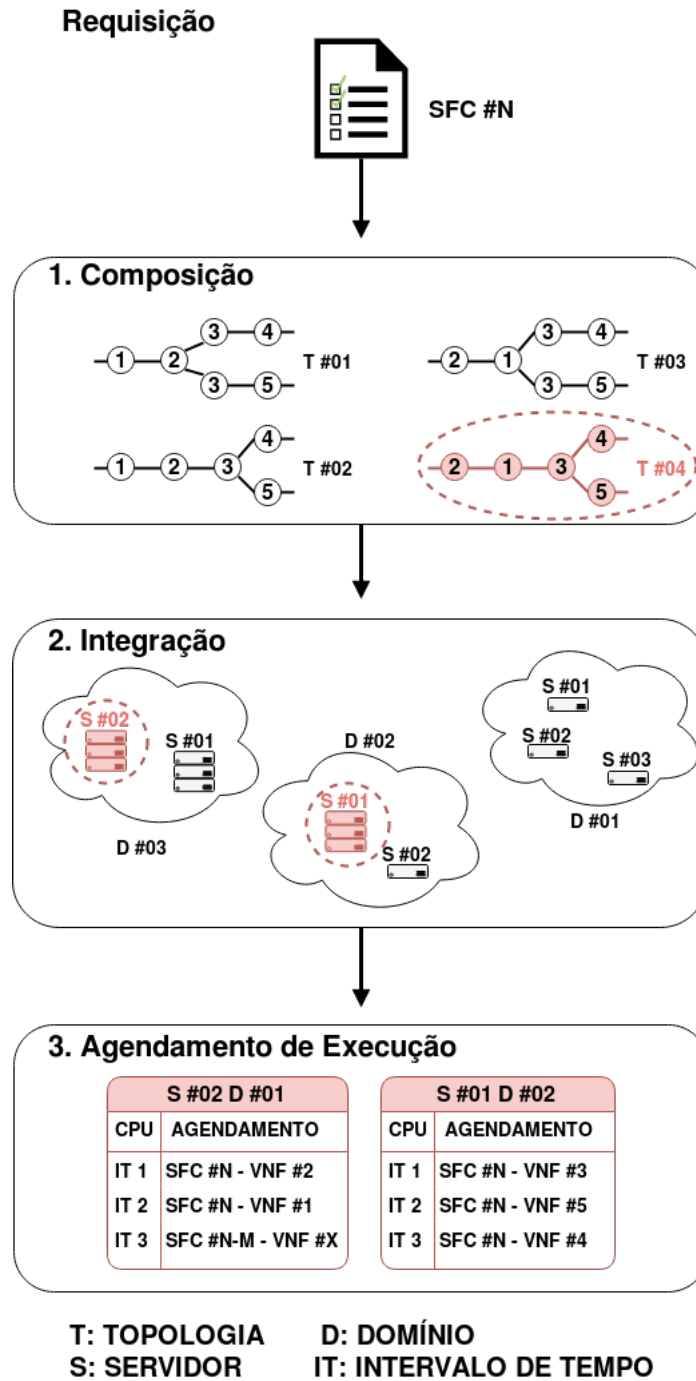
do processo como um todo. Já etapas não coordenadas retornam resultados genéricos obtidos considerando apenas seus próprios objetivos, sendo estes agnósticos e independentes a metodologia que será empregada em seguida.

Sejam as soluções que implementam as etapas do NFV-RA coordenadas ou não entre si, uma ordem de execução deve ser seguida para que o processo ocorra por completo, visto que os resultados de uma etapa servem como entrada para a sua posterior. A Figura 3 exemplifica a ordem de execução das etapas do NFV-RA. A primeira etapa fundamental é a composição, esta provê a cadeia com ordenamento absoluto a ser utilizado para o processamento na etapa de integração. A integração, por sua vez, indicará desde os domínios de execução, se mais de um estiver disponível, até as máquinas físicas que hospedarão as VNFs que formam a SFC requisitada. Finalmente, em posse dessas informações, o agendamento de execução pode ser executado a fim de modificar configurações específicas dos equipamentos apontados pela etapa de integração.

A etapa de composição realiza a definição de uma topologia SFC com ordenamento absoluto (*i.e.*, com suas funções de rede alocadas em posições definidas e imutáveis dentro de um grafo de serviço) considerando eventuais limitações observadas através de um SFCR. Dessa forma, o SFCR deve ser capaz de fornecer tanto informações das VNFs e suas relações, quanto as características manipuláveis e de exceção relacionadas ao serviço em si (*e.g.*, ordenamentos parciais, dependências, políticas). Para que as soluções de composição possam atuar sobre essas características, também são fornecidas através do SFCR os dados de avaliação das NFs que compõem o serviço, estes últimos podem ser obtidos através da instanciação, de testes de execução ou de análises de comportamento das funções. Algoritmos de composição, como os apresentados em (MEHRAGHDAM; KELLER; KARL, 2014; GIL-HERRERA; BOTERO, 2017; OCAMPO et al., 2017), usam esses dados para avaliar as formações de cadeia possíveis e retornar a (as) que melhor se adéqua (adéquam) as funções objetivo presentes em seus algoritmos.

A etapa de integração objetiva alocar a topologia com ordenamento absoluto, fornecida pela etapa de composição, na infraestrutura disponível. Essa etapa pode utilizar diferentes técnicas para alcançar a melhor distribuição de VNFs. Três técnicas adequadas a essa etapa são chamadas de divisão e mapeamento de SFC (para distribuição multidomínio), seleção de VNFs (para compartilhamento de VNFs) e alocação em *hardware* (para instanciação de VNFs na infraestrutura física). A divisão e mapeamento de SFCs é uma técnica utilizada para distribuir

Figura 3 – Execução da Alocação de Recurso em NFV



a cadeia sobre múltiplos domínios. Essa faz uso de funções de objetivo (*e.g.*, minimização de custos financeiros, de latência, de distância geográfica) para fragmentar a SFC em sub-cadeias e mapear as mesmas nos domínios disponíveis, como exemplificado em (DIETRICH; ABU-JODA; PAPADIMITRIOU, 2015; RIERA et al., 2016). A seleção de VNFs, como apresentado em (MEDHAT et al., 2015), consiste na detecção VNFs já alocadas na infraestrutura, mas que operam para outras SFCs, e são comuns a aquelas presentes na requisição atual, fazendo des-

sas instâncias compartilhadas, esse tipo de abordagem também é apresentada no contexto de mercados de VNF (GU et al., 2016). Por fim, a alocação em *hardware*, como demonstrada em (MOENS; DE TURCK, 2014; BAEK et al., 2017; ALLYBOKUS et al., 2018), é executada internamente ao domínio. Esse processo mapeia as VNFs nos dispositivos físicos disponíveis (*i.e.*, nós de virtualização e hospedagem de VNFs) de acordo com os seus requisitos de execução (*e.g.*, CPU, memória, interfaces de rede e tecnologia de virtualização).

Por fim, a etapa de agendamento de execução distribui e organiza o tempo de execução internamente aos nós que hospedam instâncias de VNFs. Essa etapa tipicamente objetiva reduzir o tempo total necessário para realizar a execução das VNFs sem degradar o desempenho das mesmas, mas outros objetivos também são comuns, como a redução da carga de CPU decorrente do uso de aceleradores de pacotes presentes nas plataformas de VNF (ALAN ANTHONY SHIHABUR CHOWDHURY, 2018). A capacidade computacional de cada nó e a complexidade de execução das VNFs hospedadas são informações essenciais para alcançar uma programação da execução justa e eficiente. Algumas soluções estão presentes na literatura, como em (MIJUMBI et al., 2015; KULKARNI et al., 2017), e em geral garantem que os recursos físicos estão alocados adequadamente através da análise contínua do custo de instanciação e do tempo médio de processamento de cada VNF.

2.4 SUMARIZAÇÃO E DISCUSSÃO

O paradigma NFV justifica-se pela necessidade de evolução das técnicas de processamento e gerência das mais diferentes funções de rede. Essa demanda é impulsionada pelo rápido crescimento do tráfego mundial ocasionado pela contínua popularização da Internet, seja para uso comercial ou seja para uso doméstico, e de novas aplicações para a mesma, como a Internet das Coisas (*Internet of Things* - IoT) (HAZAS et al., 2016). Assim, suportar e gerenciar esses volumes crescentes de dados se tornou uma tarefa complexa e desencadeou a busca de alternativas aos comumente utilizados *physical appliances* que, apesar de obterem bom desempenho de processamento (*e.g.*, baixa latência de processamento e alto *throughput*), apresentam alto custo de manutenção, baixa escalabilidade e baixa flexibilidade para suportar novos protocolos e operações sem que haja necessidade de troca, ossificando o núcleo da Internet (HANDLEY, 2006).

De acordo com que esforços são empregues em pesquisas que abordam o paradigma NFV, novas estruturas e ferramentas são desenvolvidas para suportarem sua utilização em ce-

nários sofisticados e reais. Por exemplo, o conceito de SFC foi criado para que, a partir de um conjunto de funções de rede virtualizadas, seja possível criar serviços de rede complexos (*e.g.*, estruturas ramificadas e funções de rede compartilhadas entre SFCs heterogêneas) que apresentam desde analisadores de tráfego até topologias completas de segurança. Essas evoluções contribuíram para que o paradigma NFV passasse a ser estudado pela sua viabilidade econômica, sendo esta explorada através da disponibilização do mesmo em modelos "*as-a-service*" (DING et al., 2015) (SHEN et al., 2015) e na criação de mercados de funções de rede (XILOURIS et al., 2014) (BONDAN et al., 2018). Com isso, a operacionalização de SFCs se tornou um processo tão importante quanto sua criação, resultando em soluções que visam a eficiente composição, integração e programação de execução dos serviços virtualizados (HERRERA; BOTERO, 2016). Alguns dos principais trabalhos que tratam das etapas de composição e integração em ambientes multi-domínio são discutidas no Capítulo 3.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta e discute os trabalhos diretamente relacionados ao tema desta dissertação. Inicialmente, os modelos de especificação de topologias SFC, utilizados como parte de um SFCR para a execução das etapas do NFV-RA, são revisados a fim de identificar lacunas de definição que reduzem a expressividade do operador de rede e o poder de análise das soluções de alocação de recursos. Após, soluções que abordam as etapas do NFV-RA são analisadas aprofundadamente, é importante ressaltar que o escopo deste trabalho está limitado as etapas de composição e integração multi-domínio, sendo a segunda investigada através da técnica de divisão e mapeamento.

3.1 ESPECIFICAÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO

A especificação de SFCs tem por objetivo principal estruturar o elemento central a ser processado durante as etapas do NFV-RA (*i.e.*, a topologia SFC). Para isso, existem diferentes modelos de especificação a serem considerados, desde aqueles de baixo nível (*e.g.*, grafos) até os mais abstraídos (*e.g.*, linguagens formais ou de marcação). As decisões de projeto que permeiam a utilização de um modelo ou outro também determinam a quantidade de esforço empregado por operadores de rede durante a criação e implantação das topologias. Tipicamente, modelos de baixo nível são facilmente processados durante as etapas de implantação, estes são capazes de indicar explicitamente operações para a realização dessas etapas, um exemplo deste tipo de definição ocorre, fora do paradigma NFV, na definição topológica de redes simuladas Mininet (LANTZ; HELLER; MCKEOWN, 2010). Por outro lado, modelos de alto nível são facilmente compreendidos pelos operadores de rede, reduzindo a curva de aprendizado de seus usuários, além de, desde de que sejam formalizados, poderem ser traduzidos para meta-modelos de baixo nível e então empregados no processo de implantação.

De forma geral, modelos de especificação topológica são tipicamente construídos com a finalidade de facilitar o entendimento, projeto e desenvolvimento de serviços em NFV. Ao mesmo tempo, é conveniente que esses modelos sejam arquitetados visando reduzir os esforços de implementação de validadores e processadores desses. Dessa forma, estruturas formais amplamente difundidas, como gramáticas e linguagens de marcação (*e.g.*, XML, JSON, YAML), apresentam-se como boas alternativas devido a ampla disponibilidade de validado-

res léxicos, sintáticos e estruturais (*e.g.*, NLTK⁵, PyXML⁶, SimpleJSON⁷, PyYAML⁸, RapidJSON⁹, PefectXML¹⁰), permitindo aos desenvolvedores de soluções que adotam tais modelos concentrarem-se na análise semântica dos mesmos.

Além disso, é desejável que um modelo de especificação de SFCs assegure uma série de capacidades de definição ao operador de rede (*e.g.*, fixação de posição, grupos parcialmente ordenados, ramificações genéricas, dependências de localização, dependências entre VNFs, acoplamento entre VNFs). Essas capacidades permitem que as categorias de SFCs sejam mapeadas e consideradas para determinar o modo de operação de soluções das etapas do NFV-RA, como estratégias de tratamento de ramificações ou para assegurar políticas (*e.g.*, tamanho máximo da cadeia). Exemplos de categorias topológicas de SFCs que podem ser analisadas são:

- **Organização:** a organização de uma topologia SFC pode ser classificada como linear, ou seja, formada apenas por VNFs limitadas a um ponto de entrada e a um ponto de saída de dados, essas contidas entre dois nós de borda (DING et al., 2015). Ainda, para serviços tipicamente mais complexos, a topologia pode apresentar a organização de um grafo arbitrário (*i.e.*, ramificada), apresentando tanto ramificações terminais (conectadas a um nó de borda) quanto não terminais (conectadas a uma VNF agregadora).
- **Largura e Tamanho:** a largura de uma SFC corresponde ao maior número de VNFs entre dois nós de borda presentes na topologia (*i.e.*, o maior caminho) (CARPIO; DHAHRI; JUKAN, 2017). Já o tamanho de uma SFC, por sua vez, pode ser medido de duas maneiras: como o número total de VNFs ou como o número total de conexões internas (tanto entre duas VNFs, quanto entre VNFs e nós de borda). É importante ressaltar que a largura e o tamanho são categorias que se inter-relacionam em cadeias lineares.
- **Compartilhamento:** duas classes podem ser consideradas quanto a categoria de compartilhamento: isolada ou compartilhada. Topologias isoladas empregam apenas VNFs dedicadas a um único serviço, enquanto que topologias compartilhadas apresentam pelo menos uma VNF que opera para mais de um serviço ao mesmo tempo. O compartilhamento de VNFs é previsto e suportado pela arquitetura padrão de implementação de SFCs (HALPERN; PIGNATARO, 2015).

⁵ <https://nltk.org>

⁶ <https://pypi.org/project/PyXML>

⁷ <https://pypi.org/project/simplejson>

⁸ <https://pypi.org/project/PyYAML/>

⁹ <http://rapidjson.org>

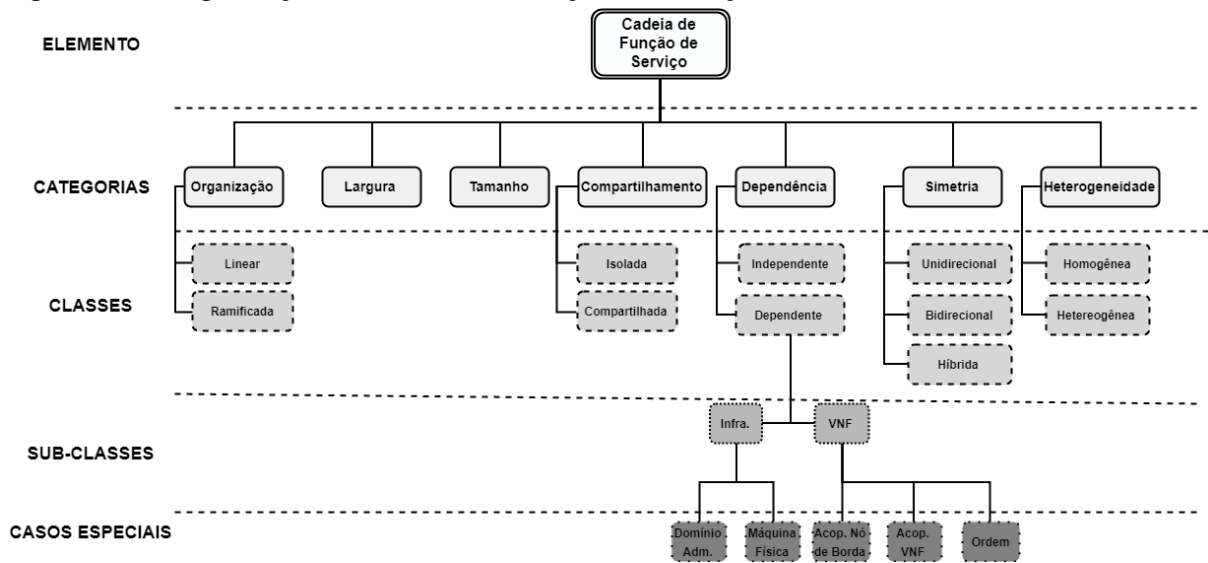
¹⁰ <http://perfectxml.com>

- **Dependências:** uma SFC pode ou não apresentar dependências entre suas VNFs, nós de borda e infraestrutura. Uma SFC independente pode organizar livremente os elementos presentes em segmentos parcialmente ordenados, além de ser capaz de ter sua topologia implantada em qualquer ponto do substrato físico disponível. Já SFCs dependentes (*i.e.*, uma ou mais VNFs com dependências topológicas e/ou de infraestrutura) precisam cumprir regras de organização de segmentos parcialmente ordenados, como as de ordem (analisando a precedência obrigatória entre NFs) e de acoplamento (verificando o elemento – NF ou nó de borda – imediatamente anterior a outro), e de distribuição na infraestrutura (instanciando elementos em domínios administrativos ou máquinas físicas rigidamente definidos). O posicionamento absoluto de VNFs, seja a nível topológico ou de substrato físico, deve ser feito de maneira cautelosa, visto que este tipo de ação restringe o escopo de atuação das soluções que implementam as etapas do NFV-RA e podem degradar o desempenho geral de um serviço (QUINN; NADEAU, 2015).
- **Simetria:** o encaminhamento de tráfego em uma SFC pode ser assimétrico (*i.e.*, unidirecional ou híbrido) ou simétrico (*i.e.*, bidirecional) (HALPERN; PIGNATARO, 2015). Quando o tráfego é encaminhado apenas em uma direção, a topologia é dita unidirecional. Já uma topologia híbrida é capaz de processar, além do tráfego de chegada, o tráfego de retorno, porém este último é encaminhado apenas a um subconjunto das funções de rede. Finalmente, a topologia é intitulada como bidirecional quando o tráfego de retorno é processado em ordem contrária por todas as funções de rede presentes em uma SFC.
- **Heterogeneidade:** duas classificações podem ser aplicadas a SFCs quanto a categoria de heterogeneidade: homogênea e heterogênea. Uma topologia homogênea apresenta apenas VNFs em seus elementos operacionais. Por outro lado, topologias heterogêneas permitem a utilização de *hardware appliances* – as PNFs (NFVISG, 2014) – operando em conjunto com VNFs para compor um serviço NFV completo.

Considerando as categorias descritas, sumarizadas pela Figura 4, trabalhos que propõem modelos de especificação de topologias SFC são apresentados e discutidos a seguir. Após, uma tabela comparativa é exposta para evidenciar as categorias abordadas por cada um dos modelos de especificação dos trabalhos analisados, ressaltando assim os pontos fortes e fracos de cada um deles.

Em (GARAY et al., 2016) um modelo de baixo nível baseado em uma notação de grafos,

Figura 4 – Categorização de Cadeias de Função de Serviço



chamado de Grafo de Serviço (*Service Graph - SG*) é proposto. O principal objetivo desse modelo é facilitar a integração de serviços 5G nas infraestruturas físicas que proveem recursos computacionais. Essa infraestrutura é também definida através de um grafo e é chamada de Grafo de Recursos (*Resource Graph - RG*). Os autores ressaltam que ambos os grafos (*i.e.*, SG e RG) são intimamente relacionados e devem ser compatíveis para simplificar o mapeamento entre eles e obter melhores resultados durante a integração dos serviços virtualizados.

O modelo do SG apresenta três componentes principais: as Funções de Rede (*Network Functions - NF*) que representam um nó operacional que executa em ambiente virtualizado (*i.e.*, VNF), estas devem estar acompanhadas de informações de implantação e de gerência de ciclo de vida; os Pontos de Acesso de Serviço (*Service Access Points - SAP*) que representam ligações entre o SG e elementos fora de seu escopo. Os SAPs, além de encaminhar dados para as NFs, também podem atuar como nós de processamento realizando análises e classificações (*e.g.*, prioridade de clientes); e as Ligações de Serviço (*Service Link - SL*) que representam conexões lógicas entre elementos (*i.e.*, NFs e SAPs). Também é possível delegar aos SLs tarefas de classificação ou de seleção de tráfego.

Como um grafo topológico é usado para descrever SGs, este modelo pode ser facilmente analisado quanto a características de nós e conexões, como a sua organização (aceitando a criação de topologias lineares e ramificadas), largura e tamanho. Os autores informam que dados relacionados a políticas e requisitos gerais podem ser anexados as SFCs. Esses dados podem apresentar restrições quanto a possíveis compartilhamentos de NFs ou assinalar dependências

referentes a implantação da topologia na infraestrutura disponível, porém não há indicação de como essas informações são inseridas no modelo. Além disso, o SG não apresenta nenhuma discussão quanto a possíveis dependências entre VNFs, simetria ou heterogeneidade.

Em (MEHRAGHDAM; KELLER; KARL, 2014) são definidos modelos de especificação da infraestrutura de rede e topologias SFC - estes modelos são posteriormente empregados como entrada de um método de composição e alocação em *hardware* otimizado para redução de latência e de utilização de recursos computacionais. A especificação de topologias SFC ocorre através de uma Gramática Livre de Contexto (GLC), esta é utilizada devido sua capacidade de expressar ordenamento entre as funções de rede, além de apresentar regras de produção formalizadas e ser de fácil compreensão humana. As regras de produção que compõem a gramática adotada permitem a manipulação da topologia através das funções de rede, dos nós de borda, de segmentos parcialmente ordenados, de ramificações terminais e de execuções paralelas (*i.e.*, replicação de segmentos).

A GLC é formalizada por uma quintupla de componentes $G = (\nu, \tau, \rho, \varsigma)$, onde cada componente representa elementos e regras de uma SFC que juntos formam topologias genéricas. O componente ν contém os símbolos não terminais da gramática, enquanto τ abrange o grupo de símbolos terminais da mesma. Os símbolos terminais caracterizam os elementos de uma SFC (*i.e.*, funções de rede e nós de borda). Em ρ são descritas as regras de produção, indicando o relacionamento entre os símbolos terminais e não terminais. Finalmente, ς indica o símbolo não terminal pelo qual a derivação das regras de produção deve começar. Este modelo pode fazer uso de *lexers* e *parsers* para realizar a validação das topologias.

Uma vez que os componentes operacionais e nós de borda são apresentados de forma explícita, uma topologia SFC especificada pela GLC pode ser naturalmente categorizada pela sua organização, largura e tamanho. Os identificadores utilizados em símbolos não terminais podem corresponder tanto a uma função de rede (de forma mais abstrata), quanto a instância ou equipamento de uma função de rede. Caso a segunda opção seja considerada, é possível então identificar SFCs com VNFs compartilhadas através da análise de todas as topologias que executam em determinado ambiente. Também, a GLC não apresenta nenhum mecanismo de definição de dependências, sejam elas topológicas ou de infraestrutura. Finalmente, não existe maneira de explicitar se uma NF é uma instância virtualizada ou um *hardware appliance* (impossibilitando a análise de sua heterogeneidade) e nem mesmo se as conexões entre elas são ou não simétricas.

A adaptação de linguagens já existentes com semântica, sintaxe e vocabulário orientadas a áreas correlatas é outra opção para especificar SFCs. Considerando isso, a linguagem formal que opera sobre processos, recursos e eventos, intitulada Álgebra de Comunicação de Recursos Compartilhados (*Algebra of Communicating Shared Resources - ACSR*) (KWAK et al., 1998), foi remodelada para essa finalidade. O ACSR de Pacotes (*Packet ACSR - pACSR*) (SHIN et al., 2015) objetiva demonstrar de forma detalhada o processo de encaminhamento e processamento de pacotes entre funções de rede, para isso uma sintaxe baseada em predicados e funções é empregada.

Para descrever processos em ACSR, um conjunto de operadores (regras semânticas e mapas) e regras algébricas que manipulam o processo são definidos. O pACSR estende o ACSR ao inserir o elemento pacote e permitir o encaminhamento do mesmo através de portas - simulando topologias SFC. Existem duas operações principais em pACSR: passagem de valor em portas (*i.e.*, encaminhamento de pacotes) e a utilização de valores para processar parâmetros (*i.e.*, processamento de pacotes). Algumas operações que podem ser aplicadas durante a modelagem de uma topologia em pACSR são: ação nula (NIL), execução paralela (*//*), processo condicional (\rightarrow), processo restritivo (*/*) e escolhas não determinísticas entre dois processos (*+*).

Apesar da linguagem pACSR ser capaz de definir o processamento de pacotes em baixa granularidade, características topológicas (*e.g.*, organização, largura e tamanho) não podem ser recuperadas através da mesma. Além disso, uma topologia não pode ser categorizada quanto ao seu compartilhamento ou heterogeneidade, visto que os processos especificados podem ser executados por VNFs ou PNFs, sejam elas isoladas ou compartilhadas. Os serviços em pACSR são naturalmente unidirecionais já que existe a necessidade de explicitar um ordenamento absoluto para a execução dos processos. Como consequência da inexistência de segmentos parcialmente ordenados, também não há suporte a definição de dependências topológicas. Finalmente, não são apresentadas discussões quanto a dependências de infraestrutura, nesse modelo essa situação é agravada pela falta de delimitação das NFs (*i.e.*, quais processos pertencem a quais funções).

Os modelos de especificação de SFCs apresentados divergem quanto aos objetivos e recursos utilizados. O modelo GLC apresenta características intimamente ligadas a topologia da SFC, tornando possível analisar as mesmas através das categorias previamente descritas. Apesar da GLC apresentada em (MEHRAGHDAM; KELLER; KARL, 2014) ser limitada quanto a especificação de compartilhamentos, dependências, simetria e heterogeneidade, esta pode ser

estendida para suportar tais características através de mudanças em sua estrutura básica (quádrupla G). Também, a GLC apresenta forte formalização descritiva devido as suas regras de formação, essa característica fecha o escopo de construções topológicas suportadas e dá previsibilidade das possíveis especificações que podem ser geradas pela gramática. Isso é desejável quando o processamento dessas especificações é condicionado as possíveis características presentes nas mesmas (*e.g.*, o tratamento de segmentos parcialmente ordenados e ramificações em (MEHRAGHDAM; KELLER; KARL, 2014) e (DRÄXLER; KARL, 2017)), sendo possível determinar as ações que devem ser realizadas na presença de tais características ainda durante o desenvolvimento de soluções que adotam o modelo de GLC.

O SG, por sua vez, apresenta-se como o mais flexível dos modelos analisados, isso ocorre uma vez que não são apontados formalizações descritivas para tal. Dessa forma, um SG é tipicamente especificado através de linguagens de marcação (estruturadas como um grafo) para então ser validado. A validação de um SG tipicamente estabelece as regras de rejeição das especificações topológicas, diferente da GLC que define um conjunto de regras de aceitação. Sendo assim, um SG se torna mais adequado quando se deseja garantir maior liberdade de especificação ao operador de rede, um exemplo disso são os grafos de relacionamentos possíveis entre VNFs utilizados nas soluções de composição apresentadas em (OCAMPO et al., 2017), (GIL-HERRERA; BOTERO, 2017) e (WANG et al., 2017).

Diferente da modelagem topológica que ocorre na GLC e SG, o pACSR tem como objetivo realizar uma modelagem de processos extremamente detalhada, indicando por quais computações um pacote deve ser submetido durante a execução de um serviço de rede. Com isso, o pACSR atua de forma complementar a uma especificação puramente topológica, já que um ou mais processos descritos através dessa linguagem podem ser mapeados aos nós ou elementos terminais dos outros modelos. Por fim, é importante ressaltar também que, apesar de apresentarem diferenças de modelagem e de cenários de aplicação, o modelo de SG pode ser traduzido para o modelo de GLC (sendo o contrário também verdadeiro), visto que ambos apresentam compatibilidade em seu objetivo final (*i.e.*, a especificação topológica de uma SFC).

3.2 COMPOSIÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO

Diferentes topologias podem ser construídas para prover um mesmo serviço através de um grupo comum de funções de rede (HERRERA; BOTERO, 2016). Dessa forma, a primeira etapa do NFV-RA (*i.e.*, a composição) destina-se justamente a análise dessas diferentes possibi-

lidades de formação para detectar a(s) topologia(s) mais adequada(s) considerando uma função objetivo. Com isso, a entrada das soluções que implementam essa etapa tipicamente contém um esboço da topologia a ser composta, seja este modelado através de uma cadeia parcialmente ordenada, ou seja através da descrição dos possíveis relacionamentos entre as funções de rede requisitadas. Também, esse esboço topológico pode incluir informações sobre dependências e políticas a serem aplicadas durante a etapa de composição, assegurando características obrigatórias do serviço (*e.g.*, que um compressor de dados execute antes de um descompressor) e garantindo o correto funcionamento do mesmo.

Soluções de composição de SFCs recebem, normalmente através de um SFCR, informações básicas que possibilitam a sua execução (*e.g.*, VNFs que compõem a topologia, seus comportamentos operacionais e suas inter-relações). Entretanto, o SFCR provido deve ser compatível ao modelo adotado pela solução desejada tanto em sua estrutura, quanto na especificação da SFC e nos dados esperados pela função objetivo. Eventualmente, recomendações de implantação (*i.e.*, dados resultantes de uma investigação prévia realizada pelo desenvolvedor de um serviço de rede, estes sugerindo configurações ideais para implantação do serviço em diferentes ambientes físicos e virtualizados) e informações obtidas através de *benchmarking* de VNFs (ROSA; ROTHENBERG; SZABO, 2015) também podem ser utilizadas de maneira complementar ao SFCR. A seguir os principais trabalhos presentes na literatura que abordam a etapa de composição de SFCs são analisados e discutidos:

1. **Solução de Mehraghdam (MEHRAGHDAM; KELLER; KARL, 2014):** a solução de Mehraghdam apresenta um método formal para especificar e compor uma SFC através de uma gramática livre de contexto utilizada para criar topologias parcialmente ordenadas. Sendo assim, o SFCR utilizado contém uma topologia, descrita através desse modelo, em conjunto com a definição da taxa de tráfego (*i.e.*, o quociente entre a quantidade de dados encaminhados para os pontos de saída e a quantidade de dados recebidos nos pontos de entrada) de cada uma das VNFs participantes. A função objetivo utilizada pela solução de composição almeja reduzir a taxa de tráfego agregada a cada salto na SFC, para isso funções com características de descarte são alocadas antes das demais. Porém, essa solução de composição não considera dependências em segmentos parcialmente ordenados, forçando a fixação de VNFs para garantir características funcionais do serviço. Também, a modelagem topológica permite apenas a construção de ramificações terminais, isso se mostra insuficiente para serviços complexos, como no tratamento de tráfego de baixa

prioridade em soluções de segurança como VGuard (FUNG; MCCORMICK, 2015) e DeMONS (GARCIA et al., 2018), onde ramificações não terminais são empregadas para criar indireções no encaminhamento dos pacotes.

2. **Solução de Draxler (DRÄXLER; KARL, 2017):** na solução de Draxler um modelo de especificação baseado em YANG (BJÖRKLUND, 2016) (*i.e.*, linguagem para modelagem de dados) é utilizado para especificar topologias com ramificações genéricas e ordenamento parcial, além de informações de dependências das mesmas. Esse modelo é empregado como SFCR pela solução de composição a fim de tornar absoluta a ordem dos segmentos parcialmente ordenados. Assim sendo, são utilizadas funções objetivo que avaliam o tamanho da cadeia, a taxa de tráfego das VNFs e a quantidade de recursos necessários para a implantação da SFC. O algoritmo desenvolvido objetiva determinar a melhor relação de custo/benefício entre essas três características através de um modelo de otimização de Pareto e simulações de alocação em *hardware*. Apesar da flexibilidade do modelo YANG, como acontece em (MEHRAGHDAM; KELLER; KARL, 2014), não é possível explicitar dependências particulares dentro dos segmentos parcialmente ordenados, forçando fixações de VNFs que resultam na não consideração de todas as possibilidades de composição.
3. **Solução de Ocampo (OCAMPO et al., 2017):** a solução de Ocampo não processa uma topologia SFC explícita, mas sim os possíveis relacionamentos entre as VNFs que executam determinado serviço. Esses relacionamentos são descritos através de um modelo de grafo em um SFCR, onde também consta a taxa de tráfego e os requisitos computacionais para cada uma das VNFs participantes. O problema de composição é descrito através de Programação Linear Inteira (*Integer Linear Programming* - ILP), esta descrição resulta em uma função objetivo estática que visa reduzir a necessidade total de largura de banda para a implantação de uma SFC.
4. **Solução de Gil (GIL-HERRERA; BOTERO, 2017):** assim como a solução de Ocampo (OCAMPO et al., 2017), a solução de Gil também opera sobre os possíveis relacionamentos entre VNFs. O SFCR provido como entrada da solução deve conter a taxa de tráfego para cada uma das VNFs a fim de gerar uma função objetivo capaz de reduzir a necessidade total de largura de banda da SFC. A solução é desenvolvida através de uma meta-heurística baseada em busca tabu, esta investiga as possíveis vizinhanças no espaço

de busca (no caso, os relacionamentos possíveis a partir de uma VNF) para determinar a melhor transição local.

5. **Solução de Wang (WANG et al., 2017):** a solução de Wang apresenta um método de recuperação automática de dependências de SFCs ao utilizar o conhecimento prévio do comportamento operacional (*e.g.*, escrita, classificação, descarte, balanceamento de carga) das VNFs que as compõem, estes providos através de seus descritores. Esses comportamentos são mapeados para definir tanto as dependências de ordem, quanto eventuais conflitos que podem ser verificados em caso de descumprimento dessas dependências. Assim, um algoritmo de composição desenvolvido através da utilização de diagramas Hasse (um conjunto parcialmente ordenado e finito em forma de grafo) para representar e processar as cadeias é proposto. Esse algoritmo fixa a ordem dos elementos em segmentos parcialmente ordenados através da avaliação de uma prioridade atribuída a cada uma das VNFs, gerando uma topologia final com ordenamento absoluto.

Em geral, as soluções de composição apresentadas preocupam-se principalmente com a taxa de tráfego das VNFs e os requisitos de largura de banda das SFCs. Essas soluções tipicamente apresentam funções objetivo predefinidas e acopladas em seus algoritmos. Apesar de haver indicações da extensibilidade dos métodos adotados pelas soluções para a avaliação de funções objetivo diferentes daquelas prefixadas (OCAMPO et al., 2017; WANG et al., 2017), não existe uma implementação genérica que possibilite que essa mudança seja feita sem necessidade de manipulação de código e nem formalizações para descrever tais funções. A Tabela 2 sumariza as principais características dos trabalhos apresentados, nela o símbolo ✓ indica que a solução avaliada apresenta determinadas capacidades ou características de maneira completa, enquanto que o símbolo * indica a parcialidade e ✗ a ausência das mesmas. Por fim, a numeração indicando os trabalhos referenciados nesta mesma tabela corresponde a ordem utilizada durante a apresentação dos mesmos.

Os modelos de especificação de SFCs adotados pelas soluções indicam o tamanho da intervenção do operador de rede para a composição de um serviço. Modelos onde são especificados os segmentos e características da topologia estão presentes nos trabalhos [1] e [2]. Já modelos mais flexíveis, onde a capacidade de especificação do operador de rede é reduzida em decorrência da automatização da etapa de composição, estão presentes na descrição dos possíveis relacionamentos individuais entre VNFs ([3] e [4]) ou na inferência dessas relações através da análise comportamental das mesmas ([5]). Porém, essa automatização da etapa pode

Tabela 2 – Sumarização de Características das Soluções de Composição

	Trabalho Referenciado				
	[1]	[2]	[3]	[4]	[5]
Modelo de Especificação de SFC	GLC	SG (YANG)	SG (Possíveis Relacionamentos)	SG (Possíveis Relacionamentos)	SG (Possíveis Relacionamentos)
Solução/Formulação do Problema	Heurística Redutiva	Análise de Pareto	ILP	Pesquisa Tabu	Algoritmo Hasse e Algoritmo de Prioridades
Função Objetivo	Estática	Estática	Estática	Estática	Estática
Métricas Avaliadas	Taxa de Tráfego	Taxa de Tráfego Requisitos de Recursos Tamanho da SFC	Requisitos de Banda	Requisitos de Banda	Prioridade
Suporte a Criação de Ramificações	*	✓	✓	✓	✓
Suporte a Ordenamento Parcial	✓	✓	✓	✓	✓
Suporte a Dependência de Ordem	x	x	✓	✓	✓
Suporte a Dependência de Acoplamento	x	x	*	*	*

resultar em fatores limitantes indesejados que incidem sobre as possibilidades de atuação dos operadores de rede, ou seja, necessidades e conhecimentos específicos das VNFs que compõem o serviço e da rede onde este será integrado (*e.g.*, pontos de monitoramento, necessidade de determinada VNF processar todo o tráfego de entrada) podem ser desconsiderados durante a composição da topologia. Além disso, ações como a fixação de VNFs em posições absolutas na cadeia, a criação de ramificações obrigatórias ou o acoplamento de VNFs que não apresentam dependências operacionais entre si se tornam ações complexas, sendo necessária a definição de um grupo dependências artificiais para a efetivação das mesmas.

Como todos os trabalhos analisados utilizam funções objetivo estáticas, as formulações e soluções das etapa de composição são otimizadas para obterem os resultados mais adequados (sejam eles locais ou globais) para a o grupo fechado de métricas consideradas. Os trabalhos [1] e [2] utilizam-se de soluções que visam encontrar a melhor topologia dentro das combinações possíveis de seus segmentos parcialmente ordenados. Enquanto a solução [1] simplesmente ordena tais elementos a fim de minimizar a taxa de tráfego agregada, a solução [2] utiliza uma análise de Pareto para identificar as topologias que maximizam o custo/benefício entre as três métricas consideradas. Já os trabalhos [3], [4] e [5] não operam sobre a fixação e combinação de segmentos parcialmente ordenados, mas na criação das topologias por inteiro, aumentando seu espaço de busca. Dessa forma, as soluções adotados por esses trabalhos visam otimizar as métricas assistidas ao mesmo tempo que a topologia é formada, para isso diferentes abordagens podem ser adotados, como a análise de cada transição através de ILP ([3]), a decisão da transição através de pesquisa Tabu ([4]) ou a análise de todas as possíveis transições descritas em um diagrama Hasse ([5]).

Finalmente, é importante ressaltar que o retorno de uma única topologia como resultado da etapa de composição pode ocasionar eventuais dificuldades de integração. Isso ocorre já que determinadas composições podem ser mais adequadas aos objetivos e políticas de integração, tipicamente diferentes daqueles de composição. Por exemplo, enquanto que a solução de composição utilizada objetiva otimizar o balanceamento de carga (retornando topologias com muitas VNFs), uma solução de integração pode atuar para reduzir os custos financeiros calculados sobre a quantidade de VNFs instanciadas em Provedores de Infraestrutura (*Infrastructure Providers* - InP). Nesse caso, não só existem dois objetivos diferentes, como estes são conflitantes. Para solucionar cenários como o descrito, abordagens de conciliação, que indicam a topologia com o melhor custo e benefício entre diferentes objetivos, podem operar em uma lista de topologias gerada a partir de uma solução de composição e avaliada por uma solução de integração.

3.3 DIVISÃO E MAPEAMENTO DE CADEIAS DE FUNÇÃO DE SERVIÇO

A técnica de divisão e mapeamento é realizada durante a etapa de integração do NFV-RA quando existem múltiplos domínios ou Provedores de Infraestrutura (*Infrastructure Provider* - InP) disponíveis. Normalmente, soluções que a implementam optam entre dois tipos de arquitetura operacional: centralizada ou distribuída. Soluções centralizadas são executados em um único ponto de processamento, estas recuperam todas as informações necessárias dos domínios, avaliam as mesmas e retornam uma distribuição adequada as políticas avaliadas. Por outro lado, soluções distribuídas espalham a requisição de integração dos elementos virtualizados entre os domínios disponíveis através de um protocolo de comunicação comum aos mesmos. Neste caso, cada domínio avalia a requisição para determinar segmentos que podem ser hospedados por eles, encaminhando o restante aos domínios vizinhos. A seguir são detalhados e discutidos trabalhos que implementam soluções de divisão e mapeamento distribuídas (ABUJODA; PAPADIMITRIOU, 2016; ZHANG et al., 2016) e centralizadas (DIETRICH; ABUJODA; PAPADIMITRIOU, 2015; RIERA et al., 2016; WANG et al., 2017) em NFV.

1. **Nestor (DIETRICH; ABUJODA; PAPADIMITRIOU, 2015):** Nestor é uma solução centralizada de divisão e mapeamento de SFCs para múltiplos InPs. Esta solução introduz o conceito de pesos utilizados para caracterizar a adequabilidade da instanciação de uma VNF em determinado provedor através da análise de suas necessidades de re-

cursos computacionais e políticas. Dessa forma, os InPs devem divulgar um único dado numérico que indica desde a adequabilidade total a infraestrutura até a incapacidade de alocação na mesma. Após, a distribuição da topologia é realizada em vista da redução dos custos financeiros de processamento e transmissão, dos pesos divulgados e do número de provedores diferentes a serem utilizados. Porém, apesar do uso de informações de alto nível (*i.e.* pesos) ser benéfica por manter a privacidade de informações sensíveis aos provedores (*e.g.*, sobrecarga de infraestrutura, rotas disponíveis, capacidade de enlace), essa abordagem depende de padronizações no cálculo dos pesos para que resultados de origens diferentes sejam comparáveis, além da necessidade de cooperação dos InPs através da disponibilização de serviços para realização desses cálculos sob demanda. Por fim, a solução proposta recebe uma topologia a ser processada como um argumento imutável, sendo esta agnóstica a etapa de composição previamente executada.

2. **DistNSE (ABUJODA; PAPADIMITRIOU, 2016)**: o DistNSE é uma solução distribuída de divisão e mapeamento de SFCs que opera sobre uma topologia com ordenação absoluta. Esse algoritmo é executado através de trocas de mensagens entre InPs com um protocolo de comunicação particular. De maneira geral, as mensagens informam VNFs disponíveis para a alocação, além dos requisitos mínimos de cada uma delas (*e.g.*, recursos computacionais e dependências de infraestrutura). A decisão de alocação ou não das VNFs informadas cabe ao próprio provedor que processa as mensagens recebidas, caso o provedor opte pela não alocação de uma ou mais VNFs, uma nova rodada de mensagens repassando as mesmas é disparada aos seus provedores vizinhos. Após a conclusão das comunicações, o conjunto de distribuições possíveis é retornado a solução que, por sua vez, as analisa em vista da redução do custo financeiro total e balanceamento de carga inter-domínio. Uma característica importante desta solução é a possibilidade de alocação de múltiplas VNFs a cada sessão de troca de mensagens entre provedores, reduzindo assim a necessidade de comunicação entre os mesmos.
3. **Solução de Zhang (ZHANG et al., 2016)**: a solução de Zhang atua de maneira distribuída e utiliza uma metodologia centrada em vértices para realizar a divisão e mapeamento de uma topologia SFC. Esta solução considera a existência de segmentos parcialmente ordenados nas topologias requisitadas como maneira flexibilizar os resultados através do aumento do escopo de análise. Assim como o DistNSE, a decisão pela alocação ou não das VNFs presentes na topologia requisitada é transferidas aos InPs envolvidos, po-

rém, é dada uma limitação de alocação de uma VNF a cada troca de mensagem disparada, evitando a monopolização da SFC por um único provedor. Além disso, o processo não é realizado em vista de nenhuma política específica, sendo retornado ao operador de rede apenas o conjunto de possíveis distribuições que atendem aos requisitos de instanciação das VNFs que compõem a topologia. Por fim, os impactos decorrentes de mudanças na topologia (realizadas em segmentos parcialmente ordenados) não são avaliados pela solução, dessa forma a responsabilidade da verificação de custo/benefício entre os resultados da função objetivo aplicada na etapa de composição e de eventuais políticas que serão avaliadas durante a etapa de integração são transferidas ao operador de rede.

4. **TeNOR (RIERA et al., 2016)**: TeNOR é uma plataforma de orquestração para SFCs implantadas em ambientes multi-domínios que apresenta uma solução de divisão e mapeamento nativa. Essa solução centraliza informações sobre custos financeiros, atraso de transmissão e utilização de recursos computacionais, e as usa como entradas de sua política de minimização que decide a distribuição de uma topologia SFC. A solução de divisão e mapeamento presente no TeNOR permite que o operador de rede modifique pesos que ponderam cada uma das três métricas que compõem a política, alterando a prioridade de cada uma delas previamente a execução da técnica de distribuição. Entretanto, esta mesma solução não é capaz de verificar dependências de localização, utilizando toda a infraestrutura conhecida como candidata a hospedar qualquer uma das VNFs requisitadas. Finalmente, a requisição deve ser necessariamente formada por apenas uma topologia com posicionamento de VNFs absoluto, não abordando casos onde um mesmo serviço pode ser estruturado através de diferentes organizações topológicas.
5. **Solução de Wang (WANG et al., 2017)**: a solução de Wang considera um cenário híbrido onde tanto domínios privados, quanto InPs provêm recursos relacionados a redes ópticas. Nesse caso, a abordagem para a realização da divisão e mapeamento de SFCs apresenta duas políticas: a redução dos custos financeiros gerais e a redução de *slots* de frequência necessários. A formulação do problema favorece a utilização de domínios privados, uma vez que esses tipicamente apresentam custos financeiros reduzidos quando comparados aos InPs tradicionais. Ao considerar apenas *slots* de frequência como as requisições por banda, essa solução torna-se adequada somente quando redes ópticas são utilizadas. Além disso, nenhum tipo de dependência de SFCs é examinada durante o processo e não são realizadas avaliações multi topológicas.

De maneira geral, as soluções apresentadas concentram esforços na efetiva redução de custos financeiros relativos a utilização de diferentes InPs, além de outras políticas complementares que tipicamente objetivam aprimorar a Qualidade de Serviço (*Quality of Service - QoS*) das SFCs requisitadas. A Tabela 3 sumariza as soluções de divisão e mapeamento analisadas. Para a coluna de formulação do problema são apresentados, além de protocolos distribuídos particulares as soluções, dois modelos de otimização: Programação Linear Inteira (*Integer Linear Programming - ILP*) e Aprendizagem por Reforço (*Reinforcement Learning - RL*). Ainda, o símbolo ✓ indica que a solução avaliada apresenta determinadas capacidades ou características de maneira completa, enquanto que o símbolo ✗ indica a ausência das mesmas. Por fim, a numeração indicando os trabalhos referenciados nesta tabela corresponde a ordem utilizada durante a apresentação dos mesmos.

Tabela 3 – Sumarização de Características das Soluções de Divisão e Mapeamento

	Trabalho Referenciado				
	[1]	[2]	[3]	[4]	[5]
Arquitetura	Centralizada	Distribuída	Distribuída	Centralizada	Centralizada
Solução / Formulação do Problema	ILP	Protocolo de Integração	Protocolo Centrado em Vértices	ILP / RL	ILP
Heurísticas	Mapeamento de Subgrafo	<i>Time to Live</i>	N/A	N/A	<i>K-Cut</i>
Método de Avaliação	Estático	Estático	Estático	Estático	Estático
Políticas	Custos Financeiros Quantidade de InPs Pesos de Adequabilidade Recursos Computacionais	Custos Financeiros Balanceamento de Carga	N/A	Custos Financeiros Atraso Inter-domínios Recursos Computacionais	Recursos Computacionais <i>Slots</i> de Frequência
Suporte a Ordenamento Parcial	✗	✗	✓	✗	✗
Suporte a Dependências de Localização	✓	✓	✓	✗	✗

Os trabalhos que propõem técnicas de divisão e mapeamento distribuídas ([2] e [3]) estruturam suas soluções a partir do desenvolvimento de protocolos de comunicação entre domínios. Já as soluções centralizadas apresentadas em [1], [4] e [5] operam sobre um grafo representativo dos domínios disponíveis para verificar formas de distribuir a topologia requisitada. Em ambas as situações, o número de possíveis distribuições cresce de acordo com a quantidade de domínios disponíveis, a quantidade de elementos operacionais na topologia requisitada e a quantidade de elementos operacionais livres (*i.e.*, quando não existem dependências de infraestrutura ou quando estas não são computadas – como nos trabalhos [4] e [5]). Uma maneira de reduzir o esforço computacional e o tempo de execução dessas soluções consiste na adoção de heurísticas que, mesmo não garantindo resultados ótimos globais, buscam resultados considerados satisfatórios de acordo com um determinado critério (*e.g.*, tempo de execução, número de iterações, margem mínima de aceitação). Dentre os trabalhos considerados, essas heurísticas estão presentes na solução [1], [2] e [5] e realizam a redução do espaço de busca por, respecti-

vamente, divisão do problema, restrição da quantidade de provedores participantes e aumento da unidade de avaliação (segmentos da topologia ao invés de elementos operacionais).

As soluções distribuídas apresentadas em [2] e [3] usam dos orquestradores disponíveis nos InPs para coordenar o processo de divisão e mapeamento de SFCs. Entretanto, a distribuição desse processo requer cooperação inter-domínios através da implementação de protocolos comuns a eles. Apesar disso resultar em mínima necessidade de divulgação de informações intra-domínio, essa cooperação pode não ser concretizada devido a questões administrativas e interesses econômicos, como o aumento da área de competição de mercado e a criação de conglomerados de InPs cooperados. Uma alternativa a essas soluções é o uso de métodos centralizados para a execução da técnica de divisão e mapeamento. Algumas soluções centralizadas, como as presentes nos trabalhos [4] e [5], necessitam de informações internas dos InPs para avaliarem suas políticas. Porém, essas informações podem não serem divulgadas devido a acordos de confidencialidade entre os provedores envolvidos. Dessa forma, uma avaliação alternativa é apontada pelo trabalho [1], este transfere a operação análise de métricas confidenciais aos próprios InPs, restringindo as políticas da solução ao uso dos resultados agregados e obscurecidos retornados por eles e de outras métricas públicas.

Finalmente, cabe destacar que existem dois momentos de análise durante a execução de uma divisão e mapeamento: o primeiro é realizado com caráter eliminatório e indica se um domínio/provedores tem capacidades técnicas para hospedar uma determinada VNF. Essa constatação é tipicamente realizada visto a disponibilidade de recursos computacionais e tecnologias presentes no ambiente de virtualização, verificando se tais capacidades suportam requisitos mínimos de instanciação tipicamente anotados em Descritores de VNF (*VNF Descriptor - VNFD*); o segundo momento apresenta caráter classificatório, neste políticas previamente definidas são utilizadas para determinar qual dentre das distribuições possíveis é a mais adequada. Nas soluções de divisão e mapeamento investigadas a definição dessas políticas não é feita pelo operador de rede, mas sim inseridas de maneira estática pelos desenvolvedores das soluções. Sendo assim, essa limitação pode resultar na utilização de soluções que não refletem completamente as necessidades dos operadores, ocasionando distribuições sub-otimizadas devido ao excesso, falta ou avaliação inapropriada de políticas para cenários com necessidades particulares de integração.

3.4 SUMARIZAÇÃO E DISCUSSÃO

A alocação de recursos consiste de um conjunto de etapas para prover um serviço de rede virtualizado visando, entre outros, QoS e alto desempenho (*e.g.*, baixa latência, alto *throughput*, recomendações de implantação). Além disso, é necessário assegurar o cumprimento de dependências e de políticas de implantação que garantem tanto o correto funcionamento do serviço, quanto o cumprimento de regras de negócio, como o uso de domínios particulares ou de InPs e os custos associados a essa escolha. Dessa forma, para que a alocação de recursos em NFV ocorra de maneira personalizada, respeitando as necessidades e particularidades dos ambientes de implantação e serviços requisitados, é necessário que os modelos e soluções envolvidas no processo assegurem uma série de capacidades descritivas aos operadores de rede, tornando possível assim indicar em fina granularidade desde características topológicas a políticas de implantação.

As soluções de composição, divisão e mapeamento analisadas e discutidas no decorrer deste capítulo, apesar de serem eficientes para determinar resultados consistentes às análises que se propõem, não permitem ao operador de rede requisitar avaliações personalizadas de suas topologias SFC no que diz respeito as funções objetivo (composição) e políticas (divisão e mapeamento). Além disso, o desenvolvimento de soluções estáticas para implementar etapas de implantação (em especial do NFV-RA) gera como efeito colateral a utilização de modelos de especificação de SFCs também limitados, visto que estes representam as necessidades mínimas para permitir a execução da ferramenta escolhida, ignorando a especificação de diversas características de SFC que poderiam influenciar diretamente nos resultados obtidos (*e.g.*, organizações topológicas, dependências, compartilhamento). Também, como decorrência da construção de soluções estáticas para o NFV-RA, os modelos de requisição de SFC apresentam, além de sua topologia, apenas os valores de entrada para as métricas de avaliação previamente definidas, não sendo prevista nenhuma maneira de descrever tais métricas em si e nem mesmo as funções e políticas relacionadas a elas. Sendo assim, o Capítulo 4 apresenta modelos que flexibilizam a especificação de topologias SFC e a construção de SFCRs e, de posse desses modelos, soluções personalizáveis que implementam as etapas de composição e integração multi-domínio são detalhadas, respectivamente, nos Capítulos 5 e 6.

4 MODELOS PARA ALOCAÇÃO DE RECURSOS EM NFV

Este capítulo detalha os modelos de especificação de cadeias de função de serviço e de requisição criados para possibilitar a descrição de características, métricas, políticas e objetivos a serem considerados por soluções flexíveis e personalizáveis que implementam etapas do NFV-RA. Primeiramente, um novo modelo de especificação de topologias SFC baseado em GLC é apresentado. Este modelo possibilita desde a inserção de dependências de VNFs e de infraestrutura, até a criação de segmentos parcialmente ordenados e ramificações genéricas. Em seguida, um modelo de requisição de SFC que adota a especificação topológica criada também é apresentado. Nesse caso, o SFCR é estruturado para não só suportar a topologia SFC e os dados computados por soluções do NFV-RA, mas também a descrição de funções objetivo e políticas a serem avaliadas por elas.

4.1 MODELO DE ESPECIFICAÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO

A topologia da SFC é o principal elemento a ser processado durante a execução das etapas do NFV-RA. Nesse contexto, os principais desafios residem na criação de modelos de especificação topológica facilmente compreendidos pelos operadores de rede e que, ao mesmo tempo, são capazes de serem validados, mapeados e processados por soluções de composição, integração e agendamento de execução. Além disso, quanto maior a expressividade do modelo, maior a quantidade de informações que podem ser diretamente colhidas ou indiretamente inferidas durante o seu processamento por soluções de implantação (*e.g.*, tamanho e largura da topologia, possíveis ordenamentos absolutos para segmentos parcialmente ordenados). O modelo deve expandir suas capacidades de especificação, como ao possibilitar a definição do posicionamento absoluto de VNFs, ordenamentos parciais, ramificações e dependências, de forma a permitir análises abrangentes que retornem resultados corretos e consistentes as peculiaridades de diferentes serviços de rede.

Para assegurar ao operador de rede um maior número de capacidades de especificação em relação aos modelos existentes, discutidos anteriormente no Capítulo 3, a gramática livre de contexto intitulada SCAG (*Service Chain Grammar*) é proposta. A utilização de uma GLC é motivada devido sua formalização através de regras de produção, tornando fechado e conhecido o escopo de topologias aceitas pela mesma. Também, a alta disponibilidade de soluções capazes

de gerar validadores léxicos e sintáticos para GLCs (*e.g.*, Lex¹¹, Yacc¹², NLTK¹³) fortalece a decisão pela adoção de tal formalização. Por fim, as regras de produção de uma GLC podem ser planejadas em vista de sua legibilidade, tornando-as mais adequadas a compreensão de seus usuários. A seguir as premissas que dão origem ao modelo de especificação de SFC proposto são apresentadas na Subseção 4.1.1, após a formalização do modelo através de seus símbolos e regras de produção é descrita na Subseção 4.1.2, por fim um validador léxico, sintático e semântico para a gramática SCAG é detalhado na Subseção 4.1.3.

4.1.1 Premissas do Modelo

O modelo SCAG é fundamentado em premissas que definem sua estrutura e regras de relacionamento entre os seus componentes. Essa premissas dão previsibilidade e garantias quanto as características que estarão presentes nas topologias especificadas. Sendo assim, elas devem ser consideradas durante o processo de formalização da gramática, tornando-se guias para a criação de suas regras de produção. O conjunto de componentes do modelo engloba nós de borda, elementos operacionais e conexões lógicas. Sobre esses componentes podem ser definidos ordenamentos parciais e dependências de VNF e infraestrutura.

Os nós de borda são divididos em entrada (NE) e saída (NS), o primeiro recebe o tráfego que deve ser introduzido no serviço, o encaminhando para ser processado pelos elementos operacionais (ELEMO) presentes no mesmo; já o segundo recebe o tráfego já processado pelos elementos operacionais do serviço e o encaminha para fora do escopo do mesmo. A quantidade desses componentes presentes em uma SFC são representadas pelas notações *NNE*, *NNS* e *NELEMO*. Da mesma forma, uma conexão lógica pode ser classificada como de entrada (CLE), que provê o tráfego a ser processado por um ELEMO ou encaminhado por um NS, bem como de saída (CLS), que encaminha o tráfego proveniente de um NE ou já processado por um ELEMO. O número de conexões de entrada e saída em um componente é indicado respectivamente por *NCLE* e *NCLS*.

Segmentos parcialmente ordenados são utilizados para aumentar a área de atuação de soluções de composição durante o NFV-RA. Para esses segmentos, chamados de ORDEMP, uma ou mais dependências de VNF (DV) podem ser definidas. Essas dependências podem indicar tanto ordenamentos quanto acoplamentos entre pares de elementos operacionais. Além

¹¹ <http://dinosaur.compilertools.net/lex/index.html>

¹² <http://dinosaur.compilertools.net/yacc/index.html>

¹³ <https://nltk.org>

disso, dependências de infraestrutura (DI) atuam sobre elementos operacionais individuais e fixam estes a um determinado domínio físico. A quantidade de segmentos parcialmente ordenados presentes em uma SFC é dado por $NORDEMP$, já o número total de dependências de VNF que atuam sobre esses segmentos e o número de dependências de infraestrutura sobre um elemento operacional são indicados, nessa ordem, por NDV e NDI . A sumarização das notações utilizadas nas formulações de premissas discutidas a seguir é apresentada na Tabela 4. Ainda, é importante ressaltar que os subscritos presentes nas notações definidas significam uma ação de recuperação de informações, estas podem ser aplicadas tanto a uma topologia completa especificada segundo a SCAG (SFC), quanto a elementos específicos previsto na mesma, como nós de borda de entrada (NE), nós de borda de saída (NS), elementos operacionais (ELEMO) e segmentos parcialmente ordenados (ORDEMP).

Tabela 4 – Notações para a Definição das Premissas do Modelo

Notação	Descrição
NNE_x	Número de nós de entrada de dados presentes em x
NNS_x	Número de nós de saída de dados presentes em x
$NELEMO_x$	Número de elementos operacionais presentes em x
$NORDEMP_x$	Número de segmentos parcialmente ordenados presentes em x
$NCLE_x$	Número de conexões lógicas de entrada presentes em x
$NCLS_x$	Número de conexões lógicas de saída presentes em x
NDV_x	Número de dependências de VNF que atuam sobre x
NDI_x	Número de dependências de infraestrutura que atuam sobre x

A Equação 4.1 restringe em um a quantidade de nós de entrada de dados em uma SFC. Essa restrição vincula um serviço de rede ao seu ponto inicial, tendo seus caminhos orientados a fluxo e permitindo determinar exatamente quais as operações realizadas pelo serviço no decorrer de sua execução. A definição de um único nó de entrada para o serviço também facilita a identificação de intersecções topológicas, o que indica o compartilhamento de uma ou mais funções de rede entre diferentes serviços. Já as Equações 4.2 e 4.3 asseguram a existência de pelo menos um nó de saída de dados e um elemento operacional na especificação de uma SFC. Dessa forma, é possível determinar que a topologia mínima (*i.e.*, menor cadeia de serviço possível) deste modelo é constituída por um nó de entrada de dados, um elemento operacional e um nó de saída de dados.

$$NNE_{SFC} = 1 \quad (4.1)$$

$$NNS_{SFC} > 0 \quad (4.2)$$

$$NELEMO_{SFC} > 0 \quad (4.3)$$

A Equação 4.4 determina a obrigatoriedade da existência de uma, e apenas uma, conexão lógica de saída para o encaminhamento de dados entre NE e elementos operacionais. Assim sendo, o NE não apresenta nenhuma característica de processamento e qualquer necessidade de balanceamento de carga, espelhamento de tráfego ou encaminhamento através de classificações é atendida utilizando um elemento operacional que se segue a ele. Também, a Equação 4.5 garante que o NE não receberá tráfego de retorno, caracterizando uma topologia SFC unidirecional. Cadeias de retorno devem ser definidas através de novas topologias, uma vez que o encaminhamento e os comportamentos operacionais destas são diferentes de sua cadeia originária. Dessa forma, nessas novas topologias o NE pode passar a ser um NS, herdando as características deste componente que consistem do não encaminhamento interno de dados (Equação 4.6), além do suporte a diversas conexões lógicas de entrada para recebimento do tráfego processado pelos elementos operacionais da SFC (Equação 4.7).

$$NCLS_{NE} = 1 \quad (4.4)$$

$$NCLE_{NE} = 0 \quad (4.5)$$

$$NCLS_x = 0, \forall NS x \in SFC \quad (4.6)$$

$$NCLE_x > 0, \forall NS x \in SFC \quad (4.7)$$

Os elementos operacionais presentes na topologia SFC podem conter várias conexões lógicas de entrada e de saída de dados (Equações 4.8 e 4.9). Múltiplas conexões lógicas de entrada em um mesmo elemento operacional ocorrem na intersecção de ramos provenientes de ramificações não terminais, onde o primeiro elemento operacional após a mesma recebe o tráfego de seus anteriores. Já múltiplas conexões lógicas de saída ocorrem ao se definir o início de uma ramificação qualquer, neste caso o elemento operacional que da origem a mesma encaminha dados ao começo de cada um dos ramos especificados. Finalmente, é importante ressaltar que os elementos operacionais não indicam uma instância de determinada função de rede, mas sim a figura abstrata de uma função de rede que pode ser instanciada (ou ser física), além das operações de ciclo de vida aplicáveis a ela (*e.g.*, monitoramento, escala, migração). Isto posto, a reutilização de um elemento operacional não equivale necessariamente a reutilização de uma instância ou equipamento, e nem mesmo categoriza a topologia como compartilhada.

$$NCL E_x > 0, \forall ELEM O x \in SFC \quad (4.8)$$

$$NCL S_x > 0, \forall ELEM O x \in SFC \quad (4.9)$$

Também, a SCAG suporta a especificação de qualquer quantidade de segmentos parcialmente ordenado como indicado pela Equação 4.10. Esses segmentos, entretanto, devem conter pelo menos dois elementos operacionais para serem considerados válidos (Equação 4.11). As dependências de VNF atuam sobre os segmentos parcialmente ordenados e também podem ocorrer em qualquer quantidade para cada um deles (Equação 4.12). Essas dependências reduzem o espaço de busca por ordenamentos absolutos possíveis, porém garantem que as funcionalidades e características do serviço sejam mantidas independentemente do ordenamento escolhido durante uma futura etapa de composição. Já dependências de infraestrutura apresentam um limite de ocorrências na SFC, este definido pelo número de elementos operacionais contidos na mesma (Equação 4.13). Isso ocorre já que, como mostra a Equação 4.14, um elemento operacional pode hospedar apenas uma dependência de localização, permitindo assim a fixação do mesmo a infraestrutura física de onde sua execução deve ocorrer.

$$NORDEMP_{SFC} \geq 0 \quad (4.10)$$

$$NELEM O_x > 1, \forall ORDEMP x \in SFC \quad (4.11)$$

$$NDV_x \geq 0, \forall ORDEMP x \in SFC \quad (4.12)$$

$$NDI_{SFC} \leq NELEM O_{SFC} \quad (4.13)$$

$$NDI_x \geq 0 \wedge NDI_x \leq 1, \forall ELEM O x \in SFC \quad (4.14)$$

O mapeamento entre algumas premissas e regras de produção da gramática pode ocorrer de maneira explícita, como na representação de nós de borda, elementos operacionais e domínios através de símbolos terminais, ou na definição de delimitadores e modificadores para a criação de segmentos parcialmente ordenados, ramificações e dependências. Por outro lado, outras premissas podem ser construídas de maneira implícita, como na condução das transições entre regras de produção para que assegurem a coerência das conexões lógicas firmadas entre nós de borda e elementos operacionais. De qualquer forma, independentemente da maneira como ocorre a apresentação ao usuário, as premissas devem ser garantidas para que exista previsibilidade nas possíveis especificações geradas pelo modelo, o que é essencial para o desen-

volvimento de novos métodos e soluções que realizam o processo de implantação de topologias SFC.

4.1.2 Formalização do Modelo

A formalização da GLC é realizada através da quádrupla $SCAG = (\nu, \tau, \rho, \varsigma)$. Nesse caso, ν representa o conjunto de símbolos não terminais (INICIO, BLOCOOP, BLOCOOPR, BLOCOTP, ORDEMP, EXCECAO, RAMIFICACAOT, PRAMIFICACAOT, RAMIFICACAONT, RAMIFICACAONTINT, PRAMIFICACAONT, PELEMOI, ELEMOI, ELEMOM, NS, DOMINIO) e τ o conjunto de símbolos terminais (símbolos de controle - '<', '>', '{', '}', '(', ')', '[', ']', '/', '*', 'NE' - e símbolos variáveis - presentes nas regras de produção indicadas por ELEMOM, NS e DOMINIO) que formam a gramática. Esses símbolos são empregados na criação de regras de produção, sendo que o conjunto destas regras é representado por ρ e definido na Figura 5. É importante ressaltar que, por motivos de compatibilidade de implementação da SCAG através de verificadores léxicos e sintáticos existentes, nenhum símbolo de vazío foi introduzido em suas regras de produção. Finalmente, ς indica, no contexto de ρ , o símbolo não terminal INICIO e representa a regra de origem da gramática.

Os símbolos terminais presentes na SCAG representam delimitadores e metadados da especificação. Esses símbolos, que tem suas funções sumarizadas na Tabela 5, se dividem em duas categorias: controle e variável. Os símbolos de controle são elementos estruturantes essenciais da topologia que nunca são modificados. Já os símbolos terminais variáveis podem ser inseridos de maneira personalizada na gramática, uma vez que estes retratam os identificadores de elementos operacionais, nós de borda de saída e domínios de instanciação disponíveis. A obtenção desses símbolos variáveis pode ser feita, por exemplo, através da análise de informações em bancos de dados centralizados (*e.g.*, solicitando informações da topologia ao mercado de onde esta foi adquirida) ou de modelos de requisições que disponibilizam uma listagem dos elementos utilizados na topologia. Ainda, os desenvolvedores de soluções que adotam este modelo podem optar por definir padronizações para a nomenclatura dos símbolos variáveis, possibilitando a inserção estática dos mesmos na gramática.

As regras de produção da gramática asseguram a correta especificação das topologias SFC. A SCAG contempla um conjunto de 16 regras, apresentadas na Figura 5, onde a primeira (INICIO, linha 1) indica o momento inicial da especificação que ocorre através da inserção de um nó de borda destinado a entrada de dados (NE), seguido do direcionamento da produção

Tabela 5 – Símbolos Terminais da SCAG

Símbolo Terminal	Categoria	Descrição
'NE'	Controle	Nó de borda obrigatório que indica o ponto de entrada de dados da SFC.
'<', '>'	Controle	Delimitadores que abarcam um ID de domínio, indicam uma dependência de infraestrutura.
'(',')'	Controle	Delimitadores que abarcam dois IDs de elementos operacionais onde sua ordem de aparecimento apresenta a precedência esperada entre eles, indicam uma dependência de VNF.
'**'	Controle	Modificador de dependências de VNF, é alocado antes do símbolo ')', sua ausência indica uma dependência de ordem, sua presença indica um acoplamento.
'[,]'	Controle	Delimitadores que abarcam IDs de elementos operacionais, indicam um segmento parcialmente ordenado.
'{,}'	Controle	Delimitadores que abarcam IDs de elementos operacionais, indicam uma ramificação na topologia. Esta estrutura pode ocorrer de maneira recursiva.
'/'	Controle	Delimitador utilizado internamente a uma ramificação, indica a separação entre seus ramos.
IDs em ELEMTO	Variável	Identificadores de VNFs e PNFs, utilizados para construir o serviço da topologia especificada.
IDs em NS	Variável	Identificadores de nós de borda de saída, utilizados para delimitar a topologia especificada.
IDs em DOMINIO	Variável	Identificadores de domínios, utilizados em dependências de infraestrutura.

a regra BLOCOOP (linha 2). Na BLOCOOP ocorre a definição dos blocos operacionais da cadeia através de quatro possíveis encaminhamentos: inserção de ramificações terminais (RAMIFICACAOT, linha 7); inserção de ramificações não terminais (RAMIFICACAONT, linha 9); definição de um elemento operacional (BLOCOTP, linha 4) seguido por uma recursividade em BLOCOOP; e definição de um elemento operacional (BLOCOTP, linha 4) seguido de um nó de borda de saída (NS, linha 15). Os elementos operacionais representam as funções de rede disponíveis (VNFs e PNFs) com ou sem dependências de infraestrutura (ELEMTOI na linha 13, sendo os domínios especificados pelos símbolos terminais presentes em DOMINIO na linha 16), estes elementos são identificados pelos símbolos terminais apontados na regra ELEMTO (linha 14).

O posicionamento dos elementos operacionais pode ser absoluto ou parcial, segmentos de elementos operacionais parcialmente ordenados podem ser criados a partir da regra ORDEMP (linha 5). Dependências de VNF podem ser definidas apenas para esses segmentos e sua especificação ocorre através da regra EXCECAO (linha 6). Essas dependências podem acontecer de duas maneiras: ordem, com precedência de execução fraca, podendo existir outras NFs

entre as envolvidas na dependência de que a sua ordem seja mantida; ou acoplamento, com precedência de execução forte, onde não podem existir outras NFs entre aquelas envolvidas no acoplamento. Do ponto de vista de notação, tanto dependências de ordem quanto acoplamentos utilizam os mesmos delimitadores, porém dependências de ordem não apresentam o terminal '*', sendo este reservado para indicar acoplamentos.

Figura 5 – Regras de Produção da SCAG

- 1 **INICIO** → 'NE' BLOCOOP
- 2 **BLOCOOP** → RAMIFICACAOT | RAMIFICACAONT | BLOCOTP BLOCOOP | BLOCOTP NS
- 3 **BLOCOOPR** → RAMIFICACAONTINT | BLOCOTP BLOCOOPR | BLOCOTP
- 4 **BLOCOTP** → ORDEMP | ELEMROI
- 5 **ORDEMP** → '[' ELEMROI PELEMROI ']' EXCECAO | '[' ELEMROI PELEMROI ']'
- 6 **EXCECAO** → '(' ELEMROI ELEMROI ')' EXCECAO | '(' ELEMROI ELEMROI ')' | '(' ELEMROI ELEMROI '*' ')' EXCECAO | '(' ELEMROI ELEMROI '*' ')' EXCECAO
- 7 **RAMIFICACAOT** → BLOCOTP '{' BLOCOOP PRAMIFICACAOT '}'
- 8 **PRAMIFICACAOT** → '/' BLOCOOP PRAMIFICACAOT | '/' BLOCOOP
- 9 **RAMIFICACAONT** → BLOCOTP '{' BLOCOOPR PRAMIFICACAONT '}' BLOCOOP
- 10 **RAMIFICACAONTINT** → BLOCOTP '{' BLOCOOPR PRAMIFICACAONT '}' BLOCOOPR
- 11 **PRAMIFICACAONT** → '/' BLOCOOPR PRAMIFICACAONT | '/' BLOCOOPR
- 12 **PELEMROI** → ELEMROI PELEMROI | ELEMROI
- 13 **ELEMROI** → ELEMROI '<' DOMINIO '>' | ELEMROI
- 14 **ELEMROI** → 'EO1' | 'EO2' | 'EO3' | ... | 'EO_n'
- 15 **NS** → 'NS1' | 'NS2' | 'NS3' | ... | 'NS_n'
- 16 **DOMINIO** → 'DOM1' | 'DOM2' | 'DOM3' | ... | 'DOM_n'

Já ramificações presentes na topologia podem ser tanto terminais quanto não terminais. Em ramificações terminais todos os ramos são finalizados por um nó de borda de saída, tornando possível a presença de qualquer bloco operacional nos mesmos. A especificação de ramificações terminais segue as regras RAMIFICACAOT (linha 7, que define a estrutura geral básica) e PRAMIFICACAOT (linha 8, que define a construção de ramos). Por outro lado, ramificações não terminais possuem necessariamente uma intersecção entre seus ramos, indicando o ponto final da mesma. Essa estrutura é formada pela sua organização básica, apontada pela regra RAMIFICACAONT (linha 9), e pela geração de seus ramos como descrito na regra PRAMIFICACAONT (linha 11). Devido a característica de intersecção das ramificações não terminais, nenhum nó de borda de saída pode ser inserido em seus ramos, para isso torna-se necessária a utilização de um bloco operacional reduzido (BLOCOOPR, linha 3) e uma meta-ramificação não terminal (RAMIFICACAONTINT, linha 10) contendo apenas estruturas que não levam a regra NS. Ademais, é importante ressaltar que a notação utilizada para a especificação de rami-

ficações é compartilhada por ambos os tipos possíveis para a mesma.

A quádrupla que formaliza a SCAG não só define as notações utilizadas para a construção de topologias SFC, mas também assegura, através de suas regras de produção, o cumprimento das premissas do modelo definidas na Subseção 4.1.1. Através dessa formalização é possível avaliar a gramática frente as capacidades de especificação garantidas ao operador de rede. Essas capacidades são intimamente relacionadas ao número de categorias de SFC que são abordadas direta ou indiretamente pelo modelo. Levando em consideração as sete categorias de SFC previamente discutidas (*i.e.*, organização, largura, tamanho, dependências, simetria, heterogeneidade e compartilhamento), a SCAG apresenta as seguintes características:

- **Organização:** completamente suportada e explícita na topologia. O modelo permite a especificação de cadeias lineares e ramificadas;
- **Largura e Tamanho:** completamente suportada e explícita na topologia. Tanto a largura quanto o tamanho pode ser recuperada através da análise da cadeia uma vez que os elementos operacionais, nós de borda e conexões lógicas estão declarados na mesma;
- **Dependências:** completamente suportada e explícita na topologia. O modelo prevê a inserção de dependências de VNF (ordem e acoplamento) e de infraestrutura (domínio administrativo);
- **Simetria:** parcialmente suportada e implícita no modelo. De acordo com as suas premissas topológicas, o modelo suporta apenas a criação de cadeias assimétricas e unidirecionais;
- **Heterogeneidade:** não suportada e implícita no modelo. O modelo não distingue se um elemento operacional consiste de uma VNF ou PNF;
- **Compartilhamento:** não suportada e implícita no modelo. O modelo não distingue se um elemento operacional é compartilhado por mais de um serviço.

Dessa forma, o modelo SCAG é capaz de especificar cadeias que atendem aos principais requisitos de análise topológica das soluções que implementam etapas do NFV-RA discutidas no Capítulo 3. Para as soluções de composição, a especificação de ramificações, segmentos parcialmente ordenados e dependências de VNF eram fatores analisados, juntos a função objetivo, para otimizar o posicionamento absoluto das funções de rede na topologia. Já para a técnica

de divisão e mapeamento, o suporte a segmentos parcialmente ordenados e a dependências de localização era requerido pelas soluções investigadas. Finalmente, a formalização do modelo SCAG pode ser adaptada através de seus símbolos terminais, símbolos não terminais e regras de produção para permitir a especificação das categorias não suportadas.

4.1.3 Validação do Modelo

As topologias SFC criadas a partir da SCAG são tipicamente representadas por uma cadeia de caracteres. O processo de validação pode ocorrer através do exame dessa cadeia frente a quadrupla SCAG (para análise léxica e sintática) e uma série de avaliações extras que indicam a validade semântica da topologia especificada (*e.g.*, inexistência de ciclos em uma mesmas função de rede, de coexistências impossíveis de dependências e de utilizações inválidas de elementos operacionais). Para isso, um validador léxico, sintático e semântico é proposto e implementado utilizando a linguagem Python 3 como parte deste trabalho. Para a apresentação do processo de validação, o grupo de elementos operacionais (ELEMO) formado pelos símbolos terminais 'EO1', 'EO2', 'EO3', 'EO4' e 'EO5', o grupo de nós de borda de saída (NS) constituído por 'NS1' e 'NS2' e o grupo de domínios (DOMINIO) formado por 'DOM1' são utilizados. As topologias exemplo a seguir são empregadas para exemplificar o processo:

1. NE EO1 EO2 EO3
2. NE EO1 EO2 EO3 (EO2 EO3) NS1
3. NE [EO1 EO2 EO3] (EO3 EO4) EO4 NS1
4. NE [EO1 EO2 EO3 EO4] (EO1 EO2) (EO2 EO1) EO5 NS1
5. NE EO1 < DOM1 > { EO2 / EO3 / EO4 } EO5 NS1
6. NE EO1 { [EO2 EO3 EO4] (EO4 EO3 *) NS1 / EO5 NS2 }

O validador implementa as regras de produção da SCAG através da biblioteca NLTK Python 3. A NLTK realiza a verificação léxica e sintática da gramática através de análise redu-tiva tabular com a técnica empilha e reduz. Sendo assim, as árvores de derivação são construídas das folhas a raiz, verificando passo a passo os símbolos terminais de entrada face as regras de produção disponíveis. Correspondências entre conjuntos de símbolos no topo da pilha, sejam

eles terminais ou não, com regras de produção resultam na substituição dos mesmos pelo símbolo não terminal representante da regra constatada. Um erro é verificado quando o conjunto de símbolos no topo da pilha não permite a realização de nenhuma redução válida e nenhuma transição (*i.e.*, inserção de um novo símbolo no topo da pilha) é capaz de remover o processo deste estado.

Sendo assim, a validação léxica e sintática das topologias exemplo deve retornar negativo apenas para duas delas: [1] e [2]. A primeira topologia não apresenta nenhum nó de borda de saída, o que indica que algum bloco operacional (BLOCOOP) não foi encerrado. Já a segunda topologia insere uma dependência de VNF sem que haja um segmento parcialmente ordenado, uma vez que dependências desse tipo só podem ser derivadas a partir da regra ORDEMP, a gramática invalidará essa estrutura dado a requisição ilegal da regra EXCECAO. As demais topologias são classificadas como válidas quanto a suas formações léxicas e sintáticas. As seqüências de derivação das regras de produção para as especificações topológicas de exemplificação válidas são exibidas a seguir:

1. N/A
2. N/A
3. (INICIO **NE** (BLOCOOP (BLOCOTP (ORDEMP [(ELEMOI (ELEMOMO **EO1**)) (PELEMOI (ELEMOI (ELEMOMO **EO2**)) (PELEMOI (ELEMOI (ELEMOMO **EO3**))))] (EXCECAO ((ELEMOMO **EO3**) (ELEMOMO **EO4**)))) (BLOCOOP (BLOCOTP (ELEMOI (ELEMOMO **EO4**)) (NS **NS1**))))
4. (INICIO **NE** (BLOCOOP (BLOCOTP (ORDEMP [(ELEMOI (ELEMOMO **EO1**)) (PELEMOI (ELEMOI (ELEMOMO **EO2**)) (PELEMOI (ELEMOI (ELEMOMO **EO3**)) (PELEMOI (ELEMOI (ELEMOMO **EO4**))))] (EXCECAO ((ELEMOMO **EO1**) (ELEMOMO **EO2**) (EXCECAO ((ELEMOMO **EO2**) (ELEMOMO **EO1**))))) (BLOCOOP (BLOCOTP (ELEMOI (ELEMOMO **EO5**)) (NS **NS1**))))
5. (INICIO **NE** (BLOCOOP (RAMIFICACAONT (BLOCOTP (ELEMOI (ELEMOMO **EO1**) < (DOMINIO **DOM1**) >)) { (BLOCOOPR (BLOCOTP (ELEMOI (ELEMOMO **EO2**)))) (PRAMIFICACAONT / (BLOCOOPR (BLOCOTP (ELEMOI (ELEMOMO **EO3**)))) (PRAMIFICACAONT / (BLOCOOPR (BLOCOTP (ELEMOI (ELEMOMO **EO4**)))))) } (BLOCOOP (BLOCOTP (ELEMOI (ELEMOMO **EO5**)) (NS **NS1**))))

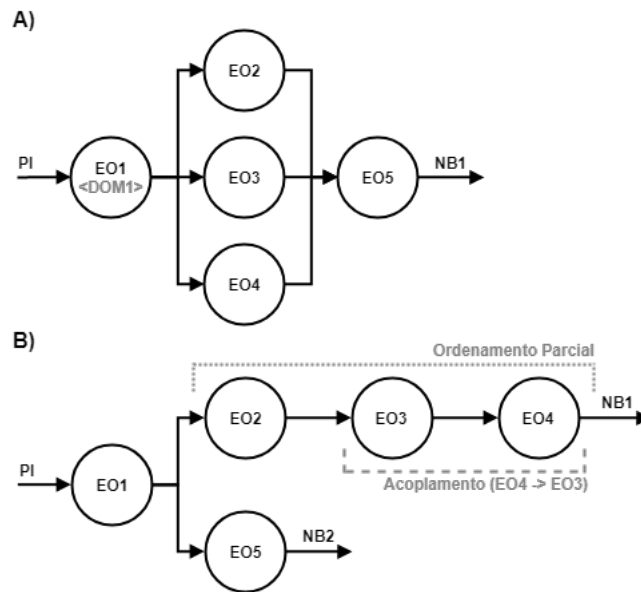
6. (INICIO **NE** (BLOCOOP (RAMIFICACAOT (BLOCOTP (ELEMOI (ELEMO **EO1**)))
 { (BLOCOOP (BLOCOTP (PORDER [(ELEMOI (ELEMO **EO2**)) (PELEMOI (ELE-
 MOI (ELEMO **EO3**)) (PELEMOI (ELEMOI (ELEMO **EO4**)))] (EXCECAO ((ELEMO **EO4**) (ELEMO **EO3**) *)))) (NS **NS1**)) (PRAMIFICACAOT / (BLOCOOP
 (BLOCOTP (ELEMOI (ELEMO **EO5**))) (NS **NS2**))) })))

Mesmo sendo válidas quanto a sua estrutura léxica e sintática, as topologias [3], [4], [5] e [6] ainda podem apresentar formações incoerentes. Dessa forma, um segundo processo de validação é executado com o objetivo de verificar a coesão semântica da especificação. Esse processo é realizado sobre as dependências de VNF e visa verificar se as mesmas são corretas em relação ao segmento parcialmente ordenado ao qual se referem, e se podem coexistir com as demais dependências aplicadas sobre este mesmo segmento. Dessa forma, o processo de validação semântica é dividido em dois passos: o primeiro analisa a consistência entre os elementos que formam as dependências e os elementos do segmento parcialmente ordenado ao qual se referem; o segundo verifica possíveis conflitos de dependência, como cruzamentos inválidos (*i.e.*, conjunto de dependências de ordem ou acoplamento que geram cenários impossíveis), dupla dependência de acoplamento (*i.e.*, um elemento operacional figura na mesma posição em mais de uma dependência de acoplamento) e auto dependência (*i.e.*, um elemento operacional depende dele mesmo).

A execução da validação semântica nas topologias de exemplo retorna resultados negativos para duas delas: [3] e [4]. A topologia [3] contém uma dependência que inclui um elemento operacional inválido (EO4), sendo esta descartada no primeiro passo do processo. Já a topologia [4] apresenta um cruzamento de dependências detectado no segundo passo da validação semântica, neste a primeira dependência de ordem determina a alocação de EO1 antes de EO2, já a segunda exige o caso contrário (alocação de EO2 antes de EO1), caracterizando assim um cenário impossível.

Após a conclusão do processo de validação, duas topologias são apontadas como corretas – [5] e [6] – e são ilustradas, respectivamente, na Figura 6.A e 6.B. A especificação [5] indica uma dependência de infraestrutura entre EO1 e DOM1, além disso, uma ramificação não terminal de três ramos, tendo início em EO1 e sendo unificada em EO5, está presente na topologia descrita. Por fim, em [6] a especificação contém uma ramificação terminal (iniciada em EO1) que consiste de dois ramos, sendo que o primeiro inclui um segmento parcialmente ordenado e um acoplamento indicando que EO4 deve localizar-se imediatamente antes de EO3. A vali-

Figura 6 – Topologias Completamente Validadas



dação realizada nas topologias garante que as mesmas estão adequadas para serem finalmente processadas por soluções de implantação de SFC.

4.2 MODELO DE REQUISIÇÃO DE CADEIAS DE FUNÇÃO DE SERVIÇO

A requisição de cadeias de função de serviço é uma das principais fontes de informações que alimentam os processos de implantação, em seus modelos de especificação é comum estar presente a topologia do serviço e descrições comportamentais e de desempenho dos elementos operacionais que a compõem. O modelo de uma requisição é tipicamente desenvolvido orientado as necessidades particulares de determinados métodos de composição, integração e programação de execução. Esse fenômeno é motivado devido a inflexibilidade e a baixa capacidade de personalização das avaliações realizadas por essas soluções de implantação uma vez que, normalmente, estas características são determinadas durante o desenvolvimento das mesmas, determinando assim um conjunto fechado de dados de desempenho e políticas a serem fornecidos pelos operadores de rede.

Considerando este cenário, uma requisição deve prover um conjunto de dados bem delimitado, imutável e previamente conhecido pelas soluções de implantação adotadas. Essa situação pode ser constatada nos modelos utilizados nos trabalhos discutidos no Capítulo 3. Nesses trabalhos constatam-se dois cenários quanto a estruturação de suas requisições: descrição dos dados solicitados pela solução, porém ausência de um modelo de dados para o fornecimento dos

mesmos (MEHRAGHDAM; KELLER; KARL, 2014; OCAMPO et al., 2017; GIL-HERRERA; BOTERO, 2017; DIETRICH; ABUJODA; PAPADIMITRIOU, 2015); ou indicação dos dados necessários para o processamento das soluções através de um modelo de dados especificado (RIERA et al., 2016; DRÄXLER; KARL, 2017).

O modelo de requisição utilizado para a solução TeNOR (RIERA et al., 2016) é estruturado através da linguagem de marcação *Java Script Object Notation*¹⁴ (JSON), sendo este aplicado para descrever tanto o serviço de rede, quanto os elementos operacionais que o compõem. Este modelo adota os parâmetros de descrição de VNF e serviços sugeridos pela ETSI (MANSNG, 2014) em adição a campos extras que indicam configurações do próprio TeNOR. Já o modelo utilizado em (DRÄXLER; KARL, 2017) é estruturado por meio da linguagem *Yet Another Next Generation* (YANG) (ENNS et al., 2011) que, além de ser capaz de especificar a topologia da cadeia requisitada, facilita o processo de definição de suas dependências e *Service Level Agreements* (SLA) dado a sua adequabilidade a criação de políticas (SCHONWALDER; BJORKLUND; SHAFER, 2010).

Apesar dos modelos de requisição existentes suportarem a especificação de serviços e de regras que atuam sobre eles (*e.g.*, dependências e políticas), estes não permitem a definição de dados de processamento customizados (*e.g.*, métricas que serão avaliadas em uma função objetivo ou utilizadas na definição de políticas) e nem mesmo de características operacionais relacionadas a avaliação e atualização dos mesmos (*e.g.*, objetivos como minimização/maximização e ponderações). Sendo assim, os modelos atuais suportam apenas um conjunto estático de informações previamente estabelecido pelas soluções de implantação que os adotam. Como maneira de flexibilizar este cenário, o presente trabalho propõem um modelo de requisição expansível e personalizável que, além de conter as especificações topológicas, é capaz de descrever completamente a função objetivo e as políticas a serem consideradas durante a execução das etapas do NFV-RA. A seguir, as Subseções 4.2.1 e 4.2.2 apresentam, respectivamente, as estruturas de dados adotadas no modelo de requisição proposto e a implementação de um validador para o mesmo.

¹⁴ <https://json.org/>

4.2.1 Modelo de Dados

O modelo de dados do SFCR proposto é definido através da linguagem de serialização *Yet Another Markup Language* (YAML)¹⁵. Essa linguagem é baseada no conceito de chave-valor, de forma que a chave representa o identificador utilizado para acessar elementos que podem ser desde variáveis e listas até objetos complexos. Diferente de outras linguagens semelhantes, como Markdown e JSON, o YAML objetiva definir claramente as estruturas de dados, sendo facilmente interpretado tanto por máquinas, quanto por programadores e operadores humanos. Além disso, o YAML se popularizou no contexto de NFV devido a sua adoção no modelo de especificação TOSCA NFV¹⁶, tornando-se amplamente conhecido e utilizado na área.

O modelo de requisição proposto é estruturado em cinco blocos principais (*i.e.*, metadados, serviço, função objetivo, políticas e implantação) capazes de especificar desde a topologia a ser instanciada até as suas métricas de avaliação, políticas e requisitos de instanciação. O bloco de metadados (chave METADADOS) contém elementos de identificação da requisição que podem ser utilizados tanto pelas soluções (*e.g.*, gerenciamento de duplicatas de requisições, manutenção de *logs* de implantação), quanto pelos operadores de rede (*e.g.*, sumarização dos objetivos da requisição). Já o bloco de serviço (chave SERVICIO) anota a topologia a ser instanciada além de especificar seus elementos operacionais e nós de saída. O bloco de função objetivo (chave FUNCAO_OBJETIVO), por sua vez, define as métricas a serem consideradas para a implantação do serviço, além das operações de avaliação e atualização das mesmas, estas informações são empregadas durante a geração da função objetivo utilizada no processo de composição de uma topologia SFC. O bloco de políticas (chave POLITICAS) é tipicamente avaliado por soluções de integração e pode especificar desde requisitos de desempenho (*e.g.*, latência máxima, vazão mínima) até regras de negócio de alto nível (*e.g.*, custo máximo por instanciação, número máximo de elementos operacionais em um provedor). Finalmente, o bloco de implantação (chave IMPLANTACAO) estabelece informações operacionais e de desempenho relacionadas a topologia requisitada, contendo desde previsões de recursos computacionais até *benchmarks* de execução dos elementos operacionais e comportamentos esperados para o serviço.

O bloco de metadados é a menor estrutura presente no modelo de dados proposto. Esse

¹⁵ <http://yaml.org/>

¹⁶ <http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/tosca-nfv-v1.0.html>

bloco, ilustrado na Figura 7, é constituído de apenas dois atributos, sendo suas chaves denominadas ID e DESCRICAO. O atributo representado pela chave ID consiste de uma cadeia de caracteres única que identifica uma requisição, permitindo assim que soluções, plataformas e/ou operadores de rede as gerenciem de maneira catalogada. O armazenamento das requisições é justificado devido a eventual necessidade de reavaliação da implantação do serviço, isto pode ocorrer, por exemplo, devido a mudanças no estado da infraestrutura física (*e.g.*, indisponibilidade ou congestionamento de um canal de comunicação) ou nas características de funcionamento esperadas para o serviço (*e.g.*, um determinado tipo de tráfego torna uma VNF um gargalo). Além disso, caso seja previsto o compartilhamento de elementos operacionais entre diferentes SFCs, as informações de políticas e SLAs contidas nas requisições devem estar disponíveis para alimentar verificadores de conflitos administrativos. Finalmente, a chave DESCRICAO indica um texto multilinear que descreve detalhes e apresenta explicações dos demais blocos da requisição. Uma vez que SFCs podem ser fornecidas como parte de um serviço durante sua aquisição em mercados de VNFs e SFCs, a DESCRICAO cumpre um papel importante ao prover informações sintetizadas para o operador de rede, permitindo ao mesmo realizar a avaliação da adequabilidade da requisição frente aos seus próprios objetivos e assim detectar eventuais necessidades de adaptações na mesma.

Figura 7 – Bloco de Metadados

```

1 METADADOS :
2   ID:!!str
3   DESCRICAO: >
4   ...
5   ...

```

O bloco de serviço, ilustrado na Figura 8, contempla as informações topológicas da SFC requisitada. Esse bloco apresenta quatro atributos principais cujo as chaves são identificadas por TOPOLOGIA, NO_ENTRADA, NOS_SAIDA e ELEMENTOS_OPERACIONAIS. O atributo TOPOLOGIA indica a estrutura do serviço através das conexões entre elementos operacionais e nós de borda. Não existe padronização de modelo de especificação de SFC a ser adotado por este modelo de requisição, entretanto a representação da topologia deve ocorrer através de uma cadeia de caracteres. Os demais atributos são utilizados para reconhecimento dos componentes topológicos, sendo assim, NOS_ENTRADA carrega a identificação do nó de borda de entrada de dados por meio de uma cadeia de caracteres, NOS_SAIDA contém uma lista de cadeias de caracteres representando todos os nós de borda de saída de dados e, finalmente, ELEMEN-

TOS_OPERACIONAIS consiste também de uma lista de cadeias de caracteres, porém esta indicando os elementos operacionais atuantes na topologia especificada.

Figura 8 – Bloco de Serviço

```

1 SERVICIO:
2   TOPOLOGIA:!!str
3   NO_ENTRADA:!!str
4   NOS_SAIDA:
5     -!!str
6     ...
7   ELEMENTOS_OPERACIONAIS:
8     - &eol!!str
9     ...

```

O bloco de função objetivo é apresentado na Figura 9 e destina-se a apresentar os parâmetros de execução a serem aplicados durante a etapa de composição do serviço. Esse bloco é constituído por uma única lista de objetos, sendo que cada dos objetos presentes na mesma descreve uma métrica de avaliação utilizada para a geração da função objetivo. Uma métrica é especificada através de seis atributos com chaves denominadas ID, OBJETIVO, PONDERACAO, ENTRADA, AVALIACAO e ATUALIZACAO. O atributo ID é uma cadeia de caracteres que identifica a métrica, este deve ser único dentro da requisição já que pode ser referenciado posteriormente no próprio bloco de função objetivo, além da sua presença obrigatória nos *benchmarks* do bloco de implantação. Como forma de padronização, é possível determinar um apelido para os atributos ID das métricas utilizando o marcador "&", este apelido pode ser empregado para indicar o conteúdo do atributo em futuras referências (marcador "*"). O atributo OBJETIVO também é estruturado sobre uma cadeia de caracteres e representa propósito de avaliação da métrica (tipicamente visto como a maximização ou minimização de seus resultados avaliativos). Já o atributo PONDERACAO é utilizado para adaptar a relevância de uma métrica em relação as demais definidas no mesmo bloco, este atributo consiste de um valor numérico em ponto flutuante maior ou igual a zero e menor ou igual que um. Métricas assinaladas com ponderação zero são usadas apenas para realizar a atualização de suas semelhantes, enquanto que uma métrica com valor de ponderação um representa que o processo de composição da topologia SFC ocorrerá unicamente em relação a mesma. É importante ressaltar que o somatório dos atributos PONDERACAO de todos os objetos presentes no bloco de função objetivo deve resultar em um.

O atributo ENTRADA recebe um valor numérico (inteiro ou ponto flutuante) que é

Figura 9 – Bloco de Função Objetivo

```

1 FUNCAO_OBJETIVO:
2   - ID: &metrical!!str
3   OBJETIVO:!!str
4   PONDERACAO:!!float
5   ENTRADA:!!int ou !!float
6   AVALIACAO:!!str
7   ATUALIZACAO:!!str ou *metric(n-m)
8   ...

```

computado como o parâmetro inicial das operações de avaliação e atualização descritas para a métrica (*e.g.*, valores de estimativa como de taxa de tráfego esperada ou valores de acúmulo como para o tamanho de uma topologia SFC). A entrada é modificada tanto ao final de cada iteração de elementos operacionais (momento de atualização) ou na presença de um ponto de ramificação ou intersecção topológica (respectivamente, divisão e agregação) durante a execução da etapa de composição. O atributo AVALIACAO consiste de uma cadeia de caracteres que apresenta a função a ser calculada entre o valor de entrada e o valor correspondente a métrica de um elemento operacional, realizando a avaliação do mesmo. Essa função pode ser descrita de diferentes maneiras, como através da indicação de simples operações matemáticas (*e.g.*, soma, subtração, multiplicação e divisão) até equações mais complexas, a liberdade de descrição da função depende das capacidades e padronizações utilizadas pela solução de composição adotada. Finalmente, o atributo ATUALIZACAO é formado por uma cadeia de caracteres que corresponde a um ID de métrica previamente definidos (o seu próprio ou outro proveniente do bloco de função objetivo) ou a uma função de atualização disponibilizada pelas soluções. No caso de processos de atualização vinculados a uma métrica, a operação de avaliação desta deve ser aplicada para atualizar o valor de entrada a ser utilizado na próxima iteração, por exemplo, se o uso de memória é diretamente proporcional a taxa de tráfego, a entrada da métrica de memória deve ser atualizada através da função de avaliação da métrica taxa de tráfego.

O bloco de políticas, exemplificado na Figura 10, tem por objetivo estabelecer os parâmetros de cômputo a serem utilizados durante a etapa de integração do serviço. Esse bloco é constituído por duas subestruturas que identificam a maneira de avaliação de uma política. A primeira diz respeito a políticas de avaliação imediata (chave IMEDIATAS), ou seja, os valores são analisados localmente para cada alocação realizada durante a execução de uma solução de integração (*e.g.*, custo financeiro máximo para instanciar uma VNF). A segunda se refere a políticas de avaliação agregada (chave AGREGADAS), para elas o acúmulo dos resultados pro-

venientes das alocações já realizadas por uma solução de integração é considerado (*e.g.*, custo financeiro total para a instanciação de um serviço). Ambas as subestruturas são compostas por uma lista de objetos, sendo que cada objeto descreve uma política que deve ser observada durante a distribuição, seleção e/ou alocação em *hardware* da topologia SFC requisitada. O objeto de políticas, por sua vez, é composto de seis atributos com chaves denominadas ID, PONDERACAO, VERIFICACAO, MINIMO, MAXIMO e OBJETIVO. O atributo ID é uma cadeia de caracteres identificadora, este deve ser único já que, apesar de não ser reutilizado em nenhum outro campo da requisição como ocorre para os IDs de métricas do bloco função objetivo, será posteriormente empregado pelas soluções de integração para reconhecimento das políticas especificadas.

Figura 10 – Bloco de Políticas

```

1 POLITICAS :
2   IMEDIATAS :
3     - ID:!!str
4     PONDERACAO:!!float
5     TIPO:!!str
6     MINIMO:!!int ou !!float
7     MAXIMO:!!int ou !!float
8     OBJETIVO:!!str
9     ...
10  AGREGADAS :
11    ...

```

O atributo PONDERACAO deve ser um valor em ponto flutuante maior ou igual a zero e menor ou igual a um, diferente da ponderação aplicada as métricas que formam uma função objetivo, onde zero identifica métricas não avaliadas mas utilizadas como meio de atualização de outras métricas, a ponderação igual a zero em políticas determina que elas serão apenas consideradas em etapas de análise eliminatória de distribuições (verificando máximos e mínimos), não sendo consideradas em etapas posteriores de classificação das mesmas. Por fim, o somatório do atributo PONDERACAO de todas as políticas definidas deve resultar em um. O atributo TIPO é formado por uma cadeia de caracteres e representa o ponto onde determinada política deve ser analisada. Esses pontos podem variar de acordo com a técnica de integração a ser utilizada, por exemplo, no caso de uma divisão e mapeamento os pontos de verificação podem ocorrer internamente a um domínio (*e.g.*, custos de instanciação para um provedor) ou durante uma transição entre domínios (*e.g.*, latência inter-provedores). Os atributos MAXIMO e MINIMO indicam os limites de aceitação da política, ambos são indicados através de números

inteiros ou pontos flutuantes. Por fim, o atributo OBJETIVO consiste de uma cadeia de caracteres que guia as soluções de integração quanto aos resultados ideais para a avaliação de uma política. Um objetivo pode indicar, por exemplo, uma minimização, sendo o resultado mais apropriado aquele que se aproxima do atributo MINIMO. A análise decorrente do atributo OBJETIVO pode ser utilizada como critério de desempate quando diferentes ações são possíveis em uma única iteração das soluções de integração.

Por último, o bloco de implantação, ilustrado na Figura 11, consiste de um conjunto de informações relacionadas aos elementos operacionais e ramificações da topologia SFC requisitada. Esse bloco é composto por duas subestruturas denominadas DESCRICAO_EOS e DESCRICAO_RAMIFICACOES. A primeira subestrutura (*i.e.*, DESCRICAO_EOS) constitui-se por um dicionário em que as chaves representam o atributo ID dos elementos operacionais existentes (informados pelo bloco de serviço). Cada chave indica um objeto traduzido por dois outros objetos chamados de FLAVOUR e BENCHMARK. O objeto FLAVOUR caracteriza a quantidade de recursos computacionais necessários para a instanciação de um dado elemento operacional, para isto são previstos três atributos intitulados MEMORIA, REDE e CPU. Todos os três atributos em FLAVOUR armazenam valores inteiros que significam a quantidade de memória volátil a ser reservada (MEMORIA), a quantidade de interfaces de rede necessárias (REDE) e o número de núcleos a serem alocados (CPU) para a instanciação do elemento operacional ao qual se referem. Já o objeto identificado como BENCHMARK é utilizado para prover dados relacionados as métricas presentes no bloco de função objetivo, estes computados durante a etapa de composição do serviço. Dessa forma, o BENCHMARK é composto por um grupo de atributos com chaves equivalentes ao atributo ID presente nas métricas previamente especificadas no bloco de função objetivo e conteúdos representados por valores inteiros ou em ponto flutuante.

Já a subestrutura DESC_RAMIFICACOES é formada por um dicionário cujo as chaves são as métricas que compõem o bloco de função objetivo. Cada uma dessas chaves aponta a um objeto que determina como ocorre a divisão dos valores de entrada (*i.e.*, valores avaliados e atualizados para uma métrica durante a execução da etapa de composição do serviço) em cada ramo na eminência do início de uma ramificação na topologia. Esse objeto é composto por uma função de divisão exposta através do atributo DIVISAO (descrito de maneira semelhante ao atributo AVALIACAO dos objetos de métricas presente no bloco de função objetivo), além de uma lista contida no atributo FATORES com argumentos utilizados para o cálculo desta função.

Figura 11 – Bloco de Implantação

```

1 IMPLANTACAO:
2   DESCRICAO_EOS:
3     *eol:
4     FLAVOUR:
5       MEMORIA:!!int
6       REDE:!!int
7       CPU:!!int
8     BENCHMARK:
9       *metrical:!!int ou !!float
10      ...
11     *metrican:!!int ou !!float
12    ...
13  DESCRICAO_RAMIFICACOES:
14    *metrical:
15      DIVISAO:!!str
16      FATORES:
17        - [!!int, !!int, !!int]
18        ...
19        - [!!float, !!float]
20    ...

```

Os fatores devem estar ordenados de acordo com a ordem de ocorrência dos pontos iniciais de ramificações na topologia através da leitura linear da cadeia de caracteres representante da mesma (semelhante a uma busca em profundidade no grafo de serviço) e devem conter um argumento do tipo inteiro ou ponto flutuante a ser calculado pela função de divisão para cada ramo presente em sua estrutura.

A união dos cinco blocos apresentados constitui um modelo de dados que contempla tanto a definição dos componentes topológicos e valores destinados a avaliação da SFC requisitada, quanto os processos que devem ser executados a partir dessas informações. Modelos de requisição capazes de descrever processos são ferramentas fundamentais que possibilitam o desenvolvimento de soluções de implantação flexíveis e personalizáveis, permitindo aos operadores de rede configurar suas avaliações de acordo com o cenário de implantação e necessidades particulares de cada cliente.

4.2.2 Validação do Modelo

As requisições criadas a partir do modelo proposto são validadas estruturalmente (*i.e.*, especificação adequada a linguagem YAML) e semanticamente (*i.e.*, validade dos dados infor-

dados para a definição dos blocos). Um validador para o modelo, desenvolvido¹⁷ por meio da linguagem Python 3 em conjunto a biblioteca PyYAML, é apresentado como parte deste trabalho. Esse validador realiza a análise estrutural das requisições através de rotinas genéricas dispostas pela PyYAML, enquanto que a análise semântica é implementada por um grupo de rotinas que verificam os valores contidos em cada uma das chaves presentes na requisição. Para que isso seja possível, coleções de valores possíveis são vinculadas a alguns atributos, gerando padronizações sumarizadas pela Tabela 6 e discutidas no decorrer desta subseção.

Tabela 6 – Padronização de Valores de Validação do Modelo

Bloco	Atributo	Valores Possíveis
SERVICO	TOPOLOGIA	SCAG
FUNCAO_OBJETIVO	OBJETIVO (Objeto de Métrica)	"MAX", "MIN"
	AVALIACAO (Objeto de Métrica)	"SOM", "SUB" "MUL", "DIV"
	ATUALIZACAO (Objeto de Métrica)	"ESTATICO"
POLITICAS	OBJETIVO (Objeto de Política)	"MAX", "MIN"
	TIPO (Objeto de Política)	"LOCAL" "TRANSICAO"
IMPLANTACAO	DIVISAO (Objeto de Ramificação)	"SOM", "SUB" "MUL", "DIV"

Os atributos que não têm seu conjunto de valores definido devem seguir as especificações de tipagem e estrutura definidas pelo modelo de dados. Nesse sentido, o bloco de metadados é o único que não segue nenhum tipo de padronização para seus atributos, enquanto os blocos de serviço, função objetivo, políticas e implantação apresentam atributos pontuais com restrição de valores: TOPOLOGIA para o bloco de serviço; OBJETIVO, AVALIACAO e ATUALIZACAO para o bloco de função objetivo; OBJETIVO e TIPO para o bloco de políticas; e DIVISAO para o bloco de implantação. Alguns desses atributos apresentam o mesmo conjunto de valores suportados, isso ocorre visto que operações similares podem ser aplicadas a contexto diferentes durante a implantação do serviço.

O atributo TOPOLOGIA presente no bloco de serviço pode ser definido através de qualquer topologia válida para o modelo SCAG, sendo o validador deste último empregado durante o processo de validação da requisição. Os valores "MAX" e "MIN" são destinados a descrição

¹⁷ Disponível em <https://github.com/ViniGarcia/NFV-FLERAS>

do objetivo de avaliação das métricas que compõem a função objetivo e dos critérios de análise de políticas para tomada de decisão durante a etapa de integração do serviço, em ambas as situações esses valores estão presentes nos atributos intitulados OBJETIVO de seus respectivos objetos. Nesse caso, o valor "MAX" indica a opção pela maximização dos resultados calculados para métricas e a preferência por opções que se aproximem aos valores contidos no atributo MAXIMO das políticas, pelo contrário, o valor "MIN" indica a minimização das métricas e a inclinação pela proximidade do atributo MINIMO para políticas.

Os valores "SOM", "SUB", "MUL" e "DIV" são alocados nos atributos denominados por AVALIACAO (no objeto de métricas do bloco de função objetivo) e DIVISAO (no objeto de ramificação no bloco de implantação). Esse conjunto de valores representa operações matemáticas a serem executadas entre dois argumentos, sendo "SOM" para somatório, "SUB" para subtração, "MUL" para multiplicação e "DIV" para divisão. O valor "ESTATICO" pode ser utilizado no atributo ATUALIZACAO do objeto de métricas presente no bloco de função objetivo, este é empregado para indicar que o atributo ENTRADA do mesmo objeto não sofre nenhuma atualização entre as iterações avaliativas, além desta cadeia de caracteres, o atributo ATUALIZACAO pode conter qualquer identificador de métricas previamente definidas. Os valores "LOCAL" e "TRANSICAO" são reservados para definir os tipos de políticas que podem ser aplicadas. Uma política "LOCAL" é verificada toda vez que uma operação interna a um determinado domínio ou equipamento é executada (*e.g.*, integração de uma VNF a um provedor ou equipamento específico), enquanto que políticas do tipo "TRANSICAO" ocorrem na eminência de uma mudança do local de avaliação (*e.g.*, transição entre equipamentos, redes ou provedores). Por fim, é importante evidenciar que, apesar do validador desenvolvido operar apenas com topologias do tipo SCAG e com os conjuntos de valores apresentados, o modelo de requisição é flexível, expansível e independente para adotar valores diferentes dos determinados para esta solução.

Ainda, o processo de validação, além de realizar a verificação do conteúdo presente nos atributos que seguem algum tipo de padronização de valores, também é responsável por averiguar se os valores presentes nos demais atributos correspondem ao tipo previsto no modelo de dados. Além disso, outras análises incidem sobre o conteúdo dos atributos definidos, como por exemplo a verificação se todos os componentes da topologia (bloco de serviço) estão contidos nos atributos NO_ENTRADA, NOS_SAIDA e ELEMENTOS_OPERACIONAIS; se todas as ramificações indicadas pela topologia (bloco de serviço) estão corretamente descritas no atributo RAMIFICACOES (bloco de implantação); se os valores em MAXIMO e MINIMO

(bloco de políticas) são coerentes entre si; se o somatório dos atributos PONDERACAO (tanto no bloco de políticas, quanto no de função objetivo) resultam em um; além da verificação da existência de todos os blocos e de todos os atributos previstos para os mesmos e para os objetos que os compõem.

4.3 SUMARIZAÇÃO E DISCUSSÃO

Os modelos de especificação de topologias e de requisição descrevem serviços virtualizados de rede e suas principais características (*e.g.*, funções de rede, conexões, *benchmarks*, ponderações). Atualmente, a especificação desses serviços pode ser realizada tanto através de modelos de baixo nível, que remetem diretamente ao grafo de serviço (GARAY et al., 2016), quanto pela descrição de topologias em alto nível, empregando um conjunto de regras e capacidades descritivas bem definidas, como ocorre em (MEHRAGHDAM; KELLER; KARL, 2014), ou definindo as rotinas de processamento de tráfego e seus momentos de aplicação, a exemplo da álgebra de pacotes em (SHIN et al., 2015). Modelos de baixo nível são normalmente usados como meta-modelos, mapeando especificações mais abstraídas e as tornando adequadas ao processamento realizado por algoritmos de implantação e/ou de operações de ciclo de vida. Já os de mais alto nível são diretamente utilizados por operadores e gerentes de rede. Nesse caso, modelos que explicitam a topologia de um serviço carregam informações essenciais sobre a sua organização, tamanho, largura e dependências, sendo usualmente adotados durante o processo de implantação.

Gramáticas livres de contexto são recursos amplamente conhecidos na área de linguagens formais. Essas podem ser elaboradas com o intuito de especificarem serviços virtualizados de rede, formalizando capacidades descritivas e assegurando aspectos léxicos e sintáticos através de regras de produção. A aplicação de GLCs no contexto de SFCs foi inicialmente explorada em (MEHRAGHDAM; KELLER; KARL, 2014). Nele, uma gramática é proposta permitindo a especificação de topologias com diversas funções de rede e nós de borda dispostos em segmentos com ordenação absoluta, segmentos parcialmente ordenados, ramificações terminais e execuções paralelas. Entretanto, estruturas de ramificações não terminais, requisitadas por serviços como a solução de segurança DeMONS (GARCIA et al., 2018), e dependências de VNF e infraestrutura não são abordadas pela mesma. Considerando isso, este trabalho apresenta a SCAG, uma nova GLC que, além de abranger todas as capacidades de especificação da primeira gramática proposta, adiciona a possibilidade de inserção de ramificações não terminais

em qualquer ponto de uma topologia, de dependências de ordem e acoplamentos em segmentos parcialmente ordenados e de dependências de infraestrutura (domínio) em elementos operacionais específicos.

Os modelos de requisições existentes são normalmente orientados as necessidades particulares de soluções que implementam alguma das etapas de implantação. Esses podem ser organizados como uma lista de argumentos sem definições de uma estrutura formal, como acontece em (MEHRAGHDAM; KELLER; KARL, 2014; DIETRICH; ABUJODA; PAPADIMITRIOU, 2015; OCAMPO et al., 2017; GIL-HERRERA; BOTERO, 2017), ou através de um modelo de dados bem definido, usando recursos como linguagens de marcação ou de serialização (RIERA et al., 2016; DRÄXLER; KARL, 2017). Entretanto, os modelos de requisição existentes permitem a especificação apenas de serviços, regras gerais (*e.g.*, políticas e regras de negócio) e *benchmarks* de funções de rede para métricas previamente estabelecidas por soluções de implantação, não permitindo, por exemplo, a indicação e customização de novas métricas, funções objetivo e configurações de avaliação (*e.g.*, ponderações, objetivos). Sendo assim, um novo modelo de requisições, planejado utilizando a linguagem de serialização de dados YAML, foi desenvolvido no contexto deste trabalho. Esse modelo é dividido em cinco blocos (metadados, serviço, função objetivo, políticas e implantação) os quais o de metadados e de serviço se assemelham aos dos demais modelos formalizados, o de políticas tem seu potencial expandido para permitir a descrição de objetivos e ponderações de avaliação das mesmas, o de função objetivo é inédito e assegura a descrição de métricas e seus modos de avaliação e, finalmente, o de implantação é flexibilizado para comportar dados de teste de funções de rede e seus requisitos computacionais de maneira personalizada considerando os demais blocos.

Os modelos desenvolvidos no decorrer deste capítulo são genéricos, ou seja, podem ser aplicados a diferentes soluções de qualquer etapa do NFV-RA. Porém, é necessário incluir nessas soluções a capacidade de explorar as diferentes estruturas e características contempladas pelo modelo de especificação de topologias e a flexibilidade e poder de configuração do modelo de requisições. Para isso, metodologias capazes de avaliar um conjunto imprevisível e heterogêneo de métricas e políticas também precisam ser concebidas. Dessa forma, duas soluções, uma de composição e outra de integração pela técnica de divisão e mapeamento, empregando os novos modelos construídos são detalhadas, respectivamente, nos Capítulos 5 e 6.

5 COMPOSIÇÃO ORIENTADA A FUNÇÃO OBJETIVO PERSONALIZADA

Mesmo existindo diversas soluções de composição de SFCs na literatura, como demonstrado no Capítulo 3, estas apresentam baixa capacidade de personalização, realizando um conjunto de análises previamente determinadas. Em outras palavras, as soluções existentes trabalham com uma função objetivo prefixada durante o processo de desenvolvimento das mesmas, não permitindo ao operador de rede determinar métricas de interesse que reflitam cenários de implantação específicos. A exemplo disso estão as soluções de composição apresentadas no Capítulo 3 que usam apenas um grupo estático de métricas (*i.e.*, taxa de tráfego, requisitos de recursos computacionais, requisitos de banda e tamanho da SFC) para construir suas funções objetivo. Além disso, cada solução apresenta um tipo de retorno particular e não protocolado, o que dificulta a utilização de tais resultados por outras ferramentas que executam etapas de implantação distintas. Dessa forma, uma solução de composição com suporte a personalização de função objetivo e com um processo de avaliação genérico e padronizado é proposta no contexto deste trabalho. Para tanto, os modelos definidos no decorrer do Capítulo 4 são utilizados para a especificação topológica e da requisição submetidas como argumentos à solução. Os procedimentos de construção da função objetivo e avaliação da topologia requisitada são apresentados Seção 5.1, nesta também evidência-se decisões de projeto e detalhes de implementação da solução que é avaliada na Seção 5.2. Por fim, a Seção 5.3 apresenta uma sumarização e uma discussão geral sobre o capítulo.

5.1 METODOLOGIA DA SOLUÇÃO

A solução de composição é dividida em três procedimentos principais: expansão de topologias, construção da função objetivo e avaliação de topologias. O primeiro procedimento (*i.e.*, expansão de topologias) expande as possibilidades de composições através da resolução de ordenamentos parciais e processos de redução de ramificações. O segundo procedimento (*i.e.*, construção da função objetivo) identifica e processa as métricas presentes no SFCR, estas são utilizadas para construir uma função objetivo a ser analisada durante a avaliação topológica. Já o terceiro procedimento (*i.e.*, avaliação de topologias) consiste da aplicação da função objetivo gerada em cada uma das composições candidatas resultantes da expansão de topologias, atribuindo a elas um índice de adequabilidade que caracteriza as diferentes organizações

topológicas de um mesmo serviço em relação as métricas requisitadas. O índice de adequabilidade, por sua vez, é utilizado para ranquear as topologias analisadas, possibilitando a solução sugerir uma organização a ser adotada para próximas etapas de implantação do serviço. Cada um desses procedimentos apresenta um conjunto de passos particulares a serem cumpridos, a descrição de cada um deles é apresentada no decorrer da presente seção.

Figura 12 – Bloco de Serviço do Exemplo Padrão

```

1 SERVICIO:
2 TOPOLOGIA: !!str NE [ EO1 EO2 EO3 ] (
    EO2 EO3 ) { EO4 < DOM1 > EO5 NS1 /
    EO4 < DOM1 > EO6 NS2 }
3 NO_ENTRADA: NE
4 NOS_SAIDA:
5 - NS1
6 - NS2
7 ELEMENTOS_OPERACIONAIS:
8 - EO1
9 - EO2
10 - EO3
11 - EO4
12 - EO5
13 - EO6

```

Figura 13 – Bloco de Função Objetivo do Exemplo Padrão

```

1 FUNCAO_OBJETIVO:
2 - ID: TxTrafego
3 OBJETIVO: MIN
4 PONDERACAO: 0,5
5 ENTRADA: 40
6 AVALICAO: MULT
7 ATUALIZACAO: TxTrafego
8 - ID: Memoria
9 OBJETIVO: MIN
10 PONDERACAO: 0,3
11 ENTRADA: 40
12 AVALICAO: MULT
13 ATUALIZACAO: TxTrafego
14 - ID: Tamanho
15 OBJETIVO: MAX
16 PONDERACAO: 0,2
17 ENTRADA: 0
18 AVALICAO: SOM
19 ATUALIZACAO: ESTATICO

```

Figura 14 – Bloco de Implantação do Exemplo Padrão (*Benchmark* - Início)

```

1 IMPLANTACAO:
2 DESCRICAO_EOS:
3 EO1:
4 BENCHMARK:
5 TxTrafego: 1,1
6 Memoria: 5
7 Tamanho: 1
8 EO2:
9 BENCHMARK:
10 TxTrafego: 1,0
11 Memoria: 5
12 Tamanho: 1
13 EO3:
14 BENCHMARK:
15 TxTrafego: 0,5
16 Memoria: 10
17 Tamanho: 1

```

Figura 15 – Bloco de Implantação do Exemplo Padrão (*Benchmark* - Continuação)

```

18 EO4:
19 BENCHMARK:
20 TxTrafego: 0,2
21 Memoria: 5
22 Tamanho: 1
23 EO5:
24 BENCHMARK:
25 TxTrafego: 0,5
26 Memoria: 10
27 Tamanho: 1
28 EO6:
29 BENCHMARK:
30 TxTrafego: 1,0
31 Memoria: 5
32 Tamanho: 1

```

Figura 16 – Bloco de Implantação do Exemplo Padrão (Ramificações)

```

1 IMPLANTACAO:
2 RAMIFICACOES:
3 TxTrafego:
4 DIVISAO: MULT
5 FATORES:
6 - [0,5, 0,5]
7 Memoria:
8 DIVISAO: MULT
9 FATORES:
10 - [0,5, 0,5]
11 Tamanho:
12 DIVISAO: MULT
13 FATORES:
14 - [0,5, 0,5]

```

A exemplificação dos procedimentos de composição ocorre a partir de uma topologia com seis elementos operacionais distintos (identificados como EO1, EO2, EO3, EO4, EO5 e

EO6), dois nós de borda de saída (NS1 e NS2), um segmento parcialmente ordenado (contendo EO1, EO2 e EO3) e uma ramificação terminal. Sobre essa topologia incide dois tipos de dependências: de ordem envolvendo EO2 e EO3 no segmento parcialmente ordenado; e de infraestrutura envolvendo as duas ocorrências de EO4 com o domínio DOM1. O bloco de serviço que apresenta tais especificações é ilustrado pela Figura 12. A função objetivo requisitada, apresentada na Figura 13, é composta por três métricas: taxa de tráfego denominada "TxTrafego", consumo de memória para processamento do tráfego denominada "Memoria" e tamanho da topologia denominada "Tamanho". As métricas TxTrafego e Memoria compartilham do mesmo objetivo de minimização de seus valores, enquanto a métrica Tamanho tem como objetivo a maximização de seus resultados. O bloco de implantação apresenta as informações utilizadas para a avaliação da topologia do serviço, este é representado pelas Figuras 14 e 15 (cortes contendo os dados de *benchmark*) e pela Figura 16 (corte contendo os dados de ramificações).

A solução descrita nesta seção foi desenvolvida como prova de conceito através da linguagem de programação Python 3. A implementação¹⁸ utiliza os validadores dos modelos de topologia SFC e de requisição apresentados no Capítulo 4, herdando todas as premissas, capacidades e restrições, assim como bibliotecas não nativas, definidas para os mesmos.

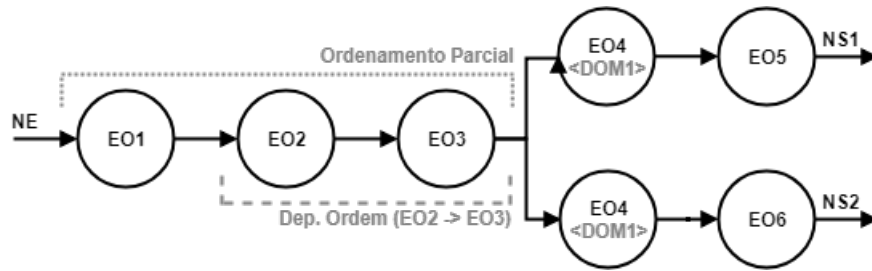
5.1.1 Expansão de Topologias

Um mesmo serviço de rede pode ser provido através de diferentes topologias SFC. Sendo assim, uma especificação topológica pode ser construída através da definição parcial da ordenação de seus elementos operacionais (criando estruturas conhecidas como segmentos parcialmente ordenados). Além disso, as ramificações existentes em uma topologia podem ser exploradas em vista das intersecções entre fragmentos comuns a todos os seus ramos, permitindo, por exemplo, a redução da quantidade de ocorrências de uma função de rede em comum dentro de um mesmo serviço. Sendo assim, o procedimento de expansão de topologias tem por objetivo analisar segmentos parcialmente ordenados, dependências de VNF, dependências de infraestrutura e ramificações objetivando criar um conjunto de topologias válidas com ordenamento absoluto (chamadas de composições candidatas).

O procedimento de expansão de topologias é dividido em dois passos: resolução de ordenamentos parciais e redução de ramificações. Esses passos recebem como argumentos os dados fornecidos pelo bloco de serviço presente em uma requisição, nele os atributos NO_ENTRADA,

¹⁸ Disponível em <https://github.com/ViniGarcia/NFV-FLERAS>

Figura 17 – Topologia Geral do Exemplo Padrão

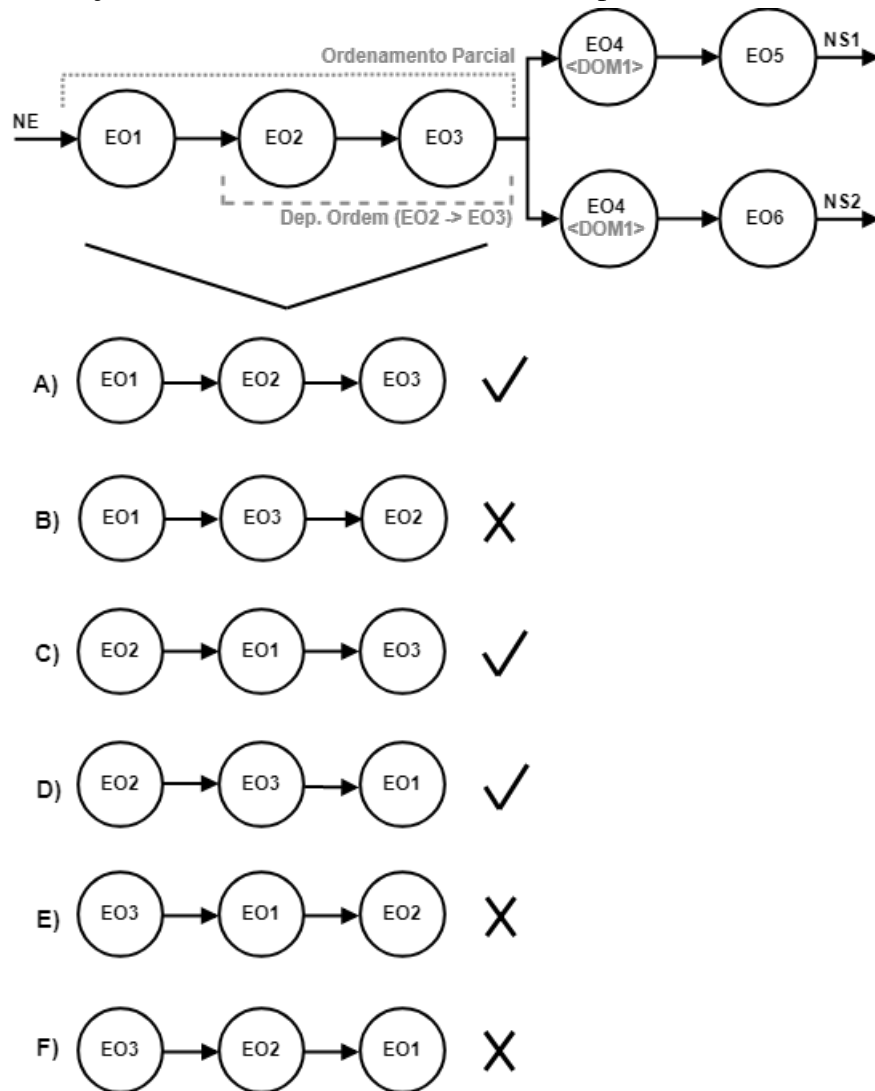


NOS_SAIDA e ELEM_OPERACIONAIS são empregados para distinguir a função dos diferentes identificadores presentes na topologia requisitada (atributo TOPOLOGIA), além de serem diretamente utilizados pela solução de validação da mesma. O atributo TOPOLOGIA, por sua vez, fornece as informações necessárias para a execução deste procedimento (*i.e.*, dependências, organização e ordenamentos topológicos) que, após processadas, resultam no conjunto de composições candidatas que implementam o serviço. A Figura 17 ilustra a topologia geral da SFC requisitada através do bloco de serviço indicado na Figura 12, este utilizado como o exemplo padrão no decorrer desta seção.

A resolução de ordenamentos parciais é processada analogamente a uma permuta com restrições onde as restrições são equivalentes as dependências de VNF apresentadas como de ordenamento ou acoplamento. Dependências de ordenamento implicam na redução do número de possíveis posições aplicáveis a um determinado elemento operacional, sendo o tamanho desta redução relativa a quantidade de posições que antecedem aquela ocupada pelo primeiro elemento participante da dependência. Já uma dependência de acoplamento implica na fixação do segundo elemento operacional nela presente na posição logo posterior ao primeiro já alocado, sendo estes compreendidos como um bloco indivisível e tratados como um único elemento operacional pela solução.

A solução de composição desenvolvida gera todas as permutas possíveis para um segmento parcialmente ordenado e verifica uma a uma a validade das mesmas (*i.e.*, força bruta). Dessa maneira, a quantidade total de ordenamentos absolutos analisados para um segmento parcialmente ordenado é determinado pelo fatorial do número de elementos operacionais presentes no mesmo, enquanto que a quantidade de combinações possíveis constatadas ao final deste passo corresponde ao produto entre o número de ordenamentos absolutos válidos gerados a partir de cada segmento processado. Isso posto, a permuta realizada na topologia exemplo é ilustrada pela Figura 18, nela um segmento parcialmente ordenado com três elementos operacionais (EO1, EO2 e EO3) resulta em seis segmentos com ordenamento absoluto, destes quais

Figura 18 – Resolução de Ordenamentos Parciais no Exemplo Padrão



apenas três são considerados válidos (A, C, D) por cumprirem a dependência de ordem definida (alocação de E02 antes de E03).

A redução de ramificações consiste na análise dos ramos presentes em uma topologia absolutamente ordenada a fim de detectar segmentos que podem ser alocados em uma seção comum da cadeia, antes do início ou após a conclusão de uma ramificação analisada. Ademais, segmentos equivalentes localizados em posições intermediárias dos ramos também podem ser unidos. Para isso, uma ramificação qualquer é decomposta em duas ramificações, uma de característica não terminal, estabelecida do ponto inicial da ramificação original até sua conclusão que ocorre antes do início do segmento unificado (sendo o primeiro elemento operacional deste segmento o ponto de intersecção dos ramos), e outra com a mesma característica da ramificação original, iniciada após o segmento unificado e perpetuada até a conclusão do restante da rami-

ficação original, ou seja, até os pontos de saída em caso de uma ramificação terminal ou até o elemento operacional de intersecção em caso de uma ramificação não terminal.

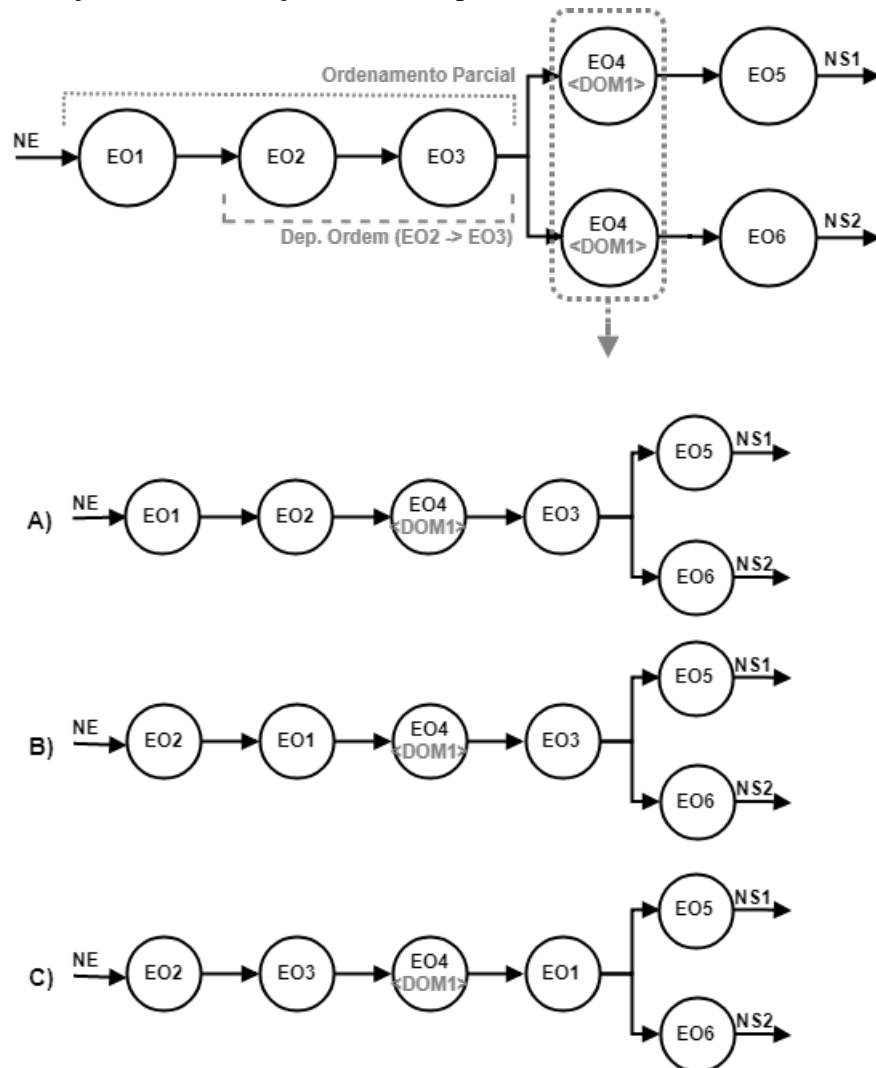
A solução desenvolvida, por simplificação de implementação, realiza somente a redução de ramificações em segmentos iniciais e finais dos ramos, desconsiderando o caso de segmentos intermediários. Para tanto, as composições candidatas produzidas durante o passo de resolução de ordenamentos parciais são processadas a fim de identificar possibilidades de redução em cada uma das ramificações especificadas. Uma ramificação apresenta segmentos que podem ser reduzidos quando os elementos operacionais destes são comuns a todos os ramos (tanto em função quanto em posição), além de, quando existirem, tais elementos apresentarem as mesmas dependências de infraestrutura. Dessa forma, a quantidade de novas composições candidatas geradas pela redução de uma ramificação por um segmento máximo (*i.e.*, maior segmento de elementos operacionais comuns entre os ramos) é igual a quantidade de elementos operacionais presentes no mesmo, já que seus múltiplos elementos pode ser decomposto em subsegmentos menores. Ainda, a quantidade total de composições candidatas resultantes do passo de redução de ramificações é igual ao produto entre o tamanho de todos os segmentos máximos detectados nas ramificações analisadas, uma vez que as reduções de diferentes ramificações podem ser aplicadas ao mesmo tempo.

Para o exemplo adotado, as três composições candidatas obtidas através da resolução de ordenamentos parciais possibilitam uma redução de ramificação. Essa redução é aplicada a única ramificação existente na topologia e compreende um segmento com apenas um elemento operacional (EO4) cujo cumpre todas as requisições do processo. Dessa forma, o elemento operacional é transferido para a posição logo anterior ao elemento que dá origem a ramificação, passando a atuar na seção principal da cadeia. Ao final deste processo, ilustrado pela Figura 19, o conjunto final de composições candidatas tem três possibilidades adicionadas (A, B, C), passando a contemplar um total de seis topologias válidas que serão analisadas durante o procedimento de avaliação de topologias, sendo este último realizado através de uma função objetivo que abrange todas as métricas requisitadas.

5.1.2 Construção da Função Objetivo

A função objetivo representa os dados a serem avaliados para cada uma das composições candidatas geradas durante a etapa de expansão de topologias. A solução desenvolvida adota uma função objetivo padrão retratada pela Equação 5.1, nela CC é uma composição can-

Figura 19 – Redução de Ramificações no Exemplo Padrão



didata presente no conjunto de composições retornadas pela expansão de cadeias (CET). Essa função objetivo sugere a utilização da topologia com maior (MAX) Índice de Adequabilidade Topológica (IAT). Este índice, por sua vez, reflete o grau de proximidade da avaliação de uma composição candidata ao resultado ótimo referente a todas as métricas informadas pelo operador de rede. O resultado ótimo é considerado como conjunto de resultados máximos constatados para as métricas com objetivo de maximização, e de resultados mínimos verificados para as métricas com objetivo de minimização, esse conjunto é determinado após a avaliação das métricas requisitadas em todas as composições candidatas.

$$MAX(IAT_{CC}^{CET}), \forall CC \in CET \quad (5.1)$$

Uma vez que o conjunto de métricas é volátil entre diferentes requisições, a construção da equação que calcula o IAT também ocorre de maneira dinâmica, sendo assim, uma sequên-

cia de três passos denominados como geração das funções parciais, normalização dos resultados parciais e construção da função IAT são executados. No decorrer desses passos, os dados provenientes dos objetos de métrica (informados no bloco de função objetivo de uma requisição) são utilizados para alimentar a criação e atualização da função geradora do IAT. Dessa forma, os atributos considerados durante a construção da função objetivo são: ENTRADA, variável utilizada como argumento de cálculo de uma métrica em determinado elemento operacional da topologia, que é atualizada a cada iteração durante a avaliação topológica (representada pelo símbolo E); AVALIACAO, constante que indica a operação matemática a ser aplicada entre a variável ENTRADA e o valor de *benchmark* assinalado para um elemento operacional avaliado sob determinada métrica; OBJETIVO, constante que indica a pretensão pela maximização (MAX) ou minimização (MIN) da métrica avaliada; e PONDERACAO, valor contido no intervalo [0;1] que descreve a importância relativa de uma métrica em relação as demais (representada pelo símbolo P). O conjunto de notações utilizadas durante esta seção são sumarizadas na Tabela 7.

Tabela 7 – Notações Utilizadas na Construção da Função Objetivo

Notação	Descrição
CET	Conjunto de composições candidatas (topologias absolutamente ordenadas) gerado durante a expansão de cadeias.
CC	Composição candidata composta por um conjunto de elementos operacionais e nós de borda absolutamente ordenados.
E_{mtc}	Valor de entrada referente a métrica "mtc".
P_{mtc}	Peso referente a métrica "mtc".
EO_{mtc}	Valor da métrica "mtc" informado para o elemento operacional "EO".
AVA_{mtc}^{CC}	Resultado da avaliação da função parcial referente a métrica "mtc" para a composição candidata "CC".
AVA_{mtc}^{CET}	Conjunto de resultado de avaliação da função parcial referente a métrica "mtc" para todas as composições candidatas existentes em "CET".
DA_{mtc}^{CET}	Distância absoluta máxima entre os resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
$DA_{mtc}^{CC CET}$	Distância absoluta máxima entre o resultado da avaliação da composição candidata "CC" e o conjunto de resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
$NORM_{mtc}^{CC CET}$	Normalização do resultado da avaliação da composição candidata "CC" em relação ao conjunto de resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
$COMP_{mtc}^{CC CET}$	Complemento da normalização da composição candidata "CC" em relação ao conjunto de resultados de avaliação das composições candidatas existentes em "CET" pela métrica "mtc".
IAT_{CC}^{CET}	Índice de adequabilidade topológica da composição candidata "CC" em relação ao conjunto de composições candidatas "CET".

O primeiro passo a ser executado para a construção da função objetivo é chamado de geração das funções parciais. Uma função parcial é utilizada para avaliar as composições candidatas em relação a uma única métrica requisitada. Logo, cada uma das métricas presentes no bloco de função objetivo do SFCR é transcrita para uma função parcial a ser calculada durante as iterações de avaliação das topologias do serviço. O cálculo das funções parciais retornam valores agregados brutos que consistem do somatório dos resultados da avaliação de cada um dos elementos operacionais, estes devem estar disponíveis para todas as composições candidatas do conjunto CET antes da execução próximo passo (*i.e.*, normalização dos resultados parciais). As Equações 5.2, 5.3 e 5.4 indicam, respectivamente, as funções parciais para as métricas TxTráfego (tt), Memória (mem) e Tamanho (tam), apontadas no bloco de função objetivo exemplo, aplicadas sobre os elementos operacionais (EO) de uma composição candidata (CC).

$$AVA_{tt}^{CC} = \sum_{EO}^{CC} E_{tt} * EO_{tt} \quad (5.2)$$

$$AVA_{mem}^{CC} = \sum_{EO}^{CC} E_{mem} * EO_{mem} \quad (5.3)$$

$$AVA_{tam}^{CC} = \sum_{EO}^{CC} E_{tam} + EO_{tam} \quad (5.4)$$

Entretanto, a grandeza numérica relacionada a uma determinada métrica pode ser completamente diferente das demais, impossibilitando a utilização direta dos resultados de avaliação (AVA) para a geração do índice de adequabilidade topológica. Por exemplo, considerando a função objetivo requisitada na Figura 13, a métrica de tamanho da cadeia de serviço tem seus resultados em um intervalo de valores muito inferior quando comparados aos resultados obtidos para avaliações da taxa de tráfego e consumo de memória. Sendo assim, se esses resultados são naturalmente agrupados para a geração do IAT de uma composição candidata, o impacto avaliativo do tamanho da cadeia é relativamente inferior ao das demais métricas. Para contornar isso, o processo de normalização de resultados parciais é aplicado a cada uma das métricas iteradas sobre as composições candidatas. Esse processo é executado sobre todos resultados provenientes das avaliações realizadas pelas funções parciais e objetiva mapear seus valores em um intervalo contínuo [0;1].

A normalização das avaliações obtidas de uma métrica consiste em determinar a representatividade de um resultado qualquer em relação ao intervalo contínuo mínimo que compreende todos os resultados obtidos através da função parcial desta métrica. De maneira prática,

esse intervalo é compreendido pelo conjunto de valores reais contidos entre o menor e o maior resultado verificado durante a avaliação de uma função parcial para um conjunto de composições candidatas, estes resultados são chamados, respectivamente, de Borda Inferior (BI) e Borda Superior (BS). Dessa forma, a normalização tem por objetivo mapear o intervalo [BI;BS] para o intervalo [0;1], sendo BI equivalente a 0, BS equivalente a 1 e resultados intermediárias equivalentes a valores reais entre 0 e 1 representantes da sua proximidade à borda superior. Para fazer isso, o intervalo de dados [BI;BS] é deslocado para que o ponto de origem deste convirja com o ponto de origem do intervalo objetivo (*i.e.*, 0). Esse processo é realizado pela determinação da distância absoluta máxima entre os resultados de avaliação de uma métrica, obtidos através de uma função parcial, para um determinado conjunto de composições candidatas (DA_{mtc}^{CET} , definida na Equação 5.5), indicando assim a maior diferença absoluta possível entre dois resultados quaisquer provenientes de uma mesma métrica avaliada sobre um mesmo conjunto. Nessas circunstâncias, todas as demais diferenças absolutas calculadas devem necessariamente estar contidas no intervalo $[0;DA_{mtc}^{CET}]$.

$$DA_{mtc}^{CET} = \max(AVA_{mtc}^{CET}) - \min(AVA_{mtc}^{CET}) \quad (5.5)$$

Assim, os resultados de avaliação de composições candidatas são mapeados através da Equação 5.6 para o intervalo padrão de distância absoluta máxima (*i.e.*, $[0;DA_{mtc}^{CET}]$). Esse mapeamento é realizado com o objetivo de determinar a proximidade de um dado resultado avaliativo à borda superior do conjunto do qual ele faz parte, ou seja, quanto maior o valor de $DA_{mtc}^{CC\ CET}$ mais próximo está o resultado de "CC" do maior resultado obtido durante a avaliação do conjunto de composições candidatas "CET" pela função parcial representante da métrica "mtc". Logo, o processo de normalização é executado de forma a determinar a razão entre a distância absoluta de uma composição candidata e a distância absoluta máxima, como demonstra a Equação 5.7. O resultado dessa normalização é um valor existente no intervalo contínuo [0;1] que mantém a característica da proximidade de um resultado à borda superior, porém este representado através de um intervalo de valores comuns as demais métricas quando normalizadas. A cargo de exceção, em cenários onde todas as composições candidatas avaliadas apresentam o mesmo valor para uma mesma métrica ($DA_{mtc}^{CET} = 0$), a Equação 5.8 é utilizada visto que todos os resultados são considerados como as próprias bordas superiores.

$$DA_{mtc}^{CC\ CET} = AVA_{mtc}^{CC} - \min(AVA_{mtc}^{CET}) \quad (5.6)$$

$$NORM_{mtc}^{CC\ CET} = \frac{DA_{mtc}^{CC\ CET}}{DA_{mtc}^{CET}}, DA_{mtc}^{CET} \neq 0 \quad (5.7)$$

$$NORM_{mtc}^{CC\ CET} = 1, DA_{mtc}^{CET} = 0 \quad (5.8)$$

Após a normalização dos resultados provenientes das funções parciais, a função de cálculo do IAT pode ser construída. O IAT é representado por um valor contido no intervalo [0;1] que indica a proporcionalidade da avaliação geral de uma composição candidata ao resultado ótimo considerando o conjunto de composições ao qual ela pertence e de métricas pelas quais ela foi avaliada. Isto posto, uma composição adquire o maior IAT possível (*i.e.*, 1) quando esta apresenta os resultados de borda inferior para todas as métricas com objetivo de minimização e resultados de borda superior para todas as métricas cujo objetivo é sua maximização. Porém, como o processo de normalização indica a proximidade de um dado resultado à borda superior, este pode ser naturalmente utilizado apenas para métricas cujo objetivo é sua maximização. Por outro lado, métricas com o objetivo de minimização devem ter seus resultados de normalização conciliados para que passem a representar a sua proximidade à borda inferior, para isto o complementar do valor normalizado (Equação 5.9) é empregue de forma que a borda inferior normalizada para o valor 0 é complementada para o valor 1 e, analogamente, a borda superior normalizada para o valor 1, terá complementação igual a 0.

$$COMP_{mtc}^{CC\ CET} = 1 - NORM_{mtc}^{CC\ CET}, \quad (5.9)$$

Finalmente, cada métrica é ponderada de acordo com seu peso. Os pesos consistem de valores reais entre ou igual 0 e 1 que determinam o grau de relevância de uma métrica em relação as demais. Além disso, como previsto no modelo de requisição apresentado na Seção 4, o somatório destes pesos devem resultar em 1, criando assim um ponto de referência padrão que indica a máxima relevância possível de uma métrica. Desse modo, o somatório dos resultados normalizados (e complementados quando necessário) das funções parciais ponderados pelos pesos das suas respectivas métricas determinam o IAT de uma composição candidata. Considerando o exemplo padrão adotado, o cálculo do IAT de uma composição cujo as métricas são taxa de tráfego (tt), memória (mem) e tamanho (tam) é demonstrado pela Equação 5.10, nela a notação P simboliza o peso especificado para uma determinada métrica.

$$IAT_{CET}^{CC} = P_{tt} * COMP_{tt}^{CC\ CET} + P_{mem} * COMP_{mem}^{CC\ CET} + P_{tam} * NORM_{tam}^{CC\ CET} \quad (5.10)$$

Após construída, a função de cálculo do IAT é mapeada para a função objetivo padrão adotada por esta solução de composição (Equação 5.1). É importante ressaltar que, apesar da função objetivo ser responsável por apontar a composição candidata mais adequada aos objetivos do procedimento de avaliação de topologias, apenas ela não é suficiente para a execução completa do mesmo. Em outras palavras, a função objetivo deve ser provida em conjunto com as funções parciais, ambas definidas durante o procedimento de construção de função objetivo. As funções parciais, por sua vez, são aplicadas para caracterizar individualmente elementos operacionais e composições candidatas em relação as métricas requeridas, sendo estes dados então utilizados como parâmetros para o posterior cômputo e avaliação do IAT.

5.1.3 Avaliação de Topologias

A avaliação de topologias consiste na análise do conjunto de composições candidatas em vista das métricas informadas na requisição do serviço. Este procedimento é o último a ser executado pela solução de composição, isto acontece já que os resultados dos demais procedimentos são utilizados como entradas para a avaliação topológica. Essa avaliação, por sua vez, ocorre sobre os elementos operacionais de uma composição candidata e utiliza os dados de *benchmark* e ramificações (providos através do bloco de implantação presente no SFCR) para calcular os resultados individuais dos mesmos. Além desses dados, a avaliação de topologias usa informações presentes no bloco de função objetivo do SFCR durante seu processamento. Essas últimas dizem respeito aos atributos ENTRADA e ATUALIZACAO indicados para cada uma das métricas requisitadas, o primeiro aponta o valor de entrada inicial a ser considerado para o cálculo das funções parciais, já o segundo representa o método de atualização a ser aplicado ao valor de entrada após cada iteração da solução sobre um elemento operacional, a atualização pode ser executada de três formas: como a persistência do valor de entrada inicial (ESTATICO); como o resultado de avaliação da métrica corrente; ou como o resultado da aplicação de uma função de avaliação de qualquer outra métrica também definida na requisição para o valor de entrada da métrica a ser atualizada.

Apesar da função objetivo indicar o propósito geral a ser considerado durante o procedimento de avaliação de topologias, esta não se destina a avaliar os elementos operacionais das composições candidatas para caracterizar as mesmas em relação as métricas requisitadas. Dessa forma, o processo de avaliação ocorre em três passos distintos: avaliação de elementos operacionais, avaliação de composições candidatas e aconselhamento. O primeiro passo é realizado

sobre os elementos operacionais das composições candidatas, este tem por objetivo determinar e agregar resultados de todas as funções parciais para elas. Já a avaliação de composições candidatas utiliza o conjunto de resultados agregados obtidos durante a execução do primeiro passo como parâmetros de execução dos processos de normalização e ponderação (além da complementação, quando necessária), gerando o IAT para cada uma das topologias analisadas. Finalmente, o aconselhamento aplica a função objetivo padrão aos resultados provenientes do segundo passo, estabelecendo assim qual das composições candidatas é capaz de maximizar relação de custo/benefício ponderada entre as métricas avaliadas.

O primeiro passo (*i.e.*, avaliação de elementos operacionais) itera sobre todas as posições onde elementos operacionais estão presentes nas composições candidatas. Para isso, a solução desenvolvida realiza a avaliação desses elementos linearmente de acordo com a especificação em SCAG das topologias geradas, efetuando um processo semelhante a uma busca em profundidade na abstração do grafo de serviço. Uma exceção no curso normal de execução ocorre quando ramificações não terminais começam a ser avaliadas, sendo estas completamente processadas antes do retorno aos seus segmentos de origem. Os elementos operacionais são avaliados através das funções parciais determinadas durante o procedimento de construção da função objetivo, originando a agregação métrica a métrica de resultados que compõem os resultados parciais a serem computados durante o segundo passo deste procedimento. Após a conclusão da avaliação de um elemento operacional em específico, o processo de atualização dos valores de entrada é executado para cada métrica avaliada durante a iteração recém concluída.

Durante a atualização, o valor de entrada utilizado pode tanto ser mantido igual ou ser substituído pelo resultado proveniente da utilização do mesmo como argumento de alguma das funções de avaliação disponíveis, sendo esta função representada pelo identificador de sua métrica correspondente nos campos ATUALIZACAO do bloco de função objetivo presente no SFCR. O valor de entrada atualizado pode ser tanto repassado de maneira integral a próxima iteração, em caso de ausência de ramificações, ou de maneira dividida, caso exista uma ramificação. A divisão fragmenta o valor de entrada entre os diversos ramos de uma ramificação, neste caso, as operações previstas nos objetos de ramificação presentes no bloco de implantação são aplicadas. Esses objetos refletem cada uma das métricas avaliadas e contêm uma operação (DIVISAO) e uma lista de fatores (FATORES) de divisão, sendo suas sub-listas representantes de uma ramificação (em ordem de ocorrência na especificação) e cada valor inserido nestas de

um fator de divisão para um de seus ramos (também em ordem de ocorrência). Por fim, da mesma forma que os valores de entrada são dividido na eminência de uma ramificação, em ramificações não terminais estes são agregados antes da avaliação do elemento operacional que intersecta os seus ramos.

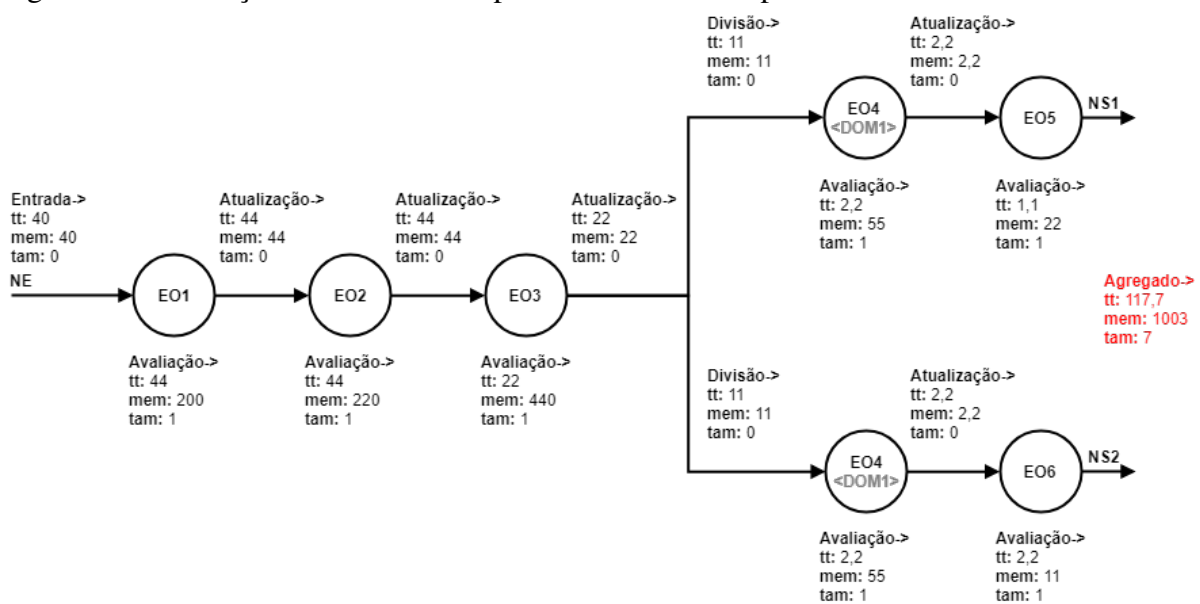
$$E_{tt}^{RAM_n SEG_m} = E_{tt} * FD_{tt}^{RAM_n SEG_m} \quad (5.11)$$

$$E_{mem}^{RAM_n SEG_m} = E_{mem} * FD_{mem}^{RAM_n SEG_m} \quad (5.12)$$

$$E_{tam}^{RAM_n SEG_m} = E_{tam} * FD_{tam}^{RAM_n SEG_m} \quad (5.13)$$

O exemplo padrão adotado apresenta três funções de divisão, estas indicadas pelas Equações 5.11 (TxTrafego), 5.12 (Memoria) e 5.13 (Tamanho), nelas "E" simboliza o valor de entrada, "RAM" uma ramificação de uma composição candidata, "SEG" um segmento de uma ramificação e "FD" o fator de divisão a ser praticado. Todo o arcabouço de funções e dados descritos para o primeiro passo do procedimento de avaliação de topologias é aplicado para as seis composições candidatas determinadas no exemplo padrão, sendo os processos de avaliação de elemento operacional e atualização de valores de entrada para uma delas exibido através da Figura 20 e os valores agregados obtidos para as demais expostos na Tabela 8.

Figura 20 – Avaliação de Elementos Operacionais no Exemplo Padrão



O segundo passo do procedimento de avaliação topológica (*i.e.*, avaliação de composições candidatas) utiliza os resultados agregados provenientes do primeiro passo para realizar a

normalização, complementação e ponderação destes a fim de determinar – através da função IAT gerada durante o procedimento de construção da função objetivo – o índice de adequabilidade topológica das composições candidatas. Com isso, o terceiro passo pode em fim ser executado, nele as composições candidatas são organizadas em ordem decrescente de IAT de modo que aquela que ocupa a primeira posição da lista corresponde a topologia aconselhada considerando a função objetivo geral adotada pela solução (*i.e.*, maximização do IAT). É importante ressaltar que, apesar da composição candidata com maior IAT ser considerada como o principal resultado da solução de composição desenvolvida, as demais topologias e seus respectivos índices não são descartados, isto ocorre já que o conjunto de composições retornadas pode ser integralmente utilizado como entrada para outras soluções de implantação de SFCs, sendo assim possível determinar a topologia que gera a melhor relação custo benefício entre diferentes resultados. A Tabela 8 apresenta o processo de cálculo do IAT das composições candidatas e o ranqueamento de acordo com a função objetivo realizados durante, respectivamente, o segundo e o terceiro passo do presente procedimento.

Tabela 8 – Resultados Gerais de Avaliação do Exemplo Padrão

Composição Candidata	Agregados	Normalização	Complementação	Ponderação	IAT	Função Objetivo
NE EO2 EO3 EO4 <DOM1 > EO1 { EO5 NS1 / EO6 NS2 }	tt: 71,7 mem: 753 tam: 6	tt: 0 mem: 0,046 tam: 0	tt: 1 mem: 0,954 tam: -	tt: 0,5 mem: 0,2862 tam: 0	0,7862	1°
NE EO2 EO3 EO1 { EO4 <DOM1 > EO5 NS1 / EO4 <DOM1 >EO6 NS2 }	tt: 89,7 mem: 843 tam: 7	tt: 0,391 mem: 0,389 tam: 1	tt: 0,609 mem: 0,611 tam: -	tt: 0,3045 mem: 0,1833 tam: 0,2	0,6978	2°
NE EO2 EO1 EO4 <DOM1 > EO3 { EO5 NS1 / EO6 NS2 }	tt: 100,5 mem: 741 tam: 6	tt: 0,626 mem: 0 tam: 0	tt: 0,374 mem: 1 tam: -	tt: 0,187 mem: 0,3 tam: 0	0,487	3°
NE EO1 EO2 EO4 <DOM1 > EO3 { EO5 NS1 / EO6 NS2 }	tt: 104,5 mem: 761 tam: 6	tt: 0,713 mem: 0,076 tam: 0	tt: 0,287 mem: 0,924 tam: -	tt: 0,1435 mem: 0,2772 tam: 0	0,4207	4°
NE EO2 EO1 EO3 { EO4 <DOM1 > EO5 NS1 / EO4 <DOM1 >EO6 NS2 }	tt: 113,7 mem: 983 tam: 7	tt: 0,913 mem: 0,924 tam: 1	tt: 0,087 mem: 0,076 tam: -	tt: 0,0435 mem: 0,0228 tam: 0,2	0,2663	5°
NE EO1 EO2 EO3 { EO4 <DOM1 > EO5 NS1 / EO4 <DOM1 >EO6 NS2 }	tt: 117,7 mem: 1003 tam: 7	tt: 1 mem: 1 tam: 1	tt: 0 mem: 0 tam: -	tt: 0 mem: 0 tam: 0,2	0,2	6°

Finalmente, uma vez que a solução de composição desenvolvida é hierarquizada em uma série de procedimentos e passos que, por mais que sejam interconectados, apresentam uma estrutura de execução independente e podem ser alterados ou substituídos desde que mantenham o modelo padrão de geração de resultados. Em outras palavras, apesar de haver uma dependência relacionada a passagem dos resultados de um procedimento/passo como argumento para aquele que se segue, não há influência entre eles quanto ao algoritmo aplicado para a geração destes

resultados. Dessa forma, heurísticas podem ser utilizadas no procedimento de expansão topológica sem que modificações sejam necessárias nos demais procedimentos, da mesma forma que as funções parciais e a função objetivo podem ser alteradas sem que haja a necessidade de mudanças no procedimento de avaliação de composições candidatas.

5.2 AVALIAÇÃO DA SOLUÇÃO

Nesta seção um cenário de testes é estudado a fim de avaliar a solução de composição desenvolvida, para isto são utilizados resultados empíricos obtidos para uma cadeia de função de serviço real. Primeiramente, a Subseção 5.2.1 descreve o cenário através das características do serviço a ser implantado, dos resultados de *benchmark* obtidos a partir de seus elementos operacionais, da requisição a ser processada pela solução e dos equipamentos e programas empregues durante a avaliação. Em seguida, na Subseção 5.2.2, as composições candidatas resultantes da execução da solução de composição são instanciadas e testadas em condições isonômicas de infraestrutura e tráfego. Os dados quantitativos decorrentes desse processo são finalmente expostos e discutidos. De maneira geral, os experimentos têm por objetivo atestar a eficiência da solução proposta para a tomada de decisão por uma organização topológica que melhor concilie e otimize métricas de interesse, sendo estas definidas de maneira personalizada pelo operador de rede.

5.2.1 Ferramental e Cenário de Teste

A experimentação realizada consiste dos processos de composição e instanciação de uma cadeia de função de serviço que provê segurança, controle de conteúdo e balanceamento de carga para servidores HTTP e HTTPS. Esse serviço, por sua vez, abrange sete funções de rede¹⁹ denominadas filtro de protocolo/porta (FPP), classificador de tráfego (CT), DPI de assinaturas (DPIA), DPI de conteúdo (DPIC), filtro de marcação (FM), IPS HTTP/S (IPSH) e balanceador de carga (BC) das quais as características operacionais são sumarizadas na Tabela 9. De maneira geral, o serviço virtualizado de teste intermedeia as conexões entre os clientes e os servidores HTTP e HTTPS disponíveis, descartando quaisquer fluxos que apresentem características diferentes das previstas para tais aplicações (*i.e.*, TCP/IP, porta 80 para HTTP e porta 443 para HTTPS). Os fluxos que condizem com os protocolos e demais características

¹⁹ Disponível em <https://github.com/ViniGarcia/ViNeFuR>

suportadas são então processados por funções de controle de conteúdo e segurança, descartando o tráfego que contenha requisições consideradas ilegais ou que seja visto como malicioso para o sistema. Finalmente, um balanceador de carga equilibra a quantidade de conexões firmadas entre diferentes instâncias virtuais e/ou equipamentos físicos que atuam como os servidores *Web*.

A topologia do serviço de teste é especificada através do modelo SCAG e contempla sete elementos operacionais distintos identificados como EO1, EO2, EO3, EO4, EO5, EO6 e EO7, sendo a função de rede virtualizada alocada a eles respectivamente indicada por FPP, CT, DPIA, DPIC, FM, IPSH e BC. Alguns desses elementos são definidos em dois segmentos parcialmente ordenados (um contendo EO3, EO5 e EO6 e outro contendo EO4, EO5 e EO6), cada um alocado em um ramo da única ramificação presente na cadeia. Essa ramificação, por sua vez, é do tipo terminal, iniciada em EO2 e concluída pela intersecção de seus ramos em EO7. Ademais, sobre ambos os segmentos parcialmente ordenados incide uma dependência de VNF que indica o acoplamento entre EO3 e EO5 para o primeiro, e entre EO4 e EO5 para o segundo, garantindo assim que a verificação de eventuais marcações de pacotes sejam realizadas após o processamento dos mesmos pelos elementos marcadores. Finalmente, a topologia descrita tem seu escopo operacional delimitado por dois nós de borda, um de entrada (NE) e um de saída (NS) conectados, nesta ordem, ao EO1 e EO7. A Figura 21 expõe a cadeia de caracteres construída a partir do modelo SCAG, além da ilustração da topologia do serviço de teste especificada por ela.

Figura 21 – Topologia do Serviço de Teste

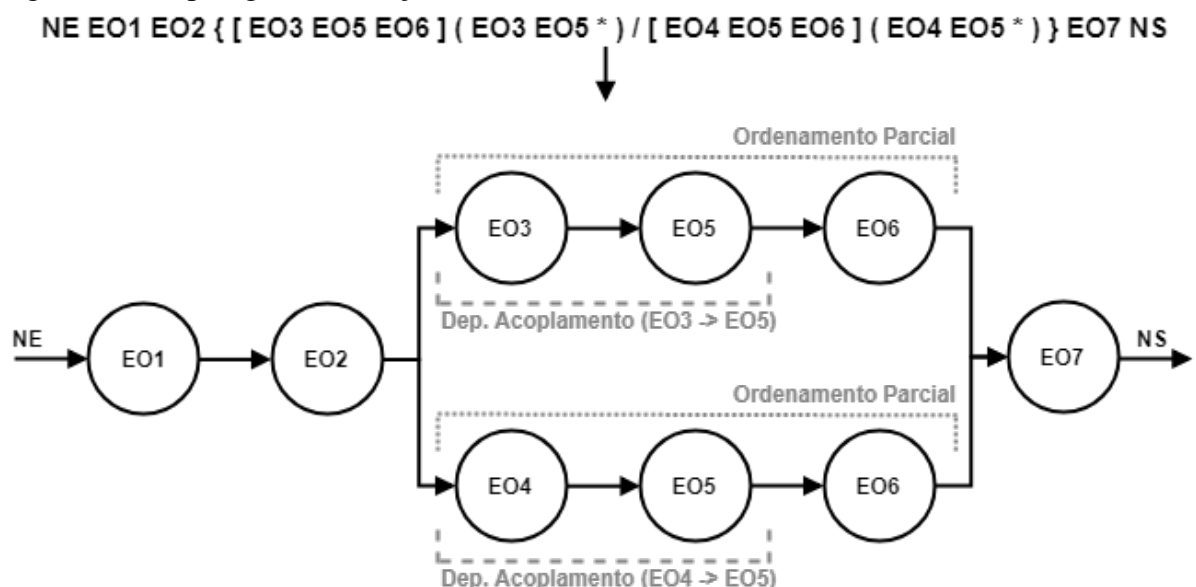


Tabela 9 – Descrição das Funções de Rede do Serviço de Teste

Função de Rede	Descrição	Implementação
Filtro de Protocolo/Porta	Encaminha pacotes ARP e TCP/IP com destino a porta 80 ou 443 para a interface de saída, descartando todos os demais. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Click Modular Router
Classificador de Tráfego	Encaminha pacotes HTTP e ARP para uma determinada interface de saída e pacotes HTTPS para uma segunda interface de saída. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Click Modular Router
DPI de Assinatura	Verifica pacotes em busca de sequências de <i>bytes</i> em posições pré-determinadas de seus conteúdos e os marca em caso de correspondência. Todos os pacotes são encaminhados para a interface de saída. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Python 3
DPI de Conteúdo	Verifica pacotes em busca de conteúdos proibidos (palavras-chave) e os marca em caso de correspondência. Todos os pacotes são encaminhados para a interface de saída. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Python 3
Filtro de Marcação	Encaminha pacotes não marcados pelo DPI de Assinatura para a interface de saída, descartando todos os demais. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Python 3
IPS HTTP/S	Verifica o tráfego em busca de anomalias no tamanho dos pacotes de requisições HTTP e HTTPS, bloqueando aqueles considerados maliciosos. O tráfego considerado benigno é encaminhado a interface de saída, já o tráfego considerado malicioso é descartado. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Python 3
Balancedor de Carga	Encaminha pacotes provenientes de uma mesma origem (MAC) para uma das interfaces de saída disponíveis. Encaminha incondicionalmente o tráfego de retorno para a interface de entrada.	Python 3

As funções de rede foram analisadas individualmente no cenário de *benchmarking* ilustrado pela Figura 22, esta análise determinou valores para as métricas a serem posteriormente avaliadas pela solução de composição. Três máquinas virtuais foram utilizadas para viabilizar o *benchmarking* das funções de rede: um servidor web HTTP na porta 80 e HTTPS na porta 443 criado a partir da solução webfs²⁰; um cliente que realiza requisições HTTP e HTTPS; e uma função de rede virtualizada que intermedeia a conexão cliente/servidor. Essas máquinas virtuais foram instanciadas em uma máquina física com 8GB de memória RAM DDR3, processador Core I5 3330, sistema operacional Debian 8²¹ e virtualizador KVM²². Cada instância contém 512MB de memória RAM, um núcleo virtual de processamento e executa o sistema operacional Debian 8, sendo estas conectadas por pontes criadas através do pacote bridge-utils²³. Ademais, as máquinas de cliente e servidor pertencem a uma mesma rede /24, dispensando a atividade de roteamento durante a execução dos testes.

O processo de *benchmarking* consiste de um teste de estresse dos servidores HTTP e HTTPS executado pelo programa http-perf²⁴ na máquina virtual representante do cliente. O

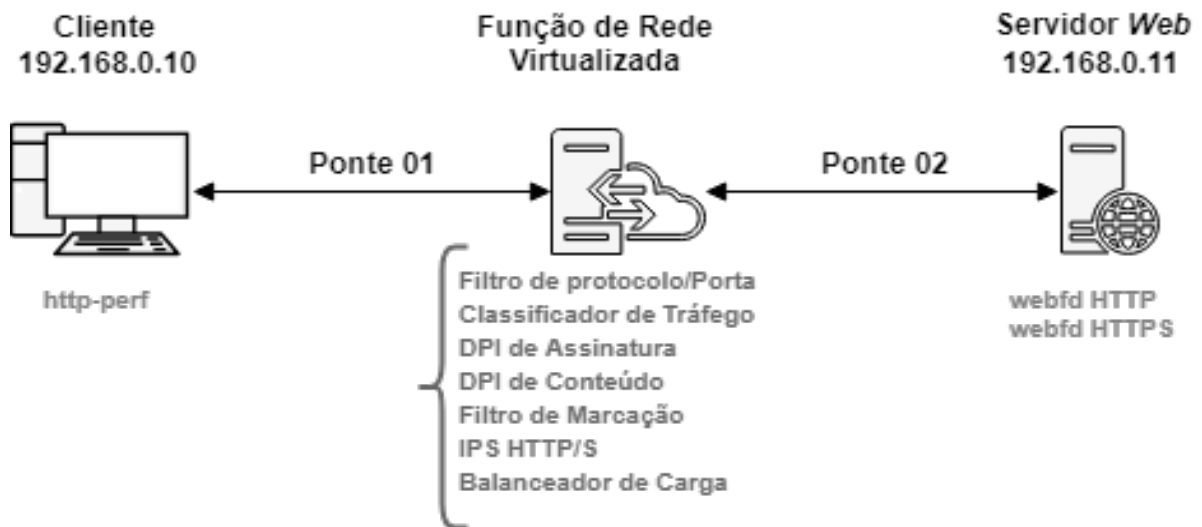
²⁰ <http://linux.bytesex.org/misc/webfs.html>

²¹ <https://www.debian.org/releases/jessie/>

²² https://www.linux-kvm.org/page/Main_Page

²³ <https://packages.debian.org/pt-br/jessie/bridge-utils>

²⁴ <https://www.npmjs.com/package/http-perf>

Figura 22 – Cenário de *Benchmarking*

programa foi configurado para recuperar a taxa de requisições atendidas (*i.e.*, requisições por segundo) entre cliente e servidor, para isso 500 requisições foram disparadas com um índice de concorrência igual a 5 (*i.e.*, cinco requisições simultâneas), tendo o tempo de duração de suas conexões mensurados individualmente. As funções de rede do serviço de teste foram instanciadas na máquina virtual mediadora das conexões e submetidas ao processo de *benchmarking* uma por vez, as requisições HTTP e HTTPS foram construídas de modo a executar a função tanto de maneira minimalista (*i.e.*, menor número possível de verificações e ausência de modificações nos pacotes recebidos), quanto de maneira completa, disparando todas as operações nelas previstas (*e.g.*, verificações e alterações de conteúdo, recálculos e alterações de cabeçalhos). Da mesma forma, funções de rede que, segundo a topologia do serviço, processam tráfego HTTP e HTTPS foram submetidas ao *benchmarking* para ambos os protocolos. Quando mais de uma execução do http-perf foi realizada para uma mesma função de rede, o resultado final adotado consiste da média aritmética entre as diferentes medidas de taxa de requisição recuperadas. A Tabela 10 sumariza os cenários e os protocolos examinados para as funções de rede do serviço de teste, além de apresentar os resultados da métrica objetivada.

Através dos resultados do *benchmarking* é possível analisar as diferenças comportamentais das funções de rede em decorrência da diversidade de características operacionais e de desenvolvimento verificadas entre as mesmas, como a heterogeneidade de linguagens de programação e de complexidade das rotinas de processamento de pacotes. A exemplo disso estão as funções FPP e CP que, por serem implementadas em *Click Modular Router* (linguagem orientada a aplicação, baseada em C, compilada e executada em nível de usuário), apresentam

Tabela 10 – *Benchmarking* HTTP e HTTPS das Funções de Rede do Serviço de Teste

Função de Rede	Cenários	Aplicação	Taxa de Requisições Atendidas (req/s)	Média Aritmética
Filtro de Protocolo/Porta	Único	HTTP	1937,9844	1023,3991
		HTTPS	108,8139	
Classificador de Tráfego	Único	HTTP	1968,5039	1038,8371
		HTTPS	109,1703	
DPI de Assinatura	Análise positiva	HTTPS	35,8217	44,4051
	Análise negativa	HTTPS	52,9885	
DPI de Conteúdo	Análise positiva	HTTP	81,6859	95,9090
	Análise negativa	HTTP	110,1321	
Filtro de Marcação	Único	HTTP	114,3118	78,9368
		HTTPS	58,4112	
IPS HTTP/S	Único	HTTP	112,0824	84,4564
		HTTPS	54,8305	
Balanceador de Carga	Único	HTTP	116,2520	87,7749
		HTTPS	59,2979	

alta vazão de requisições quando comparadas a funções que executam operações semelhantes (*e.g.*, FM e BC), porém são implementadas em Python 3 (linguagem genérica, com alto nível de abstração e interpretada). Ademais, funções de rede responsáveis por tarefas mais complexas, como análise e alteração de dados (*e.g.*, DPIA, DPIC), manifestaram uma taxa de requisições atendidas relativamente inferior quando comparadas aos resultados do restante das funções executadas nos mesmos cenários e/ou com os mesmos protocolos. Finalmente, o protocolo de aplicação utilizado também pode interferir nas medidas obtidas para a taxa de tráfego, isto ocorre principalmente devido ao processo de criptografia e descriptografia realizado tanto pelo cliente quanto pelo servidor em conexões HTTPS, além do tráfego de dados extra decorrente da inserção de uma camada de apresentação nos pacotes e da troca de certificado. Apesar dessa interferência não ser originada pela execução de uma função de rede em si, esta pode prejudicar a qualidade do serviço prestado pelo ponto de vista de seus usuários, sendo assim também considerada como objeto de análise durante o processo de *benchmarking*.

As funções de rede também são categorizadas a partir de sua capacidade ou não de descartar tráfego, originando a segunda métrica a ser considerada durante a etapa de composição do serviço de teste. Essa métrica, chamada de taxa relativa de tráfego, representa uma expectativa de atuação das funções de rede no descarte de pacotes em um cenário genérico, podendo ou não se concretizar de acordo com as características do tráfego submetido ao serviço em um dado momento de sua execução. Por exemplo, o filtro de protocolo/porta pode descartar completamente uma inundação ICMP, porém, em uma outra situação, pode não realizar nenhum

descarte uma vez que todo o tráfego de entrada apresenta os atributos exigidos (*i.e.*, ARP ou TCP/IP 80/443) para ser encaminhado a próxima função. Dessa forma, quatro categorias são definidas generalizando a capacidade de descarte das funções de rede: pleno, alto, moderado e inexistente.

No caso do serviço de teste utilizado, nenhuma função apresenta a característica de descarte pleno (taxa de tráfego relativa igual a 0), por outro lado a função FPP é categorizada como de descarte alto (taxa de tráfego relativa igual a 0,33), isto ocorre pois esta função é alocada de maneira absoluta na primeira posição da cadeia de serviço, o que implica no processamento indiscriminado de todo o tráfego de entrada, além disso, a filtragem aplicada nas camadas de rede 3 e 4 é bastante restritiva, permitindo apenas o encaminhamento de pacotes TCP/IP (porta 80 ou 443) ou ARP. As funções de rede FM e IPSH foram categorizadas como de descarte moderado (taxa de tráfego relativa igual a 0,66) já que atuam em pacotes filtrados por, pelo menos, uma função anterior e realizam descartes somente na ocorrência de indícios de ataques ou de características específicas verificadas nos dados de camada 5 ou superior. Por fim, as demais funções de rede (*i.e.*, CT, DPIA, DPIC e BC) são categorizadas como de descarte inexistente (taxa de tráfego relativa igual a 1) pois processam e encaminham incondicionalmente qualquer pacote submetido a elas.

Por último, as composições candidatas são avaliadas pelo seu tamanho (*i.e.*, quantidade de elementos operacionais empregues pela mesma). Apesar da especificação topológica provida na requisição do serviço ser apresentada através de um número fixo de elementos operacionais, o procedimento de expansão de topologias é capaz de modificar o mesmo durante a execução do passo de redução de ramificações. Tipicamente, um maior número de elementos operacionais resulta em um aumento na quantidade mínima de recursos computacionais necessários para a implantação do serviço. Entretanto, a redução na quantidade desses elementos não resulta necessariamente em uma economia direta de recursos durante a instanciação e execução dos mesmos, isto ocorre pois elementos operacionais em ramos unificados tendem a processar individualmente uma maior quantidade de tráfego se comparados a aqueles que atuam na ramificação não modificada, sujeitando os primeiros a requisitarem por processos de escala horizontal ou vertical mais frequentemente. Além disso, o tamanho da topologia é uma característica considerada diretamente proporcional a complexidade de execução e gerenciamento da mesma (RIERA et al., 2015; VU; DINH; KIM, 2016; BAEK et al., 2017), sendo assim a redução do total de elementos operacionais necessários (empregando, por exemplo, VNFs com

múltiplos componentes de rede) é desejável a fim de simplificar os serviços.

A partir da cadeia de função de serviço descrita, dos resultados de *benchmarking* coletados e das categorizações realizadas nas funções de rede, um SFCR foi construído agrupando os dados necessários para a execução da solução de composição desenvolvida. A função objetivo especificada ambiciona pela minimização da taxa de tráfego (TxTrafego - tt), da diferença entre a taxa de requisição atendidas mensurada e a taxa de requisições atendidas desejada para o serviço (DifTxRequisição - dtr) e do tamanho (Tamanho - tam) da topologia. A taxa de tráfego é avaliada percentualmente, sendo o atributo de entrada iniciado com valor 100 indicando a totalidade do tráfego encaminhado ao serviço em um dado momento. A métrica de diferença de taxa de requisições atendidas apresenta como valor de entrada a capacidade definida em requisições por segundo que deve ser suportada pelo serviço, sendo esta fixada em 100. As métricas apresentam o mesmo grau de relevância para a geração do IAT, desta forma todas recebem valores de ponderação virtualmente iguais. Também, não houve preferência na divisão de tráfego no ponto de origem da ramificação presente na topologia, prevendo equivalência entre tráfego HTTP e HTTPS o que resulta em uma distribuição igualitária entre os seus ramos. As Figuras 23, 24, 25 e 26 ilustram a requisição gerada ao apresentar, respectivamente, os blocos de serviço, de função objetivo e de implantação, sendo o último segmentado em duas partes.

Figura 23 – Bloco de Serviço do Serviço de Teste

```

1 SERVICIO:
2 TOPOLOGIA: !!str NE EO1 EO2 { [
      EO3 EO5 EO6 ] ( EO3 EO5 * ) /
      [ EO4 EO5 EO6 ] ( EO4 EO5 * )
    } EO7 NS
3 NO_ENTRADA: NE
4 NOS_SAIDA:
5 - NS
6 ELEMENTOS_OPERACIONAIS:
7 - EO1
8 - EO2
9 - EO3
10 - EO4
11 - EO5
12 - EO6
13 - EO7

```

Figura 24 – Bloco de Função Objetivo do Serviço de Teste

```

1 FUNCAO_OBJETIVO:
2 - ID: TxTrafego
3 OBJETIVO: MIN
4 PONDERACAO: 0,3334
5 ENTRADA: 100
6 AVALICAO: MULT
7 ATUALIZACAO: TxTrafego
8 - ID: DifTxRequisicao
9 OBJETIVO: MIN
10 PONDERACAO: 0,3333
11 ENTRADA: 100
12 AVALICAO: SUB
13 ATUALIZACAO: ESTATICO
14 - ID: Tamanho
15 OBJETIVO: MIN
16 PONDERACAO: 0,3333
17 ENTRADA: 0
18 AVALICAO: SOM
19 ATUALIZACAO: ESTATICO

```

O cenário de implantação das composições candidatas obtidas, ilustrado na Figura 27, consiste de duas máquinas físicas sendo a primeira hospedeira das máquinas virtuais que as-

Figura 25 – Bloco de Implantação do Serviço de Teste (*Benchmark* - Início)

```

1 IMPLANTACAO:
2 DESCRICAO_EOS:
3   EO1:
4     BENCHMARK:
5       TxTrafego: 0,33
6       DifTxRequisicao: 1023,3991
7       Tamanho: 1
8   EO2:
9     BENCHMARK:
10      TxTrafego: 1,0
11      DifTxRequisicao: 1038,8371
12      Tamanho: 1
13   EO3:
14     BENCHMARK:
15      TxTrafego: 1,0
16      DifTxRequisicao: 44,4051
17      Tamanho: 1
18   EO4:
19     BENCHMARK:
20      TxTrafego: 1,0
21      DifTxRequisicao: 95,9090
22      Tamanho: 1
23   EO5:
24     BENCHMARK:
25      TxTrafego: 0,66
26      DifTxRequisicao: 78,9368
27      Tamanho: 1

```

Figura 26 – Bloco de Implantação do Serviço de Teste (*Benchmark* - Continuação)

```

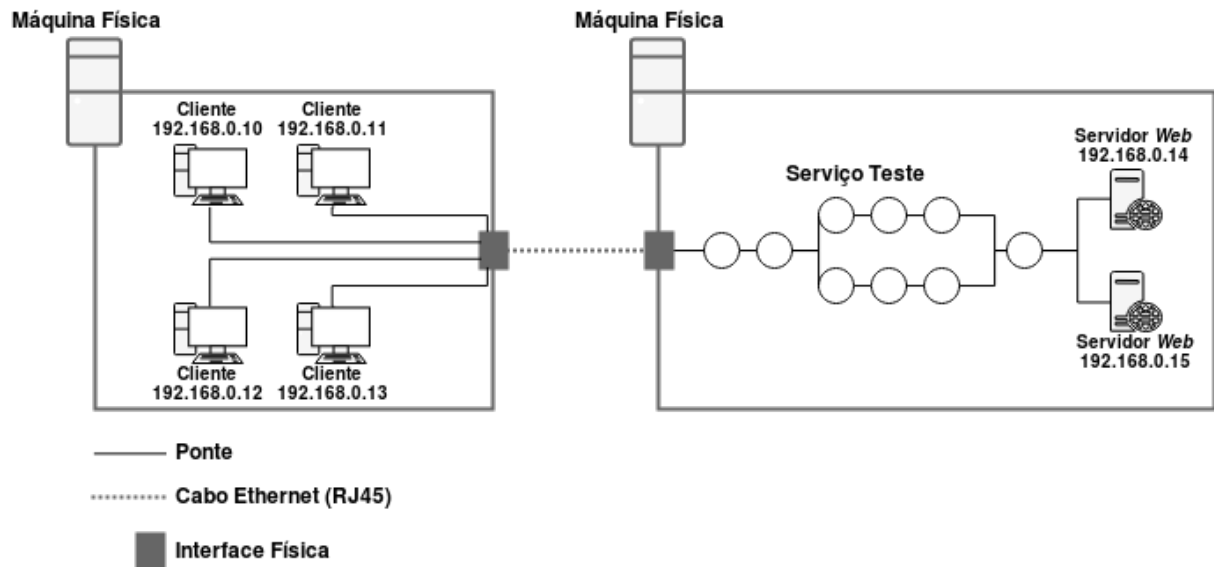
28   EO6:
29     BENCHMARK:
30      TxTrafego: 0,66
31      DifTxRequisicao: 84,4564
32      Tamanho: 1
33   EO7:
34     BENCHMARK:
35      TxTrafego: 1,0
36      DifTxRequisicao: 87,7749
37      Tamanho: 1
38   RAMIFICACOES:
39     TxTrafego:
40     DIVISAO: MULT
41     FATORES:
42     - [0,5, 0,5]
43     DifTxRequisicao:
44     DIVISAO: MULT
45     FATORES:
46     - [0,5, 0,5]
47     Tamanho:
48     DIVISAO: MULT
49     FATORES:
50     - [0,5, 0,5]

```

sumem o papel de clientes, esta contém 8GB de memória RAM DDR3, processador Core I3 4010U, sistema operacional Ubuntu 14.04 e virtualizador KVM; já a segunda é destinada a instanciação de máquinas virtuais que atuam como servidores HTTP e HTTPS ou como elementos operacionais que compõem o serviço de teste, para isto estão disponíveis 8 GB de memória RAM DDR3, processador Core I5 3330, sistema operacional Debian 8 e virtualizador KVM. Todas as máquinas virtuais utilizam o sistema operacional Debian 8 e demandam, no mínimo, 512 MB de memória RAM e 1 núcleo virtual de processamento. As máquinas físicas apresentam conexão Gigabit Ethernet, já as máquinas virtuais são conectadas através de pontes criadas utilizando o pacote bridge-utils. As máquinas virtuais representantes de cliente e servidor são alocadas em uma mesma rede /24, excluindo a necessidade de roteamento entre elas.

A infraestrutura física e virtualizada é utilizada para a execução de quatro cenários de teste: i) requisições HTTP e HTTPS normais; ii) requisições HTTP e HTTPS normais em conjunto a tráfego UDP; iii) requisições HTTP e HTTPS normais em conjunto a requisições HTTP e HTTPS irregulares; e iv) requisições HTTP e HTTPS normais em conjunto a requisições HTTP e HTTPS maliciosas. Clientes normais, no contexto deste trabalho, realizam processos

Figura 27 – Cenário de Avaliação do Serviço de Teste



de requisição HTTP e HTTPS não maliciosos, regulares (*i.e.*, apenas solicitando por serviços permitidos), completos (*i.e.*, abertura de conexão, requisição do serviço, recebimento de dados e fechamento da conexão), contínuos (*i.e.*, um novo processo de requisição é disparado logo após a conclusão do corrente) e sem concorrência (*i.e.*, não existe mais de uma conexão ativa entre um determinado cliente e servidor). Clientes considerados irregulares realizam apenas um processo de conexão e retransmitem insistentemente uma mesma requisição por um serviço não permitido. Já clientes maliciosos firmam apenas uma conexão e enviam por elas tráfego considerado regular, porém com pacotes anômalos que não implicam em nenhuma resposta, o objetivo dos clientes maliciosos é indisponibilizar o serviço de teste através da sobrecarga das funções de rede virtualizadas.

O primeiro cenário de teste apresenta quatro clientes normais, sendo dois deles HTTP e dois HTTPS, este cenário tem por finalidade avaliar o serviço de teste em condições normais de uso. O segundo cenário consiste de dois clientes normais (um HTTP e outro HTTPS) junto a dois clientes encaminhando tráfego UDP com pacotes de 1450 Bytes gerados através da ferramenta nping (LYON, 2009) a uma taxa de tráfego de aproximadamente 10Mbps (totalizando 20Mbps), este cenário objetiva avaliar o serviço quando tráfego não reconhecido pelos servidores *Web* é detectado, bloqueando o mesmo e atendendo apenas as requisições HTTP e HTTPS encaminhadas. O terceiro cenário também possui um cliente normal HTTP e um HTTPS, além de um cliente irregular HTTP e outro HTTPS, este cenário, por sua vez, objetiva avaliar o impacto da atuação dos filtros de conteúdo internos do serviço de teste. Finalmente, o quarto

cenário conta com um cliente HTTP e outro HTTPS normais executando requisições ao mesmo tempo que dois clientes maliciosos (um HTTP e outro HTTPS) realizam um ataque através do encaminhamento de pacotes anômalos de 1450 bytes a uma taxa individual de 10Mbps também gerados através da ferramenta nping, este cenário avalia o impacto da atuação da função de IPSH em diferentes composições candidatas do serviço de teste.

As composições candidatas submetidas aos cenários descritos são avaliadas quanto ao agregado e a taxa média de tráfego no serviço, estas métrica consistem, respectivamente, do somatório da quantidade de tráfego em MB e a média da taxa de tráfego em Mbps verificadas entre todas as pontes empregadas para a instanciação da topologia, estes dados são recuperados pelo monitoramento das pontes e interfaces virtuais através da ferramenta nload²⁵. Também, a taxa de requisições atendidas e o tempo médio de conexão por requisição atendida são utilizados como parte da avaliação, a ferramenta http-perf é novamente utilizada para a obtenção de tais dados. Por último, a quantidade de recursos necessários para a instanciação de uma composição candidata com configurações mínimas (*i.e.*, número mínimo de máquinas virtuais reservando a quantidade mínima de recursos computacionais para prover o serviço) é também verificada. As composições candidatas testadas são todas instanciadas com suas configurações mínimas e processam o tráfego dos clientes que começam a executar simultaneamente, sendo o ponto inicial e final dos teste definidos, nesta ordem, pelo primeiro e pelo último pacote enviado por algum dos clientes normais, sendo estes configurados para realizar 5000 processos de requisição.

5.2.2 Resultados

A avaliação da requisição do serviço de teste pela solução de composição desenvolvida retorna nove composições candidatas e seus respectivos IAT como descrito na Tabela 11. O maior IAT constatado foi retornado pela composição candidata de seis elementos operacionais, ou seja, aquela que passou por dois processos de redução de ramificações: EO6 representante da função IPSH alocado no segmento principal logo antes de EO3 (função CT); e EO5 caracterizada pela função MF, inserido como o novo ponto de intersecção dos ramos no segmento principal. Sua avaliação atinge os resultados mais apropriados para as métricas de tamanho e taxa de tráfego em decorrência da movimentação do elemento operacional IPSH para o segmento principal anterior a ramificação, desonerando a função de classificação de tráfego em caso de ataque, além disso, as duas reduções de ramificações realizadas implicam na exclusão

²⁵ <https://github.com/rolandriegel/nload>

de dois elementos operacionais duplicados (EO5 e EO6), reduzindo o tamanho final da composição candidata. Porém, esse processo de redução de ramificação afeta diretamente os resultados avaliativos da métrica de taxa de requisições desta composição candidata. Isso acontece uma vez que os elementos operacionais unificados processam uma maior quantidade de requisições já que estão presentes no segmento principal da topologia, tornando as suas capacidades de processamento individuais (78,936 para MFW e 84,456 para IPSH) insuficientes em relação ao objetivo especificado na requisição do serviço.

Tabela 11 – Composições Candidatas do Serviço de Teste

Composição Candidata	Agregados	Normalização	Complementação	Ponderação	IAT	Função Objetivo
NE EO1 EO6 EO2 { EO3 / EO4 } EO5 EO7 NS	tt: 127,1 dtr: -1853,7 tam: 7	tt: 0 dtr: 1 tam: 0	tt: 1 dtr: 0 tam: 1	tt: 0,333 dtr: 0 tam: 0,333	0,666	1º
NE EO1 EO6 EO2 { EO3 EO5 / EO4 EO5 } EO7 NS	tt: 127,1 dtr: -1932,6 tam: 8	tt: 0 dtr: 0,517 tam: 0,5	tt: 1 dtr: 0,483 tam: 0,5	tt: 0,333 dtr: 0,161 tam: 0,166	0,660	2º
NE EO1 EO2 { EO6 EO3 / EO6 EO4 } EO5 EO7 NS	tt: 138,3 dtr: -1938,2 tam: 8	tt: 0,5 dtr: 0,483 tam: 0,5	tt: 0,5 dtr: 0,417 tam: 0,5	tt: 0,167 dtr: 0,172 tam: 0,166	0,505	3º
NE EO1 EO2 { EO6 EO3 EO5 / EO6 EO4 EO5 } EO7 NS	tt: 138,3 dtr: -2017,1 tam: 9	tt: 0,499 dtr: 0 tam: 1	tt: 0,501 dtr: 1 tam: 0	tt: 0,167 dtr: 0,333 tam: 0	0,500	4º
NE EO1 EO2 { EO3 EO5 EO6 / EO6 EO4 EO5 } EO7 NS	tt: 143,9 dtr: -2017,1 tam: 9	tt: 0,749 dtr: 0 tam: 1	tt: 0,251 dtr: 1 tam: 0	tt: 0,083 dtr: 0,333 tam: 0	0,416	5º
NE EO1 EO2 { EO6 EO3 EO5 / EO4 EO5 EO6 } EO7 NS	tt: 143,9 dtr: -2017,1 tam: 9	tt: 0,749 dtr: 0 tam: 1	tt: 0,251 dtr: 1 tam: 0	tt: 0,083 dtr: 0,333 tam: 0	0,416	5º
NE EO1 EO2 { EO3 EO5 EO6 / EO4 EO5 EO6 } EO7 NS	tt: 149,5 dtr: -2017,1 tam: 9	tt: 1 dtr: 0 tam: 1	tt: 0 dtr: 1 tam: 0	tt: 0 dtr: 0,333 tam: 0	0,333	6º
NE EO1 EO2 { EO3 / EO4 } EO5 EO6 EO7 NS	tt: 149,5 dtr: -1853,7 tam: 7	tt: 1 dtr: 1 tam: 0	tt: 0 dtr: 0 tam: 1	tt: 0 dtr: 0 tam: 0,333	0,333	6º
NE EO1 EO2 { EO3 EO5 / EO4 EO5 } EO6 EO7 NS	tt: 149,5 dtr: -1932,6 tam: 8	tt: 1 dtr: 0,517 tam: 0,5	tt: 0 dtr: 0,483 tam: 0,5	tt: 0 dtr: 0,161 tam: 0,166	0,327	7º

Para a instanciação e experimentação dos cenários de teste, além da composição candidata com maior IAT (que também obtém os melhores resultados de taxa de tráfego e tamanho), outras duas topologias foram utilizadas: a que apresenta o maior IAT dentre aquelas que obtêm os melhores resultados da métrica de diferença de taxa de requisições; e aquela que obteve o pior IAT. No primeiro caso, a composição candidata com nove elementos operacionais e IAT de 0,5 é escolhida. Essa composição, além de minimizar a métrica de diferença de taxa de requisições através da alocação de elementos operacionais em ramificações que dividem o tráfego, também posiciona a função de rede IPSH logo no início de seus ramos, evitando que as posteri-

ores a ela processem tráfego malicioso. Já no segundo caso, a composição candidata conta com um IAT de 0,327 ocasionado pela alocação de funções de rede com característica de descarte nas últimas posições aplicáveis a elas, onerando o serviço como um todo com eventuais fluxos irregulares ou maliciosos. Também, existe uma diferença significativa entre a taxa de requisições da composição de IAT 0,327 em relação ao melhor resultado obtido pelas avaliações desta métrica já que redução de ramificação efetuada sobre EO6 não permite que ele atinja a meta definida, além disto os resultados de tamanho também não são os ideais dado a dupla ocorrência do elemento operacional EO5 na ramificação da topologia.

O primeiro conjunto de resultados foi colhido nas máquinas virtuais que assumem o papel de clientes e consiste da taxa de requisições e do tempo médio de conexão por requisição atendida tanto para requisições HTTPS, ilustrados nas Figuras 28 e 29, quanto para requisições HTTP, exibidos nas Figuras 30 e 31. A utilização de HTTPS tipicamente ocasiona menores taxas de requisições atendidas e maior tempo médio de conexão por requisição atendida, este fenômeno ocorre por dois motivos principais: execução de processos de criptografia e descriptografia no cliente e no servidor; e necessidade de comunicações extras para a troca do certificado que viabiliza a conexão segura. Além disso, existe uma proporcionalidade na variação dos resultados verificados entre HTTP e HTTPS quando considerados o conjunto de composições candidatas testadas nos cenários previstos (a exceção do cenário iv onde o ataque torna os resultados imprevisíveis), reforçando que tal característica está intrinsecamente ligada ao protocolo de aplicação usado.

Figura 28 – Taxa de Requisições HTTPS Atendidas

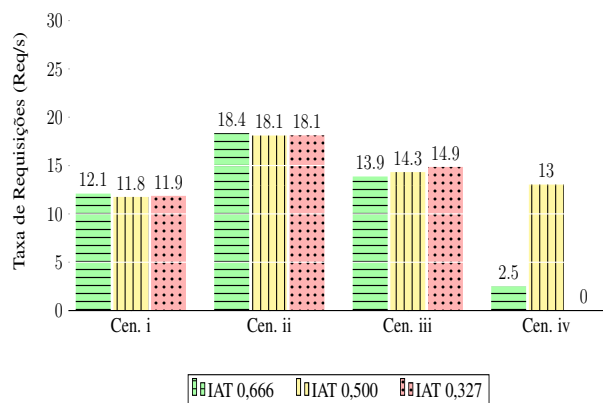
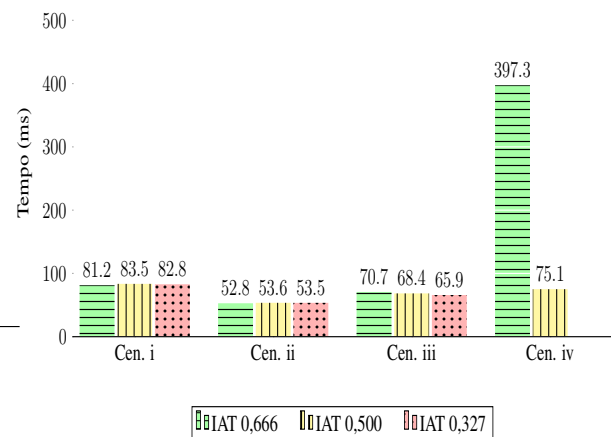


Figura 29 – Tempo Médio de Conexão por Requisição HTTPS Atendida



O primeiro cenário de teste remete a um uso contínuo e moderado do serviço, neste caso as funções de rede instanciadas operam abaixo de sua capacidade máxima, resultando em taxas

Figura 30 – Taxa de Requisições HTTP Atendidas

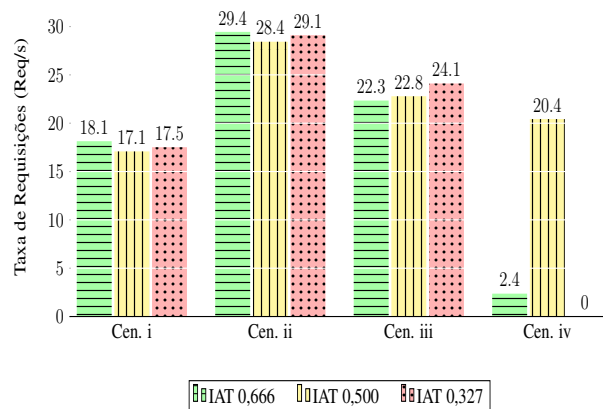
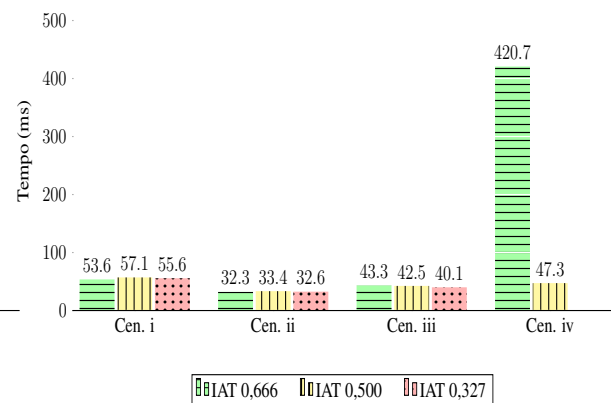


Figura 31 – Tempo Médio de Conexão por Requisição HTTP Atendida



de requisições atendidas e tempos de conexão semelhantes em todas as composições candidatas testadas. A topologia com IAT 0,666 manifesta resultados marginalmente superiores para a taxa de requisições e inferiores para o tempo de conexão em relação as demais, esta diferença é atribuída a redução da concorrência pelo uso dos núcleos físicos já que a sua configuração mínima é a mais enxuta dentre todas as composições consideradas. Analogamente, essa mesma situação ocorre, em escala reduzida, entre os resultados obtidos nas composições com IAT 0,327 e IAT 0,500 dado que a primeira apresenta uma configuração mínima com menores requisitos por núcleos virtuais em relação a segunda. O segundo cenário de teste demonstra resultados com comportamentos e justificativas semelhantes ao primeiro, neste caso todo o tráfego UDP encaminhado é descartado pelo filtro de protocolo/porta posicionado na borda de entrada do serviço. Apesar desse descarte acarretar certa sobrecarga sobre o primeiro elemento operacional da topologia, este não prejudica de maneira significativa o seu funcionamento visto que o filtro de protocolo/porta é um função com alta capacidade de processamento e não foi suficientemente sobrecarregada para se tornar um gargalo.

O terceiro cenário de teste cria um ambiente com quantidade de tráfego semelhante ao primeiro cenário, categorizando também uma condição de uso contínua e moderada. A taxa de requisições atendidas e o tempo médio de conexão neste cenário apresentam valores intermediários em relação aos dois anteriores. Essa situação surge devido o ingresso de requisições originadas por quatro clientes, porém sendo apenas o tráfego de dois deles de fato entregue e processado pelos servidores *Web* disponíveis. As requisições dos cliente irregulares são analisadas e marcadas pelos DPIs para posteriormente serem descartadas pelos filtros de marcação. Diferentemente de como ocorre para o filtro de protocolo/porta, as funções de DPI e filtragem

por marcação sobrecarregam-se com o tráfego irregular, ocasionando como efeito colateral a redução da vazão de requisições regulares. Ainda, neste cenário em específico, a composição candidata com IAT 0,327 compreende os melhores resultados em virtude da execução dos DPIs e filtros de marcação anteriormente ao IPS HTTP/S, isentando o último do processamento de tráfego irregular, porém não malicioso. Finalmente, a composição com IAT 0,666 evidenciou resultados ligeiramente inferiores em relação as demais, isto ocorre graças a presença de um único filtro de marcação alocado no segmento principal da topologia, aumentando a sobrecarga de processamento neste.

O quarto cenário consiste de um ataque de inundação de pacotes anômalos em conjunto a requisições originadas por clientes normais. Esse ataque é detectado pela função de IPS HTTP/S que descarta os fluxos maliciosos e encaminha apenas as requisições consideradas reais. É importante ressaltar também que o filtro de protocolo/porta e o classificador de tráfego apresentam capacidade de processamento suficiente para suportar o ataque e não se convertem em gargalos durante a realização do teste. Sendo assim, a composição candidata com IAT 0,500 apresentou maior eficiência na mitigação do ataque, evitando danos colaterais expressivos na prestação do serviço aos clientes normais. Esse efeito é fruto da duplicação da função de IPS e alocação destas nas primeiras posições dos ramos originados pelo classificador de tráfego que, por sua vez, realizou um balanceamento involuntário do tráfego malicioso. A composição candidata com IAT 0,666 também foi capaz de suportar o ataque e manter o serviço parcialmente operante, entretanto, a sobrecarga verificada na única instância da função de IPS HTTP/S alocada no segmento principal da topologia logo antes do classificador de tráfego foi suficiente para aumentar excessivamente o tempo médio de conexão dos clientes normais e, por consequência, diminuir a taxa de requisições atendidas. Por fim, a composição candidata com IAT 0,332 foi indisponibilizada pelo ataque, situação originada devido a alocação do IPS como ponto de intersecção da ramificação a qual atuam as funções de DPI e filtro de marcação, sendo assim toda a ramificação é sobrecarregada durante o ataque e tem esta sobrecarga agravada dado a realização de operações de busca por assinaturas e palavras-chave no *payload* dos pacotes maliciosos, considerados muito grandes para requisições HTTP e HTTPS. Ainda, evidencia-se que a indisponibilidade do serviço foi constatada devido aos subseqüentes esgotamentos do tempo de espera TCP ocorridos tanto nos clientes normais quanto nos servidores *Web*.

O segundo conjunto de resultados, exposto na Tabela 12, foi coletado através da análise do tráfego de rede entre as máquinas virtuais que hospedam os elementos operacionais do ser-

Tabela 12 – Agregado e Taxa de Tráfego do Serviço de Teste

Composição Candidata	Métricas	Cenário i	Cenário ii	Cenário iii	Cenário iv
IAT 0,666	Agregado de Tráfego Processado pelo Serviço (MB)	108,43	737,24	99,29	10495,8
	Taxa de Tráfego Média no Serviço (Mbps)	2,002	20,645	2,101	37,929
IAT 0,500	Agregado de Tráfego Processado pelo Serviço (MB)	107,73	795,38	97,81	2997,9
	Taxa de Tráfego Média no Serviço (Mbps)	1,935	21,976	2,138	59,631
IAT 0,327	Agregado de Tráfego Processado pelo Serviço (MB)	108,57	754,43	88,9	-
	Taxa de Tráfego Média no Serviço (Mbps)	1,966	20,864	2,015	-

viço de teste, os servidores *Web* e os clientes. As métricas observadas consistem do agregado de tráfego relacionado a conexões e requisições HTTP e HTTPS e a taxa de tráfego média, ambas recuperadas através do monitoramento de todas as interfaces de rede e pontes de conexão que servem ao serviço de teste após instanciado. O primeiro e o segundo cenário apontam resultados semelhantes entre todas as composições candidatas testadas, motivo disto é a inexistência de descarte (cenário i) ou o descarte de pacotes realizado num ponto comum de todas as topologias (cenário ii). Já para o terceiro cenário de teste, uma variação considerável ocorre nas métricas recuperadas devido a organização topológica da composição com IAT 0,327, nela, como explicitado anteriormente, o tráfego de requisições irregulares é descartado antes de seu processamento pelo IPS HTTP/S, o que ocasiona uma redução em ambas as métricas quando comparadas as observadas nas demais composições (aproximadamente 10% no agregado de tráfego e 5% na taxa de tráfego média).

Por último, o quarto cenário de teste é o que apresenta os resultados mais desiguais entre as composições candidatas testadas devido aos diferentes comportamentos das mesmas durante a execução de um ataque. Neste cenário, a composição com IAT 0,327 não foi considerada visto que o serviço se tornou indisponível, já a composição com IAT 0,500 obteve o menor valor agregado de tráfego uma vez que foi capaz de manter o serviço estável durante o ataque e preservar uma vazão de requisições relativamente alta, concluindo o teste em menor tempo. Porém, como nessa composição o tráfego malicioso se perpetua e é processado por três funções de rede diferentes (FPP, CT e IPSH) antes de ser descartado, sua taxa de tráfego média é expressivamente maior do que a obtida para a composição com IAT 0,666 (aproximadamente 57%) cujo bloqueia o tráfego malicioso após o processamento deste por duas funções de rede

apenas (FPP e IPSH).

Tabela 13 – Recursos Computacionais Utilizados no Serviço de Teste

Composição Candidata	Máquinas Virtuais	Memória (MB)	Núcleos Virtuais (CPU)	Interfaces de Rede Virtuais
IAT 0,666	7	3584	7	16
IAT 0,500	9	4608	9	20
IAT 0,327	8	4096	8	18

O terceiro e último conjunto de resultados é formado pela totalidade de recursos computacionais utilizados para a instanciação do serviço de teste com configurações mínimas. Esses valores, expostos na Tabela 13, são diretamente proporcionais ao tamanho da topologia utilizada, visto que cada elemento operacional contido na mesma é traduzido para uma máquina virtual contando apenas com uma quantidade fixa de recursos estritamente necessários para sua execução. Sendo assim, a composição candidata com IAT 0,666 é a que apresenta os requisitos mínimos de instanciação mais enxutos dentre as analisadas, consequência de duas reduções de ramificações realizadas que resultaram em uma topologia com apenas sete elementos operacionais. A composição com IAT 0,327, por sua vez, passou por um único processo de redução de ramificações, implicando em oito elementos operacionais em sua cadeia de serviço e necessidade de recursos intermediária, posicionada entre a composição com IAT 0,666 e a com IAT 0,500 cujo mantém seus ramos com três elementos operacionais cada, requisitando assim, no mínimo, a instanciação de nove máquinas virtuais para a viabilizar o serviço de teste.

Os resultados empíricos observados atestam a acurácia da avaliação realizada pela solução de composição desenvolvida. A composição candidata com IAT 0,327 obteve resultados intermediários para as métricas de diferença de capacidade e tamanho, além de apresentar dificuldades organizacionais para descartar tráfego indesejado. Isso se comprova na verificação da quantidade de recursos necessária para a implantação do serviço através de suas configurações mínimas, além do comportamento do mesmo durante a ocorrência do ataque de inundação de pacotes anômalos, o que gerou a indisponibilidade do serviço devido sua capacidade mediana de processamento de requisições aliada a uma postergação do descarte de tráfego malicioso. A composição candidata com IAT 0,500, por seu turno, obteve o melhor resultado para a métrica de diferença de taxa de requisições, sugerindo um serviço de alta vazão e, ao mesmo tempo, atingiu um resultado intermediário para a métrica de taxa de tráfego, sendo capaz de desonerar boa parte de suas funções de rede de operarem sobre tráfego indesejado. Porém, para que essas características se concretizem foi necessária a utilização de um maior número de elementos

operacionais que, por consequência, gerou a configuração mínima com os maiores requisitos de recursos computacionais. Essa descrição também se comprova durante o cenário de ataque, onde a vazão de requisições de clientes normais e o atendimento a elas se mantém semelhante as dos demais cenários.

Finalmente, a composição com IAT 0,666 apresenta o pior resultado para a métrica de diferença de taxa de requisição, originado pela unificação no segmento principal de duas funções originalmente presentes na ramificação da cadeia. Por outro lado, essa composição obteve os melhores resultados relativos a taxa de tráfego e tamanho devido a essas mesmas unificações e o posicionamento estratégico das funções participantes delas. Mais uma vez, essas verificações se comprovam durante a implantação e execução do serviço de teste onde o consumo reduzido de recursos impacta nas marcas de vazão e atendimento de requisições detectadas em cenários com ausência de ataque, sendo estas marginalmente melhores devido a subutilização das funções de rede (sem nenhum sinal de sobrecarga) e a necessidade reduzida por escalonamentos de núcleos virtualizados de processamento. Também, apesar de serem evidenciadas maiores dificuldades no tratamento do tráfego malicioso em relação à composição candidata com IAT 0,500 no cenários de ataque, a composição com IAT 0,666 não indisponibiliza o serviço em nenhum momento, além de ser capaz de reduzir de modo significativo a taxa de tráfego média que a percorre através do descarte prematuro dos fluxos maliciosos. É importante ressaltar também que, neste último caso, o gargalo na função IPSH ocorre devido as características de implementação desta função e pela instanciação dela através de suas configurações mínimas. Dessa forma, é possível aumentar a capacidade do IPSH em específico através do aprimoramento de qualquer uma das propriedades citadas, sendo a primeira sem consequências de aumento na utilização de recursos e a segunda com mudanças pontuais exclusivamente na quantidade de núcleos virtualizados empregados.

5.3 SUMARIZAÇÃO E DISCUSSÃO

A etapa de composição tem por objetivo determinar e avaliar topologias absolutamente ordenadas válidas a partir de especificações mais genéricas que contenham, por exemplo, segmentos parcialmente ordenados e dependências de VNF. Diversas inovações foram realizadas em relação aos tipos de análise e heurísticas empregues em soluções que abordam essa etapa, a exemplo das presentes em trabalhos como (MEHRAGHDAM; KELLER; KARL, 2014; DRÄXLER; KARL, 2017; OCAMPO et al., 2017; GIL-HERRERA; BOTERO, 2017; WANG et al.,

2017). Entretanto, as ferramentas de composição ainda são estáticas quanto a função objetivo que orienta a avaliação das topologias, não permitindo ao operador de rede adequar as mesmas para processarem métricas decorrentes de suas necessidades particulares e/ou de características específicas de determinados serviços.

Considerando esse cenário, o presente trabalho apresenta uma nova solução de composição com suporte a configuração sob demanda de métricas de análise e função objetivo. Essa solução realiza uma busca exaustiva pelo resultado ótimo e utiliza uma metodologia genérica para a geração de seus resultados. Sendo assim, a execução da avaliação de composições candidatas retorna índices (IAT) que caracterizam cada uma delas em relação a função objetivo requisitada. Um serviço de segurança, controle de conteúdo e balanceamento de carga para fluxos HTTP e HTTPS foi especificado e desenvolvido para a experimentação e validação da metodologia de indexação aplicada a solução de composição proposta. Os resultados obtidos mostraram que a avaliação das composições candidatas retrataram o comportamento real do serviço de rede virtualizado quando instanciado em um ambiente de teste. Finalmente, é importante ressaltar que, apesar da solução desenvolvida analisar todas as possibilidades de composição possíveis (força bruta), heurísticas e podas na execução podem ser planejadas em busca de reduzir o espaço de busca e/ou tempo de execução.

6 DIVISÃO E MAPEAMENTO ORIENTADA A ANÁLISE PERSONALIZADA DE POLÍTICAS

As soluções de divisão e mapeamento presentes na literatura têm como objetivo fundamental distribuir os elementos operacionais de uma topologia entre diferentes domínios disponíveis. Com isso, políticas e dependências de infraestrutura devem ser observadas para garantir que a integração do serviço requisitado cumpra com determinadas características operacionais (*e.g.*, atrasos máximos, uso de rotas específicas) e regras de negócio (*e.g.*, orçamento máximo de implantação, utilização fixa de determinados provedores) previamente especificadas. Porém, essas soluções tipicamente avaliam um conjunto predeterminado e imutável de políticas, permitindo apenas a alteração dos valores limítrofes que definem seus critérios de aceitação ou rejeição para a avaliação de topologias. Além disso, as soluções implementadas de maneira não coordenada com as demais etapas de implantação (em especial, as do NFV-RA) não consideram de nenhuma forma os resultados de avaliações prévios a sua execução, mantendo não apenas as metodologias operacionais, mas também as subsequentes distribuições geradas, independentes e agnósticas a elas.

Dessa forma, uma solução de divisão e mapeamento com suporte a customização de conjunto de políticas e das avaliações a elas aplicadas é proposta como parte deste trabalho. A solução desenvolvida utiliza as políticas de integração do serviço tanto para eliminar distribuições com características inapropriadas em relação a elas, quanto para classificar as distribuições consideradas como candidatas a execução. A solução desenvolvida também comporta a análise simultânea de múltiplas topologias que proveem o mesmo serviço, analisando e classificando as distribuições geradas de maneira isonômica a organização topológica empregada. Também, é prevista a possibilidade de harmonização entre índices de composição desde que estes estejam contidos no intervalo $[0;1]$ (*i.e.*, aos moldes do IAT definido no Capítulo 5) e os índices de distribuição gerados por essa solução, caracterizando assim a tupla formada por uma topologia e sua distribuição a fim de obter o melhor custo/benefício entre os objetivos definidos para duas etapas distintas de implantação. A seguir, a metodologia da solução proposta é apresentada na Seção 6.1 e a avaliação realizada exposta na Seção 6.2. Finalmente, uma sumarização e uma discussão geral do capítulo são incluídas na Seção 6.3.

6.1 METODOLOGIA DA SOLUÇÃO

A solução de divisão e mapeamento desenvolvida é dividida em três procedimentos: distribuição de topologia, validação de distribuições e avaliação de distribuições candidatas. O primeiro procedimento (*i.e.*, distribuição de topologia) consiste na identificação de todas as possibilidades de divisão e mapeamento tecnicamente viáveis de uma topologia nos domínios disponíveis. O segundo procedimento (*i.e.*, validação de distribuições) verifica a conformidade de cada uma das distribuições viáveis em relação as políticas especificadas pelo operador de rede, resultando em um conjunto de distribuições candidatas que não as infringem de nenhuma forma. Finalmente, terceiro procedimento (*i.e.*, avaliação de distribuições candidatas) analisa as distribuições candidatas uma a uma a fim de determinar um Índice de Adequabilidade de Distribuição (IAD) que relaciona seus resultados avaliativos particulares a todos os demais previamente determinados, além de as ranquear considerando o mesmo. Essa solução pode ser aplicada tanto a uma única topologia, quanto a um conjunto delas, no segundo caso, o terceiro procedimento é executado após a conclusão de seus dois antecessores para todas as topologias informadas, gerando assim parâmetros isonômicos utilizados na avaliação de diferentes composições que prestam o mesmo serviço. Ainda durante esse último procedimento ocorre, se requisitado, a harmonização entre os resultados de composição (IAT) e o índice de distribuição gerado.

Figura 32 – Modelo de Documento de Domínios (Domínios e Recursos)

```

1 DOMINIOS:
2   - &dominio1!!str
3   ...
4   - &dominion!!str
5 RECURSOS:
6   *dominio1:
7     MEMORIA!!int
8     REDE!!int
9     CPU!!int
10  ...
11  *dominion:
12    MEMORIA!!int
13    REDE!!int
14    CPU!!int

```

Figura 33 – Modelo de Documento de Domínios (Métricas Locais e de Transição)

```

1 METRICAS_LOCAIS:
2   - ID!!str
3     *dominio1!!int ou !!float
4     ...
5     *dominion!!int ou !!float
6     ...
7 METRICAS_TRANSICAO:
8   - ID!!str
9     *dominio1!!int ou !!float
10    *dominio2!!int ou !!float
11    *dominio5!!int ou !!float
12    ...
13    ...
14    *dominion:
15      *dominio1!!int ou !!float
16    ...

```

Para que as distribuições possam ser avaliadas em relação aos domínios disponíveis, é necessário que os provedores destes disponibilizem uma série padrão de informações relacionadas a métricas de desempenho e de negócio aplicáveis a especificação de políticas de integração. Portanto, em conjunto a requisição do serviço, a solução de divisão e mapeamento recebe um documento com formatação YAML, cujo modelo é exposto nas Figuras 32 e 33, contendo informações sobre os domínios disponíveis. A construção do documento de domínios é realizada através de quatro blocos principais: DOMINIOS, destinado a definição dos identificadores únicos de cada domínio; RECURSOS, utilizado para indicar o montante de recursos computacionais disponíveis neles; METRICAS_LOCAIS, empregado na definição de valores de métricas particulares de cada domínio (*e.g.*, custo financeiro por instanciação de VM ou de processamento); e METRICAS_TRANSICAO, que detalha valores de métricas referentes a possíveis transições entre um domínio e outro (*e.g.*, latência, distância geográfica).

Mais especificadamente, o bloco DOMINIOS é composto por uma lista de cadeias de caracteres onde cada ocorrência retrata um possível domínio a ser usado durante a distribuição do serviço. Já o bloco RECURSOS contempla um dicionário de objetos cujo as chaves são os identificadores dos domínios apontados no bloco anterior (*i.e.*, DOMINIOS). Para cada objeto presente nesse existem três atributos obrigatórios denominados MEMORIA (memória volátil em MB), REDE (interfaces de rede virtualizadas) e CPU (núcleos de processamento virtualizados), estes armazenam números inteiros indicando as suas respectivas disponibilidades no domínio ao qual se referem. Os últimos dois blocos do documento de domínios especificam informações que podem ser empregadas durante a definição de políticas de integração. Em relação a isso, os blocos METRICAS_LOCAIS e METRICAS_TRANSICAO são formado por uma lista de objetos que apresentam uma possível métrica de interesse. Essa métrica é identificado pelo atributo ID, por sua vez construído sobre uma cadeia de caracteres única no documento, e seu objeto compreende uma série de atributos que definem seu valor em cada domínio (METRICAS_LOCAIS) ou para cada transição possível (METRICAS_TRANSICAO). No primeiro caso, esses atributos contêm apenas um valor numérico inteiro ou em ponto flutuante, já no segundo caso, eles armazenam dicionários que relacionam os domínios cujo a transição é possível a valores numéricos inteiros ou em ponto flutuante.

A exemplificação dos procedimentos que fazem parte da solução de divisão e mapeamento ocorre através de um ambiente com três domínios (DOM1, DOM2 e DOM3), sendo as informações operacionais dos mesmos fornecidas por um documento de especificação como o

Figura 34 – Documento de Domínios do Exemplo Padrão (Início)

```

1 DOMINIOS:
2   - DOM1
3   - DOM2
4   - DOM3
5 RECURSOS:
6   DOM1:
7     MEMORIA: 1280
8     REDE: 10
9     CPU: 6
10  DOM2:
11   MEMORIA: 5120
12   REDE: 6
13   CPU: 4
14  DOM3:
15   MEMORIA: 2048
16   REDE: 5
17   CPU: 3
    
```

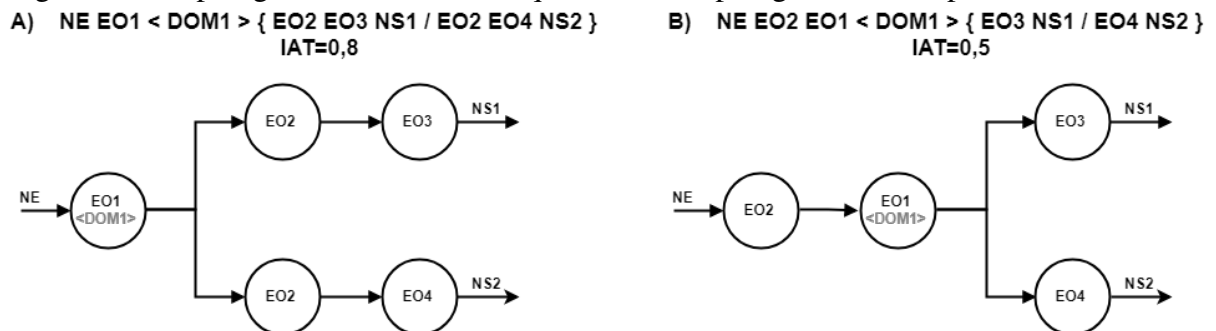
Figura 35 – Documento de Domínios do Exemplo Padrão (Continuação)

```

1 METRICAS_LOCAIS:
2   - ID: CUSTO
3     DOM1: 100
4     DOM2: 150
5     DOM3: 250
6 METRICAS_TRANSICAO:
7   - ID: RTT
8     DOM1:
9       DOM2: 100
10    DOM2:
11      DOM1: 10
12      DOM3: 20
13    DOM3:
14      DOM1: 10
15      DOM2: 15
    
```

apresentado nas Figuras 34 e 35. Sobre este ambiente duas composições simples de um mesmo serviço são divididas e mapeadas, estas são ilustradas na Figura 36 através de suas topologias e especificações SCAG e contemplam quatro elementos operacionais distintos (EO1, EO2, EO3, EO4), dois nós de borda de saída (NS1 e NS2) e uma ramificação terminal de dois ramos, cada um com dois elementos em A e um elemento em B, além disso o elemento operacional EO1 é marcado com uma dependência de infraestrutura que o fixa em DOM1. Finalmente, o valor de IAT previamente calculado durante o processo de composição é fornecido em conjunto a cada uma das topologias (0,8 para a topologia A e 0,5 para a topologia B), este é utilizado para a geração de um índice unificado que harmoniza as etapas de composição e integração.

Figura 36 – Topologias e Índices de Adequabilidade Topológica do Exemplo Padrão



Dois blocos provenientes do SFCR são usados durante a execução da divisão e mapeamento das topologias: políticas e implantação. Para o exemplo padrão, o bloco de políticas contempla duas da categoria imediata (RTT e CustoFin) e uma da categoria agregada (CustoFin). A política imediata RTT indica o *Round Trip Time* máximo aceitável em uma transição

entre domínios, já a política imediata *CustoFin* representa a faixa de preço adotada para a instanciação de um elemento operacional nos provedores de infraestrutura disponíveis. Por último, a política agregada *CustoFin* também analisa o custo financeiro associado aos domínios utilizados na distribuição da topologia, porém esta não determina o teto de valores para integração de elementos operacionais individuais, mas sim para a totalidade das integrações realizadas. O objetivo de minimização das métricas avaliadas pelas políticas é comum a todas elas. Finalmente, o bloco de implantação disponibiliza informações sobre a quantidade mínima de recursos computacionais necessários para efetivar a instanciação dos elementos operacionais. As Figuras 37 e 38 exibem, respectivamente, o bloco de políticas e o bloco de implantação através do atributo *FLAVOUR* que foram utilizados nas exemplificações.

Figura 37 – Bloco de Políticas do Exemplo Padrão

```

1 POLITICAS:
2   IMEDIATAS:
3     - ID: RTT
4     PONDERACAO: 0,5
5     TIPO: TRANSICAO
6     MINIMO: 0
7     MAXIMO: 15
8     OBJETIVO: MIN
9     - ID: CustoFin
10    PONDERACAO: 0,2
11    TIPO: LOCAL
12    MINIMO: 0
13    MAXIMO: 500
14    OBJETIVO: MIN
15  AGREGADAS:
16    - ID: CustoFin
17    PONDERACAO: 0,3
18    TIPO: LOCAL
19    MINIMO: 0
20    MAXIMO: 900
21    OBJETIVO: MIN

```

Figura 38 – Bloco de Implantação do Exemplo Padrão (*Flavours*)

```

1 IMPLANTACAO:
2   DESCRICAO_EOS:
3     EO1:
4       FLAVOUR:
5         MEMORIA: 512
6         REDE: 3
7         CPU: 1
8     EO2:
9       FLAVOUR:
10      MEMORIA: 1536
11      REDE: 2
12      CPU: 1
13    EO3:
14      FLAVOUR:
15      MEMORIA: 512
16      REDE: 2
17      CPU: 1
18    EO4:
19      FLAVOUR:
20      MEMORIA: 1024
21      REDE: 2
22      CPU: 1

```

A solução reportada nesta seção foi desenvolvida como prova de conceito através da linguagem de programação Python 3. A implementação desta²⁶ foi configurada para processar nativamente os resultados obtidos pela solução de composição apresentada no Capítulo 5, porém seus módulos podem ser facilmente desmembrados e utilizada individualmente ou como parte de outras soluções de implantação.

²⁶ Disponível em <https://github.com/ViniGarcia/NFV-FLERAS>

6.1.1 Distribuição de Topologia

Uma topologia SFC já composta (*i.e.*, com ordenamento absoluto) pode ser integrada a uma infraestrutura física multi-domínio de diferentes maneiras. Sendo assim, os domínios devem garantir os recursos computacionais e virtualizados necessários para hospedar um ou um grupo de elementos operacionais destinados a eles. O primeiro procedimento de integração consiste em determinar todas as distribuições de uma topologia suportadas pela infraestrutura física multi-domínio disponível, este conjunto de possibilidades, chamado de distribuições viáveis, é construído a partir de dois passos: permuta de domínios e checagem de viabilidade física. Esses passos recebem como argumentos de processamento as informações estruturais e as dependências de infraestrutura das topologias a serem distribuídas, além dos requisitos de recursos computacionais presentes no bloco de implantação do SFCR e a disponibilidade dos mesmos localizados no documento de domínios (*i.e.*, bloco RECURSOS).

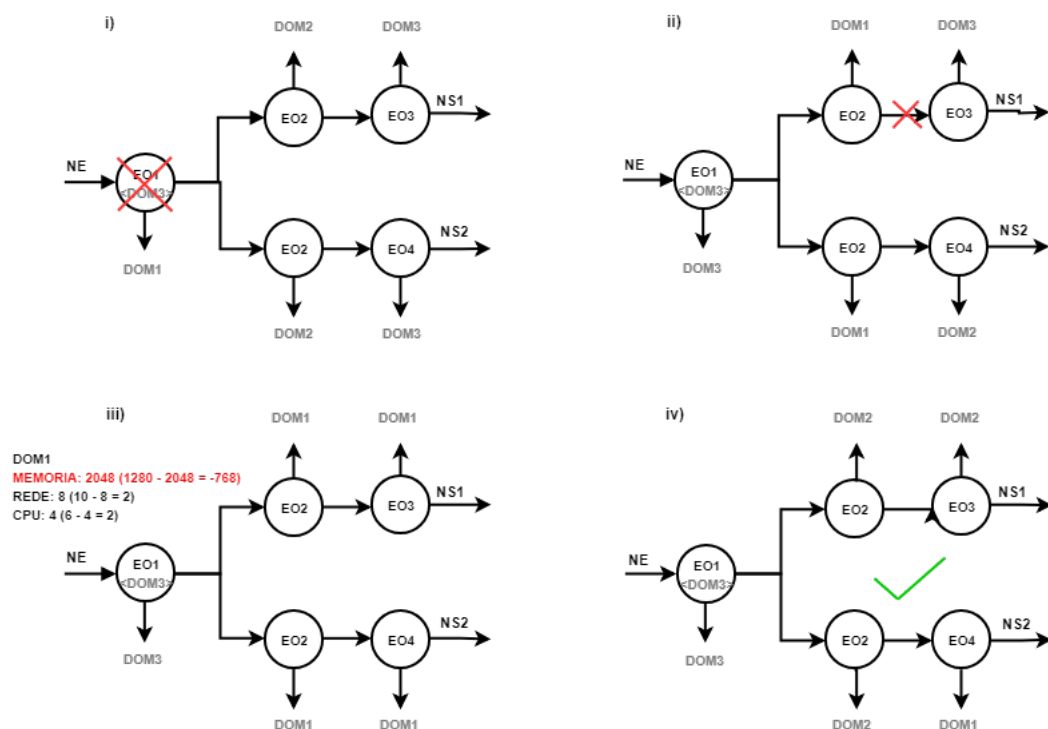
A permuta de domínios tem por objetivo determinar as opções de divisão e mapeamento de uma topologia na infraestrutura física disponível. Entretanto, devido a possível presença de elementos operacionais com dependências de infraestrutura, restrições são adicionadas ao processo a fim de eliminar distribuições previamente reconhecidas como inválidas, ou seja, que não observam as dependências especificadas. Dessa forma, a quantidade total de distribuições retornadas deste passo remete ao resultado da multiplicação entre o número de possibilidades de integração de cada um dos elementos operacionais, sendo que este é traduzido em um de dois valores possíveis: quantidade de domínios disponíveis, para aqueles elementos operacionais que não apresentam nenhuma dependência de infraestrutura e, portanto, podem ser alocados em qualquer domínio; ou 1 para os elementos operacionais que contêm dependências de infraestrutura.

O total de distribuições resultantes do passo de permuta de domínios (TDP) é formalizado pela Equação 6.1, nela "d" representa o número de domínios disponíveis, "e" a quantidade de elementos operacionais na topologia e "i" a quantidade de dependências de infraestrutura especificadas. Considerando as topologias A e B do exemplo padrão e os domínios utilizados no mesmo, a definição das variáveis "d" e "i" são comuns em ambos os casos, assumindo os valores de, respectivamente, 3 e 1. Já a variável "e" difere entre as topologias utilizadas, sendo que para A ela assume o valor 5 e resulta em 81 distribuições e para B é considerado o valor 4, originando 27 distribuições.

$$TDP = d^{e-i} \quad (6.1)$$

As distribuições obtidas no primeiro passo são então enviadas como argumentos a serem processados pelo passo de checagem de viabilidade física. Durante sua execução são verificadas as transições entre domínios distintos e o consumo de recursos computacionais previsto para cada um deles a fim de assegurar a viabilidade técnica de uma distribuição. Com isso, uma matriz de adjacências resultante da validação das métricas de transição é comparada as distribuições para detectar e excluir aquelas que contêm transições irregulares entre domínios (*i.e.*, que não foram definidas ou que não existem). Em seguida, os requisitos anotados para os elementos operacionais são agregados domínio a domínio de acordo com a distribuição em análise, sendo estas agregações comparadas com a disponibilidade de recursos computacionais nos domínios correspondentes através de uma subtração do segundo pelo primeiro, assim se o resultado apresentar um valor menor do que zero a distribuição é invalidada.

Figura 39 – Processos Eliminatórios no Exemplo Padrão



Com o exemplo padrão adotado nesta seção, a avaliação das transições realizada excluiu 17 resultados da topologia A e 9 da topologia B devido a, pelo menos, uma ocorrência da única transição inexistente segundo o documento de domínios utilizado (de DOM1 para DOM3). Posteriormente, a verificação de recursos descartou mais 52 distribuições referentes a topologia

A, sendo 23 delas disparadas devido a falta de interfaces de rede e 29 por insuficiência de memória, e 8 distribuições derivadas da topologia B, de onde 6 foram atribuídas a carência de memória e as demais a falta de interfaces de rede. Isso posto, o conjunto final de resultados gerados após a execução do procedimento nas topologias A e B incluem, respectivamente, 12 e 10 distribuições consideradas tecnicamente viáveis.

A execução dos processos eliminatórios do procedimento de distribuição de topologias (checagem de dependências topológicas, checagem de transições e checagem de viabilidade física) são ilustrados através da exemplificação presente na Figura 39. As análises realizadas em i, ii e iii resultam no descarte das distribuições avaliadas devido, respectivamente, ao não cumprimento da dependência de infraestrutura especificada, a existência de uma transição inexistente no documento de domínios e a inviabilidade de integração de todos os elementos operacionais destinados a um domínio. Por fim, a análise em iv indica a viabilidade da distribuição testada, sendo esta adicionada no conjunto de resultados retornado pelo procedimento.

6.1.2 Validação de Distribuições

O procedimento de validação opera sobre as distribuições viáveis retornadas pelo primeiro procedimento em vista de determinar quais delas são adequadas ao conjunto de políticas especificadas pelo operador de rede. Para isso dois passos são executados concomitantemente: exame de políticas e agregação de resultados. O objetivo final desse procedimento é construir um conjunto de composições candidatas utilizáveis na integração de uma topologia na infraestrutura multi-domínio disponível, sendo garantida a inexistência de conflitos relacionados as conexões e recursos físicos (garantido pelo primeiro procedimento) e as políticas especificadas. Além disso, o somatório dos resultados de validação das métricas empregadas nas políticas (tanto da categoria imediata quanto agregada) de cada distribuição candidata são também retornados para avaliação no subsequente procedimento.

Os passos do procedimento de validação iteram sobre todas as posições de uma topologia, percorrendo linearmente a especificação SCAG de maneira semelhante a uma busca em profundidade na abstração do grafo de serviço, uma exceção ocorre durante o processamento de ramificações não terminais que, após iniciada, têm todos os seus ramos avaliados antes do retorno da execução ao segmento principal da cadeia. Também, uma distribuição referente a uma topologia em análise é percorrida ao mesmo tempo e de maneira linear toda as vezes em que se iteram sobre elementos operacionais na topologia. As políticas especificadas apresen-

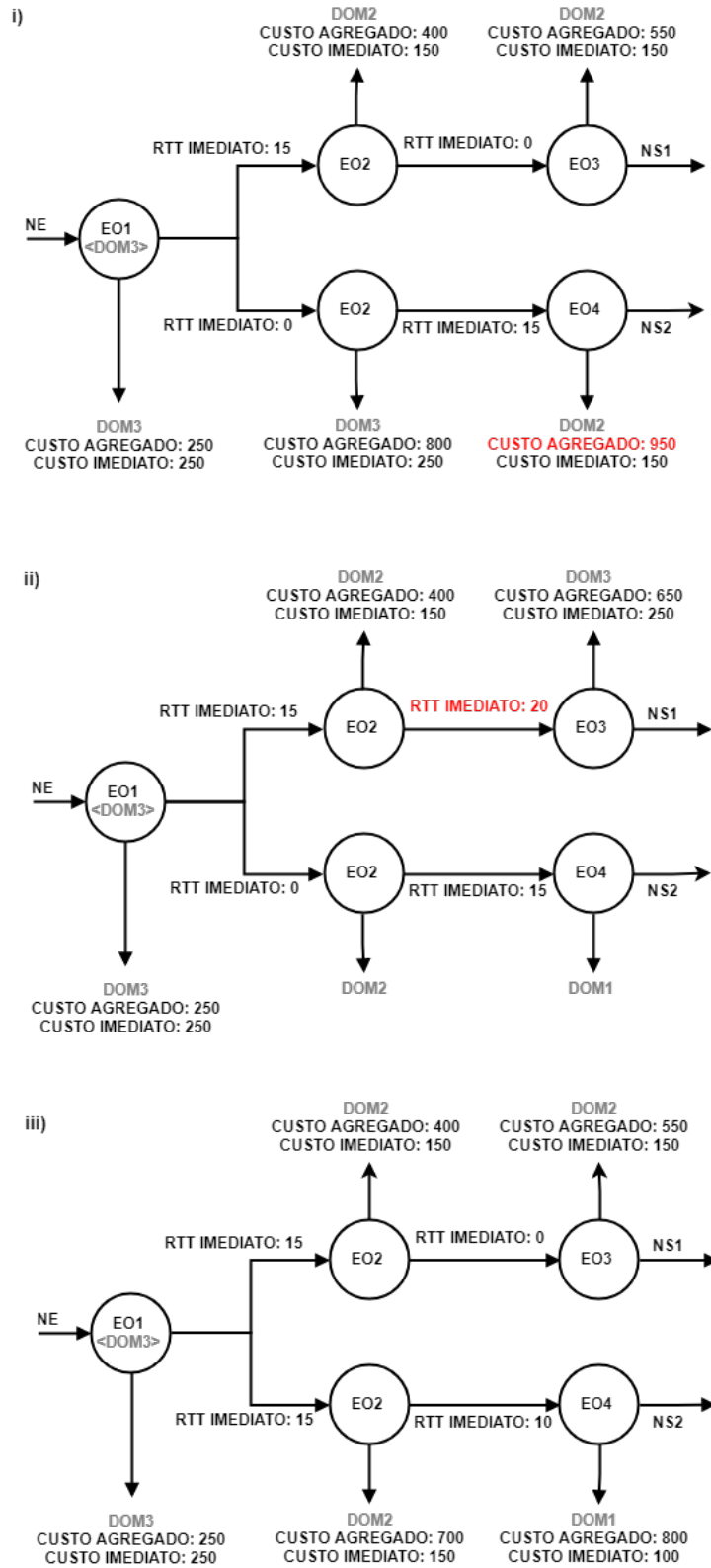
tam uma variável de acúmulo iniciada com o valor zero para cada distribuição a ser validada, nesta variável os dados numéricos verificados nas métricas disponibilizadas pelos domínios são somados passo a passo, antes mesmo da execução do exame das políticas. Políticas da categoria imediata são validadas considerando diretamente os valores presentes no documento de domínios em exercício, já políticas da categoria agregada são examinadas através dos valores contidos em suas respectivas variáveis de acúmulo.

Na validação de uma distribuição, três fenômenos que alteram o curso normal de processamento podem ocorrer durante o percorrido da topologia: início de uma ramificação, mudança de ramo e final de uma ramificação. O início de uma ramificação (demarcado pelo símbolo "{") dispara o salvamento do elemento operacional de origem da mesma. Esse dado é utilizado a cada começo de análise de um novo ramo, sinalizado pelo símbolo "/", onde o contexto de avaliação não é mais provido pelo último elemento operacional examinado, mas sim por aquele que originou a ramificação. Por último, a conclusão de uma ramificação não terminal (indicada pelo símbolo "}" seguido de um elemento operacional) pode disparar uma multi-análise de políticas de transição. Essa operação acontece quando os últimos elementos operacionais de cada ramo estão alocados em, pelo menos, dois domínios distintos daquele onde o elemento de intersecção localiza-se.

O exame das políticas especificadas na requisição do serviço é executado sobre as distribuições de uma topologia e consiste de um processo eliminatório daquelas consideradas viáveis tecnicamente, mas que não condizem com as necessidades particulares, especificadas através de políticas, de clientes e/ou operadores de rede (*e.g.*, baixa latência, limitações de custo financeiro, distribuição de carga). Assim, em uma política da categoria imediata a exclusão de uma distribuição acontece quando a métrica utilizada por ela tem um valor informado pelo domínio em análise maior ou menor que, respectivamente, o limite superior (atributo MAXIMO) e inferior (atributo MINIMO) previamente definidos. Da mesma, a verificação desses limites também é executada na validação de políticas categorizadas como agregadas, porém, neste caso a comparação ocorre usando a variável de acúmulo da mesma.

Algumas das distribuições viáveis obtidas no exemplo padrão do capítulo são empregadas na exemplificação do procedimento de validação, tendo seus respectivos resultados passo a passo expostos na Figura 40. O processo identificado por *i* demonstra a ocorrência de uma invalidação dada a inconsistência entre o valor de teto previsto na política de agregação de custo financeiro e o total encontrado durante o exame da distribuição (*i.e.*, o valor financeiro total ex-

Figura 40 – Processos de Avaliação no Exemplo Padrão



cede o permitido). O processo ii também resulta na rejeição da distribuição examinada, neste caso ocasionado pela presença de uma transição que infringe a política imediata RTT. Finalmente, em iii a distribuição é reconhecida como candidata uma vez que não viola nenhuma das

três políticas especificadas. De maneira geral, o procedimento de verificação de políticas aplicado a topologia A conduz a 6 distribuições candidatas, removendo as demais devido violações da política imediata de RTT (4) e da política agregada de custo financeiro (2). Já a execução deste procedimento para as distribuições viáveis da topologia B retorna 3 classificadas como candidatas, descartando as outras 7 possíveis devido unicamente a existência de transições que descumprem a política imediata de RTT.

6.1.3 Avaliação de Distribuições Candidatas

O terceiro e último procedimento recebe como argumento um ou mais grupos de distribuições candidatas oriundas de topologias que provenham o mesmo serviço e seus resultados acumulados provenientes do último procedimento executado, além disso o IAT das topologias divididas e mapeadas são opcionalmente informados caso se deseje que estes sejam utilizados em um processo final de harmonização entre as etapas de composição e integração. A avaliação de distribuições candidatas, por sua vez, realiza o ranqueamento destas através da geração de um Índice de Adequabilidade de Distribuição (IAD). O IAD consiste de um valor numérico adimensional contido no intervalo $[0;1]$ onde a proximidade à borda inferior (*i.e.*, 0) indica a inadequabilidade proporcional de uma distribuição em relação as demais candidatas avaliadas sobre um mesmo grupo de políticas. De maneira oposta e sob as mesmas condições, a proximidade do IAD ao valor de borda superior (*i.e.*, 1) constata maior adequabilidade relativa de uma distribuição candidata, caracterizando assim uma função objetivo geral pela maximização do índice das distribuições candidatas (DC) presentes em um mesmo conjunto (CDC), como expresso na Equação 6.2.

$$MAX(IAD_{DC}^{CDC}), \forall DC \in CDC \quad (6.2)$$

O procedimento de avaliação de distribuições é dividido em três passos, sendo o segundo de realização opcional: geração do IAD, harmonização IAT/IAD e aconselhamento. O primeiro passo mapeia os processos de normalização, complementação, ponderação e geração de índice apresentados na primeira seção do Capítulo 5 aos valores e variáveis obtidos no bloco de políticas e nos resultados dos procedimentos de divisão e mapeamento previamente desempenhados, gerando assim o IAD de uma distribuição candidata. É importante ressaltar também que, apesar do mapeamento de constantes e variáveis realizado, esses processos herdam todas as premissas de execução e características operacionais já discutidas para eles. Após, o segundo passo ex-

plora a compatibilidade do intervalo de valores entre IAT e IAD para harmonizar os mesmos em um Índice Unificado (IU), demonstrando a relação custo/benefício entre duas etapas distintas da implantação (*i.e.*, composição e integração) de uma cadeia de função de serviço. Finalmente, o aconselhamento aplica uma função objetivo de maximização do IAD e do IU (caso requisitado) estabelecendo qual topologia e qual distribuição aplicada a ela são as mais recomendadas segundo os resultados obtidos pelas soluções de implantação.

Inicialmente, para a geração do IAD, os resultados acumulados de cada distribuição candidata são reorganizados de acordo com a categoria das políticas utilizadas. Sendo assim, políticas categorizadas como agregadas utilizam esses resultados na íntegra, enquanto políticas da categoria imediata usam a média aritmética entre estas agregações e a quantidade de elementos operacionais existentes nas topologias divididas e mapeadas (para políticas do tipo local) ou a quantidade de transições realizadas entre domínios (para políticas do tipo transição). Essa reorganização e pré-processamento dos resultados de acúmulo origina o conjunto de avaliações de distribuições candidatas (AVA) que é empregado no decorrer do processamento do primeiro passo do presente procedimento.

O mapeamento de constantes e variáveis de distribuição feito sobre as equações de normalização, complementação, ponderação e geração de índices do IAT segue as notações descritas na Tabela 14. Esse processo é iniciado pela determinação da distância absoluta (DA) máxima das métricas (mtc) utilizadas nas políticas especificadas e avaliadas para o conjunto de distribuições candidatas (CDC). Para isso, os resultados máximos e mínimos das avaliações são empregues como indicado na Equação 6.3, gerando um intervalo de distância absoluta $[0; DA_{mtc}^{CDC}]$. Em seguida, os mesmos resultados de avaliação das distribuições candidatas (DC) são mapeados individualmente nos respectivos intervalos de distância absoluta máxima de suas métricas através da operação explicitada pela Equação 6.4.

$$DA_{mtc}^{CDC} = \max(AVA_{mtc}^{CDC}) - \min(AVA_{mtc}^{CDC}) \quad (6.3)$$

$$DA_{mtc}^{DC} = AVA_{mtc}^{DC} - \min(AVA_{mtc}^{CDC}) \quad (6.4)$$

Em sequencia a realização do cálculo das distâncias absolutas, é realizado a transcrição dos mesmos para o intervalo $[0;1]$ através do processo de normalização. Dois cenários podem ocorrer durante a execução de tal processo, o primeiro detém um valor de distância absoluta máxima maior que zero e resulta na transcrição de intervalos através da aplicação da Equação 6.5. O segundo cenário acontece quando a distância absoluta máxima é igual a zero, o que signi-

Tabela 14 – Notações Utilizadas na Avaliação de Distribuições Candidatas

Notação	Descrição
CDC	Conjunto de distribuições candidatas gerado durante a validação de distribuições.
DC	Distribuição candidata composta por um conjunto de domínios para onde os elementos operacionais são destinados.
P_{mtc}	Peso referente a métrica "mtc".
AVA_{mtc}^{DC}	Resultado da avaliação referente a agregação (políticas agregadas) ou média por integração (políticas imediatas) da métrica "mtc" para a distribuição candidata "DC".
AVA_{mtc}^{CDC}	Conjunto de resultados das avaliações referentes a agregação (políticas agregadas) ou média por integração (políticas imediatas) da métrica "mtc" para todas as composições candidatas existentes em "CDC".
DA_{mtc}^{CDC}	Distância absoluta máxima entre os resultados de avaliação das distribuições candidatas existentes em "CDC" pela métrica "mtc".
$DA_{mtc}^{DC CDC}$	Distância absoluta máxima entre o resultado da avaliação da distribuição candidata "DC" e o conjunto de resultados de avaliação das distribuições candidatas existentes em "CDC" pela métrica "mtc".
$NORM_{mtc}^{DC CDC}$	Normalização do resultado da avaliação da distribuição candidata "DC" em relação ao conjunto de resultados de avaliação das distribuições candidatas existentes em "CDC" pela métrica "mtc".
$COMP_{mtc}^{DC CDC}$	Complemento da normalização da distribuição candidata "DC" em relação ao conjunto de resultados de avaliação das distribuições candidatas existentes em "CDC" pela métrica "mtc".
IAD_{DC}^{CDC}	Índice de adequabilidade de distribuição da distribuição candidata "DC" em relação ao conjunto de distribuições candidatas "CDC".

fica que todas as distribuições candidatas apresentam o mesmo resultado de avaliação para uma determinada métrica, nesta situação a Equação 6.6 é responsável pela determinação dos valores de normalização. Por fim, políticas que firmam preferência por distribuições que se aproximem seus resultados de piso (atributo MINIMO), expressa através da indicação de minimização no atributo de objetivo de seus respectivos objetos de especificação, têm suas normalizações complementadas (Equação 6.7) de modo a conciliarem-se ao objetivo final de maximização do IAD.

$$NORM_{mtc}^{DC CDC} = \frac{DA_{mtc}^{DC CDC}}{DA_{mtc}^{CDC}}, DA_{mtc}^{CDC} \neq 0 \quad (6.5)$$

$$NORM_{mtc}^{DC CDC} = 1, DA_{mtc}^{CDC} = 0 \quad (6.6)$$

$$COMP_{mtc}^{DC CDC} = 1 - NORM_{mtc}^{DC CDC}, \quad (6.7)$$

Como último processo, a ponderação das normalizações e/ou das complementações calculadas é executada atribuindo a relevância própria de uma política indica pelo seu peso (P) definido no atributo de ponderação presente em seus objetos de especificação. O somatório

dos resultados ponderados de uma distribuição candidata em específico é tomado como o seu, sendo este vinculado a mesma e retornado para posterior utilização em um possível passo de harmonização IAT/IAD e de aconselhamento. Considerando o exemplo padrão adotado neste capítulo, a função de cálculo do IAD das políticas cujo o objetivo geral é de minimização dos resultados de avaliação e que examinam as métricas de *Round Trip Time* imediato (rtti), o custo financeiro imediato (cfi) e o custo financeiro agregado (cfa) é determinada pela Equação 6.8.

$$IAD_{CDC}^{DC} = P_{rtti} * COMP_{rtti}^{DC CDC} + P_{cfi} * COMP_{cfi}^{DC CDC} + P_{cfa} * COMP_{cfa}^{DC CDC} \quad (6.8)$$

Após o cálculo do IAD de todas as distribuições candidatas, concluindo o primeiro passo, a harmonização IAT/IAD pode enfim ocorrer. Esse segundo passo é disparado somente se o IAT das topologias que foram divididas e mapeadas estiver disponível, neste caso o IU é calculado pela média aritmética entre ambos os índices como determina a Equação 6.9. O IU, por sua vez, expõem a relação custo/benefício entre uma topologia do serviço e a distribuição a ela aplicada. Isso se faz necessário já que nem sempre uma organização topológica melhor avaliada é capaz de gerar uma distribuição adequada ao conjunto de políticas requisitadas. Da mesma forma, uma boa distribuição pode ser encontrada para uma topologia inadequada considerando as métricas avaliadas durante a etapa de composição. Sendo assim, uma tupla topologia/distribuição com IU elevado pode não significar a utilização de ótimos locais encontrados em etapas individuais de implantação, mas sim aqueles resultados que proveem a relação mais harmoniosa entre as mesmas.

$$IU_{CC}^{DC} = (IAT_{CC} + IAD_{DC})/2 \quad (6.9)$$

O passo de desfecho da solução de divisão e mapeamento é feito através da interpretação e organização dos índices gerados durante a mesma. Com isso uma lista de ranqueamento é construída através do ordenamento decrescente de um índice, sendo a primeira entrada desta representante da melhor avaliação obtida. O processo de ranqueamento sempre é realizado para as distribuições candidatas, utilizando como guia os valores de IAD atribuídos a elas. Um segundo ranqueamento é também gerado e fornecido de maneira independente ao primeiro caso o cálculo do IU tenha sido executado. A distribuição que obtém o maior IAD e a harmonização com maior IU são utilizadas como os resultados principais deste procedimento, sendo elas aconselhadas pela solução desenvolvida. Entretanto, destaca-se que as demais possibilidades de

distribuição e seus respectivos índices são acessíveis pelos usuários, possibilitando a utilização destes em avaliações e harmonizações efetuadas por outras soluções que utilizem métodos de geração de índices de mesma grandeza em processos ou etapas da implantação de serviços de rede (como a integração em *hardware* e a seleção de serviços, ambas prevista no NFV-RA).

Tabela 15 – Processo de Avaliação de Distribuições Candidatas do Exemplo Padrão

Topologias	Distribuições Candidatas	Agregados/ Média	Norm.	Comp.	Pond.
EO1 < DOM3 > EO2 EO3 / EO2 EO4	(DOM3, DOM2, DOM1, DOM2, DOM2)	cfa: 800 cfi: 160 rtti: 13,333	cfa: 0,333 cfi: 0 rtti: 0,333	cfa: 0,667 cfi: 1 rtti: 0,667	cfa: 0,133 cfi: 0,3 rtti: 0,333
	(DOM3, DOM2, DOM2, DOM2, DOM1)	cfa: 800 cfi: 160 rtti: 13,333	cfa: 0,333 cfi: 0 rtti: 0,333	cfa: 0,667 cfi: 1 rtti: 0,667	cfa: 0,133 cfi: 0,3 rtti: 0,333
(EO1, EO2, EO3, EO2, EO4)	(DOM3, DOM2, DOM2, DOM3, DOM1)	cfa: 900 cfi: 180 rtti: 12,5	cfa: 1 cfi: 0,5 rtti: 0	cfa: 0 cfi: 0,5 rtti: 1	cfa: 0 cfi: 0,15 rtti: 0,5
	(DOM3, DOM3, DOM1, DOM2, DOM2)	cfa: 900 cfi: 180 rtti: 12,5	cfa: 1 cfi: 0,5 rtti: 0	cfa: 0 cfi: 0,5 rtti: 1	cfa: 0 cfi: 0,15 rtti: 0,5
	(DOM3, DOM2, DOM1, DOM3, DOM2)	cfa: 900 cfi: 180 rtti: 13,333	cfa: 1 cfi: 0,5 rtti: 0,333	cfa: 0 cfi: 0,5 rtti: 0,667	cfa: 0 cfi: 0,15 rtti: 0,333
	(DOM3, DOM3, DOM2, DOM2, DOM1)	cfa: 900 cfi: 180 rtti: 13,333	cfa: 1 cfi: 0,5 rtti: 0,333	cfa: 0 cfi: 0,5 rtti: 0,667	cfa: 0 cfi: 0,15 rtti: 0,333
	(DOM3, DOM3, DOM1, DOM2)	cfa: 750 cfi: 187,5 rtti: 12,5	cfa: 0 cfi: 0,687 rtti: 0	cfa: 1 cfi: 0,313 rtti: 1	cfa: 0,2 cfi: 0,093 rtti: 0,5
(EO2, EO1, EO3, EO4)	(DOM3, DOM3, DOM2, DOM1)	cfa: 750 cfi: 187,5 rtti: 12,5	cfa: 0 cfi: 0,687 rtti: 0	cfa: 1 cfi: 0,313 rtti: 1	cfa: 0,2 cfi: 0,093 rtti: 0,5
	(DOM3, DOM3, DOM2, DOM2)	cfa: 800 cfi: 200 rtti: 15	cfa: 0,333 cfi: 1 rtti: 1	cfa: 0,667 cfi: 0 rtti: 0	cfa: 0,133 cfi: 0 rtti: 0

Todos os resultados intermediários utilizados para o cálculo dos índices referentes as topologias e distribuições do exemplo padrão, além dos valores finais e ranqueamentos obtidos para os mesmos, são respectivamente exibidos nas Tabelas 15 e 16. Entre as distribuições avaliadas, o maior IAD alcançado foi de 0,793 e foi atribuído a duas distribuições candidatas detectadas para a topologia B. Por outro lado, o cálculo do IU realizado nas tuplas topologia/-distribuição resultou em um ranqueamento diferente do obtido considerando somente o IAD. Isso acontece em decorrência do valor reduzido do IAT da topologia B quando comparado a aquele informado como o da topologia A, produzindo uma relação menos harmoniosa entre a primeira e as suas distribuições candidatas. Sendo assim, o processo de aconselhamento em

Tabela 16 – Indexação e Ranqueamento das Distribuições Candidatas do Exemplo Padrão

Topologias	Distribuições Candidatas	IAD	Função Objetivo IAD	IU	Função Objetivo IU
EO1 < DOM3 >	(DOM3, DOM2, DOM1, DOM2, DOM2)	0,766	2º	0,783	1º
EO2 EO3 / EO2 EO4	(DOM3, DOM2, DOM2, DOM2, DOM1)	0,766	2º	0,783	1º
(EO1, EO2, EO3, EO2, EO4)	(DOM3, DOM2, DOM2, DOM3, DOM1)	0,65	3º	0,725	2º
	(DOM3, DOM3, DOM1, DOM2, DOM2)	0,65	3º	0,725	2º
	(DOM3, DOM2, DOM1, DOM3, DOM2)	0,483	4º	0,641	4º
	(DOM3, DOM3, DOM2, DOM2, DOM1)	0,483	4º	0,641	4º
EO2 EO1 < DOM3 > EO3 / EO4	(DOM3, DOM3, DOM1, DOM2)	0,793	1º	0,646	3º
	(DOM3, DOM3, DOM2, DOM1)	0,793	1º	0,646	3º
(EO2, EO1, EO3, EO4)	(DOM3, DOM3, DOM2, DOM2)	0,133	5º	0,316	5º

vista do IU aponta para duas distribuições candidatas recuperadas a partir da topologia A, ambas com valor de índice igual a 0,783. É importante ressaltar também que, tanto no resultado de aconselhamento do IAD quanto no do IU, o mesmo valor de índice é constatado em mais de uma distribuição ou tupla como consequência da aplicação de uma carga isonômica de utilização a um grupo de domínios em comum, além da previsão das mesmas transições entre eles, porém realizadas em uma sequência diferente.

6.2 AVALIAÇÃO DA SOLUÇÃO

Nesta seção são realizadas simulações de distribuição do serviço apresentado como caso de estudo na segunda seção do Capítulo 5 em um ambiente multi-domínio real cujo as informações alimentam a solução de divisão e mapeamento desenvolvida. Primeiramente, a Subseção 6.2.1 ressalta as políticas e características dos elementos operacionais que norteiam a distribuição do serviço de segurança, controle de conteúdo e balanceamento de carga, nela também são descritas a organização e o documento de especificação do ambiente distribuído e das métricas de avaliação colhidas em cada domínio, assim como as ferramentas utilizadas nesta tarefa. Após, os resultados provenientes da execução da solução de divisão e mapeamento são expostos e discutidos na Subseção 6.2.2. De maneira geral, os experimentos têm por objetivo evidenciar

adaptabilidade da técnica empregada na distribuição do serviço com diferentes configurações de políticas, estas definidas sob demanda e de maneira personalizada pelo seu usuário final.

6.2.1 Ferramental e Cenário de Teste

A experimentação proposta objetiva avaliar as capacidades da solução de divisão e mapeamento desenvolvida através da obtenção e discussão dos IAD e IU do serviço de teste descrito na segunda seção do Capítulo 5. Para isso, o ambiente real de virtualização multi-domínio, chamado *Federated Ecosystem for Offering, Distribution, and Execution of Virtual Network Functions* (FENDE), foi caracterizado através de um conjunto de métricas aplicáveis à especificação de políticas de integração. O FENDE²⁷ é um projeto apoiado pela Rede Nacional de Pesquisa (RNP) que objetiva criar um ecossistema direcionado a virtualização e distribuição de funções de rede, sua infraestrutura física atual consiste de três servidores alocados em domínios diferentes, denominados FendeSM, FendePOA e FendeCTBA, contando cada um com 8GB de memória RAM DDR4, além de processador Intel Xeon E3 1220 (FendeSM e FendePOA) ou Intel Core I7 6700 (FendeCTBA). O servidor FendeSM localiza-se na Universidade Federal de Santa Maria (cidade de Santa Maria no estado de Rio Grande do Sul), o servidor FendePOA está alocado na Universidade Federal do Rio Grande do Sul (cidade de Porto Alegre no estado de Rio Grande do Sul) e, por último, o servidor FendeCTBA opera na Universidade Federal do Paraná (cidade de Curitiba no estado de Paraná).

Três métricas foram colhidas e fornecidas para a especificação de políticas: *Round Trip Time* (RTT), saltos entre roteadores da rota padrão (Saltos) e custo financeiro para utilização dos serviços de controle de ciclo de vida (CustoCV). O RTT refere-se ao tempo total em milissegundos do percurso de ida e volta para um pacote enviado de um domínio atuando como emissor para outro representando o receptor. Os valores da métrica RTT foram definidos através da média dos resultados de quinhentas transmissões realizadas pela ferramenta ping²⁸. A métrica denominada Saltos apresenta a quantidade de saltos entre roteadores realizados para viabilizar a comunicação inter-domínios através das rotas padrões definidas entre eles, apresentando indícios do esforço de roteamento necessário para efetivar as comunicações, os valores que definem tal métrica foram obtidos por meio da ferramenta traceroute²⁹. Ambas as métricas de RTT e Saltos podem ser empregadas na especificação de políticas de transição.

²⁷ <http://coral.ufsm.br/gt-fende/index.html>

²⁸ <https://linux.die.net/man/8/ping>

²⁹ <https://linux.die.net/man/8/traceroute>

Tabela 17 – Matriz de Adjacências das Métricas Coletadas nos Domínios FENDE

	FendeSM	FendePOA	FendeCTBA
FendeSM	CustoCV = 3	RTT = 7,811 Saltos = 11	RTT = 21,721 Saltos = 13
FendePOA	RTT = 4,826 Saltos = 11	CustoCV = 7	RTT = 26,002 Saltos = 16
FendeCTBA	RTT = 29,332 Saltos = 14	RTT = 28,187 Saltos = 15	CustoCV = 5

A métrica CustoCV, por sua vez, apresenta a despesa mensal de locação das ferramentas para controle de ciclo de vida de VNFs para cada instância executando em uma infraestrutura FENDE. Entretanto, evidencia-se que os valores associados ao CustoCV são hipotéticos uma vez que estes ainda não foram estabelecidos pelos proprietários dos domínios estudados (BONDAN et al., 2018). Diferente das métricas RTT e Saltos, valores de custo financeiro são associados diretamente a um domínio, sendo adequados ao uso de políticas do tipo local. Os valores mensurados ou definidos para as métricas consideradas são apresentados na matriz de adjacências ilustrada pela Tabela 17, nela as métricas de transição são alocadas em células que representam intersecções entre domínios (*i.e.*, aquelas que não fazem parte da diagonal principal da matriz), já a métrica local é exposta em células que simbolizam um único domínio (*i.e.*, diagonal principal da matriz). Finalmente, o documento de especificação de domínios empregue para a realização dos testes é delineado nas Figuras 41 e 42.

Considerando a infraestrutura multi-domínio apresentada, três configurações de políticas são submetidas à avaliação da solução de divisão e mapeamento desenvolvida. A primeira configuração analisa exclusivamente a despesa agregada de integração de um serviço virtualizado considerando os valores presentes na métrica CustoCV. Sendo assim, um intervalo de valores aceitáveis (atributos MAXIMO e MINIMO), um objetivo (minimização, ou seja, preferência por valores próximos ao mínimo aceitável) e um peso (naturalmente 1 devido a sua singular presença no bloco) são definidos e aplicados na política. Já a segunda configuração foca na avaliação das conexões inter-domínios, neste caso as métricas RTT e Saltos são adotadas em políticas do tipo de transição e categorizadas como imediatas. As políticas utilizadas no segundo cenário compartilham do objetivo de minimização de seus resultados de avaliação e apresentam mesma relevância para a geração de índices (peso igual a 0,5). A última configuração, por sua vez, mescla as primeiras duas em um único bloco, mantendo todas as características das políticas a exceção dos seus pesos, sendo estes virtualmente iguais em cada uma delas (*i.e.*,

Figura 41 – Documento de Domínios FENDE (Início)

```

1 DOMINIOS:
2   - FendeSM
3   - FendePOA
4   - FendeCTBA
5 RECURSOS:
6   FendeSM:
7     MEMORIA: 7168
8     REDE: 10
9     CPU: 7
10  FendePOA:
11    MEMORIA: 7168
12    REDE: 10
13    CPU: 7
14  FendeCTBA:
15    MEMORIA: 7168
16    REDE: 8
17    CPU: 3
18 METRICAS_LOCAIS:
19 - ID: CustoCV
20   FendeSM: 2
21   FendePOA: 5
22   FendeCTBA: 3,5

```

Figura 42 – Documento de Domínios FENDE (Continuação)

```

23 METRICAS_TRANSICAO:
24 - ID: RTT
25   FendeSM:
26     FendePOA: 7,811
27     FendeCTBA: 21,721
28   FendePOA:
29     FendeSM: 4,826
30     FendeCTBA: 26,002
31   FendeCTBA:
32     FendeSM: 29,332
33     FendePOA: 26,187
34 - ID: Saltos
35   FendeSM:
36     FendePOA: 11
37     FendeCTBA: 13
38   FendePOA:
39     FendeSM: 11
40     FendeCTBA: 16
41   FendeCTBA:
42     FendeSM: 14
43     FendePOA: 15

```

0,3333 em RTT e Saltos e 0,3334 em CustoCV). Os intervalos de valores que determinam se uma distribuição cumpre ou não a política são definidos de forma a gerar pelo menos um resultado aceito em todas as topologias e configuração de políticas. Os blocos de requisição usados para especificar as configurações descritas são exibidos, pela mesma ordem que ocorrem no parágrafo, nas Figuras 43, 44 e 45.

Três topologias (A, B e C) que implementam o mesmo serviço de rede virtualizado são enviadas para serem divididas e mapeadas sob as configurações de políticas previamente apresentadas. A primeira topologia (A) manifesta um IAT de 0,666 e contém sete elementos operacionais distintos, estando dois deles alocados na única ramificação da cadeia (dois ramos contendo um elemento operacional cada). A topologia B ostenta um IAT de 0,5 e engloba nove elementos operacionais que executam um dos sete tipos de função de rede do serviço, uma ramificação também é encontrada na cadeia e nesta estão inseridos seis elementos operacionais, sendo três em cada ramo existente. Por fim, a terceira topologia (C) ostenta um IAT de 0,327 e contém oito elementos operacionais com dois deles executando uma mesma função de rede, além de uma ramificação que contempla quatro elementos designados dois a um ramo e dois a outro. A Figura 46 mostra a especificação SCAG, uma representação topológica e o IAT de todas as topologias testadas, já as Figuras 47 e 48 apresentam os objetos de requisitos computacionais mínimos localizados no bloco de implantação do serviço de teste.

Figura 43 – Primeira Configuração Bloco de Políticas do Serviço de Teste

```

1 POLITICAS:
2   IMEDIATAS: []
3   AGREGADAS:
4     - ID: CustoCV
5     PONDERACAO:
6       1,0
7     TIPO: LOCAL
8     MINIMO: 0
9     MAXIMO: 37
10    OBJETIVO: MIN

```

Figura 44 – Segunda Configuração Bloco de Políticas do Serviço de Teste

```

1 POLITICAS:
2   IMEDIATAS:
3     - ID: RTT
4     PONDERACAO:
5       0,5
6     TIPO:
7       TRANSICAO
8     MINIMO: 0
9     MAXIMO: 27
10    OBJETIVO: MIN
11   - ID: Saltos
12   PONDERACAO:
13     0,5
14   TIPO: LOCAL
15   MINIMO: 0
16   MAXIMO: 15
17   OBJETIVO: MIN
18   AGREGADAS: []

```

Figura 45 – Terceira Configuração Bloco de Políticas do Serviço de Teste

```

1 POLITICAS:
2   IMEDIATAS:
3     - ID: RTT
4     PONDERACAO:
5       0,3333
6     TIPO:
7       TRANSICAO
8     MINIMO: 0
9     MAXIMO: 27
10    OBJETIVO: MIN
11   - ID: Saltos
12   PONDERACAO:
13     0,3333
14   TIPO: LOCAL
15   MINIMO: 0
16   MAXIMO: 15
17   OBJETIVO: MIN
18   AGREGADAS:
19     - ID: CustoCV
20     PONDERACAO:
21       0,3334
22     TIPO: LOCAL
23     MINIMO: 0
24     MAXIMO: 37
25     OBJETIVO: MIN

```

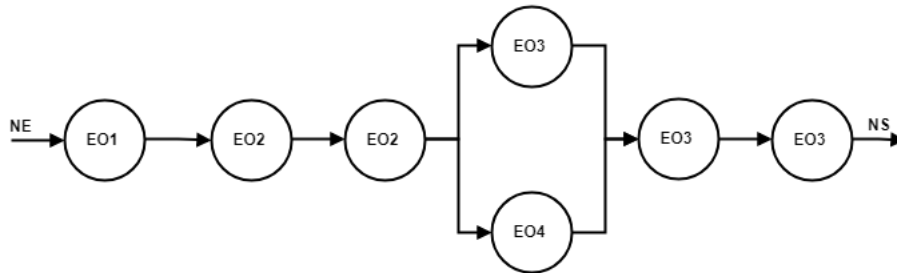
Os processos de distribuição e de avaliação aplicados as topologias testadas são detalhados na Subseção 6.2.2 através dos índices e resultados parciais gerados, ambos provenientes da execução dos procedimentos da solução de divisão e mapeamento desenvolvida e de seus respectivos passos. Individualmente, as distribuições que atingem os maiores IAD e IU locais em cada topologia e globais entre todas elas têm seus dados coletados também a cada procedimento da solução, possibilitando assim uma análise aprofundada das características esperadas para elas em relação as políticas e métricas examinadas. Todo esse conjunto de informações é discutido objetivando a validação operacional da solução proposta através da aplicação de um olhar crítico aos seus resultados em diferentes cenários e configurações de uso.

6.2.2 Resultados

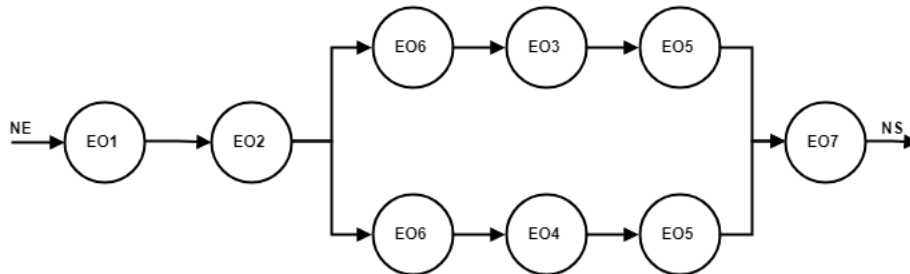
A avaliação da requisição e das topologias do serviço de teste e do documento de domínios expostos na subseção anterior pela solução de divisão e mapeamento desenvolvida retornou um total de 2061, 2044 e 401 distribuições candidatas resultantes do processamento da, respectivamente, primeira, segunda e terceira configuração de políticas como detalhado a Tabela 18.

Figura 46 – Topologias Analisadas do Serviço de Teste

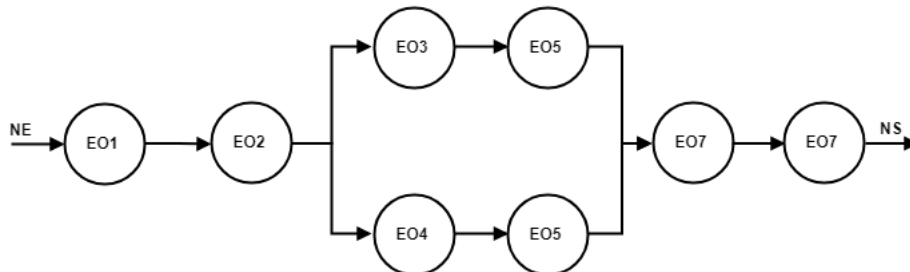
A) NE EO1 EO2 { EO3 / EO4 } EO5 EO7 NS IAT=0,666



B) NE EO1 EO2 { EO6 EO3 EO5 / EO6 EO4 EO5 } EO7 NS IAT=0,500



C) NE EO1 EO2 { EO3 EO5 / EO4 EO5 } EO6 EO7 NS IAT=0,327



O processamento realizado pela solução de divisão e mapeamento retorna o mesmo conjunto de distribuições para as três configurações de políticas nos primeiros três passos realizados (*i.e.*, permuta de domínios, checagem de conexões existentes e checagem de recursos computacionais), isto ocorre devido a persistência de características vinculadas as topologias, como a inexistência de dependências de infraestrutura e a estabilidade dos requisitos de recursos computacionais informados, e aos domínios, mantendo sempre as mesmas conexões inter-domínios declaradas através das métricas de transição.

O passo de exame de políticas é o que diferencia a coleção de resultados obtidos entre as diferentes configurações testadas. Em especial, na primeira configuração, cuja uma política de agregação é utilizada para desempenhar as operações de exame, a quantidade de distribuições

Figura 47 – Bloco de Implantação do Serviço de Teste (*Flavours* - Início)

```

1 IMPLANTACAO:
2 DESCRICAO_EOS:
3   EO1:
4     FLAVOUR:
5     MEMORIA: 512
6     REDE: 2
7     CPU: 1
8   EO2:
9     FLAVOUR:
10    MEMORIA: 512
11    REDE: 3
12    CPU: 1
13  EO3:
14    FLAVOUR:
15    MEMORIA: 512
16    REDE: 2
17    CPU: 1
18  EO4:
19    FLAVOUR:
20    MEMORIA: 512
21    REDE: 2
22    CPU: 1

```

Figura 48 – Bloco de Implantação do Serviço de Teste (*Flavours* - Continuação)

```

23   EO5:
24     FLAVOUR:
25     MEMORIA: 512
26     REDE: 2
27     CPU: 1
28   EO6:
29     FLAVOUR:
30     MEMORIA: 512
31     REDE: 2
32     CPU: 1
33   EO7:
34     FLAVOUR:
35     MEMORIA: 512
36     REDE: 3
37     CPU: 1

```

Tabela 18 – Quantidade de Distribuições do Serviço de Teste

	Primeira Configuração (CustoCV)			Segunda Configuração (RTT + Saltos)			Terceira Configuração (CustoCV + RTT + Saltos)		
	0,666	0,500	0,327	0,666	0,500	0,327	0,666	0,500	0,327
Topologia (IAT)	0,666	0,500	0,327	0,666	0,500	0,327	0,666	0,500	0,327
Permuta de Domínios	2187	19683	6561	2187	19683	6561	2187	19683	6561
Checagem de Conexões Existentes	2187	19683	6561	2187	19683	6561	2187	19683	6561
Checagem de Recursos Computacionais	1618	7980	3806	1618	7980	3806	1618	7980	3806
Exame de Políticas	1229	84	748	356	1045	643	292	5	104
Total		2061			2044			401	

candidatas é inversamente ligado ao tamanho da topologia examinada. Esse fenômeno acontece pois o teto de validação (37) é considerado baixo para dividir e mapear uma topologia com nove elementos operacionais (IAT 0,500) nos domínios disponíveis e, ao mesmo tempo, é relativamente alto para a realização da distribuição de uma topologia com sete elementos operacionais (IAT 0,666). Isso se confirma ao analisar a proporcionalidade de crescimento de acúmulo da métrica CustoCV a cada iteração avaliativa em uma distribuição viável onde, dado o intervalo de aceitação definido por [0;37], a adição de um elemento operacional na topologia representa no mínimo 8,1% (caso integrado em FendeSM) e no máximo 17% (caso integrado em FendePOA) do montante total permitido.

O número de distribuições candidatas recuperadas após o exame de políticas na segunda

configuração é, neste caso em específico, diretamente relacionado ao tamanho da topologia analisada. Nessa configuração o mesmo critério de exclusão é aplicado as distribuições viáveis de todas as topologias processadas, sendo ele originado através da comparação dos tetos de aceitação explicitados nas políticas RTT (27) e Saltos (15) com os valores definidos para suas métricas correspondentes no documento de especificação de domínios, impedindo assim a existência de transições originadas no FendeUFPR com destino ao FendeUFSM e originadas no FendePOA com destino ao FendeUFPR devido, respectivamente, a um RTT de 29,332 e a um Saltos de 16. O exame dessas políticas resulta em um número proporcionalmente inferior de distribuições candidatas identificadas entre as viáveis para as topologias de maior tamanho em relação as de menor. Isso ocorre como consequência dos elevados requisitos de recursos computacionais que induzem a realização mais frequente de transições entre domínios para a geração de distribuições viáveis. Entretanto, considerando o cenário utilizado, as maiores quantidades de distribuições candidatas ainda são obtidas a partir de topologias que acumulam mais elementos operacionais, situação ocasionada em razão do aumento exponencial de possibilidades retornadas pela permuta de domínios, além da extensiva utilização do FendeSM e FendePOA que contêm a maior parte dos recursos disponíveis (principalmente de processamento) e não apresentam nenhum impedimento de transição entre eles.

O exame de políticas aplicado a terceira configuração analisa as distribuições viáveis tanto pela política agregada CustoCV quanto pelas políticas imediatas RTT e Saltos, resultando no menor grupo de distribuições candidatas entre as configurações testadas. Como característica geral, essas distribuições estão também presentes nos conjuntos constituídos a partir do exame da primeira e segunda configurações, já que a terceira contém uma mescla de todas as políticas especificadas nas demais fazendo com que conjunto final retornado esteja necessariamente contido na intersecção entre aqueles obtidos nos respectivos exames anteriores.

Tabela 19 – Indexação das Melhores Distribuições Candidatas por Topologia do Serviço de Teste

	Primeira Configuração (CustoCV)			Segunda Configuração (RTT + Saltos)			Terceira Configuração (CustoCV + RTT + Saltos)		
Topologia (IAT)	0,666	0,500	0,327	0,666	0,500	0,327	0,666	0,500	0,327
Quantidade de Distribuições	1	84	6	3	3	3	1	3	1
IAD	1	0	0,583	1	0,972	1	0,908	0,463	0,683
IU	0,833	0,25	0,455	0,833	0,736	0,663	0,787	0,481	0,505

Todas as distribuições candidatas geradas foram qualificadas através de um IAD e as

tupla topologia/distribuição harmonizados pelo cálculo de seus respectivos IU, os melhores resultados obtidos em cada configuração processada, além da quantidade de distribuições que apresentam os mesmos índices, são demonstrados na Tabela 19. Ainda, os valores absolutos de acúmulo e/ou média das métricas avaliadas nas políticas obtidos pelas melhores distribuições de cada topologia na primeira, segunda e terceira configuração são, nesta ordem, expostos de maneira comparativa pelas Figuras 49, 50 e 51 e discutidos a seguir.

As melhores distribuições anotadas pelo processamento da primeira configuração de políticas nas diferentes topologias apresentam grande variação de IAD e IU. Nesse sentido, o melhor índice de distribuição global encontrado é 1 e foi verificado em uma única distribuição da topologia com IAT 0,666, ao mesmo tempo, o pior IAD (*i.e.*, 0) foi marcado em todas as distribuições candidatas da topologia com IAT 0,500, o que significa que o menor custo possível para a integração desta nos domínios disponíveis é o mais alto entre aqueles avaliados nas distribuições candidatas. Os lados opostos em que se encontram as topologias com IAT 0,666 e IAT 0,500 são refletidos nos dados de acúmulo na Figura 49, sendo que a primeira apresenta um valor absoluto de \$25 e a segunda de \$37 (teto do intervalo de aceitação da política). Ademais, os melhores resultados dessas topologias são intermediados pelas melhores distribuições provenientes da topologia com IAT 0,327 (\$30) que alcançam um IAD de 0,583. A harmonização dos índices topológicos e de distribuição aconselha o uso da melhor distribuição global (IU de 0,833) e ranqueia as melhores tuplas com a topologia de IAT 0,327 (IU 0,455) em posições superiores em relação as melhores tuplas onde a topologia com IAT 0,500 (IU 0,25) está presente.

Figura 49 – Agregações das Melhores Distribuições por Topologia (Primeira Configuração)

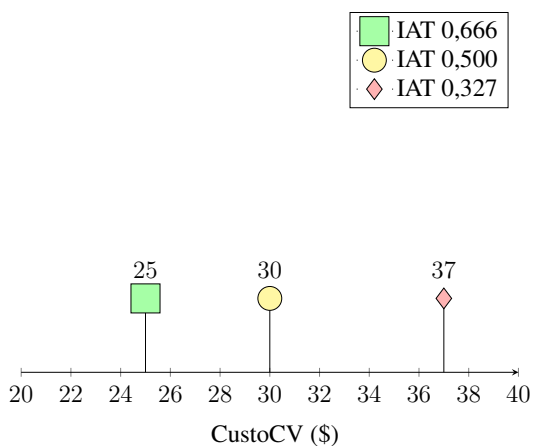
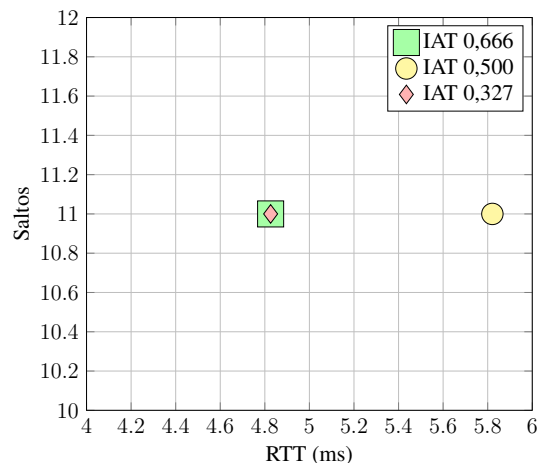
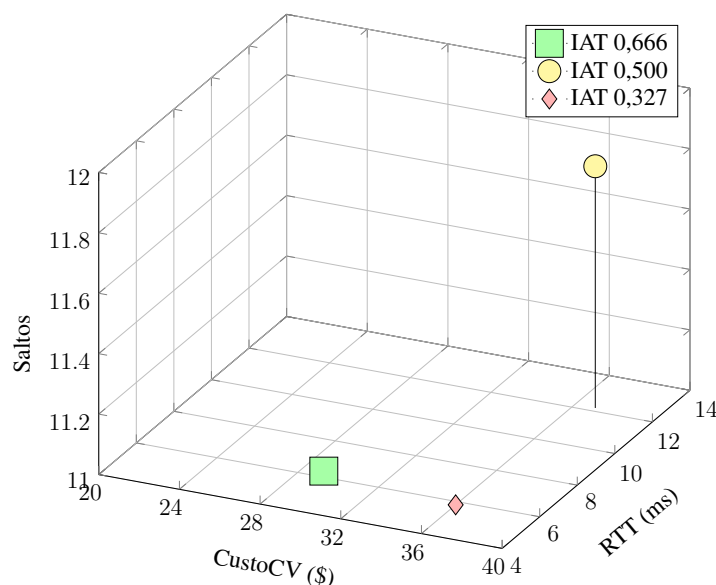


Figura 50 – Médias das Melhores Distribuições por Topologia (Segunda Configuração)



Já as melhores distribuições encontradas a partir da segunda configuração de políticas aplicada as topologias de teste apresentaram baixa variação nos índices de adequabilidade de distribuição, situação reflexo dos acúmulos e médias ilustradas na Figura 50. Esses resultados acontecem devido a realização exclusiva das transições que apresentam os menores valores possíveis de RTT e Saltos (do domínio FendePOA ao domínio FendeSM) nas topologias com IAT 0,666 e IAT 0,327, gerando as médias dos valores de acúmulo alcançáveis e, por consequência, o IAD máximo (1). Já as melhores distribuições da topologia com IAT 0,500 utilizam duas transições diferentes, originadas em FendePOA com destino a FendeSM e originadas em FendeSM com destino a FendePOA, com duas ocorrências de cada uma delas. Isso resulta no aumento da média de RTT devido ao maior valor definido para esta métrica na segunda transição, mas mantém o número de saltos constante uma vez que estes são iguais em todas as transições realizadas. Apesar disso, o IAD verificado para as distribuições da topologia com IAT 0,500 é apenas marginalmente reduzido em relação as demais devido a larga distância absoluta máxima entre as avaliações da métrica RTT (21,361 ms). Considerando as melhores distribuições de cada topologia, o processo de harmonização da segunda configuração de políticas reduz a diferença proporcional antes notada entre os IAT, porém este não tem como efeito colateral uma mudança na característica de ranqueamento das tuplas em relação ao realizado quando utilizado somente os índices topológicos (*i.e.*, maiores IAT alcançam maiores IU).

Figura 51 – Agregações e Médias das Melhores Distribuições por Topologia (Terceira Configuração)



A terceira configuração de políticas unifica a análise de todas as métricas definidas nos

domínios disponíveis. Com isso, maiores variações nos acúmulos e médias obtidas, ilustradas pela Figura 51, ocorrem em virtude da necessidade de conciliação entre as políticas requisitadas, já que elas podem ou não possuir correlação entre si. A distribuição apontada como globalmente mais adequada nessa configuração é associada a topologia com IAT 0,666 e apresenta um IAD de 0,908 originado através de um CustoCV de \$29, RTT médio de 6,318 ms e Saltos médio igual a 11. Da mesma forma, as melhores distribuições constatadas para as topologias com IAT 0,500 e IAT 0,327 manifestam, respectivamente, CustoCV de \$37 e \$36, RTT de 12,181 ms e 5,821 ms e Saltos igual a 11,8 e 11. Por fim, a harmonização realizada entre IAT e IAD novamente aponta uma tupla com a topologia de IAT 0,666 como a recomendada (IU 0,787), além de ranquear as melhores distribuições da topologia com IAT 0,327 em posições acima das melhores distribuições da topologia de IAT 0,500 por motivos análogos aos explicitados nos cenários anteriores.

Os resultados observados demonstram que a solução desenvolvida apresenta o comportamento esperado para a distribuição das topologias de teste, isto é, independentemente da configuração de políticas adotada, o procedimento de distribuição de topologias cujo realiza os passos de permuta de domínios, checagem de conexões e checagem de recursos computacionais retorna o mesmo conjunto de composições viáveis considerando apenas a topologia requisitada e o documento de domínios provido. O segundo procedimento é aquele que diferencia o grupo de distribuições gerado referente a uma topologia entre as configurações de políticas analisadas. Nele uma distribuição viável é categorizada como candidata quando apresenta valores dentro do intervalo de aceitação para todas as métricas de avaliação utilizadas. Sendo assim, o conjunto final gerado equivale a intersecção do conjunto de distribuições aceitas durante a validação individual de cada política. Em conclusão, o índice que unificou a avaliação das etapas de composição e de integração por divisão e mapeamento apontou as melhores distribuições da topologia com IAT 0,666 como as mais adequadas em todas as configurações de políticas testadas, colocando as topologias com IAT 0,500 e IAT 0,327 como as que apresentaram a segunda opção de distribuições mais harmônicas em, respectivamente, uma (segunda) e duas (primeira e terceira) dessas configurações.

6.3 SUMARIZAÇÃO E DISCUSSÃO

A etapa de integração objetiva alocar uma topologia SFC previamente composta no substrato físico disponível. Em especial, quando múltiplos domínios e provedores de infra-

estrutura estão disponíveis, técnicas como a de divisão e mapeamento podem ser empregadas para distribuir uma cadeia de serviço. A técnica de divisão e mapeamento já foi explorada em trabalhos como (DIETRICH; ABUJODA; PAPADIMITRIOU, 2015; ABUJODA; PAPADIMITRIOU, 2016; ZHANG et al., 2016; RIERA et al., 2016; WANG et al., 2017), porém as ferramentas desenvolvidas por eles suportam apenas a definição e análise de políticas que se convertem em resultados binários de aceitação ou não de uma distribuição de topologia. Após, para as distribuições aceitas, uma avaliação ocorre considerando um conjunto predefinido e imutável de métricas (*e.g.*, custos financeiros, uso de recursos computacionais, balanceamento de carga). Dessa forma, o operador de rede não é capaz de especificar e avaliar de maneira personalizada métricas e políticas que melhor se adéquem as necessidades do serviço requisitado e do ambiente distribuído ao qual ele será integrado.

Este trabalho propõe uma nova solução de integração por divisão e mapeamento com suporte a customização dos processos de avaliação de políticas. Essa solução executa uma busca exaustiva pelo resultado ótimo e faz uso da mesma metodologia de indexação aplicada na solução de composição apresentada no Capítulo 5. Com isso, todas as distribuições válidas encontradas para as composições candidatas analisadas são avaliadas, gerando índices (IAD) que as caracterizam individualmente em relação ao conjunto de políticas requisitado. Como experimentação, métricas recuperadas de um ambiente multi-domínio real (FENDE) foram usadas na distribuição de três composições candidatas de um serviço de segurança, controle de conteúdo e balanceamento de carga para fluxos HTTP e HTTPS. Os testes realizados detalharam o comportamento da solução, além de retornarem os índices alcançados para as distribuições (IAD) e para as conciliações (*i.e.*, relação custo/benefício) entre os índices topológicos e de distribuição (IU). Por fim, da mesma forma que acontece para a solução de composição proposta anteriormente, heurísticas e podas podem ser utilizadas em substituição ao método de força bruta para, desta forma, reduzir o espaço de busca e/ou o tempo de execução do processo de divisão e mapeamento.

7 CONCLUSÃO

Esta dissertação propôs um arcabouço holístico de modelos, metodologias e soluções que viabilizam a implantação de serviços de rede virtualizados através do processamento de multi-requisitos customizados pelos seus usuários. Com isso, modelos de SFC e de SFCR foram definidos para flexibilizar as possibilidades de especificação tanto das organizações topológicas de um serviço, quanto dos requisitos de implantação referentes a ele. Utilizando esses modelos, duas soluções que abordam as etapas de composição e integração multi-domínio de serviços de rede foram desenvolvidas. Essas soluções adotam uma metodologia genérica de avaliação e harmonização dos resultados de implantação baseada em indexadores, também criada como parte deste trabalho. Finalmente, um serviço de teste que provê segurança, controle de conteúdo e balanceamento de carga para servidores HTTP e HTTPS foi implementado e usado para avaliar as soluções de composição e integração propostos.

O arcabouço construído provê aos operadores de rede a capacidade de personalizar as avaliações realizadas por soluções que implementam etapas de implantação de um serviço de rede virtualizado (composição, integração e agendamento de execução). Com isso, é possível orientar a geração de resultados dessas soluções a métricas de interesse (*e.g.*, latência, uso de recursos computacionais, custos financeiros, número de conexões suportadas) e objetivos particulares (*e.g.*, minimização, maximização) de diferentes cenários e usuários, sendo ambos analisados através de operações também customizáveis (*e.g.*, operações matemáticas, funções, ponderações). Ainda, novos modelos de especificação de topologias SFC e de requisições foram formalizados com o intuito de facilitar a configuração de um processo de implantação flexibilizado. Assim, um operador de rede pode expressar suas necessidades (funções objetivo ou políticas) por meio de documentos estruturados, padronizados, de fácil compreensão humana e completamente agnósticos ao algoritmo da solução escolhida.

Um conjunto de experimentos foi realizado sobre um serviço de teste e consistiram na especificação topológica geral deste serviço utilizando o modelo de SFC formalizado pela gramática SCAG, na criação de requisições com diferentes configurações de políticas e métricas de análise de acordo com o modelo de SFCR proposto e, finalmente, no seu processamento pelas soluções de composição e integração (técnica de divisão e mapeamento) desenvolvidas. Os resultados obtidos na etapa de composição foram validados a partir da instanciação de diferentes composições candidatas em um ambiente controlado, demonstrando um comportamento

de execução como o esperado levando em consideração as avaliações que formaram os seus índices de adequabilidade. Em um segundo momento, dados que caracterizam um ambiente multi-domínio real, denominado FENDE, foram coletados e empregados pela solução de integração em um subconjunto das topologias do serviço de teste previamente compostas. As distribuições candidatas geradas foram minuciosamente examinadas e discutidas a cada etapa de validação e avaliação, garantindo o comportamento operacional esperado para a solução.

Como trabalhos futuros pretende-se expandir as capacidades do modelo topológico para suportar de forma explícita características como heterogeneidade (considerando as funções de rede) e simetria (do ponto de vista das conexões entre as funções de rede). Quanto a solução de composição desenvolvida, objetiva-se generalizar as funções de avaliação, atualização e divisão previstas através da definição de equações mais complexas que empregam uma ou mais entradas entre múltiplas e diversificadas operações matemáticas. Também, o passo de redução de ramificações do procedimento de expansão de topologias será aprimorado para computar possíveis unificações intermediárias presentes em uma ramificação. Finalmente, na solução de integração multi-domínio, um novo módulo será adicionado com o intuito de analisar dados que caracterizam os clientes que utilizam o serviço requisitado e as suas relações com os domínios disponíveis (*e.g.*, categorias de clientes e atraso de transmissão entre clientes e domínios), possibilitando assim o uso de técnicas de QoS, como a priorização de domínios e replicação de segmentos do serviço, durante a divisão e o mapeamento do mesmo.

REFERÊNCIAS

- ABUJODA, A.; PAPADIMITRIOU, P. DistNSE: distributed network service embedding across multiple providers. In: COMMUNICATION SYSTEMS AND NETWORKS (COMSNETS), 2016 8TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2016. p.1–8.
- ALAN ANTHONY SHIHABUR CHOWDHURY, T. B. R. B. J. F. UNiS: a user-space non-intrusive workflow-aware virtual network function scheduler. In: IEEE/ACM/IFIP CONFERENCE ON NETWORK AND SERVICE MANAGEMENT (CNSM) 2018. **Anais...** [S.l.: s.n.], 2018.
- ALLYBOKUS, Z. et al. Virtual function placement for service chaining with partial orders and anti-affinity rules. **Networks**, [S.l.], v.71, n.2, p.97–106, 2018.
- BAEK, H. et al. Order dependency-aware service function placement in service function chaining. In: INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2017 INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2017. p.193–195.
- BERNARDOS, C. J. et al. **Network Virtualization Research Challenges**. [S.l.]: Internet Engineering Task Force, 2018. Internet Draft.
- BJÖRKLUND, M. **The YANG 1.1 Data Modeling Language - RFC 7950**. [S.l.]: Internet Engineering Task Force, 2016. Request for Comments.
- BLANCO, B. et al. Technology pillars in the architecture of future 5G mobile networks: nfv, mec and sdn. **Computer Standards & Interfaces**, [S.l.], v.54, p.216–228, 2017.
- BONDAN, L. et al. FENDE: marketplace and federated ecosystem for the distribution and execution of vnfs. In: ACM SIGCOMM 2018 CONFERENCE ON POSTERS AND DEMOS. **Anais...** [S.l.: s.n.], 2018. p.135–137.
- CARPIO, F.; DHAHRI, S.; JUKAN, A. VNF placement with replication for Load balancing in NFV networks. In: COMMUNICATIONS (ICC), 2017 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2017. p.1–6.
- DEARIE, A. Software deployment, past, present and future. In: FUTURE OF SOFTWARE ENGINEERING, 2007. FOSE'07. **Anais...** [S.l.: s.n.], 2007. p.269–284.

DIETRICH, D.; ABUJODA, A.; PAPADIMITRIOU, P. Network service embedding across multiple providers with nestor. In: IFIP NETWORKING CONFERENCE (IFIP NETWORKING), 2015. **Anais...** [S.l.: s.n.], 2015. p.1–9.

DING, W. et al. OpenSCaaS: an open service chain as a service platform toward the integration of sdn and nfv. **IEEE Network**, [S.l.], v.29, n.3, p.30–35, 2015.

DRÄXLER, S.; KARL, H. Specification, composition, and placement of network services with flexible structures. **International Journal of Network Management**, [S.l.], v.27, n.2, p.1963:1–20, 2017.

ENNS, R. et al. **Network Configuration Protocol (NETCONF) - RFC 6241**. [S.l.]: Internet Engineering Task Force, 2011.

FARACI, G.; LOMBARDO, A.; SCHEMBRA, G. A Processor-sharing Scheduling Strategy for NFV Nodes. **Journal of Electrical and Computer Engineering**, [S.l.], v.2016, p.1:1–1:1, 2016.

FUNG, C. J.; MCCORMICK, B. VGuard: a distributed denial of service attack mitigation method using network function virtualization. In: INTERNATIONAL CONFERENCE ON NETWORK AND SERVICE MANAGEMENT (CNSM), 2015. **Anais...** [S.l.: s.n.], 2015. p.64–70.

GARAY, J. et al. Service description in the NFV revolution: trends, challenges and a way forward. **IEEE Communications Magazine**, [S.l.], v.54, n.3, p.68–74, 2016.

GARCIA, V. F. et al. DeMONS: a ddos mitigation nfv solution. In: THE 32ND IEEE INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS (AINA), 2018. **Anais...** [S.l.: s.n.], 2018. p.769–776.

GIL-HERRERA, J.; BOTERO, J. F. A scalable metaheuristic for service function chain composition. In: COMMUNICATIONS (LATINCOM), 2017 IEEE 9TH LATIN-AMERICAN CONFERENCE ON. **Anais...** [S.l.: s.n.], 2017. p.1–6.

GSNFV, E. **Network functions virtualization (NFV): architectural framework**. [S.l.]: European Telecommunications Standards Institute, 2013.

GU, S. et al. An efficient auction mechanism for service chains in the NFV market. In: INFO-COM 2016-THE 35TH ANNUAL IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS, IEEE. **Anais...** [S.l.: s.n.], 2016. p.1–9.

HALPERN, J.; PIGNATARO, C. **Service Function Chaining (SFC) Architecture - RFC 7665**. [S.l.]: Internet Engineering Task Force, 2015. Request for Comments.

HANDLEY, M. Why the Internet Only Just Works. **BT Technology Journal**, [S.l.], v.24, n.3, p.119–129, 2006.

HAZAS, M. et al. Are there limits to growth in data traffic?: on time use, data generation and speed. In: WORKSHOP ON COMPUTING WITHIN LIMITS. **Anais...** [S.l.: s.n.], 2016. p.14–18.

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

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

KULKARNI, S. G. et al. Nfvnice: dynamic backpressure and scheduling for nfV service chains. In: ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION. **Anais...** [S.l.: s.n.], 2017. p.71–84.

KWAK, H.-H. et al. Symbolic schedulability analysis of real-time systems. In: IEEE REAL-TIME SYSTEMS SYMPOSIUM, 19. **Anais...** [S.l.: s.n.], 1998. p.409–418.

LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop: rapid prototyping for software-defined networks. In: ACM SIGCOMM WORKSHOP ON HOT TOPICS IN NETWORKS, 9. **Anais...** [S.l.: s.n.], 2010. p.19:1 – 19:6.

LIU, J. et al. On dynamic service function chain deployment and readjustment. **IEEE Transactions on Network and Service Management**, [S.l.], v.14, n.3, p.543–553, 2017.

LYON, G. F. **Nmap network scanning**: the official nmap project guide to network discovery and security scanning. [S.l.]: Insecure, 2009.

MANSF, E. **Network Function Virtualisation (NFV)**: management and orchestration. [S.l.]: European Telecommunications Standards Institute, 2014.

MARCUZZO, L. d. C. et al. Click-on-OSv: a platform for running click-based middleboxes. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 2017. **Anais...** IEEE, 2017. p.885–886.

MARTINS, J. et al. ClickOS and the art of network function virtualization. In: USENIX CONFERENCE ON NETWORKED SYSTEMS DESIGN AND IMPLEMENTATION, 11. **Anais...** [S.l.: s.n.], 2014. p.459–473.

MEDHAT, A. M. et al. Near optimal service function path instantiation in a multi-datacenter environment. In: INTERNATIONAL CONFERENCE ON NETWORK AND SERVICE MANAGEMENT (CNSM), 2015. **Anais...** [S.l.: s.n.], 2015. p.336–341.

MEHRAGHDAM, S.; KELLER, M.; KARL, H. Specifying and placing chains of virtual network functions. In: CLOUD NETWORKING (CLOUDNET), 2014 IEEE 3RD INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.7–13.

MIJUMBI, R. et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions. In: NETWORK SOFTWAREZATION (NETSOFT), 2015 1ST IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2015. p.1–9.

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

MOENS, H.; DE TURCK, F. VNF-P: a model for efficient placement of virtualized network functions. In: NETWORK AND SERVICE MANAGEMENT (CNSM), 2014 10TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.418–423.

NFVISG, E. **Network Functions Virtualization**: white paper. [S.l.]: European Telecommunications Standards Institute, 2012.

NFVISG, E. **Network Function Virtualisation (NFV)**: architectural framework. [S.l.]: European Telecommunications Standards Institute, 2013.

NFVISG, E. **Network Function Virtualisation (NFV)**: terminology for main concepts in nfv. [S.l.]: European Telecommunications Standards Institute, 2014.

OCAMPO, A. F. et al. Optimal Service Function Chain Composition in Network Functions Virtualization. In: IFIP INTERNATIONAL CONFERENCE ON AUTONOMOUS INFRASTRUCTURE, MANAGEMENT AND SECURITY. **Anais...** [S.l.: s.n.], 2017. p.62–76.

QUINN, P.; ELZUR, U.; PIGNATARO, C. **Network Service Header (NSH) - RFC 8300**. [S.l.]: Internet Engineering Task Force, 2018. Request for Comments.

QUINN, P.; NADEAU, T. **Problem Statement for Service Function Chaining - RFC 7498**. [S.l.]: Internet Engineering Task Force, 2015. Request for Comments.

RIERA, J. F. et al. Modelling the NFV forwarding graph for an optimal network service deployment. In: INTERNATIONAL CONFERENCE ON TRANSPARENT OPTICAL NETWORKS (ICTON), 2015. **Anais...** [S.l.: s.n.], 2015. p.1–4.

RIERA, J. F. et al. TeNOR: steps towards an orchestration platform for multi-pop nfv deployment. In: NETSOFT CONFERENCE AND WORKSHOPS (NETSOFT), 2016 IEEE. **Anais...** [S.l.: s.n.], 2016. p.243–250.

ROSA, R. V.; ROTHENBERG, C. E.; SZABO, R. VBaaS: vnf benchmark-as-a-service. In: SOFTWARE DEFINED NETWORKS (EWSN), 2015 FOURTH EUROPEAN WORKSHOP ON. **Anais...** [S.l.: s.n.], 2015. p.79–84.

SCHONWALDER, J.; BJORKLUND, M.; SHAFER, P. Network configuration management using NETCONF and YANG. **IEEE communications magazine**, [S.l.], v.48, n.9, p.166–173, 2010.

SHEN, W. et al. vConductor: an enabler for achieving virtual network integration as a service. **IEEE Communications Magazine**, [S.l.], v.53, n.2, p.116–124, 2015.

SHERRY, J. et al. Making Middleboxes Someone else's Problem: network processing as a cloud service. **SIGCOMM Comput. Commun. Rev.**, [S.l.], v.42, n.4, p.13–24, 2012.

SHI, W.; DUSTDAR, S. The Promise of Edge Computing. **Computer**, [S.l.], v.49, n.5, p.78–81, 2016.

SHIN, M. K. et al. Verification for NFV-enabled network services. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2015. **Anais...** [S.l.: s.n.], 2015. p.810–815.

SUN, C. et al. Nfp: enabling network function parallelism in nfv. In: CONFERENCE OF THE ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION. **Anais...** [S.l.: s.n.], 2017. p.43–56.

TAYLOR, D.; TURNER, J. **Towards a diversified internet**. [S.l.]: Washington University, 2004.

VAQUERO, L. M.; RODERO-MERINO, L. Finding Your Way in the Fog: towards a comprehensive definition of fog computing. **SIGCOMM Comput. Commun. Rev.**, [S.l.], v.44, n.5, p.27–32, 2014.

VU, A.-V.; DINH, N.-T.; KIM, Y. Modeling of Service Function Chaining in Network Function Virtualization Environment. In: KOREAN INSTITUTE OF COMMUNICATION SCIENCES CONFERENCE. **Anais...** [S.l.: s.n.], 2016. p.1100–1101.

WANG, Y. et al. Enabling automatic composition and verification of service function chain. In: QUALITY OF SERVICE (IWQOS), 2017 IEEE/ACM 25TH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2017. p.1–5.

WANG, Y. et al. Cost-efficient virtual network function graph (vNFG) provisioning in multi-domain elastic optical networks. **Journal of Lightwave Technology**, [S.l.], v.35, n.13, p.2712–2723, 2017.

XILOURIS, G. et al. T-NOVA: a marketplace for virtualized network functions. In: EUROPEAN CONFERENCE ON NETWORKS AND COMMUNICATIONS (EUCNC), 2014. **Anais...** [S.l.: s.n.], 2014. p.1–5.

ZHANG, Q. et al. Vertex-centric computation of service function chains in multi-domain networks. In: NETSOFT CONFERENCE AND WORKSHOPS (NETSOFT), 2016 IEEE. **Anais...** [S.l.: s.n.], 2016. p.211–218.

ZHANG, W. et al. OpenNetVM: a platform for high performance network service chains. In: WORKSHOP ON HOT TOPICS IN MIDDLEBOXES AND NETWORK FUNCTION VIRTUALIZATION, 2016. **Anais...** [S.l.: s.n.], 2016. p.26–31.

ZHANG, Y. et al. Parabox: exploiting parallelism for virtual network functions in service chaining. In: SYMPOSIUM ON SDN RESEARCH. **Anais...** [S.l.: s.n.], 2017. p.143–149.

ZHENG, Z. et al. GEN: a gpu-accelerated elastic framework for nfv. In: ASIA-PACIFIC WORKSHOP ON NETWORKING, 2. **Anais...** [S.l.: s.n.], 2018. p.57–64.