



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Gabriel Fontes Carvalho de Queiroz

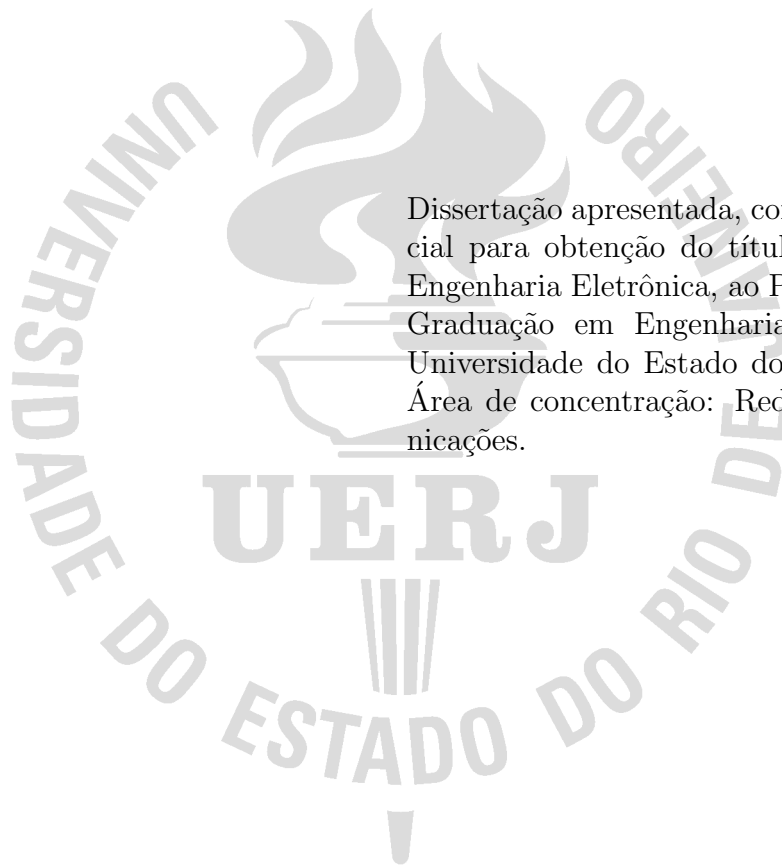
**TRELIS: Posicionamento de Funções Virtuais de Rede com
Economia de Energia e Resiliência**

Rio de Janeiro

2017

Gabriel Fontes Carvalho de Queiroz

TRELIS: Posicionamento de Funções Virtuais de Rede com Economia de Energia e Resiliência



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Engenharia Eletrônica, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Redes de Telecomunicações.

Orientador: Prof. D.Sc. Rodrigo de Souza Couto

Orientador: Prof. D.Sc. Alexandre Sztajnberg

Rio de Janeiro

2017

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

Q3 Queiroz, Gabriel Fontes Carvalho de.
TRELIS: posicionamento de funções virtuais de rede com economia de energia e resiliência / Gabriel Fontes Carvalho de Queiroz. - 2017.
79 f.

Orientadores: Rodrigo de Souza Couto e Alexandre Sztajnberg.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica. 2. Redes de telecomunicações - Alta velocidade - Dissertações. 3. Alocação de recursos - Dissertações. 4. Virtualização - Dissertações. 5. Programação heurística – Dissertações. I. Couto, Rodrigo de Souza. II. Sztajnberg, Alexandre. III. Universidade do Estado do Rio de Janeiro. IV. Título.

CDU 621.38+39

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

Assinatura

Data

Gabriel Fontes Carvalho de Queiroz

TRELIS: Posicionamento de Funções Virtuais de Rede com Economia de Energia e Resiliência

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Engenharia Eletrônica, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Redes de Telecomunicações.

Aprovada em 22 de Fevereiro de 2017.

Banca Examinadora:

Prof. D.Sc. Rodrigo de Souza Couto (Orientador)
DETEL/PEL/UERJ

Prof. D.Sc. Alexandre Sztajnberg (Orientador)
PEL/UERJ

Prof. D.Sc. Marcelo Gonçalves Rubinstein
DETEL/PEL/UERJ

Prof. Dr. Luís Henrique Maciel Kosmowski Costa
POLI/COPPE/UFRJ

Rio de Janeiro

2017

DEDICATÓRIA

Dedico este trabalho a Deus em primeiro lugar, que permitiu que eu chegasse até onde cheguei e conquistasse o que conquistei. A meus pais, meus orientadores, aos professores que me ensinaram durante a vida toda e a todos aqueles que me ajudaram e estiveram ao meu lado nos momentos mais difíceis e ali permaneceram para comemorar nos momentos de felicidade.

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter sido de sua vontade que mais esta etapa fosse concluída e por todas as experiências adquiridas ao longo desse curso. Agradeço à inestimável compreensão e apoio de meus pais por sempre me proporcionarem o esteio necessário para que eu tivesse condições de realizar o curso de mestrado. Agradeço aos meus orientadores por me ensinarem a trilhar o caminho da pesquisa e do estudo e fornecerem o conhecimento, os ensinamentos e o incentivo que me fizeram crescer pessoal e profissionalmente. Agradeço aos membros da banca examinadora que aceitaram dedicar seu tempo para analisar esta dissertação e contribuíram para o sucesso da mesma. Agradeço a todos os professores que tive em minha vida, cujos ensinamentos moldaram e lapidaram o meu caráter e me fizeram ser quem eu sou hoje. Agradeço ao apoio dos colegas de classe, que trilharam esse caminho de estudos comigo. Agradeço às agências de fomento envolvidas pelo suporte primordial para que esse trabalho fosse possível.

Foi o tempo que dedicaste à tua rosa que fez a tua rosa tão importante.

Antoine de Saint-Exupéry

RESUMO

QUEIROZ, G. F. C. *TRELIS: Posicionamento de Funções Virtuais de Rede com Economia de Energia e Resiliência*. 2017. 79 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

As redes tradicionais de telecomunicações utilizam intensivamente *middleboxes* dedicados, que são dispositivos construídos especificamente para fornecer um determinado serviço, como tradução de endereços, *firewall* e *proxy*. Esses dispositivos dedicados são custosos e tornam a infraestrutura rígida, pouco escalável e de difícil gerenciamento. Nesse cenário, a Virtualização de Funções de Rede surge para aumentar a flexibilidade e reduzir custos de infraestruturas de telecomunicações. Para tal, as funções de rede são implementadas em servidores virtualizados, de uso geral, ao invés de em equipamentos dedicados. Um dos desafios dessa abordagem é o posicionamento de funções virtuais de rede na infraestrutura, que deve ser resiliente e energeticamente eficiente. Esta dissertação formula um problema de programação inteira mista para posicionar funções virtuais de rede, escolhendo os servidores para atender às demandas de serviço. O problema formulado minimiza o uso da energia e provê resiliência aos serviços de rede através do compartilhamento das funções de rede e da replicação de funções, respectivamente. Para resolver esse problema, propõe-se a heurística TRELIS, que reduz o tamanho do problema e economiza até 35% da energia ao usar o compartilhamento de funções de rede. Os resultados mostram que a TRELIS consegue solucionar um problema quatro vezes maior que o tamanho máximo alcançado através da solução ótima em apenas um quarto do tempo.

Palavras-chave: NFV. Alocação de recursos. Resiliência.

ABSTRACT

QUEIROZ, G. F. C. *TRELIS: Virtual Network Functions Placement with Energy Saving and Resilience*. 2017. 79 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

Traditional telecommunication networks intensively employ dedicated middleboxes, which are specifically built to provide a given service, such as address translation, firewall and proxy. These dedicated devices are expensive and make the infrastructure rigid, hardly scalable and difficult to manage. In this scenario, Network Function Virtualization arises to increase flexibility and to reduce the costs of telecommunication infrastructures. To do so, network functions are implemented on virtualized generic servers, rather than on dedicated equipment. One of the challenges of this approach is the placement of virtual network functions over the infrastructure, which must be resilient and energy-efficient. This work formulates a mixed integer programming problem to place virtual network functions, choosing the servers meeting service demands. The formulated problem minimizes energy utilization and provides resilience to network services through network function sharing and function replication, respectively. To solve this problem, TRELIS heuristic is proposed, which reduces the size of the problem and saves up to 35% of the energy by employing the network function sharing. The results show that TRELIS is able to solve a problem four times bigger than the maximum size reached through the optimal solution in only a quarter of the time.

Keywords: NFV. Resource allocation. Resilience.

LISTA DE FIGURAS

Figura 1 - Arquitetura NFV.	20
Figura 2 - Componentes internos do MANO.	22
Figura 3 - Exemplos de composição de diferentes serviços.	24
Figura 4 - Compartilhamento de VNFs pelas SFCs.	26
Figura 5 - Mapeamento de uma rede virtual no substrato.	27
Figura 6 - Diferenças entre o mapeamento em VNE e o posicionamento em NFV.	28
Figura 7 - Redes tradicionais e redes SDN.	29
Figura 8 - Arquitetura SDN.	31
Figura 9 - Elementos de um SRG são afetados pela mesma falha.	41
Figura 10 - Exemplo de uma SFC qualquer com replicação de VNFs.	43
Figura 11 - Quebra na SFC quando a VNF principal e o <i>backup</i> estão no mesmo servidor.	49
Figura 12 - Quebra na SFC quando os quatro enlaces caem ao mesmo tempo.	52
Figura 13 - Quebra na SFC por uma descontinuidade espelhada.	53
Figura 14 - Remoção de ciclos e seleção de nós em uma linha feita pela TRELIS.	55
Figura 15 - Economia de energia devido ao compartilhamento de VNFs por SFCs curtas.	63
Figura 16 - Economia de energia devido ao compartilhamento de VNFs por SFCs mais longas.	65
Figura 17 - Economia de energia com VNFs de diferentes capacidades ($\mathcal{S}_i = 8000$).	67
Figura 18 - Consumo de energia para instâncias de VNFs com diferentes capacidades.	68

LISTA DE TABELAS

Tabela 1 -	Notações utilizadas no problema.	46
Tabela 2 -	Tempos de execução da solução ótima, em segundos, para SFC curtas e $\mathcal{S}_i = 1000$	60
Tabela 3 -	Tempos de execução da heurística em segundos ($\mathcal{S}_i = 1000$).	61
Tabela 4 -	Tempos de execução da heurística em segundos ($\mathcal{S}_i = 4000$).	61
Tabela 5 -	Razão entre a capacidade total requerida e a capacidade do total de VNFs.	69

LISTA DE ALGORITMOS

Algoritmo 1 - Heurística TRELIS.	56
--	----

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CAPEX	<i>CAPital EXpenditures</i>
ETSI	<i>European Telecommunications Standards Institute</i>
IDS	<i>Intrusion Detection System</i>
InP	<i>Infrastructure Provider</i>
IP	<i>Internet Protocol</i>
ISP	<i>Internet Service Provider</i>
MAC	<i>Media Access Control</i>
MANO	<i>Management ANd Orchestration</i>
MILP	<i>Mixed Integer Linear Programming</i>
NAT	<i>Network Address Translation</i>
NF	<i>Network Function</i>
NFV	<i>Network Functions Virtualization</i>
NFVI	<i>Network Functions Virtualization Infrastructure</i>
NFVO	<i>Network Functions Virtualization Orchestrator</i>
NOS	<i>Network Operating System</i>
OPEX	<i>OPerational EXpenditures</i>
QoS	<i>Quality of Service</i>
RAM	<i>Random Access Memory</i>
SDN	<i>Software-Defined Networking</i>
SFC	<i>Service Functions Chaining</i>
SLA	<i>Service Level Agreement</i>
SP	<i>Service Provider</i>
SRG	<i>Shared Risk Group</i>
TRELIS	<i>Tree Reduction with Line Solution</i>
VIM	<i>Virtual Infrastructure Manager</i>
VM	<i>Virtual Machine</i>
VNE	<i>Virtual Network Embedding</i>
VNF	<i>Virtual Network Function</i>
VNFM	<i>Virtual Network Function Manager</i>

LISTA DE SÍMBOLOS

$\%$	Porcentagem
Σ	Somatório
\in	Pertence a
\leq	Menor ou igual a
\forall	Para todos os valores de
\geq	Maior ou igual a
$=$	Igual a
\cup	União

SUMÁRIO

	INTRODUÇÃO	15
1	CONCEITOS BÁSICOS E TECNOLOGIAS RELACIONADAS	19
1.1	Arquitetura da Virtualização de Funções de Rede (<i>Network Functions Virtualization</i> , NFV)	19
1.1.1	Infraestrutura NFV (<i>NFV Infrastructure</i> , NFVI)	19
1.1.2	Funções Virtuais de Rede (<i>Virtual Network Functions</i> , VNFs)	20
1.1.3	Gerenciamento e Orquestração (<i>Management AND Orchestration</i> , MANO)	21
1.1.4	Encadeamento de Funções de Serviço (<i>Service Functions Chaining</i> , SFC)	23
1.1.4.1	Compartilhamento de VNFs	25
1.2	Mapeamento de Redes Virtuais (<i>Virtual Network Embedding</i> , VNE)	26
1.3	Redes Definidas por <i>Software</i> (<i>Software-Defined Networking</i> , SDN)	29
2	TRABALHOS RELACIONADOS	33
2.1	Estudos gerais sobre NFV	33
2.2	Posicionamento ótimo de SFCs	34
2.3	Áreas correlatas	37
2.4	Resiliência e consumo de energia em outras áreas	38
3	ESTRATÉGIA ADOTADA	39
3.1	Resiliência	39
3.2	Modelo de falhas	40
3.3	Replicação de SFCs	42
3.4	Modelo de consumo de energia	42
4	FORMULAÇÃO DO PROBLEMA	45
4.1	Função objetivo	45
4.2	Restrições de alocação de VNFs	46
4.3	Restrições de alocação de enlaces	47
4.4	Restrições de alocação de <i>backups</i>	48
4.5	Restrições de fluxo	49
4.6	Restrições de resiliência e de Grupos de Risco Compartilhado (<i>Shared Risk Groups</i> , SRGs)	51
4.7	Equações de domínio de variáveis	53
5	HEURÍSTICA TRELIS	55
6	AVALIAÇÃO DOS RESULTADOS	59
6.1	Tempos de execução da solução ótima	59
6.2	Resultados da TRELIS	60

6.2.1	<u>Posicionamento de SFCs curtas</u>	61
6.2.2	<u>Posicionamento de SFCs longas</u>	64
6.2.3	<u>Posicionamento de SFCs com diferentes capacidades de VNF</u>	66
7	CONCLUSÃO	71
7.1	Direções futuras	73
	REFERÊNCIAS	76

INTRODUÇÃO

Para prover serviços de redes, as infraestruturas de telecomunicações tradicionais utilizam equipamentos físicos dedicados, construídos para desempenhar uma determinada função ou serviço, denominados *middleboxes*. Esses *middleboxes* oferecem serviços como Tradução de Endereços de Rede (*Network Address Translation*, NAT), *proxy*, *firewall*, detecção de intrusão e outras funcionalidades (MIJUMBI et al., 2015b). Embora tais dispositivos ofereçam alto desempenho na execução de funções de rede (*Network Functions*, NF), já que são desenvolvidos especificamente para isso, os *middleboxes* dedicados costumam ser caros e apresentar arquitetura proprietária. Isso dificulta tanto a aquisição de novos dispositivos quanto a utilização e a coexistência com equipamentos de rede provenientes de outros fabricantes. Além disso, essa heterogeneidade dos *middleboxes* dificulta o gerenciamento da rede, pois cada fabricante implementa sua própria Interface de Programação de Aplicação (*Application Programming Interface*, API). Pela forma como são construídas, essas soluções são ditas verticais, o que traz pouca uniformidade ao gerenciamento da rede (SEKAR et al., 2011).

Como *middleboxes* dedicados são construídos para desempenhar uma NF específica, existe uma elevada relação entre o equipamento físico e a função executada pelos *middleboxes*, já que esses dispositivos não podem prover outros serviços senão aqueles ao quais se destinam. O uso intensivo desses *middleboxes* contribui para a ossificação da rede e eleva as despesas por parte dos operadores de rede (JOHN et al., 2013). Além disso, a alta relação entre *hardware* e *software* pode levar ao desperdício de recursos dos dispositivos. Se uma determinada função de rede é pouco requisitada, muitos dispositivos podem ficar ociosos por longos períodos de tempo, enquanto outros serviços mais requisitados podem ter seus dispositivos frequentemente saturados, operando em capacidade máxima. Isso acontece porque funções de rede diferentes não podem compartilhar o mesmo *hardware*.

Outra característica das infraestruturas baseadas em *middleboxes* dedicados é sua inflexibilidade, pois essa abordagem dificulta a expansão do serviço e a inclusão de novas funcionalidades na rede. A expansão é prejudicada por ser necessária a aquisição de um novo equipamento específico quando o atual estiver operando em sua capacidade máxima. Já a inclusão de novas funcionalidades também está diretamente relacionada com a aquisição de dispositivos novos, mas é prejudicada também pela implementação física do equipamento, considerando deslocamento e instalação. Isso se dá pois não é apenas a topologia lógica que é alterada ao incluir um serviço novo, mas também a física. Assim, o processo de administração da infraestrutura de rede se torna bastante dispendioso, o que revela a baixa escalabilidade e a inflexibilidade dessa abordagem.

Dado o exposto, a Virtualização de Funções de Rede (*Network Functions Virtualization*, NFV) se apresenta como um paradigma bastante promissor, tanto para o

cenário acadêmico quanto para o industrial, pois permite uma nova forma de lidar com as estruturas rígidas e de gerenciamento laborioso dos dispositivos de telecomunicações. Para permitir uma eficiência maior na utilização da infraestrutura e facilitar a expansão e a inovação nas redes de telecomunicações, a NFV busca contornar os problemas das infraestruturas tradicionais através da utilização de equipamentos genéricos. Nesses equipamentos as funções de rede são implementadas como soluções de *software*, não estando mais atreladas ao *hardware* (HAN et al., 2015).

Em redes de telecomunicações tradicionais, há uma forte relação entre o serviço provido e o equipamento físico no qual esse é implementado, tornando elevados os custos capitais (*CAPital EXpenditures*, CAPEX) e operacionais (*OPerational EXpenditures*, OPEX) em um sistema (HAWILO et al., 2014). Com isso, a NFV permite o desacoplamento da função exercida em termos de *software* de seu *hardware*, definindo NFs de rede que podem ser gerenciadas e orquestradas com maior flexibilidade em equipamentos físicos genéricos, reduzindo as despesas por parte dos operadores de rede e evitando que tais custos sejam repassados aos usuários finais (MIJUMBI et al., 2015b). Além disso, a NFV habilita o compartilhamento de *hardware* por NFs diferentes, o que é uma solução para o problema de desperdício de recursos.

Além de promover a separação entre o serviço provido e o equipamento físico, a NFV também viabiliza o uso não apenas de NFs, mas também de funções de rede virtuais (*Virtual Network Functions*, VNFs) desacopladas dos dispositivos. Para tal, utilizam-se máquinas virtuais (*Virtual Machines*, VMs) e instanciam-se dinamicamente as funções, que podem ser implementadas em diferentes servidores do operador de rede, sejam esses locais ou remotos. Se por um lado está se abrindo mão de equipamentos construídos especificamente para prover serviços de rede, por outro as VNFs podem oferecer os mesmos serviços de forma escalável, resiliente e com menor custo, porém com alguma redução de desempenho (HAN et al., 2015). Essa redução é esperada ao utilizar *hardware* genérico e técnicas de virtualização, sendo um dos desafios em NFV minimizar essa queda de desempenho. A utilização de técnicas de virtualização mais eficientes, algoritmos mais robustos para a implementação de VNFs e mecanismos de coleta de informações sobre o desempenho da rede são formas de reduzir a perda de desempenho.

O uso das VNFs, ou seja, das funções de rede instanciadas em máquinas virtuais, permite a criação de diferentes serviços e a possibilidade de gerenciá-los com maior facilidade. Como as VNFs podem ser instanciadas remota e dinamicamente, isso melhora a elasticidade na utilização da rede, pois as capacidades podem ser ajustadas conforme as requisições de serviços, alocando mais ou menos recursos conforme o necessário (MIJUMBI et al., 2015b). Um serviço prestado pode ser interpretado como resultado da execução de diferentes VNFs em uma sequência específica e é possível modificar esse serviço através da simples alteração da ordem na qual as VNFs são executadas ou da inserção de novas funções. Essa ordenação de funções de rede é conhecida como Encadeamento de

Funções de Serviço (*Service Functions Chaining*, SFC), responsável pelo provisionamento de serviços e alocação de recursos em NFV.

Motivação e Objetivos

Um dos desafios da abordagem NFV é a alocação ótima distribuída de cada VNF de um conjunto de SFCs em um grupo de servidores disponíveis (HAN et al., 2015). Além de um servidor ser capaz de hospedar várias VNFs, uma VNF também pode atender requisições de várias SFCs. A política de alocação deve considerar os recursos demandados para cada VNF de todas as SFCs, os recursos disponíveis dos servidores físicos individualmente e em conjunto, a largura de banda dos enlaces e valores de latência máximos permitidos. Essa política deve ainda adotar uma estratégia para aumentar a possibilidade de admitir novas SFCs, sem desperdiçar eventuais sobras de recursos de VNFs espalhadas, já que esses resquícios de recursos podem ser tão pequenos individualmente que não sejam capazes de atender a uma SFC.

Como as funções de rede em NFV não estão mais associadas a um equipamento físico, os operadores de rede necessitam que as VNFs sejam instanciadas em máquinas virtuais de seus servidores, seguindo os requisitos das SFCs de largura de banda, capacidade de processamento, latência e demais fatores. Além disso, a atribuição de VNFs às máquinas virtuais também deve levar em conta outras características, como a quantidade de funções a serem instanciadas em uma VM e em quais máquinas alocá-las, constituindo um problema de alocação de recursos. O posicionamento de SFCs é, então, o processo de escolha de quais as instâncias de VNFs que possuem recursos para atender às requisições feitas e em quais servidores essas VNFs serão alocadas. Conforme (MIJUMBI et al., 2015b), a infraestrutura NFV deve ser capaz de posicionar as VNFs nos servidores de modo ótimo, alocando os recursos físicos necessários dinamicamente para então prover o serviço através do encadeamento de funções.

No posicionamento de SFCs, dois requisitos representam desafios interessantes: a resiliência e a economia de energia. A resiliência pode ser garantida através da replicação de VNFs para manter as SFCs operantes mesmo em caso de falha (COUTO et al., 2015). Para tal, recursos adicionais são necessários para lidar com as falhas de modo a melhorar a resiliência da rede. A economia de energia pode ser alcançada por uma estratégia de compartilhamento de VNFs entre diversas SFCs, reduzindo os recursos ociosos e, consequentemente, o uso da energia.

Em princípio, resiliência e economia de energia são requisitos conflitantes, já que a replicação de VNFs consome recursos. Mas, assim como as VNFs podem compartilhar o mesmo *hardware*, o compartilhamento entre as VNFs de *backup* também ocorre, o que reduz a quantidade de recursos adicionais necessários para implementar a estratégia de

resiliência. Conforme a literatura, esta dissertação está entre os primeiros trabalhos a considerar o posicionamento ótimo de SFCs com resiliência em NFV.

Esta dissertação formula um problema de Programação Linear Inteira Mista (*Mixed Integer Linear Programming*, MILP) para posicionar SFCs em uma infraestrutura de rede. A formulação matemática do problema é resolvida para três redes acadêmicas, através do IBM CPLEX (IBM, 2013). No entanto, devido à elevada complexidade do problema, o consumo de memória e os tempos de execução inviabilizam a solução ótima. Assim, a heurística Redução em Árvore e Solução em Linha (*Tree REduction and LIne Solution*, TRELIS) é proposta para reduzir o tamanho do problema, através da seleção de um número reduzido de nós na topologia para posicionar as SFCs.

Para analisar a solução de otimização e a heurística proposta, são realizados experimentos para medir a economia de energia sob diferentes capacidades de servidores, números de SFCs e recursos alocados às VNFs. Os resultados mostram que a heurística reduz o tamanho do problema e economiza até 35% da energia dos servidores, ainda garantindo o funcionamento dos serviços provisionados se um servidor falhar. Além disso, a TRELIS é capaz de resolver um problema quatro vezes maior que o máximo possível, encontrando a solução ótima em apenas um quarto do tempo.

Organização do Texto

A dissertação está estruturada da seguinte forma. O Capítulo 1 apresenta os conceitos básicos de NFV, como a sua arquitetura, a formação dos serviços e o compartilhamento de VNFs, e também algumas das áreas correlatas ou complementares, como o Mapeamento de Redes Virtuais (*Virtual Network Embedding*, VNE) e as Redes Definidas por *Software* (*Software-Defined Networking*, SDN). Os trabalhos relacionados e a posição desta dissertação em relação à literatura são tratados no Capítulo 2. No Capítulo 3, são discutidos os modelos de falha e de consumo de energia e a estratégia de replicação de VNFs utilizados no trabalho. O problema de otimização é formulado no Capítulo 4, descrevendo a função objetivo e as restrições adotadas. O Capítulo 5 descreve a TRELIS e discute o funcionamento do algoritmo, enquanto o Capítulo 6 apresenta os experimentos realizados, discute sobre os tempos de execução obtidos através da solução ótima e da heurística e avalia os resultados obtidos. Por fim, a dissertação é concluída no Capítulo 7.

1 CONCEITOS BÁSICOS E TECNOLOGIAS RELACIONADAS

Para facilitar o entendimento da proposta, são apresentados neste Capítulo os principais conceitos sobre NFV. Conhecer as definições e os blocos funcionais da arquitetura NFV serve como base para a discussão sobre o problema de otimização. Além disso, a NFV também conta com muitas áreas complementares, como VNE e SDN, e que se valem das mesmas técnicas, como a virtualização, e até compartilham objetivos, como a alocação ótima de recursos e a capacidade de tornar a infraestrutura de rede programável e flexível. Sendo assim, embora estejam relacionadas, é preciso esclarecer os pontos de divergência que caracterizam cada uma.

1.1 Arquitetura da Virtualização de Funções de Rede (*Network Functions Virtualization*, NFV)

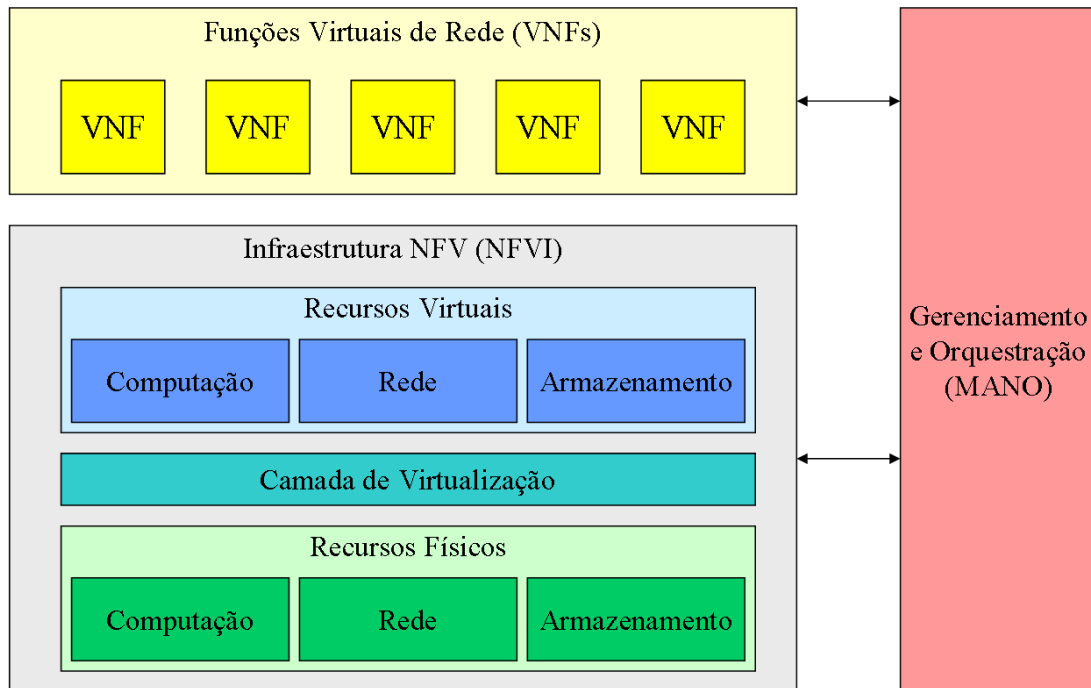
A NFV remove a relação de dependência entre *software* e *hardware* através da virtualização, que permite que as NFs sejam implementadas em servidores genéricos (HAN et al., 2015). Com isso, é empregada a virtualização de *software* para formar as VNFs. A infraestrutura física também é virtualizada, através de enlaces virtuais sobre os enlaces físicos e VMs nos servidores (MIJUMBI et al., 2015b). O enlace virtual é uma conexão lógica ponto a ponto, embora possa haver vários roteadores entre eles na topologia física. As VMs são abstrações de *software* de uma máquina física, nas quais um sistema operacional completo pode ser instalado. Isso permite que diversas VMs compartilhem o mesmo *hardware*.

A Figura 1 mostra a arquitetura NFV conforme as definições do Instituto Europeu de Padrões de Telecomunicações (*European Telecommunications Standards Institute*, ETSI), que geralmente é utilizada pelos fabricantes de soluções para NFV e serve como base para os esforços de padronização na área. A seguir detalham-se os componentes dessa arquitetura (ETSI, 2013).

1.1.1 Infraestrutura NFV (*NFV Infrastructure*, NFVI)

Sendo o componente mais básico da Arquitetura NFV, a Infraestrutura NFV (*NFV Infrastructure*, NFVI) é composta por um conjunto de recursos computacionais, de rede e de armazenamento, tanto físicos quanto virtuais. A NFVI consiste em uma coleção de recursos físicos e virtuais, estando relacionada aos dispositivos físicos e VMs que recebem as VNFs, além de envolver também os enlaces físicos e virtuais. A conexão de rede entre

Figura 1 - Arquitetura NFV.



Fonte: Adaptado de (ETSI, 2013).

os elementos e os dispositivos de rede também são considerados como parte da NFVI.

Em NFV, a camada de virtualização cria os recursos virtuais, que são abstrações dos recursos de *hardware*, e desacopla as VNFs dos equipamentos físicos. Os recursos físicos são particionados, isolados e cada parte é destinada a uma VM. A camada de virtualização permite ainda que as VNFs consigam utilizar a infraestrutura virtual subjacente, além de prover os recursos necessários à execução das funções (ETSI, 2013).

1.1.2 Funções Virtuais de Rede (*Virtual Network Functions*, VNFs)

A NFV oferece uma alternativa ao uso de equipamentos dedicados (*middleboxes*), através da alocação de VNFs. Essas funções virtuais são módulos de *software* que oferecem os serviços dos *middleboxes* em servidores genéricos, amplamente disponíveis nas diversas instalações. Dentre os serviços ofertados, estão incluídos NAT, *proxy*, balanceamento de carga, *firewall*, entre outros. As VNFs são implementações em *software* das funções de rede dos *middleboxes*, habilitadas para funcionar sobre a NFVI, situando-se no topo da arquitetura NFV da Figura 1.

Com relação ao funcionamento, tanto as funções de rede tradicionais quanto as VNFs se comportam independentemente da virtualização. Também é esperado que as

próprias interfaces externas de uma NF e de uma VNF sejam as mesmas (ETSI, 2013). Como uma VNF é um serviço de rede virtualizado, ela pode ser composta por múltiplos componentes internos, sendo implementada em uma ou várias VMs, sendo possível executar em cada VM um determinado componente da VNF. Essa estratégia é usada pela proposta de (SAHHAF et al., 2015), na qual as VNFs são serviços abstratos em alto nível descritas como uma SFC entre seus vários componentes internos.

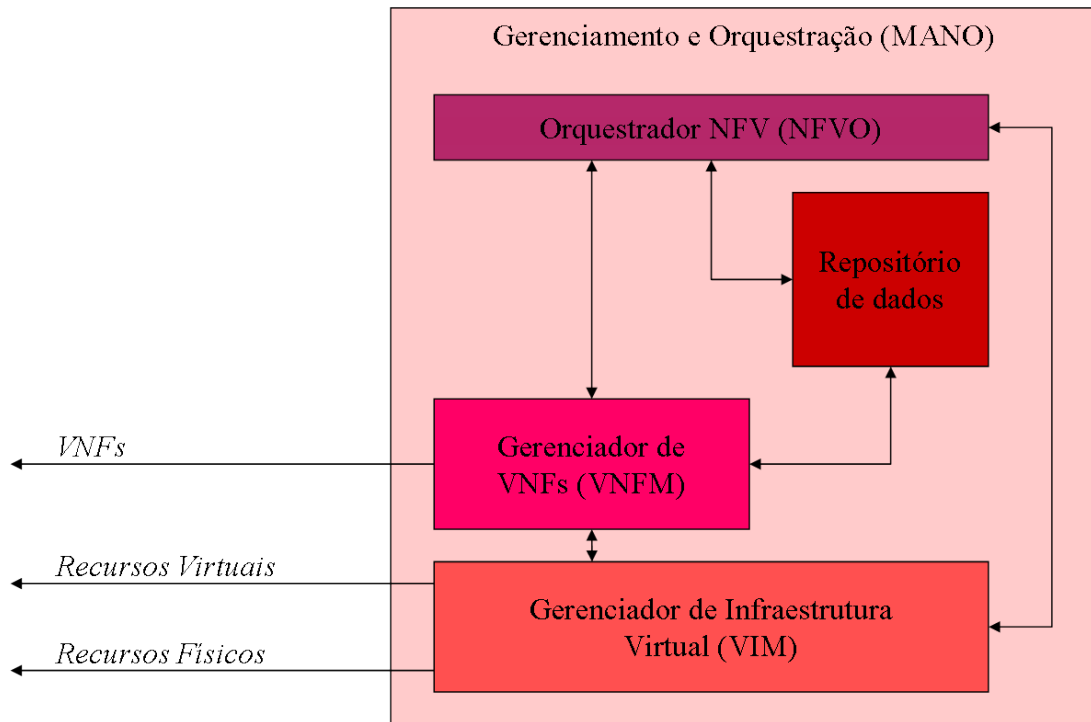
Outra característica da NFV quanto às funções de rede é citada por (MIJUMBI et al., 2015b). Embora o uso de VNFs seja um dos principais pontos a favor da NFV, a virtualização das NFs não é uma imposição. Como o ponto principal é o desacoplamento entre *hardware* e *software*, é possível usar NFs dos fabricantes instaladas diretamente na máquina, sem usar VMs. O desacoplamento não necessariamente requer a virtualização. Sendo assim, as NFs continuam independentes do *hardware*, já que podem ser instaladas em servidores genéricos, mas os benefícios oriundos da virtualização, como a elasticidade e a dinamicidade na alocação de recursos, não seriam contemplados. Além disso, também é possível construir cenários híbridos, nos quais há VNFs executando em recursos virtuais e NFs executando em recursos físicos. Esse cenário híbrido é discutido em (MOENS; TURCK, 2014).

1.1.3 Gerenciamento e Orquestração (*Management ANd Orchestration*, MANO)

O terceiro bloco da Figura 1 é o de Gerenciamento e Orquestração (*Management ANd Orchestration*, MANO), responsável pela gestão dos recursos físicos e de *software* que suportam a virtualização da infraestrutura, bem como por gerenciar o ciclo de vida das VNFs. Por isso, esse bloco interage tanto com a NFVI quanto com as VNFs. O MANO se destina a todas as tarefas de gerenciamento da virtualização necessárias na arquitetura NFV (ETSI, 2013). A Figura 2 ilustra o funcionamento interno do MANO.

Assim como a NFVI é a base da Arquitetura NFV, o bloco mais básico dentro do MANO é o Gerenciador de Infraestrutura Virtual (*Virtual Infrastructure Manager*, VIM), que envolve as funções de controle e gerenciamento da interação entre VNF e os recursos de computação, rede e armazenamento da NFVI. O VIM também é responsável pela virtualização dos recursos físicos, como um hipervisor que provê as abstrações de VMs a sistemas operacionais para que esses utilizem os recursos de *hardware* de uma máquina. É possível um cenário com múltiplos VIMs, o que indica várias NFVIs. Por exemplo, podem existir diferentes provedores de infraestrutura, cada um com um VIM designado. O gerenciamento de recursos também é feito pelo VIM, que faz um inventário das capacidades e recursos da NFVI e decide quanto à alocação de VMs. O VIM também coleta informações sobre problemas na infraestrutura, planejamento da capacidade, monitoramento e otimização (ETSI, 2013). Sendo assim, é o VIM que detecta a ocorrência

Figura 2 - Componentes internos do MANO.



Fonte: Adaptado de (ETSI, 2013).

das falhas e deve alertar aos outros blocos sobre o problema, se necessário.

O bloco intermediário é o Gerente de VNFs (*VNF Manager*, VNFM). É nesse bloco que é realizado o gerenciamento do ciclo de vida das VNFs, decidindo quando instanciá-las, quantas são necessárias para atender às requisições de serviço, se e quando remover uma VNF alocada em uma VM ou migrá-la para outra VM. Assim como o VIM, também é possível haver vários VNFMs. É possível designar um VNFM para cada VNF ou definir um VNFM para cada tipo de VNF ou até mesmo ter simplesmente um VNFM único para todas as VNFs. A presença de múltiplos VNFMs distribui o controle sobre as VNFs, que se torna mais complexo, mas também reduz a quantidade de VNFs que um único VNFM precisa gerenciar.

O bloco mais elevado é o Orquestrador NFV (*NFV Orchestrator*, NFVO), que rege tanto a NFVI quanto as VNFs. O NFVO é o elemento do MANO que orquestra os recursos, definindo o acesso dos serviços à NFVI e o compartilhamento dos recursos entre VNFs. O NFVO também orquestra os serviços, criando os serviços fim a fim, ou seja, o encadeamento das VNFs para formar SFCs, e gerenciando a topologia formada pelas instâncias de VNFs. Sendo assim, é no NFVO que a heurística TRELIS para o posicionamento das SFCs é executada, pois é nesse bloco que são tomadas as decisões de orquestração sobre os serviços, uso das VNFs e acesso às VMs. Embora a alocação de recursos às VMs ocorra no VIM e a alocação de VNFs no VNFM, é no NFVO que são

tomadas as decisões e são orquestrados os recursos e serviços.

O repositório de dados se situa entre o VNFM e o NFVO. Esse componente é um conjunto de dados, no qual estão listas e catálogos sobre as instâncias de VNFs, os serviços, os recursos e as capacidades. Por exemplo, o NFVO acessa as informações contidas no repositório e as utiliza para desempenhar suas funções. É com base nas informações do repositório que a heurística TRELIS, no NFVO, pode resolver o posicionamento das SFCs.

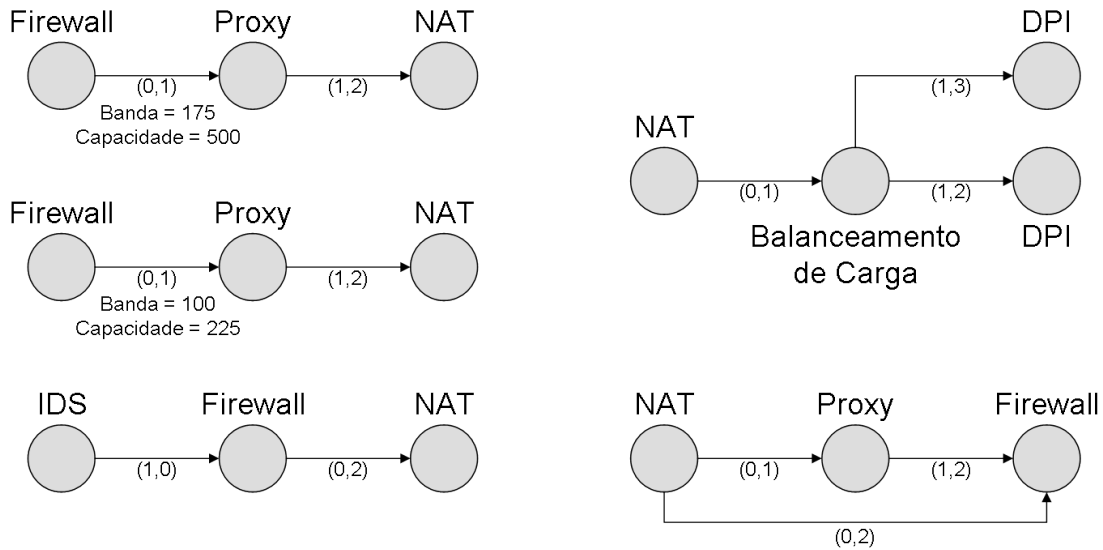
1.1.4 Encadeamento de Funções de Serviço (*Service Functions Chaining*, SFC)

Os serviços de NFV são oferecidos por uma sequência de VNFs, formando uma SFC. Esses serviços são representações em alto nível que dependem do conjunto de VNFs que os compõem e também do ordenamento delas, considerando tipo, capacidade e a sequência das VNFs. A Figura 3 mostra como diferentes serviços são compostos através das SFCs. Os pares ordenados nos exemplos identificam os enlaces virtuais, que ligam VNFs, e indicam o sentido que a SFC percorre, atravessando cada VNF. Os valores de banda e capacidade também fazem parte da requisição e, até para o mesmo encadeamento de VNFs, podem ser formados serviços diferentes. A capacidade pode ser expressa em termos de armazenamento em disco, quantidade de memória, ciclos de processamento e outros fatores.

Como um serviço é uma abstração em alto nível, um mesmo conjunto de VNFs pode ser usado para compor diferentes serviços comerciais. Como as SFCs incluem especificações, como as de largura de banda e capacidade, é possível definir um serviço único com diferentes níveis de utilização, por exemplo, com uma largura de banda menor ou maior para o cliente. Com isso, é possível definir diferentes níveis de serviço para uma mesma sequência ordenada de VNFs, aumentando valores de capacidade e banda para definir serviços com mais recursos e possivelmente mais caros. Ao mudar a ordem das VNFs, também se muda o tratamento recebido pelo fluxo de pacotes, pois esses são recebidos e tratados de forma diferente. Além disso, uma SFC também pode ser construída para atender a múltiplos pontos finais ou bifurcar o tráfego em um balanceador de carga. Isso possibilita a bifurcação do encaminhamento dos dados, utilizando, por exemplo, um balanceador de carga para segmentar um fluxo em direção a duas instâncias de VNFs distintas ou do mesmo tipo. Também é possível dividir um determinado tráfego para que parte dele passe por uma VNF específica e outra não, mesmo que tenham um ponto de destino único, convergindo ao final da SFC. Essa ampla gama de possibilidades para a criação de serviços é explorada pela NFV.

Apesar da utilização de funções virtuais em SFCs, seu conceito é ainda mais amplo e não se resume apenas à ligação de VNFs para prover um serviço. É possível que exista SFC sem funções virtuais e até mesmo sem NFV. Segundo (MIJUMBI et al., 2015b),

Figura 3 - Exemplos de composição de diferentes serviços.



como o desacoplamento entre NFs e dispositivos de *hardware* não significa que os recursos precisem ser virtualizados, então se torna possível a criação de serviços através do encadeamento até mesmo de funções de rede não virtualizadas. Sendo assim, também é possível criar SFCs sem NFV. Entretanto, essa é uma tarefa complexa, pois não é necessário apenas interligar funções para gerar um serviço, mas também posicionar os dispositivos físicos ordenadamente, visto que, sem NFV, a *software* ainda está atrelado ao *hardware* no qual é implementado. Nesse caso, o processo de modificação ou o gerenciamento de SFCs é mais dispendioso e complexo. Com isso, a NFV redefine a forma como o encadeamento de funções é realizado, facilitando o processo de gerenciamento e provisionamento de serviços.

Embora a NFV torne o processo de composição e posicionamento de SFCs mais fácil de se gerenciar e menos propenso a erros, ele também se torna mais complexo, já que, além dos recursos físicos da infraestrutura, também devem ser considerados os recursos virtuais. Além disso, o gerenciamento deve ser feito em dois níveis, para as VNFs, que devem ter recursos para atender às SFCs, e para os servidores, que devem ter recursos para receber as VNFs. Como as VNFs são soluções de *software* que desempenham funções de *middleboxes*, cada fabricante pode criar diferentes modelos, por exemplo, de *firewall*, cada um com funcionalidades diferentes. Para ser executado em uma máquina, um *software* apresenta requisitos mínimos e recomendados. Sendo assim, uma VNF de *firewall* pode requisitar, por exemplo, 2 núcleos de processamento, 4 GB de Memória de Acesso Aleatório (*Random Access Memory*, RAM) e 2 GB de armazenamento em disco. Assim, a VM que receberá essa VNF deve atender a esses requisitos para que a VNF seja implementada. Dentre as funções que esse *firewall* pode desempenhar, há desde políticas

básicas para endereços de rede, portas e protocolos e filtragem *web*, até detecção e bloqueio de *malware* e controle específico para cada aplicação.

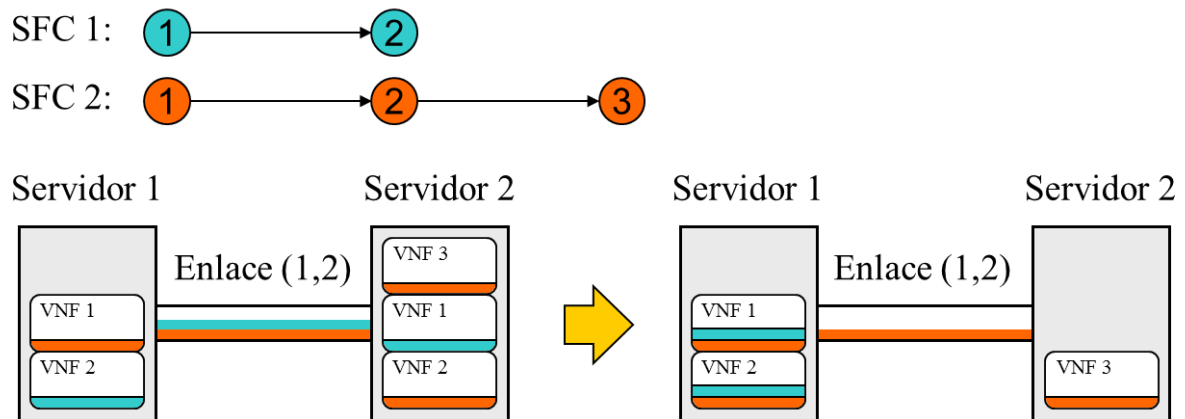
1.1.4.1 Compartilhamento de VNFs

Como os serviços podem ser constituídos por SFCs de várias formas, é possível que um provedor ofereça diferentes serviços a partir de um mesmo conjunto de VNFs. Dessa forma, frequentemente as SFCs precisam recorrer aos mesmos tipos de VNF. Como uma VNF deve sempre ter recursos suficientes para atender a uma requisição, uma VNF com muitos recursos excedentes pode ser usada para atender a mais de uma SFC. Esse é o princípio do compartilhamento de VNFs entre as SFCs. O posicionamento de novas SFCs na infraestrutura sem precisar instanciar novas VNFs facilita as funções do MANO e as alterações de VNFs alocadas na NFVI são menos frequentes, como visto a seguir nos resultados desta dissertação.

O compartilhamento de VNFs por múltiplas requisições é uma forma de melhorar a utilização das instâncias disponíveis. Muitos trabalhos sobre posicionamento de SFCs consideram o compartilhamento de VNFs pelas requisições, como (MEHRAGHDAM; KELLER; KARL, 2014; BARI et al., 2016; LUIZELLI et al., 2015; MOENS; TURCK, 2014; ADDIS et al., 2015). No entanto, para haver compartilhamento, é necessário que a VNF seja capaz de atender às requisições das SFCs. Se duas SFCs solicitam um serviço com configurações iguais de NAT, uma instância de NAT é capaz de atender as duas solicitações. Contudo, caso uma SFC solicite um *firewall* com um determinado conjunto de regras e outra SFC solicite uma configuração diferente, o mesmo *firewall* só poderá atender as duas se possuir os dois conjuntos de regras solicitados. Se isso não for possível, as SFCs deverão ser atendidas por *firewalls* diferentes e não haverá compartilhamento. Isso está previsto na formulação do problema através da indicação do tipo de VNF $m \in \mathcal{M}$, em que m pode indicar uma NAT, um *proxy*, um *firewall* com um determinado conjunto de regras ou outro *firewall*, no conjunto \mathcal{M} de tipos de VNF, conforme discutido no Capítulo 4.

A Figura 4 ilustra à esquerda uma infraestrutura sem compartilhamento de VNFs pelas SFCs e sua posterior reorganização considerando o compartilhamento. Como as duas SFCs solicitam duas VNFs do mesmo tipo, elas podem ser compartilhadas, já que as VNFs possuem capacidade para atender às requisições de ambas. Dessa forma, são utilizadas duas instâncias a menos, poupando recursos do Servidor 2. Quanto menos recursos são utilizados, menor será o uso de energia. Mesmo que a segunda máquina também esteja ligada, ter duas VNFs a menos instanciadas reduz a parcela variável do consumo de energia, referente à utilização dos recursos da máquina. Há também uma parcela fixa de consumo de energia que é o mínimo necessário para manter o servidor ligado e executando suas rotinas mais básicas. A representação das parcelas fixa e variável

Figura 4 - Compartilhamento de VNFs pelas SFCs.



do consumo de energia é discutida na Seção 3.4. Além disso, como a SFC 1 foi atendida exclusivamente pelo Servidor 1, há também economia de largura de banda no enlace.

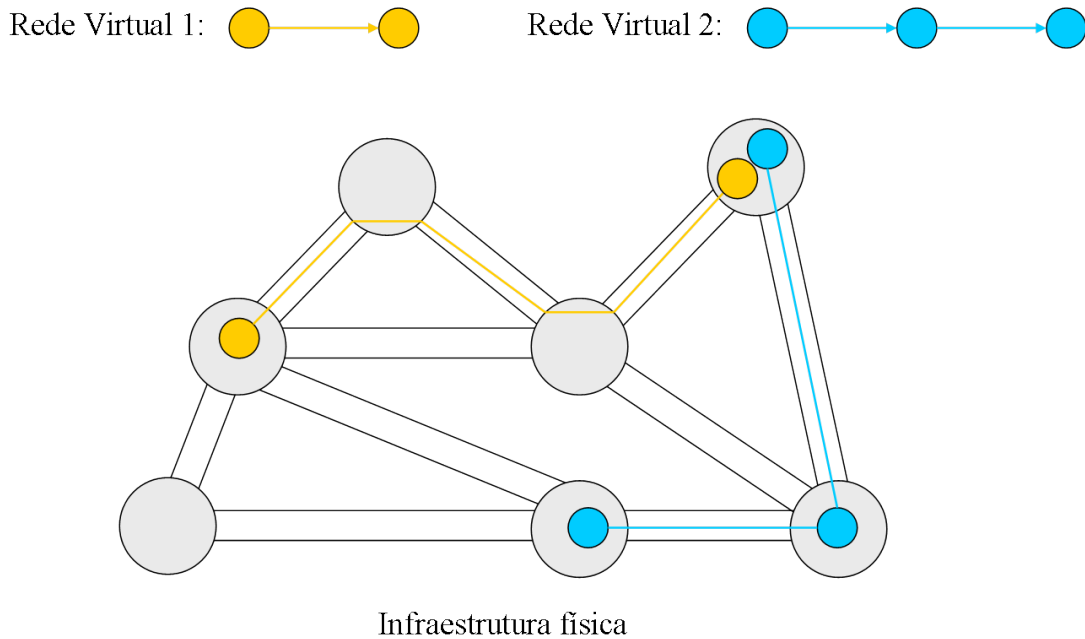
Além do compartilhamento de VNFs pelas SFCs, a Figura 4 também mostra que é possível o compartilhamento do dispositivo físico pelas VNFs. Como um servidor pode ter várias VMs e cada uma delas pode receber VNFs, o equipamento físico pode ser compartilhado para a execução de diferentes funções de rede, devido ao desacoplamento entre *software* e *hardware* proporcionado pela NFV.

1.2 Mapeamento de Redes Virtuais (*Virtual Network Embedding*, VNE)

Embora muitas inovações em termos de *software* tenham sido desenvolvidas ao longo dos anos, a evolução das infraestruturas físicas não acompanhou o mesmo ritmo. Frequentemente são encontradas barreiras que dificultam seu desenvolvimento, como a elevada dependência de *hardware* para realizar alterações na rede. Trocar o *hardware* envolve trocar a infraestrutura e isso envolve aquisição, instalação e manutenção de equipamentos. Se aplicar alterações de *software* em um número elevado de máquinas já pode incorrer em custos elevados, trocar as próprias máquinas se torna uma tarefa ainda mais laboriosa. A virtualização de redes, então, é uma tecnologia que permite mudar esse cenário e lidar com a inflexibilidade das infraestruturas físicas atuais, buscando superar a resistência às mudanças de *hardware* (FISCHER et al., 2013).

A virtualização não é um conceito novo trazido pela NFV e tem sido empregada há muitos anos, como na criação de Redes Locais Virtuais (*Virtual Local Area Network*, VLAN) e Redes Privadas Virtuais (*Virtual Private Network*, VPN) (CHOWDHURY; BOUTABA, 2009). As VLANs são redes logicamente interligadas em uma rede de nível 2 que constituem um domínio de *broadcast* independente da conectividade física. Com

Figura 5 - Mapeamento de uma rede virtual no substrato.

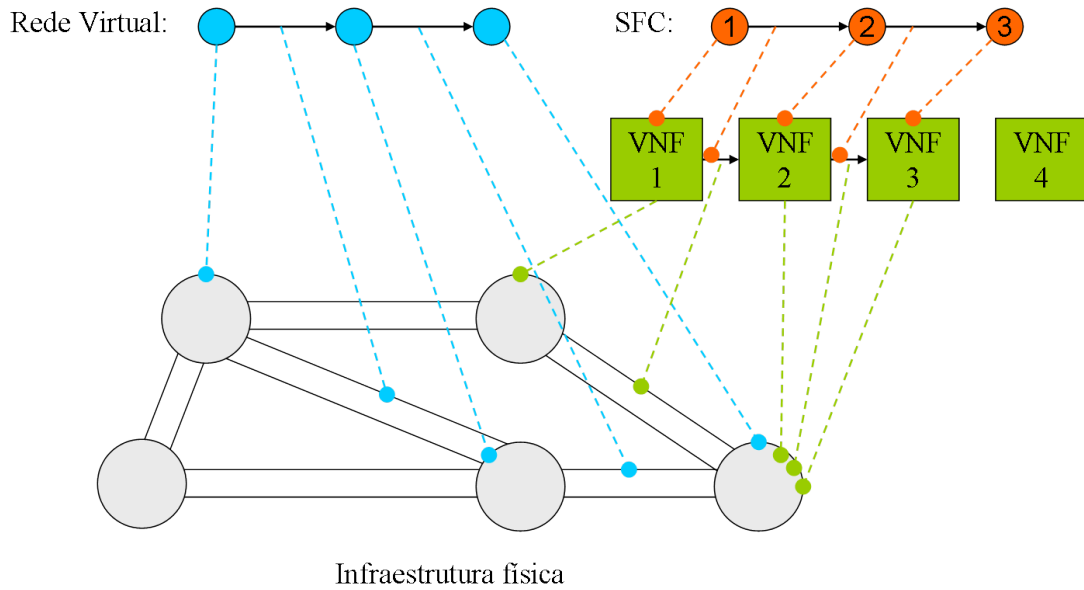


isso, uma rede local pode ser dividida em múltiplas VLANs e cada estação dentro delas só receberá quadros destinados àquela VLAN. As VPNs são redes particulares e utilizam túneis sobre a rede. Uma vez estabelecida a VPN, a estação remota passa a fazer parte do mesmo domínio de *broadcast* da rede hospedeira. Assim, é possível que dois dispositivos se reconheçam como pertencentes à mesma rede, mesmo que um pacote precise ser encaminhado por vários roteadores até ir de um para o outro. Assim, esses túneis podem ser formados sobre a própria *Internet*. Os túneis VPNs são seguros e isolam o tráfego de uma VPN do resto da *Internet*.

A NFV faz uso das técnicas de virtualização para desacoplar as funções de rede dos equipamentos físicos, criando as VNFs. A virtualização também é utilizada para instanciar VMs nos servidores, que recebem as VNFs. No entanto, antes da virtualização ser utilizada em NFV, ela já era utilizada para mapear redes virtuais em VNE.

O VNE também busca tornar as infraestruturas de rede mais flexíveis e permitir a configuração remota da rede, alocando redes virtuais sobre uma infraestrutura física, conhecida como substrato. Em (CHOWDHURY; BOUTABA, 2009), a função tradicional dos Provedores de Serviço de *Internet* (*Internet Service Providers*, ISPs) é dividida por duas entidades diferentes, o Provedor de Infraestrutura (*Infrastructure Provider*, InP) e o Provedor de Serviços (*Service Provider*, SP). Os InPs são responsáveis por gerenciar a infraestrutura física da rede, enquanto os SPs recebem requisições de serviço e criam redes virtuais, com os recursos físicos disponíveis através dos InPs para oferecer seus serviços. O mapeamento de requisições dessas redes virtuais, considerando os recursos computacionais

Figura 6 - Diferenças entre o mapeamento em VNE e o posicionamento em NFV.

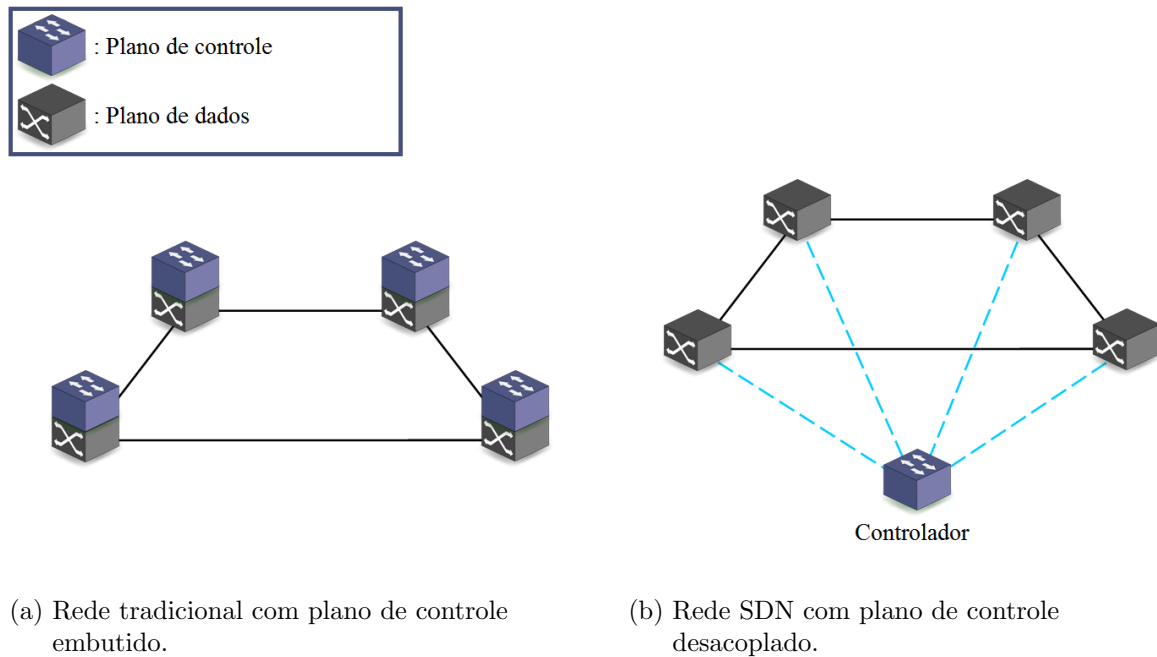


e capacidades do substrato de um InP, é o que caracteriza o VNE. A Figura 5 mostra um exemplo do problema de mapeamento de redes virtuais em um substrato.

O problema de posicionamento em VNE é muito semelhante ao problema de posicionamento de SFCs em NFV, tanto que esse deriva do primeiro. Uma das diferenças existentes é que o problema de VNE procura alocar uma rede virtual em uma rede física. Já o posicionamento de SFCs está associado a serviços de rede que estão sendo prestados, sendo a infraestrutura de rede já estabelecida previamente. Em (LUIZELLI et al., 2015), destaca-se que as soluções para o problema de VNE não são apropriadas para o de posicionamento de SFCs e outras duas diferenças são consideradas. Primeiro, o mapeamento feito em VNE é realizado em uma etapa apenas, posicionando redes virtuais em um substrato, enquanto o posicionamento de SFCs em NFV requer um mapeamento em duas etapas, alocando as VNFs aos servidores físicos e posicionando as SFCs nas VNFs alocadas. Isso torna o problema de posicionamento de SFCs muito mais complexo que o problema de VNE. Segundo, um problema de VNE considera como dispositivos físicos apenas roteadores, enquanto um problema de NFV insere uma quantidade maior de dispositivos e funções de rede coexistindo na mesma infraestrutura. A Figura 6 ilustra os níveis de mapeamento existentes em VNE e NFV.

Mesmo que o VNE e a NFV tenham objetivos semelhantes, como alocar recursos em uma infraestrutura de modo a reduzir, por exemplo, o consumo de energia ou o número de enlaces e dispositivos utilizados, o problema de posicionamento em VNE é mais simples, já que apresenta um nível a menos de complexidade no mapeamento e considera apenas dispositivos de encaminhamento de pacotes na formação das redes virtuais. Sendo assim,

Figura 7 - Redes tradicionais e redes SDN.



o posicionamento de VNE pode ser compreendido como um precursor do mesmo problema em NFV.

1.3 Redes Definidas por *Software* (*Software-Defined Networking*, SDN)

Embora NFV e SDN possuam semelhanças com relação ao processo de inovação das redes, tornando-as programáveis e mais flexíveis, a SDN o faz de outra forma. Enquanto a NFV permite o desacoplamento entre a função de rede, como um balanceador de carga ou um Sistema de Detecção de Intrusão (*Intrusion Detection System*, IDS), e o equipamento físico no qual ela é implementada, a SDN realiza o desacoplamento entre o plano de dados e o plano de controle dos roteadores (KREUTZ et al., 2015).

O plano de dados encaminha os dados recebidos através das interfaces do equipamento de rede, verificando uma tabela de encaminhamento presente no dispositivo, que indica por qual interface encaminhar os dados para que cheguem ao seu destino. O plano de controle orienta o processo de encaminhamento, utilizando protocolos para preencher as tabelas de encaminhamento dos elementos no plano de dados. Como nas redes tradicionais cada dispositivo de rede possui o seu plano de controle, ele também é capaz de comunicar com os de outros dispositivos de rede, aprendendo sobre caminhos dos vizinhos e definindo o conteúdo da tabela. Há ainda um terceiro plano, acima do plano de controle, conhecido como plano de gerenciamento, que configura e monitora o funcionamento

da rede. O plano de gerenciamento pode definir políticas como nível de Qualidade de Serviço (*Quality of Service*, QoS) e bloqueio de tráfego; esse é o plano mais próximo do administrador da rede (JAIN; PAUL, 2013).

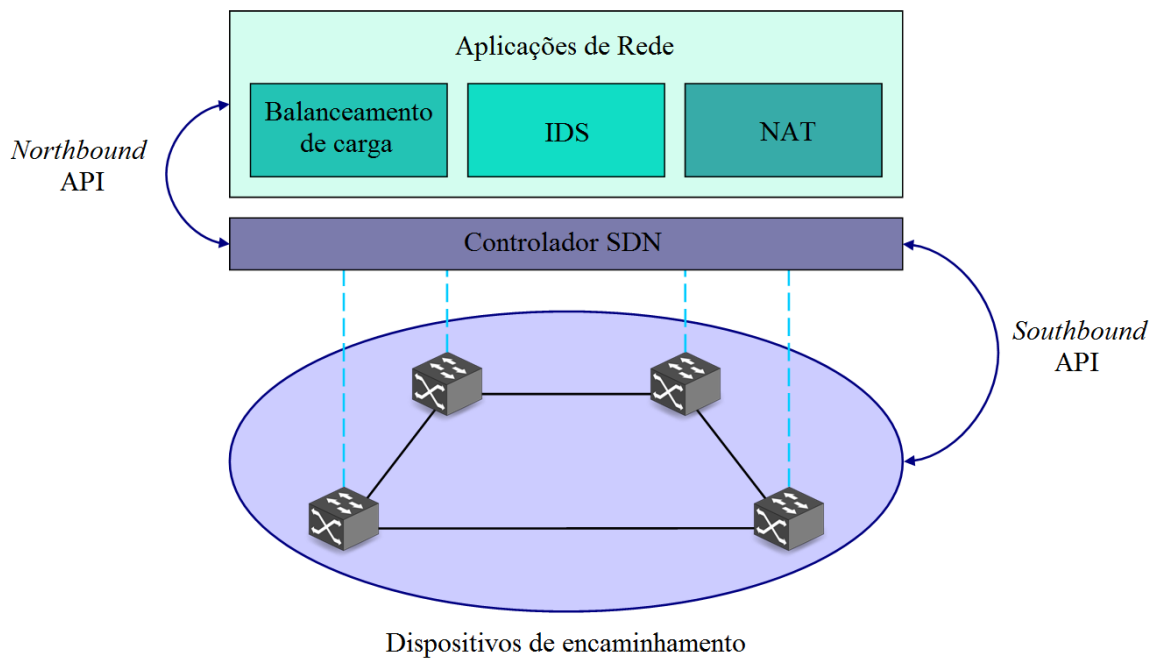
Nas redes tradicionais, os comutadores possuem ambos os planos atrelados, de modo que as informações de controle são distribuídas e a configuração de cada dispositivo torna-se complexa e propensa a erros em redes maiores. Além disso, cada dispositivo possui seu próprio *software* de configuração, então roteadores de fabricantes diferentes não podem ser configurados da mesma forma. Sendo assim, a integração vertical dos planos dificulta a inserção de mudanças no plano de controle. Em uma rede SDN, os dispositivos se tornam encaminhadores de pacotes e a inteligência associada ao plano de controle é direcionada a um controlador SDN. A Figura 7 mostra a diferença entre as redes tradicionais e as redes SDN, representando o desacoplamento do plano de controle e o atribuindo a um elemento controlador.

A Figura 8 ilustra a arquitetura básica da SDN com seus componentes principais. O controlador SDN é um plataforma de *software* que executa em dispositivos comuns, como um servidor, que se comunica com os comutadores através de *Southbound* APIs e fornece abstrações e recursos para programar a rede (KREUTZ et al., 2015). A programação das regras de encaminhamento dos comutadores é feita pelo controlador, que possui um *software*, conhecido como Sistema Operacional de Rede (*Network Operating System*, NOS), capaz de instruir os comutadores sobre as decisões a serem tomadas para os fluxos de pacotes. Então, o controlador instala as regras de fluxo nos comutadores simples com base nas aplicações de rede. As aplicações de rede podem criar regras para programar a rede através do controlador, utilizando as *Northbound* APIs.

Os encaminhadores de pacotes podem ser diferentes tipos de dispositivos de rede, como *middleboxes* ou roteadores, que mantêm tabelas de encaminhamento com as regras aprendidas através do controlador, como descartar ou repassar um fluxo. Os comutadores simples não tomam decisões autônomas. Quando um deles recebe um fluxo novo, ou seja, para o qual não há uma diretriz definida em sua tabela de encaminhamento, o comutador envia uma mensagem ao controlador para aprender a ação que ele deve tomar para lidar com aquele fluxo. Ao receber a resposta do controlador, o comutador cria um registro em sua tabela e passa a saber que ação tomar caso receba aquele fluxo novamente.

Outro aspecto básico da SDN é a forma como as decisões de encaminhamento são tomadas. Em redes convencionais, os quadros são encaminhados de acordo com o endereço de Controle de Acesso ao Meio (*Medium Access Control*, MAC) em nível de enlace e o roteamento de pacotes considera o endereço de destino do Protocolo de *Internet* (*Internet Protocol*, IP). As decisões de encaminhamento em SDN são baseadas em fluxos, que utilizam campos do cabeçalho do pacote como diferentes critérios de classificação e associam uma ação a ser tomada pelo dispositivo de encaminhamento. Por exemplo, uma determinada combinação de porta e endereços IP de origem e de destino pode categorizar

Figura 8 - Arquitetura SDN.



um fluxo SDN e encaminhá-lo através de uma interface específica ou rejeitá-lo. Dessa forma, a SDN reúne diferentes campos dos vários cabeçalhos, como o de quadro e o de pacote, e essa combinação de informações caracteriza o fluxo. Então, o comutador irá tomar a mesma ação sempre que encontrar um fluxo com uma mesma combinação de campos de cabeçalho.

A SDN busca então centralizar logicamente o plano de controle, mantendo nos dispositivos de encaminhamento apenas o plano de dados. Assim, obtém-se vantagens como a facilidade de configuração da rede, uma menor suscetibilidade a erros de configuração ao aplicar essas mudanças de forma centralizada apenas no controlador e a possibilidade de configurar dispositivos de diferentes fabricantes de forma uniforme através de APIs. Assim como a NFV, a SDN também promove a melhora na flexibilidade e na configuração remota da rede. No entanto, as duas tecnologias realizam essas melhorias de formas diferentes.

Com relação à flexibilidade, a NFV facilita o processo de gerenciamento e orquestração das VNFs, bem como dos serviços providos na forma de SFCs, utilizando técnicas de virtualização para que não haja mais uma dependência intrínseca do *hardware*. Já as redes SDN melhoram esse aspecto através do desacoplamento entre o plano de dados dos roteadores, responsável pelo encaminhamento dos pacotes, e o plano de controle, que insere as informações de controle sobre as ações a serem tomadas para cada fluxo de pacotes. Isso transforma os roteadores em comutadores simples e delegando as informações de controle a um elemento controlador. Assim, a flexibilidade é melhorada pois a modificação

de um comportamento da rede só precisa ser feita em apenas um dispositivo controlador, não mais em todos os nós da rede (NUNES et al., 2014).

No tocante à configuração remota da rede, em NFV, as VNFs podem ser instanciadas remotamente, bem como o gerenciamento das VMs alocadas nos servidores físicos. Nas redes SDN, o administrador da rede é capaz de programar a rede configurando o elemento controlador. Sendo assim, o plano de controle pode ser configurado por *software* remotamente no controlador e as novas regras serão difundidas aos comutadores simples.

Tendo em vista as diferenças entre NFV e SDN, é possível notar que a NFV está mais associada às funções de rede e serviços implementados nos dispositivos de rede, estando em um patamar acima da área de atuação da SDN, que está mais associada ao encaminhamento e roteamento dos dados entre esses dispositivos de rede. De forma geral, a SDN está mais associada ao processo de decisão e encaminhamento dos pacotes, redefinindo a forma como a programação da rede é realizada. Por sua vez, a NFV busca virtualizar as aplicações de rede, os serviços implementados pelos *middleboxes* dedicados, estando associada ao processo de provisionamento dos serviços.

2 TRABALHOS RELACIONADOS

A fim de posicionar esta dissertação em relação aos trabalhos relacionados disponíveis na literatura, as seções a seguir classificam os trabalhos listados conforme suas contribuições. São considerados, então, estudos gerais sobre NFV, artigos dedicados ao posicionamento ótimo de SFCs, áreas correlatas e resiliência e consumo de energia em trabalhos de outras áreas.

2.1 Estudos gerais sobre NFV

Como a NFV representa um paradigma novo e promissor no cenário das telecomunicações, há elevado interesse tanto por parte da comunidade científica quanto do mercado em sua evolução e muitos trabalhos de pesquisa têm sido desenvolvidos acerca desse tema (COSTA-PÉREZ et al., 2013). O trabalho (ROSA et al., 2014) traça um panorama geral sobre NFV e destaca oportunidades de pesquisa, como a formação e o posicionamento das SFCs e a resiliência. Assim, esse trabalho apresenta a NFV como uma forma de tornar as redes mais simples e flexíveis, minimizando as dependências de *hardware*. Em (HAN et al., 2015), alguns dos principais desafios em NFV são listados, bem como casos de uso e aspectos que devem ser considerados na implementação das VNFs, como a confiabilidade e a continuidade do provimento do serviço na forma de SFCs, que não pode ser impactada caso ocorra uma falha. Em (MIJUMBI et al., 2015b), é realizado um estudo abrangente sobre o estado da arte em NFV e sua relação com áreas correlatas, como a computação em nuvem e as redes SDN. Esse trabalho destaca o problema da alocação de recursos e a eficiência de energia como desafios de pesquisa, que são dois dos aspectos tratados no problema de posicionamento de SFCs desta dissertação.

Com relação à alocação de recursos em NFV, o estudo de (HERRERA; BOTERO, 2016) formaliza o problema de alocação de recursos em NFV em várias etapas, como a composição das cadeias de VNFs, o posicionamento das SFCs e o escalonamento das VNFs para execução no processador. Além disso, classifica os trabalhos realizados nessa área de acordo com a estratégia de otimização escolhida, como programação linear e heurísticas, as métricas utilizadas e a quais etapas do problema se destinam. Segundo as classificações desse *survey*, esta dissertação tem como cenário redes genéricas de NFV, podendo ser redes de centros de dados ou de provedores de serviços de telecomunicações. A etapa do problema de alocação tratada é a de posicionamento de SFCs. As estratégias providas são solução exata e heurística. Já a principal contribuição está na formulação do problema MILP e a solução do mesmo através da TRELIS. Minimiza-se o consumo de energia dos servidores e é garantido o provisionamento dos serviços de todas as SFCs

para até uma falha de servidor, utilizando a heurística TRELIS para problemas maiores.

Em (JOHN et al., 2013), são discutidas algumas das oportunidades de pesquisa acerca das SFCs em NFV, incluindo a descrição e programação das SFCs, implementação das VNFs, continuidade no provimento do serviço de rede e segurança.

2.2 Posicionamento ótimo de SFCs

O posicionamento de SFCs tem sido considerado em várias frentes de pesquisa. Em (LUIZELLI et al., 2015), é formalizado o problema de posicionamento de SFCs, com o objetivo de reduzir o número de VNFs alocadas, comparando valores de atraso fim-a-fim entre a solução proposta e infraestruturas tradicionais de *middleboxes* dedicados. A heurística utilizada busca o menor número de instâncias de VNFs possível para solucionar o problema de forma iterativa. Ela remove a função objetivo e a reescreve como uma das restrições, de modo que o número de VNFs instanciadas seja sempre menor ou igual ao número de VNFs fornecidas, para resolver o problema naquela iteração. Além disso, também é incluído um parâmetro de limite de tempo para encontrar uma solução possível, evitando que o tempo de execução da heurística se torne inviável devido às sucessivas iterações.

O trabalho de (SAHHAF et al., 2015) considera SFCs formadas por serviços descritos em alto nível, que são decompostos em VNFs de baixo nível também ligadas em SFCs. Por exemplo, uma SFC pode conter uma VNF de *firewall*, que pode ser decomposta na forma de uma VNF de filtragem *web* e uma de detecção e bloqueio de *malware*. Assim, a VNF de *firewall* pode ser decomposta em uma SFC com duas VNFs. A heurística proposta resolve o problema formulado em duas etapas. Primeiro, são formadas todas as decomposições possíveis para um dado serviço, resultando em múltiplas soluções possíveis. Na segunda etapa, verifica-se o custo de cada decomposição e a que tiver o menor custo é mapeada na rede, alocando as VNFs e posicionando as conexões lógicas entre elas.

O estudo de (BARI et al., 2016) trata do problema de orquestração de VNFs para reduzir OPEX, minimizando os custos de implementação, consumo de energia e penalidades por violações em Acordos de Nível de Serviço (*Service Level Agreement*, SLA). Por apresentar múltiplos objetivos, são utilizados parâmetros de peso para ajustar a relevância de cada componente da função objetivo. Esta dissertação se dedica apenas a reduzir o consumo de energia, sem depender de parâmetros de relevância na função objetivo. A heurística deixa de considerar explicitamente a fragmentação de recursos para fins de simplificação e é resolvida em duas etapas. Fragmentos de recursos são partes ociosas da capacidade do servidor que não podem ser aproveitadas por serem muito pequenas e representam um desperdício dos recursos disponíveis. Primeiro, são modelados os grafos de encaminhamento, que são representações de nós e enlaces associados a um

custo para ir de um vértice a outro, e depois recorre-se ao algoritmo de Viterbi (FORNEY, 1973) para resolver o posicionamento de modo quase ótimo. O algoritmo de Viterbi é utilizado para prever a sequência de estados mais provável de um sistema a partir de um conjunto de estados já conhecido. Não há considerações de resiliência na formulação do problema, mas essa possibilidade é mencionada como parte dos trabalhos futuros.

Outro trabalho a considerar o consumo de energia na alocação das VNFs na infraestrutura é o de (BRUSCHI; CARREGA; DAVOLI, 2016). No entanto, isso é feito através de uma solução baseada em teoria dos jogos, na qual as VNFs são jogadores que competem pelos recursos da infraestrutura. Sendo assim, não há uma heurística proposta, já que a solução não sofre a escalabilidade limitada dos problemas MILP. Esse trabalho considera uma modelagem de consumo de energia diferente, alterando o *clock* das máquinas, que influencia tanto no consumo quanto na taxa de processamento das tarefas pelos processadores.

O trabalho de (MEHRAGHDAM; KELLER; KARL, 2014) também se destina à alocação de recursos em NFV, mapeando as requisições de VNFs na rede para vários objetivos, como minimizar a latência nos enlaces criados, maximizar a largura de banda restante nos enlaces físicos ou minimizar o número de sítios com VNFs instanciadas. Para simplificar o problema, a heurística fornecida não computa todas as ordenações possíveis de VNFs na composição da SFC: as funções são ordenadas com base no consumo da taxa de dados e as funções que mais reduzem a taxa de dados dos fluxos são alocadas primeiro. Dentre os grafos possíveis para uma requisição, a heurística escolhe o que tiver o menor custo de taxa de dados. Como as requisições de SFCs são uma entrada do problema desenvolvido nesta dissertação, a TRELIS realiza o posicionamento das SFCs na infraestrutura e não a composição das SFCs, como a heurística desse artigo faz.

Em (MOENS; TURCK, 2014), também é proposto um problema de posicionamento de SFCs, porém voltado para cenários híbridos, nos quais os serviços são providos tanto por *middleboxes* dedicados quanto por VNFs instanciadas em servidores genéricos. Apenas a solução ótima é fornecida e a proposta de uma heurística é mencionada em seus trabalhos futuros.

O estudo de (MA; MEDINA; PAN, 2015) aborda o problema de alocação de VNFs considerando a variação de tráfego, já que as VNFs podem tanto inserir tráfego em uma SFC quanto consumi-lo. Considera-se que os fluxos reais chegam um por vez, o que torna o problema mais fácil de solucionar. Assim, apenas a solução ótima é provida, sendo desenvolvida em duas etapas. Primeiro, o caminho do fluxo é determinado e os *middleboxes* são posicionados e, em seguida, os melhores caminhos entre as funções são determinados com o auxílio de algoritmos de roteamento. Como o trabalho considera um cenário com NFV e SDN, o algoritmo proposto é implementado como um módulo do controlador SDN Floodlight (FLOODLIGHT, 2013), de código aberto, e os experimentos são conduzidos na plataforma Mininet (LANTZ; HELLER; MCKEOWN, 2010).

O trabalho de (RIERA et al., 2015) formaliza um problema de escalonamento de VNFs com o objetivo de minimizar o tempo de execução dos serviços implementados. No entanto, apenas a formulação matemática é discutida e não há uma solução ótima ou uma heurística. Como as funções são *softwares* executados em uma VM, o escalonamento se refere ao processo de atribuição dos núcleos de processamento às VNFs, de modo que os processadores não fiquem ociosos e nem sobrecarregados com VNFs em espera para serem executadas.

O escalonamento de VNFs também é discutido em (MIJUMBI et al., 2015a), que resolve simultaneamente o problema de posicionamento de SFCs e o escalonamento de VNFs através de uma meta-heurística, que é um método genérico para resolução de problemas que não está atrelado especificamente a um problema, como uma heurística usada para construir heurísticas. Nesse artigo, considera-se que as VMs já estejam mapeadas na infraestrutura física, alocando, então, as VNFs nas VMs. São utilizados três algoritmos gananciosos, alocando preferencialmente nos servidores com melhores tempos de execução, nos servidores cujos processadores ficarão disponíveis mais cedo após executar uma VNF e nos servidores com mais capacidade disponível. Além disso, é proposto um quarto algoritmo baseado em um método de pesquisa tabu, que compara uma possível solução local com a solução de vizinhos, buscando por soluções com resultados melhores. Se uma solução viola alguma restrição ou foi visitada anteriormente, ela pode ser marcada como um tabu, uma solução proibida para que o algoritmo não recorra a ela repetidas vezes.

Em (CLAYMAN et al., 2014) é proposta uma arquitetura baseada em um elemento orquestrador que aloca os serviços de rede em nós virtuais. Apenas a solução ótima é fornecida e a proposta de uma heurística para um futuro problema de programação linear é mencionada nos trabalhos futuros.

Com relação à resiliência em problemas de alocação de recursos em NFV, há poucos trabalhos disponíveis na literatura, sendo ainda um desafio pouco explorado. Em (LEE et al., 2015), é formulado um problema de posicionamento de fluxos de SFCs, distribuindo a carga das requisições por múltiplas instâncias de VNFs para prover serviços de modo escalável e tolerante a falhas. Isso é realizado através do espalhamento das funções às quais as SFCs são direcionadas. Embora a replicação de VNFs reduza a quantidade de SFCs afetadas por uma falha, essas ainda são interrompidas caso ocorra alguma falha de enlace ou servidor e não há um caminho alternativo ou uma VNF de *backup* para prover resiliência, de modo que os benefícios da estratégia adotada estão mais associados aos de balanceamento de carga. Nesta dissertação, as VNFs não são espalhadas, mas sim concentradas nos servidores para reduzir o consumo de energia, a resiliência é garantida através de caminhos e VNFs de *backup*, que nunca ficam inoperantes ao mesmo tempo que os caminhos e VNFs principais.

O trabalho de (BOUTEN et al., 2016) desenvolve um conjunto de restrições de

afinidade e antiafinidade para especificar em quais localizações as SFCs podem ser posicionadas. Isso melhora a eficiência, alocando VNFs com alta transferência de dados entre si próximas uma da outra, a resiliência, através do espalhamento das VNFs pelos centros de dados, e demais fatores. Esse trabalho adota uma estratégia similar a uma das utilizadas nesta dissertação, pois uma de suas restrições de antiafinidade diz que dois caminhos virtuais associados não podem ter nenhum enlace físico em comum. Além disso, o artigo inclui outra restrição de antiafinidade para que duas funções, como um *firewall* e uma Inspeção Profunda de Pacotes (*Deep Packet Inspection*, DPI), não estejam no mesmo servidor. No entanto, essas restrições ainda assim podem não impedir o rompimento de uma SFC quando vários enlaces virtuais são cortados, como é o caso da descontinuidade espelhada vista na Seção 4.6. Outra diferença é que o conjunto de restrições de afinidade e antiafinidade não forma caminhos secundários entre pares de VNFs, como é feito nesta dissertação.

Nesta dissertação, o problema de posicionamento de SFCs é abordado através da formulação de um problema MILP, de sua solução ótima e da heurística TRELIS, que obtém melhores tempos de execução e consegue chegar a uma solução quase ótima. A TRELIS funciona em apenas uma etapa e busca reduzir o número de servidores fornecidos ao problema de otimização, o que reduz consideravelmente o número de variáveis do problema, permitindo alcançar as soluções mais rapidamente. As discussões acerca da heurística são realizadas no Capítulo 5.

2.3 Áreas correlatas

Como as técnicas de virtualização também são empregadas em redes SDN, aumentando a flexibilidade e melhorando o gerenciamento da infraestrutura de rede, NFV e SDN podem ser usadas em conjunto de forma complementar. Em (JAIN; PAUL, 2013), destaca-se a utilização da NFV e da SDN na computação em nuvem, reforçando o aspecto complementar, embora não dependente, dessas áreas. Por exemplo, em (WOOD et al., 2015), é proposta uma forma de combinar NFV e SDN, utilizando a virtualização e a separação dos planos de controle e de dados dos roteadores. Essa combinação é feita através da plataforma de servidores virtuais proposta para otimizar para a execução de serviços de rede, a NetVM, que deve ser interligada a controladores SDN para programar a rede.

Muitas das técnicas utilizadas em NFV para alocação de VNFs e posicionamento de SFCs na infraestrutura são anteriores ao surgimento da NFV. O trabalho de (HAIDER; POTTER; NAKAO, 2009) apresenta diferentes técnicas de alocação de recursos utilizadas em redes virtuais anteriores à NFV, mas que se tornaram fundamentais para a mesma, como a alocação de múltiplas redes virtuais na mesma rede física e considerações de tráfego

e formação de enlaces. Essas técnicas já são utilizadas desde o VNE, uma estratégia anterior à NFV para o posicionamento de redes virtuais em uma infraestrutura física. O VNE pode ser interpretado como um precursor do problema de posicionamento de SFCs na NFV, conforme discutido na Seção 1.2.

Os avanços em VNE são listados em (FISCHER et al., 2013). Em (BOTERO et al., 2012), são considerados, por exemplo, aspectos de eficiência energética na alocação de recursos em VNE. Em (SU et al., 2012), é formulado um modelo de consumo de energia e desenvolvido um algoritmo energeticamente eficiente para VNE. No entanto, esses trabalhos consideram apenas que os dispositivos estejam ligados ou desligados e a redução do consumo de energia advém da minimização do número de componentes energizados. Já esta dissertação considera que os servidores estejam sempre ligados, porém ociosos ou ativos, e o consumo é proporcional à quantidade de recursos em uso. Manter os servidores desligados diminui as chances de que novas requisições sejam atendidas. A inicialização das máquinas consome mais energia e repetir esse processo constantemente pode sobrepor a economia obtida ao mantê-las desligadas, caso elas sejam reinicializadas constantemente e fiquem desligadas por pouco tempo. Além disso, um serviço pode ser atrasado ou interrompido caso os servidores que contenham os *backups* estejam desligados. O modelo de consumo de energia adotado é discutido na Seção 3.4.

2.4 Resiliência e consumo de energia em outras áreas

Embora a resiliência seja um aspecto pouco discutido no posicionamento de SFCs em NFV, os trabalhos de outras áreas, como a computação em nuvem e o VNE, tratam dessa questão. Considerando a resiliência, uma estratégia baseada em replicação também é utilizada em (COUTO et al., 2015), no qual são utilizados *backups* de centro de dados em redes geodistribuídas para prover resiliência em cenários de desastre, visando a sobrevivência do conteúdo armazenado nesses centros de dados. Em (COUTO et al., 2014) são propostas diretas para o projeto de centros de dados resilientes.

Em (RAHMAN; BOUTABA, 2013), é formulado um problema de alocação de recursos para VNE e são considerados aspectos de resiliência, já que o provisionamento das redes virtuais leva em conta falhas de enlaces para incentivar a sobrevivência da rede, diferente desta dissertação, que considera falhas de servidores em um cenário NFV.

Em (GUO et al., 2011), são propostos mecanismos de *backup* compartilhado para lidar com falhas na infraestrutura física sobre a qual as redes virtuais são mapeadas. A sobrevivência das redes virtuais em VNE também é discutida em (RAHMAN; AIB; BOUTABA, 2010). Nesta dissertação, também é utilizada uma estratégia baseada em *backups* de VNFs que também são compartilhados pelas SFCs. Os impactos dessa estratégia são discutidos na Seção 3.3.

3 ESTRATÉGIA ADOTADA

Como não é possível construir sistemas perfeitos ou prevenir desafios à operação normal (STERBENZ et al., 2013), estratégias devem ser adotadas para garantir a resiliência. Dessa forma, neste trabalho utiliza-se a replicação de caminhos e VNFs para realizar o posicionamento de SFCs. Para resolver o problema de posicionamento de SFCs com resiliência, é preciso caracterizar que tipo de falha deve ser considerada e suas consequências. Em uma infraestrutura de rede, é comum acontecerem falhas de enlaces e falhas de servidores, seja por algum tipo de configuração inapropriada, queda de energia ou porque algum dispositivo parou de funcionar.

Para lidar com as possíveis falhas e prover resiliência, uma solução possível é adaptar a SFC requerida para que, além do caminho operacional e principal, estejam previstos caminhos secundários e VNFs de *backup*. Dessa forma, o tráfego direcionado a uma VNF primária pode ser encaminhado ao seu *backup* em caso de falha. Essa estratégia, então, consiste em prever a necessidade de caminhos e funções de *backup*, também os posicionando na infraestrutura física para que, na ocorrência de uma falha, a SFC já tenha um caminho alternativo para que o serviço seja provido.

A princípio, prover resiliência com base em replicação pode implicar em um compromisso com a economia de energia, visto que mais recursos são necessários para garantir o provimento do serviço. Então, também é necessário considerar uma forma de alocar os recursos necessários às VNFs de *backup* sem prejudicar a economia de energia.

3.1 Resiliência

Um dos pontos ainda pouco explorados na literatura de NFV é a resiliência no posicionamento de SFCs. Embora a otimização na alocação de recursos e a resiliência sejam aspectos independentes, a minimização dos recursos alocados, como largura de banda, e valores de OPEX também pode considerar falhas. A alocação de instâncias de VNFs nos servidores também pode considerar requisitos de resiliência, que possam contribuir para o provisionamento de um determinado serviço. Dessa forma, a resiliência pode ser definida como a capacidade de se provisionar um serviço de modo aceitável na presença de falhas e outros desafios à operação normal, de modo que essas não tenham impacto significativo no resultado final (STERBENZ et al., 2013).

Segundo (FRY et al., 2010), as redes resilientes são aquelas que devem ser capazes de sobreviver aos desafios à operação normal, tomando ações para manter um nível de operação aceitável. Sendo assim, a alocação das instâncias de VNFs deve levar em consideração os aspectos de resiliência para efetuar o posicionamento das SFCs, como incluir

funções de *backup* em outros servidores para que a SFC tenha um caminho alternativo para contornar uma possível falha. Em (STERBENZ et al., 2010), uma falha é definida como o desvio de um serviço funcional e desejado para um funcionamento não esperado ou que não esteja de acordo com o especificado. Por exemplo, como o funcionamento normal de um servidor envolve a execução correta de cada VNF nele, a falha do servidor implica no desvio do funcionamento normal das VNFs instanciadas nele.

De acordo com (ETSI, 2015), a continuidade do serviço não é apenas uma expectativa do usuário, mas muitas vezes é um requisito legal, já que as empresas podem ser penalizadas pela descontinuidade do serviço. Assim, a continuidade do serviço é uma questão mais importante do que focar na disponibilidade da plataforma.

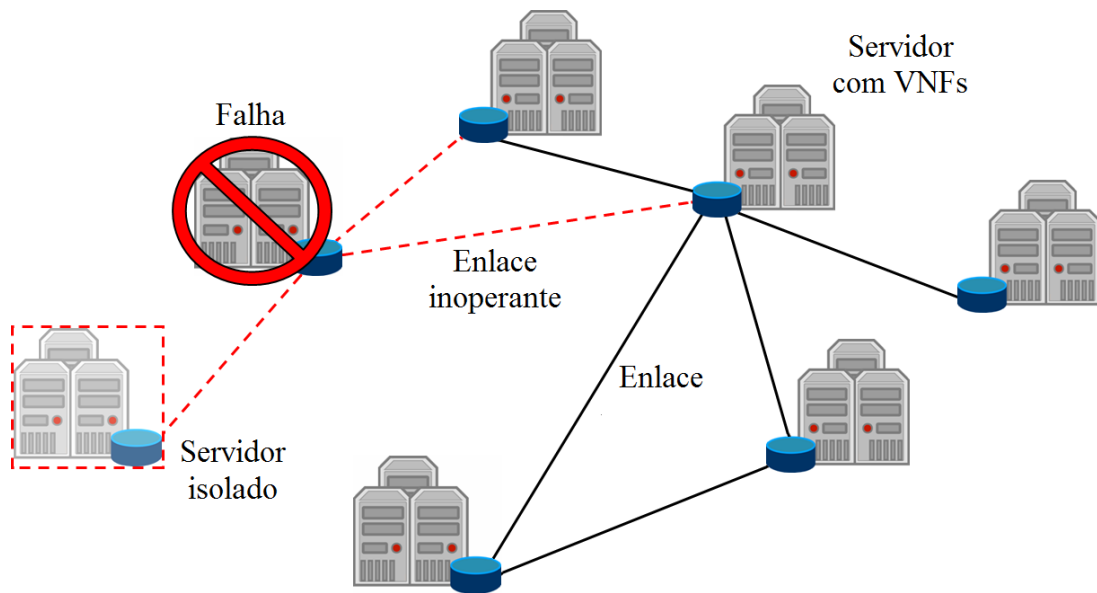
Ainda segundo (ETSI, 2015), o MANO, entidade responsável por gerenciar e orquestrar a NFVI e as VNFs associadas aos dispositivos físicos, também pode tomar ações corretivas nas VNFs. Um exemplo de ação é redirecionar, dentro de uma SFC, o encaminhamento para uma função em *standby* se for detectada uma falha de alguma VNF daquele serviço. Além disso, a principal função dessas entidades para preservar a disponibilidade de um serviço é minimizar os efeitos de falhas na infraestrutura. Sendo assim, uma alternativa para prover resiliência é a replicação de VNFs e caminhos em uma SFC, utilizando VNFs de *backup* que possam substituir uma VNF operacional e caminhos alternativos. Como o VIM é o componente do MANO que monitora a NFVI, ele é o responsável por detectar falhas nos enlaces e nos servidores e tomar as ações necessárias, como avisar aos demais componentes do MANO da ocorrência de uma falha.

Com relação às SFCs, é importante notar que uma falha em um servidor pode comprometer o provisionamento de todos os serviços que passem por ele, ou porque há VNFs instanciadas nele ou porque ele simplesmente está no caminho entre duas VNFs. Sendo assim, é necessário garantir que haja um caminho alternativo para dar continuidade à SFC. Nesse caso, é relevante considerar a questão da redundância, posicionando VNFs de *backup*, para dar sequência ao encadeamento de funções daquele serviço. Mesmo com a resiliência, as restrições de largura de banda e valores de latência fim a fim especificados nas requisições do serviço ainda devem ser mantidas, visto que é importante que o serviço seja provido de modo que a falha não impacte o funcionamento normal.

3.2 Modelo de falhas

O modelo de falha adotado considera falhas em servidores físicos. Isso pode se dar por falha do *hardware*, sistema operacional ou da virtualização. Falhas exclusivamente de enlaces físicos não são consideradas porque já são contempladas nas consequências da falha de servidor e para simplificar o modelo. Dessa forma, falhas de servidores são mais críticas, pois acarretam a interrupção do funcionamento de todas as VNFs alocadas nele

Figura 9 - Elementos de um SRG são afetados pela mesma falha.



e implicam na queda de todos os enlaces ligados ao servidor em questão.

Considera-se ainda que, na ocorrência de uma falha, um servidor se torna inoperante, ou seja, é uma falha silenciosa ou do tipo *fail-stop* (TANENBAUM; STEEN, 2007). Como as VNFs nele hospedadas param de funcionar e todos os enlaces físicos se tornam inoperantes, as SFCs que atravessam o servidor que falhou são particionadas. Se houver algum servidor ligado ao restante da rede apenas por um enlace com o servidor que falhou, esse também se torna inalcançável. No entanto, considera-se que esse outro servidor continua funcionando, apenas perdendo a conectividade com o restante da rede. Nesta dissertação, considera-se que as funções de roteamento e encaminhamento de pacotes estejam atreladas ao servidor. As VNFs instanciadas no servidor isolado continuam operacionais e podem ainda ser utilizadas nas SFCs, contanto que as mesmas não dependam de outro segmento da rede. Como a falha de um servidor pode afetar outros elementos da rede, utiliza-se o conceito de Grupo de Risco Compartilhado (*Shared Risk Group*, SRG) (COUTO et al., 2014). Um SRG é composto por todos os elementos em uma determinada região afetados pela mesma falha. A Figura 9 ilustra esse conceito. Nessa figura, o SRG associado ao servidor que falhou é ilustrado pelos enlaces ligados a ele, tracejados, que ficam inalcançáveis quando a falha ocorre. O servidor dentro das linhas tracejadas também está inalcançável para o restante da rede, mas continua operacional, portanto não falha. Isso significa que uma SFC não pode ter VNFs no servidor isolado e no restante da rede ao mesmo tempo, pois não há como passar pelo servidor que falhou. No entanto, a SFC pode ser posicionada inteiramente no servidor isolado ou apenas no restante da rede, ambos os casos são possíveis.

3.3 Replicação de SFCs

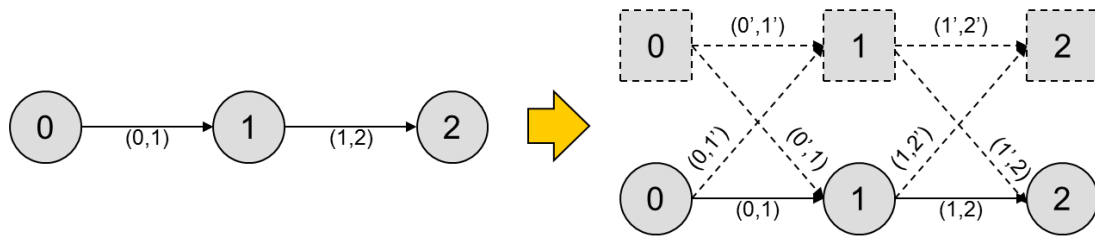
O número de elementos necessários para tolerar k falhas silenciosas de servidor, do tipo *fail-stop*, é dado por $n = k + 1$ (TANENBAUM; STEEN, 2007). Como esse tipo de falha resulta apenas na ausência de resposta, ou seja, não envolvendo respostas incorretas ou em atraso, basta que se tenha um único elemento a mais que o número de falhas que se pretende suportar. Por exemplo, com 6 elementos iguais, é possível que até 5 deles falhem que ainda haverá um capaz de responder corretamente. No modelo considerado nesta dissertação, apenas uma falha ($k = 1$) é considerada, embora o mesmo possa ser expandido para tolerar mais falhas. Assim, para cada VNF operacional na SFC, deve haver uma réplica, isto é, uma VNF de *backup*, além de enlaces alternativos suficientes para preservar a SFC. A instância de *backup* deve ter igual capacidade e deve estar preparada para substituir a VNF operacional correspondente que falhou. Com isso, a resiliência no posicionamento de SFCs permite que se continue com o mesmo nível de serviço.

A Figura 10 mostra como a replicação de VNFs de *backup* é aplicada a uma SFC e como os enlaces são construídos entre os pares de VNFs. No lado esquerdo, é representada a SFC que de fato o cliente está solicitando e, no lado direito, a SFC que deve ser alocada para garantir a resiliência a uma falha. As linhas tracejadas indicam VNFs e enlaces virtuais de *backup*, enquanto as linhas cheias representam a SFC principal. Sendo assim, para cada VNF operacional há uma VNF de *backup* correspondente e, para cada enlace virtual da SFC principal, há três enlaces replicados interligando as VNFs que conectam. As VNFs dentro dos círculos são as funções de rede encadeadas que um usuário solicita ao seu provedor de serviço, como *firewall*, *proxy* e tradução de endereços (MIJUMBI et al., 2015b). Já as VNFs em quadrados são *backups* e permitem que o serviço continue sendo provido, mesmo que um servidor da rede se torne inoperante. Portanto, para tornar a SFC resiliente na ocorrência de uma falha de servidor, a estratégia adotada consiste no posicionamento de uma réplica para cada VNF operacional solicitada na SFC, instanciadas em servidores diferentes dos que receberam sua VNF operacional correspondente. Por exemplo, mesmo que as três VNFs 0, 1 e 2 da Figura 10 estejam hospedadas no mesmo servidor e ele falhe, os *backups* das três continuam ativos e a SFC pode ser provida pelos enlaces $(0', 1')$ e $(1', 2')$, já que a resiliência é garantida para até uma falha de servidor.

3.4 Modelo de consumo de energia

O consumo de energia de uma rede é proporcional à quantidade de recursos ativos na infraestrutura, como servidores, roteadores e outros dispositivos. Por exemplo, à medida que mais unidades de processamento são utilizadas, mais energia é necessária

Figura 10 - Exemplo de uma SFC qualquer com replicação de VNFs.



por parte do *hardware*. Uma forma simples de reduzir o consumo de energia em uma rede é desligar os dispositivos que não estão em uso, gerando gastos de eletricidade desnecessários. Em (BOTERO et al., 2012), a proposta de redução de energia para um problema de otimização em VNE busca minimizar a quantidade de roteadores e enlaces ativos, concentrando as redes virtuais posicionadas no substrato atravessando a menor quantidade de dispositivos e enlaces possível. Apesar de a redução do número de dispositivos ligados reduzir os gastos de OPEX relacionados a energia, isso nem sempre pode ser possível, pois a ocorrência de uma falha pode requerer a pronta inicialização de uma máquina e o tempo até ela estar ativa pode violar algum termo de SLA ou interromper o provisionamento do serviço. Além disso, caso os servidores ativos estejam todos operando em capacidade máxima em um cenário real, a chegada de uma nova requisição de serviço pode ser rejeitada ou atrasada até que uma nova máquina seja ligada.

Dado o exposto, o modelo de energia adotado considera que todos os servidores da infraestrutura física estão ligados, porém em dois estados, ativos ou em *idle*. Um servidor em *idle*, ou seja, ocioso, tem um consumo de energia mínimo, realizando apenas as funções mais básicas da máquina. Um servidor ativo possui uma ou mais VNFs operacionais instanciadas nele. Em cenários reais, o consumo máximo de um servidor pode ser 34 vezes maior que o consumo em *idle* (BARI et al., 2016).

Como um servidor com várias VNFs alocadas pode potencialmente consumir mais energia que um servidor com poucas VNFs, já que seus recursos estão sendo utilizados mais intensamente, adotou-se como base o modelo de consumo de energia de (BARI et al., 2016). Esse modelo define um nível mínimo e um nível máximo de consumo de energia nos servidores baseados em dados de medições reais. O valor do consumo de energia de um servidor, então, pode variar entre esses dois valores, sendo proporcional à quantidade de recursos computacionais em uso nos servidores. Caso o servidor esteja ligado e ocioso, executando suas rotinas mais básicas, seu consumo de energia é igual ao valor mínimo, já que ele gasta o mínimo necessário para se manter ligado. Por outro lado, uma máquina rodando intensamente com todos os processadores trabalhando e utilização máxima de memória tem um consumo de energia máximo, pois ela está saturada, no limite de seu funcionamento. Dessa forma, o objetivo é manter o consumo de energia dos servidores

sempre o mais próximo possível do mínimo, ainda atendendo a todas as requisições de serviço. Isso garante o posicionamento mais eficiente em termos de energia. A Equação 1 mostra o consumo de energia dos servidores. Quando não há VNFs instanciadas no servidor, a $Capacidade_{alocada}$ é zero, e o consumo de energia é \mathcal{E}_{min} , ou seja, é o consumo mínimo, no estado de *idle*. Quando a $Capacidade_{alocada}$ é igual a $Capacidade_{total}$, o consumo de energia é \mathcal{E}_{max} , ou seja, é o consumo máximo, com todos os recursos em uso.

$$Consumo_{servidor} = \left((\mathcal{E}_{max} - \mathcal{E}_{min}) \cdot \frac{Capacidade_{alocada}}{Capacidade_{total}} \right) + \mathcal{E}_{min} \quad (1)$$

Neste trabalho o consumo de energia é proporcional à quantidade de VNFs executando no servidor. Assim, a estratégia de economia de energia consiste em aproveitar de forma ótima as capacidades dessas VNFs. Usar apenas uma VNF para atender requisições de várias SFCs contribui para que não seja necessário instanciar novas VNFs. Esse compartilhamento de VNFs melhora o aproveitamento dos recursos disponíveis e não eleva o consumo de energia. Da mesma forma, aproveitar a capacidade das VNFs ao máximo, isto é, sem desperdiçar seus recursos ou ter fragmentos de capacidade não aproveitáveis também contribui para economizar energia.

O consumo de energia é calculado com base na quantidade de recursos alocados, haja SFCs consumindo esses recursos das VNFs ou não. Se uma VNF está executando no servidor, o consumo de energia é relacionado ao quanto de recursos da máquina a VNF utiliza, não importando se ela atende a uma ou várias SFCs. O consumo não é calculado pela parcela de recursos solicitados, mas sim pelos recursos alocados. É por isso que se considera o consumo de energia devido aos fragmentos de capacidade desperdiçados. Além disso, uma VNF só é instanciada em um servidor se ela for servir a uma SFC. Por exemplo, caso se disponha de uma instância para NAT, *proxy* e *firewall* e uma SFC solicite apenas uma NAT, não há razão para também alocar as outras duas VNFs.

4 FORMULAÇÃO DO PROBLEMA

O problema de otimização formulado tem como objetivo reduzir o consumo de energia nos servidores, aproveitando as capacidades ociosas das VNFs operacionais e realizando o compartilhamento de VNFs por diferentes SFCs. Além disso, o problema replica VNFs, garantindo resiliência para uma falha *fail-stop*. A minimização do consumo de energia também reduz a quantidade de VNFs alocadas, devido ao compartilhamento. Além disso, reduz-se o OPEX, já que os gastos com despesas de energia elétrica e manutenção de *hardware* e *software* diminuem.

Embora necessitem estar alocadas em servidores com recursos suficientes para seu funcionamento, as VNFs de *backup* não precisam estar ativas durante a operação normal da rede, haja vista as técnicas de *cold standby* (LEVITIN; XING; DAI, 2014). Por exemplo, um servidor estará em seu consumo de energia mínimo quando possui apenas VNFs de *backup* instanciadas, equivalente ao estado de *idle*. Isso ocorre pois essas VNFs estão apenas alocadas e não em execução. Assim, durante a operação dos sistemas, apenas o consumo de energia associado às VNFs operacionais é considerado, como discutido na Seção 3.4.

A Tabela 1 lista as variáveis, parâmetros e conjuntos utilizados na formulação do problema, bem como as notações utilizadas e suas descrições. A formulação matemática é composta pela função objetivo e pelas restrições de alocação de VNFs, de alocação de enlaces, de alocação de *backups*, de fluxo e de resiliência e SRGs. Em seguida, são listadas as equações de domínio de variáveis.

4.1 Função objetivo

O modelo de energia é baseado em (BARI et al., 2016), conforme discutido na Seção 3.4. Nesse modelo, o consumo de energia pode variar entre o valor mínimo (\mathcal{E}_{min_i}), referente ao estado de *idle*, até o máximo (\mathcal{E}_{max_i}), que corresponde a 100% de uso do servidor. Como cada servidor pode ter um valor diferente de consumo mínimo e máximo, o parâmetro i serve para identifica o servidor. A Equação 2 é a função objetivo do problema MILP, que visa minimizar a parcela variável do consumo de energia mostrado na Equação 1 da Seção 3.4, já que o consumo em *idle* é considerado fixo. Em seguida, são apresentadas as restrições para solução do problema.

$$\text{minimizar } \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}_m} (\mathcal{E}_{max_i} - \mathcal{E}_{min_i}) \cdot \frac{\mathcal{V}_{m,j}}{\mathcal{S}_i} \cdot y_{i,m,j} \quad (2)$$

Tabela 1 - Notações utilizadas no problema.

Notação	Descrição	Tipo
\mathcal{I}	Servidores disponíveis	Conjunto
\mathcal{M}	Tipos de VNF disponíveis	Conjunto
\mathcal{J}_m	Instâncias disponíveis para a VNF m	Conjunto
\mathcal{R}_m	<i>Backups</i> disponíveis para a VNF m	Conjunto
\mathcal{Q}	Requisições de serviço na forma de SFCs	Conjunto
\mathcal{U}	Enlaces virtuais ligando apenas VNFs operacionais	Conjunto
\mathcal{U}'	Enlaces virtuais ligando uma VNF operacional e uma de <i>backup</i> , (u, v') e (u', v) , ou duas VNFs de <i>backup</i> , (u', v')	Conjunto
\mathcal{E}_{max_i}	Consumo de energia máximo do servidor i	Parâmetro
\mathcal{E}_{min_i}	Consumo de energia mínimo do servidor i	Parâmetro
\mathcal{S}_i	Capacidade dos recursos computacionais do servidor i	Parâmetro
$\mathcal{V}_{m,j}$	Recursos computacionais utilizados pela instância j da VNF do tipo m	Parâmetro
$\mathcal{K}_{m,r}$	Recursos computacionais utilizados pelo <i>backup</i> r da VNF do tipo m	Parâmetro
$\mathcal{F}_{m,q}$	Recursos computacionais da VNF do tipo m requeridos pela SFC q	Parâmetro
$\mathcal{B}_{i,n}$	Largura de banda do enlace físico (i, n)	Parâmetro
$\mathcal{W}_{q,u,v}$	Largura de banda solicitada pela SFC q no enlace virtual (u, v)	Parâmetro
\mathcal{T}_q	Atraso total permitido pela SFC q	Parâmetro
$\mathcal{D}_{i,n}$	Atraso de propagação associado ao enlace físico (i, n)	Parâmetro
\mathcal{P}_m	Atraso de processamento associado à VNF do tipo m	Parâmetro
m_q^o, m_q^f	Primeira VNF (o) de uma SFC q e última VNF (f) de uma SFC q	Parâmetro
m_q^u, m_q^v	VNF que está no nó u ou no nó v de um enlace virtual (u, v) qualquer da SFC q	Parâmetro
$y_{i,m,j}$	Valor binário indicando se a instância j da VNF do tipo m está alocada no servidor i	Variável
$z_{i,m,r}$	Valor binário indicando se o <i>backup</i> r da VNF do tipo m está alocada no servidor i	Variável
$a_{i,m,q}$	Valor binário indicando se a SFC q solicita uma VNF do tipo m no servidor i	Variável
$c_{i,m,q}$	Valor binário indicando se a SFC q solicita um <i>backup</i> da VNF do tipo m no servidor i	Variável
$b_{i,n,q,u,v}$	Valor binário indicando se a SFC q solicita um enlace operacional (u, v) no enlace físico (i, n)	Variável
$d_{i,n,q,u,v}$	Valor binário indicando se a SFC q solicita um enlace de <i>backup</i> (u, v) no enlace físico (i, n)	Variável
$x_{i,n,q,u,v}$	Valor binário indicando se algum dos enlaces de <i>backup</i> (u, v') , (u', v) ou (u', v') da SFC q está no enlace físico (i, n)	Variável
$g_{i,q,u,v}$	Valor binário indicando se o enlace virtual qualquer (u, v) da SFC q está no SRG afetado por uma possível falha no servidor i	Variável

4.2 Restrições de alocação de VNFs

A Equação 3 assegura que o consumo de recursos computacionais de todas as VNFs, operacionais e de *backup*, em um servidor não ultrapasse a capacidade do mesmo. De forma similar, a Equação 4 garante que o total de recursos demandados por todas as SFCs que utilizem uma VNF não ultrapasse a capacidade dela.

A Equação 5 afirma que, se uma SFC solicita recursos de uma VNF específica, então deve haver algum servidor para alocar essa VNF. Isso obriga a solução do problema a hospedar todas as VNFs necessárias nos servidores, de modo que não haja requisições de serviço não atendidas. A Equação 6 afirma que uma instância de uma determinada VNF não pode ser atribuída a mais de um servidor. Considerando que um provedor de serviços disponha de três licenças para instanciar três VNFs de um dado tipo, como um *firewall* ou um IDS, cada uma delas só pode estar alocada em uma máquina por vez, não podendo ser instanciada em vários servidores simultaneamente.

$$\text{sujeito a } \sum_{m \in \mathcal{M}} \left(\sum_{j \in \mathcal{J}_m} \mathcal{V}_{m,j} \cdot y_{i,m,j} + \sum_{r \in \mathcal{R}_m} \mathcal{K}_{m,r} \cdot z_{i,m,r} \right) \leq \mathcal{S}_i \quad \forall i \in \mathcal{I} \quad (3)$$

$$\sum_{q \in \mathcal{Q}} \mathcal{F}_{m,q} \cdot a_{i,m,q} \leq \sum_{j \in \mathcal{J}_m} \mathcal{V}_{m,j} \cdot y_{i,m,j} \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M} \quad (4)$$

$$\sum_{i \in \mathcal{I}} a_{i,m,q} = \begin{cases} 1, & \text{se } F_{m,q} \geq 0 \\ 0, & \text{se } F_{m,q} = 0 \end{cases} \quad \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (5)$$

$$\sum_{i \in \mathcal{I}} y_{i,m,j} \leq 1 \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J}_m \quad (6)$$

4.3 Restrições de alocação de enlaces

A Equação 7 representa a função lógica OU entre as três variáveis de enlace de *backup*, que é utilizada na Equação 8 para alocar a requisição de banda de apenas um dos três enlaces de *backup*. Ou seja, essa equação força $x_{i,n,q,u,v} = 1$ se pelo menos um dos enlaces de *backup* da SFC q estiver no enlace físico (i, n) .

A Equação 8 assegura que a largura de banda total utilizada pelos enlaces virtuais em um determinado enlace físico não ultrapasse a banda disponível no mesmo. Vale notar que, como é possível que dois ou três enlaces de *backup* de um mesmo enlace virtual operacional atravessem um *link* físico em comum, a reserva de banda ocorreria para cada um dos dois ou três enlaces de *backup* no cenário com uma falha. Isso não é necessário, já que, em caso de falha, basta que seja reservada a capacidade de apenas um desses enlaces de *backup*, ao invés de alocar desnecessariamente o dobro ou o triplo desse valor. Por exemplo, um enlace virtual principal (u, v) requer 50 Mbps. Consequentemente, seus enlaces de *backup* também devem requerer a mesma largura de banda para atender o serviço caso precisem ser utilizados. Se os três enlaces virtuais de *backup* de (u, v) estiverem no enlace físico (i, n) ao mesmo tempo, seria feita uma reserva de 150 Mbps naquele enlace, o que é um desperdício de largura de banda. Sendo assim, nesses casos, apenas a reserva de um dos enlaces de *backup* é necessária, que é de 50 Mbps no exemplo.

A Equação 9 garante que o atraso total em uma SFC não exceda o atraso máximo aceitável daquela SFC. Esse atraso é dado pela soma do atraso de processamento de cada VNF requerida e o atraso de propagação em cada um dos enlaces físicos que a SFC atravessa. Como as requisições de serviço também incluem valores máximos de latência, isso garante que as especificações de atraso não sejam violadas. Como a utilização das VNFs de *backup* ocorre apenas quando uma falha acontece e impede que o serviço seja provido pelo caminho principal da SFC, apenas a latência máxima do caminho principal

é considerada. Dessa forma, a única degradação do serviço possível é exceder o valor máximo de latência ao usar os *backups*, mas o provisionamento do serviço é garantido. Isso é razoável porque, enquanto a SFC segue pelos caminhos e VNFs de *backup*, a falha no caminho principal deve ser corrigida para que o caminho operacional principal volte a ser utilizado.

$$0 \leq 3 \cdot x_{i,n,q,u,v} - d_{i,n,q,u,v'} - d_{i,n,q,u',v} - d_{i,n,q,u',v'} \leq 2 \quad \forall i, n \in \mathcal{I}, \quad \forall q \in \mathcal{Q}, \quad \forall (u, v) \in \mathcal{U}' \quad (7)$$

$$\sum_{q \in \mathcal{Q}} \sum_{(u,v) \in \mathcal{U}} \mathcal{W}_{q,u,v} \cdot (b_{i,n,q,u,v} + x_{i,n,q,u,v}) \leq \mathcal{B}_{i,n} \quad \forall i, n \in \mathcal{I} \quad (8)$$

$$\sum_{i \in \mathcal{I}} \left(\sum_{n \in \mathcal{I}} \sum_{(u,v) \in \mathcal{U}} \mathcal{D}_{i,n} \cdot b_{i,n,q,u,v} + \sum_{m \in \mathcal{M}} \mathcal{P}_m \cdot a_{i,m,q} \right) \leq \mathcal{T}_q \quad \forall q \in \mathcal{Q} \quad (9)$$

4.4 Restrições de alocação de *backups*

A Equação 10 tem o mesmo propósito da Equação 4, porém se destina às VNFs de *backup* e às SFCs que as utilizam, garantindo que a capacidade disponível nas VNFs de *backup* não seja excedida. Da mesma forma, a Equação 11 é similar à Equação 5, mas está relacionada às VNFs de *backup*, obrigando todas as requisições de serviço a serem atendidas.

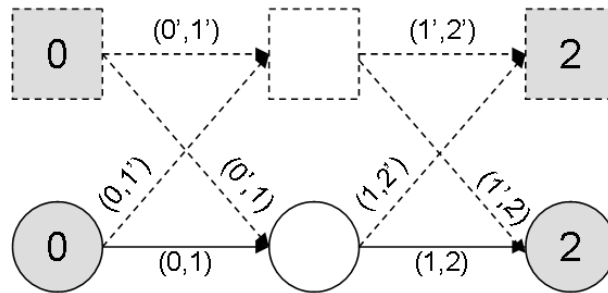
A Equação 12 impede que uma VNF operacional e o *backup* associado a ela estejam no mesmo servidor, o que poderia levar ao problema mostrado na Figura 11, no qual a falha de um servidor quebra a SFC e sua réplica. Isso impede que a SFC seja particionada e o serviço interrompido.

A Equação 13 tem a mesma função da Equação 6, porém voltada para os *backups* das VNFs. Uma instância específica só deve estar alocada em um servidor por vez.

$$\sum_{q \in \mathcal{Q}} \mathcal{F}_{m,q} \cdot c_{i,m,q} \leq \sum_{r \in \mathcal{R}_m} \mathcal{K}_{m,r} \cdot z_{i,m,r} \quad \forall i \in \mathcal{I}, \quad \forall m \in \mathcal{M} \quad (10)$$

$$\sum_{i \in \mathcal{I}} c_{i,m,q} = \begin{cases} 1, & \text{se } F_{m,q} \geq 0 \\ 0, & \text{se } F_{m,q} = 0 \end{cases} \quad \forall m \in \mathcal{M}, \quad \forall q \in \mathcal{Q} \quad (11)$$

Figura 11 - Quebra na SFC quando a VNF principal e o *backup* estão no mesmo servidor.



$$a_{i,m,q} + c_{i,m,q} \leq 1 \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (12)$$

$$\sum_{i \in \mathcal{I}} z_{i,m,r} \leq 1 \quad \forall m \in \mathcal{M}, \forall r \in \mathcal{R}_m \quad (13)$$

4.5 Restrições de fluxo

A Equação 14 especifica que o somatório de todos os fluxos que entram e saem de um servidor é igual ao somatório de todos os fluxos que surgem nele e se destinam a ele. Dessa forma, todo fluxo que chega a um servidor deve sair, a menos que seja direcionado a ele. A recíproca também é verdadeira, todo fluxo que sai de um servidor deve ter entrado, a menos que tenha se originado nele. É dito que um fluxo se origina ou se destina a um servidor se o mesmo contém a primeira ou a última VNF de uma SFC, respectivamente.

A Equação 15 afirma que um enlace virtual interno deve ser formado se duas VNFs operacionais de uma SFC estiverem no mesmo servidor. Esse enlace, por ser estabelecido internamente no servidor, possui banda infinita e atraso zero.

As Equações 16 e 17 asseguram que um enlace virtual, operacional ou de *backup*, só pode estar alocado em um sentido do enlace físico. Já as Equações 18 e 19 afirmam que um enlace virtual operacional só pode sair ou chegar, respectivamente, a um servidor por um único enlace físico. As Equações 20 e 21 fazem o mesmo para os enlaces de *backup*. Esse conjunto de restrições impede a formação de ciclos e também de enlaces virtuais

desnecessários, obrigando-os a serem posicionados corretamente sobre os enlaces físicos.

$$\sum_{n \in \mathcal{N}} (b_{i,n,q,u,v} - b_{n,i,q,u,v}) = a_{i,m_q^o,q} - a_{i,m_q^f,q} \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U} \quad (14)$$

$$0 \leq a_{i,m_q^u,q} + a_{i,m_q^v,q} - 2 \cdot b_{i,i,q,u,v} \leq 1 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U} \quad (15)$$

$$b_{i,n,q,u,v} + b_{n,i,q,u,v} \leq 1 \quad \forall i, n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U} \quad (16)$$

$$d_{i,n,q,u,v} + d_{n,i,q,u,v} \leq 1 \quad \forall i, n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U}' \quad (17)$$

$$\sum_{n \in \mathcal{I}} b_{i,n,q,u,v} \leq 1 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U} \quad (18)$$

$$\sum_{i \in \mathcal{I}} b_{i,n,q,u,v} \leq 1 \quad \forall n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U} \quad (19)$$

$$\sum_{n \in \mathcal{I}} d_{i,n,q,u,v} \leq 1 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U}' \quad (20)$$

$$\sum_{i \in \mathcal{I}} d_{i,n,q,u,v} \leq 1 \quad \forall n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U}' \quad (21)$$

As Equações 22, 23 e 24 têm a mesma função da Equação 14, estando relacionadas aos enlaces de *backup* que partem de uma VNF operacional para um *backup*, que partem de um *backup* para uma VNF operacional e que ligam duas VNFs de *backup*, respectivamente.

As Equações 25, 26 e 27 formam um paralelo com a Equação 15. São criados enlaces virtuais internos de *backup* quando estão em sequência, no mesmo servidor, uma VNF operacional e o *backup* da próxima, um *backup* e a VNF operacional seguinte e duas

VNFs de *backup*, respectivamente.

$$\sum_{n \in \mathcal{N}} (d_{i,n,q,u,v'} - d_{n,i,q,u,v'}) = a_{i,m_q^o,q} - c_{i,m_q^f,q} \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v') \in \mathcal{U}' \quad (22)$$

$$\sum_{n \in \mathcal{N}} (d_{i,n,q,u',v} - d_{n,i,q,u',v'}) = c_{i,m_q^o,q} - a_{i,m_q^f,q} \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u', v) \in \mathcal{U}' \quad (23)$$

$$\sum_{n \in \mathcal{N}} (d_{i,n,q,u',v'} - d_{n,i,q,u',v'}) = c_{i,m_q^o,q} - c_{i,m_q^f,q} \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u', v') \in \mathcal{U}' \quad (24)$$

$$0 \leq a_{i,m_q^u,q} + c_{i,m_q^v,q} - 2 \cdot d_{i,i,q,u,v'} \leq 1 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v') \in \mathcal{U}' \quad (25)$$

$$0 \leq c_{i,m_q^u,q} + a_{i,m_q^v,q} - 2 \cdot d_{i,i,q,u',v} \leq 1 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u', v) \in \mathcal{U}' \quad (26)$$

$$0 \leq c_{i,m_q^u,q} + c_{i,m_q^v,q} - 2 \cdot d_{i,i,q,u',v'} \leq 1 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u', v') \in \mathcal{U}' \quad (27)$$

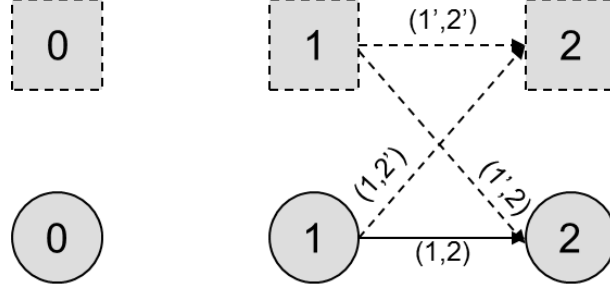
4.6 Restrições de resiliência e de Grupos de Risco Compartilhado (*Shared Risk Groups, SRGs*)

As Equações 28 e 29 definem uma função lógica OU que indica se um determinado enlace virtual pertence ao SRG de um servidor, ou seja, se uma possível falha no servidor em questão implica a queda do enlace virtual. Assim, sempre que um enlace virtual chegar, sair ou for um enlace interno em um determinado servidor, ele estará contido no grupo de enlaces afetados pela falha naquele servidor. Vale ressaltar que um enlace virtual pertence a tantos SRGs distintos quantos forem os servidores pelos quais ele passar.

Assim como a Equação 12, as Equações 30, 31 e 32 também garantem que o serviço não seja interrompido. A Equação 30 define que os quatro enlaces virtuais relacionados, isto é, o enlace operacional e seus três enlaces de *backup*, não podem estar simultaneamente no mesmo SRG, o que poderia levar ao problema mostrado na Figura 12.

Nas Equações 31 e 32, para um dado enlace virtual qualquer (u, v) , (v, z) é o enlace seguinte, tendo a VNF em v como origem, ao invés de destino. Supondo uma

Figura 12 - Quebra na SFC quando os quatro enlaces caem ao mesmo tempo.



SFC que comece em um *firewall*, siga para um *proxy* e termine em uma NAT, o enlace (u, v) é o que leva do *firewall* ao *proxy* e o enlace seguinte, (v, z) , segue do *proxy* para a NAT. Essas equações evitam uma ruptura espelhada na SFC. Isso ocorre quando os dois enlaces que chegam a uma VNF de *backup* e os dois enlaces que partem de sua VNF operacional correspondente e vice-versa ficam inacessíveis ao mesmo tempo, como mostrado na Figura 13. A SFC é interrompida porque, em ambos os casos, não há continuidade na VNF 1. O fluxo processado pelas VNFs do tipo 0 chega a uma VNF do tipo 1, mas não conseguem chegar a uma VNF do tipo 2. Da mesma forma, embora um fluxo processado pela VNF 1 possa alcançar a VNF 2, ele não foi processado anteriormente por uma VNF do tipo 0.

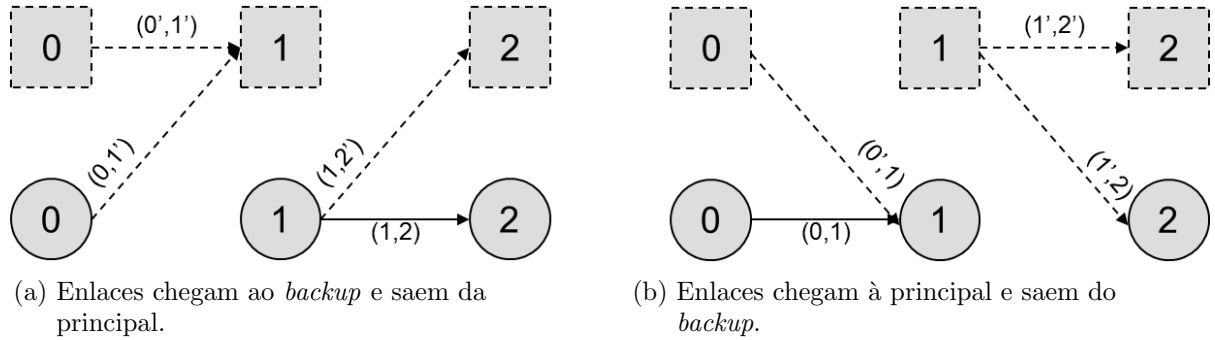
$$0 \leq (1 + (2 \cdot (n - 1))) \cdot g_{i,q,u,v} - \sum_{n \in \mathcal{N}} (b_{i,n,q,u,v} + b_{n,i,q,u,v}) + b_{i,i,q,u,v} \leq 2 \cdot (n - 1) \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v) \in \mathcal{U} \quad (28)$$

$$0 \leq (1 + (2 \cdot (n - 1))) \cdot g_{i,q,u,v} - \sum_{n \in \mathcal{N}} (d_{i,n,q,u,v} + d_{n,i,q,u,v}) + d_{i,i,q,u,v} \leq 2 \cdot (n - 1) \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v) \in \mathcal{U}' \quad (29)$$

$$g_{i,q,u,v} + g_{i,q,u,v'} + g_{i,q,u',v} + g_{i,q,u',v'} \leq 3 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v) \in \mathcal{U} \cup \mathcal{U}' \quad (30)$$

$$g_{i,q,u,v} + g_{i,q,u',v} + g_{i,q,v',z} + g_{i,q,v',z'} \leq 3 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v), (v, z) \in \mathcal{U} \cup \mathcal{U}' \quad (31)$$

Figura 13 - Quebra na SFC por uma descontinuidade espelhada.



$$g_{i,q,u,v'} + g_{i,q,u',v'} + g_{i,q,v,z} + g_{i,q,v,z'} \leq 3 \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v), (v,z) \in \mathcal{U} \cup \mathcal{U}' \quad (32)$$

4.7 Equações de domínio de variáveis

Da Equação 33 até a 40, são definidos os domínios das oito variáveis do problema de posicionamento de SFCs com resiliência. Todas as variáveis utilizadas são binárias e estão listadas na Tabela 1.

$$y_{i,m,j} \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \forall j \in \mathcal{J}_m \quad (33)$$

$$z_{i,m,r} \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \forall r \in \mathcal{R}_m \quad (34)$$

$$a_{i,m,q} \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (35)$$

$$c_{i,m,q} \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall m \in \mathcal{M}, \forall q \in \mathcal{Q} \quad (36)$$

$$b_{i,n,q,u,v} \in \{0,1\} \quad \forall i, n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u,v) \in \mathcal{U} \quad (37)$$

$$d_{i,n,q,u,v} \in \{0, 1\} \quad \forall i, n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v) \in \mathcal{U}' \quad (38)$$

$$x_{i,n,q,u,v} \in \{0, 1\} \quad \forall i, n \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v) \in \mathcal{U} \quad (39)$$

$$g_{i,q,u,v} \in \{0, 1\} \quad \forall i \in \mathcal{I}, \forall q \in \mathcal{Q}, \forall (u, v) \in \mathcal{U} \cup \mathcal{U}' \quad (40)$$

5 HEURÍSTICA TRELIS

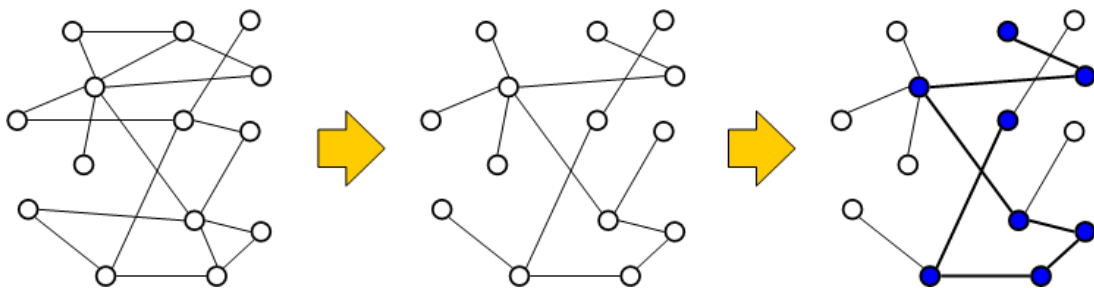
Dada a complexidade do problema MILP formulado, a sua solução não é escalável, como será visto na Seção 6.1. Assim, conforme o número de SFCs a serem posicionadas na infraestrutura cresce, o aumento do tempo de execução e da utilização de memória inviabilizam a obtenção da solução ótima. Essa questão da escalabilidade é um problema conhecido nos problemas MILP (HERRERA; BOTERO, 2016).

Como parte da solução proposta nesta dissertação, é desenvolvida uma heurística para aumentar a escalabilidade no posicionamento de SFCs. A ideia fundamental é reduzir o número de variáveis. De acordo com a Tabela 1, o número de servidores da infraestrutura é um índice presente em todas as variáveis e, sendo assim, reduzir a quantidade de servidores que podem hospedar VNFs diminui o número de variáveis e, conseqüentemente, de restrições do problema.

Em linhas gerais, a TRELIS recebe a rede original, remove enlaces que fecham ciclos e nós de sua topologia e cria uma rede em linha para ser utilizada no problema de posicionamento de SFCs. Nessa rede em linha, todos os nós possuem apenas dois enlaces, exceto os nós nas duas extremidades da linha, os quais possuem apenas um enlace cada. Sendo assim, a rede em linha não apresenta ciclos nem bifurcações em sua topologia. Como mostra a Figura 14, para a formação da rede em linha, a TRELIS recebe a rede original e então remove os ciclos da rede, transformando-a em uma *spanning tree*, que contém todos os nós da rede original. A partir dessa árvore, é selecionada uma quantidade inicial de nós que possam atender ao posicionamento das SFCs requeridas. Esse nós estão, então, conectados por uma topologia em linha. A Figura 14 mostra, em azul, uma rede em linha formada por 8 servidores a partir da rede original. Após a formação da rede em linha, a TRELIS fornece essa rede como entrada do problema de otimização. Caso o problema não seja solucionado, todo o procedimento anterior é repetido para um servidor a mais na rede em linha, até que o problema tenha solução.

O Algoritmo 1 detalha a heurística TRELIS. Da linha 1 até a 4, são definidos os

Figura 14 - Remoção de ciclos e seleção de nós em uma linha feita pela TRELIS.



valores iniciais de número de servidores, consumo de energia e posicionamento. O número inicial de servidores é baseado em uma estimativa do número de VNFs que podem ser alocadas neles, considerando as capacidades dos servidores e das VNFs. O consumo de energia é iniciado com E_{max} , que representa o maior gasto possível de energia na rede, definido como o custo máximo da rede ao ter todos os servidores operando em capacidade máxima, com 100% de seus recursos em uso. O posicionamento inicial é tido como vazio, já que não há uma solução para o problema até que o problema seja resolvido na linha 12. Para resolver o problema, a TRELIS calcula inicialmente a quantidade mínima de servidores da rede em linha necessária para a alocação das VNFs operacionais e de *backup*,

Algoritmo 1 - Heurística TRELIS.

DOCUMENTAÇÃO

TÍTULO

Heurística TRELIS

PROPÓSITO

Prover uma solução quase ótima ao problema de posicionamento de SFCs.

MÉTODO

Remover os ciclos da rede original, reduzir o número de servidores considerados na infraestrutura física e iterar para que o resultado se aproxime do valor ótimo.

ENTRADAS

redeOriginal: Rede original
requisicoes: Requisições de SFCs
 ϵ : Ganho mínimo

SAÍDAS

melhorPosicionamento: Posicionamento das SFCs
energia: Consumo de energia

OBSERVAÇÕES, RESTRIÇÕES, REQUISITOS

A iteração ocorre até que seja encontrada uma solução possível e o ganho entre uma rodada e outra seja inferior ao parâmetro de ganho ϵ . Caso uma nova iteração venha a fornecer resultados piores que a anterior, os valores antigos de saída são mantidos.

ALGORITMO TRELIS

declarar *minun, ganho, energia, posicionamento*

declarar *arvore, melhoresLinhasPossiveis*

1. $minun \leftarrow \text{arredondar}\left(\frac{2 \cdot nr \cdot VnfsRequisitadas}{nr \cdot MedioDeVnfsPorServidor}\right)$;
2. $ganho \leftarrow \epsilon + 1$;
3. $energia \leftarrow E_{max}$;
4. $posicionamento \leftarrow Vazio$;

— continua —

Algoritmo 1 - Heurística TRELIS. (continuação)

```

— continuação —
5. arvore ← spanningTree(redeOriginal);
6. melhoresLinhasPossiveis ← avaliarMelhoresCaminhosTodosPares(arvore);
7. enquanto (ganho >  $\epsilon$ ), fazer    {Até quando a TRELIS deve iterar}
8.   | ganhoAnterior ← ganho;
9.   | energiaAnterior ← energia;
10.  | posicionamentoAnterior ← posicionamento;
11.  | rede ← extrairLinha(minun, melhoresLinhasPossiveis);
12.  | energia, posicionamento ← resolverProblema(rede);
13.  | se (posicionamento ≠ Vazio), então    {Caso haja solução possível}
14.  |   | ganho ← energiaAnterior − energia;
15.  |   | senão
16.  |   |   | ganho ← ganhoAnterior;
17.  |   |   | energia ← energiaAnterior;
18.  |   |   | posicionamento ← posicionamentoAnterior;
19.  |   | fim se
20.  |   | minun += 1;
21.  | fim enquanto
22.  | se (ganho > 0), então    {Verifica se a última solução é a melhor}
23.  |   | melhorPosicionamento ← posicionamento;
24.  |   | senão
25.  |   |   | melhorPosicionamento ← posicionamentoAnterior;
26.  |   | fim se
FIM ALGORITMO
FIM DOCUMENTAÇÃO

```

com base nos recursos disponíveis. Por exemplo, se cada servidor é capaz de hospedar apenas 8 VNFs e as requisições precisam de 64 VNFs, então são necessários, no mínimo, 8 servidores na rede em linha. Esse tamanho mínimo é representado por *minun*. O número de VNFs requisitadas é multiplicado por 2 na linha 1 para considerar também as VNFs de *backup*.

Na linha 5, a *spanning tree* é formada a partir da rede original, ou seja, os ciclos são removidos. No entanto, a remoção de enlaces aqui descrita não significa que a topologia original seja alterada. Os enlaces removidos pela TRELIS são apenas desconsiderados no problema, sendo atribuídas a eles largura de banda zero e atraso de propagação infinito. Isso simplifica o problema e reduz o número de soluções possíveis.

Na linha 6, a função *avaliarMelhoresCaminhosTodosPares* calcula todos os caminhos mais curtos entre os pares de servidores, que consistem nas redes em linha utilizadas. A partir desses caminhos, seleciona-se um tamanho de rede com *i* nós e comparam-se todas as possibilidades de redes de tamanho *i*, registrando como melhor rede aquela cujo somatório da largura de banda de todos os enlaces tiver maior valor. Assim, tem-se a

melhor rede para cada tamanho possível, variando de 1 nó até a maior rede em linha possível de ser gerada.

A TRELIS itera da linha 7 até a 21, selecionando, na função `extrairLinha`, na linha 11, uma rede em linha com *minun* servidores. Essa rede é escolhida a partir dos caminhos calculados na linha 6 e o valor *minun* é incrementado a cada iteração.

Na função `resolverProblema`, na linha 12, a rede em linha escolhida é então fornecida como entrada do problema formulado no Capítulo 4. Em cada iteração, calcula-se o ganho na economia de energia em relação à iteração anterior. Se o *ganho* for menor que ϵ , a TRELIS termina a execução e escolhe o posicionamento da última iteração, já que, embora o resultado não seja suficientemente melhor a ponto de justificar uma nova iteração, ele ainda é um resultado melhor que o da penúltima iteração. Se o *ganho* for menor que zero, escolhe-se o posicionamento da penúltima iteração, nas linhas 22 a 26, já que esse resultado é melhor que o da última iteração. Isso garante que o resultado final obtido pela TRELIS seja sempre o melhor dentre todos os resultados obtidos nas iterações anteriores, evitando que o último resultado seja escolhido caso ele seja pior que o anterior.

6 AVALIAÇÃO DOS RESULTADOS

Para analisar os resultados obtidos pela TRELIS, utiliza-se o IBM CPLEX como ferramenta de otimização e são realizados três experimentos. Cada um utiliza três redes diferentes, seguindo as topologias das redes acadêmicas da RNP (RNP, 2014), GEANT (GEANT, 2016) e RENATER (RENATER, 2014).

O primeiro experimento envolve o posicionamento de SFCs curtas, isto é, com duas VNFs, e foi realizado tanto através da solução ótima quanto da TRELIS, a fim de comparar os tempos de execução obtidos em cada caso. O segundo e o terceiro experimentos envolvem, respectivamente, o posicionamento de SFCs longas, isto é, com quatro VNFs, e o posicionamento de SFCs com diferentes capacidades de VNF. Esses últimos experimentos foram realizados apenas para a heurística TRELIS. Foram escolhidas 2 e 4 VNFs para formar as SFCs como um caso de uso dos experimentos. As requisições de SFCs usam como pontos de entrada e saída os próprios servidores nos quais a primeira e a última VNF são instanciadas, respectivamente.

Os valores de energia dos servidores foram retirados de (BARI et al., 2016). Para todos os servidores, o consumo de energia em *idle* é de 80,5 W e o consumo de energia em carga máxima é de 2735 W.

Utiliza-se como rede física as redes acadêmicas RNP, GEANT e RENATER, empregando a mesma topologia, banda e latência de (COUTO et al., 2015). Emprega-se nos experimentos uma máquina com processador Intel Xeon E3-1270 V2 @ 3.50GHz e 32 GB de memória RAM DDR3 1333MHz.

6.1 Tempos de execução da solução ótima

Para avaliar os resultados da solução ótima, utiliza-se a configuração com SFCs curtas (isto é, 2 VNFs) e capacidade $\mathcal{S}_i = 1000 \forall i \in \mathcal{I}$. São utilizados dois tipos de VNF para representar esse caso de uso e cada instância requer $\mathcal{V}_{m,j} = 125 \forall m \in \mathcal{M}, \forall j \in \mathcal{J}_m$ unidades de capacidade de um servidor. Cada SFC possui duas VNFs e essas requisições são homogêneas, cada uma consumindo $\mathcal{F}_{m,q} = 100 \forall m \in \mathcal{M}, \forall q \in \mathcal{Q}$ unidades de recurso de uma VNF. Esses valores de capacidade são abstrações para as características dos servidores, como disco, memória, poder de processamento, *clock* e outros fatores, representando numericamente seu desempenho ou capacidade de forma abstrata.

No problema de otimização, a quantidade de nós da topologia tem grande impacto na quantidade de variáveis do problema. As redes da RNP, GEANT e RENATER possuem, respectivamente, 28, 41 e 48 nós. A Tabela 2 mostra como o tempo de execução cresce significativamente já para os primeiros valores de número de SFCs. Para posi-

Tabela 2 - Tempos de execução da solução ótima, em segundos, para SFC curtas e $\mathcal{S}_i = 1000$.

Topologia	Número de SFCs		
	1	2	4
RNP	0.8626	2.3002	7.9395
GEANT	4.6154	8.6275	43.0281
RENATER	5.6197	15.4545	59.6814

cionar apenas 4 SFCs na rede da RENATER, a solução ótima já leva um minuto para ser alcançada. A execução só é possível para até 4 SFCs. Isso pode ser explicado já que com 4 SFCs existem 49 mil variáveis e 64 mil restrições, justificando a proposta de uma heurística que reduza o tamanho do problema. A partir de 8 SFCs, o problema não alcança uma solução e consome toda a memória da máquina utilizada.

Como a solução ótima rapidamente se torna inviável já para o posicionamento de SFCs curtas, que é o experimento mais simples em termos de composição das SFCs, com apenas 2 VNFs em cada uma, os outros dois experimentos não foram realizados para essa abordagem.

Como os servidores da rede são considerados homogêneos, isto é, possuem os mesmos recursos e configurações, eles são indistinguíveis em termos de capacidade e consumo de energia. Dessa forma, o resultado da TRELIS, apresentado a seguir, é obrigatoriamente igual ao da solução ótima, já que os servidores que a heurística seleciona são equivalentes aos que a solução ótima utiliza para posicionar as SFCs. A TRELIS resolve o problema tendo conhecimento de apenas uma parte dos servidores disponíveis, isso poderia levá-la a não considerar os melhores servidores para realizar o posicionamento. Como não há diferença entre os servidores, o resultado não é influenciado. Portanto, os resultados da solução ótima foram omitidos aqui e estão disponíveis na Seção 6.2.1. Caso os servidores fossem heterogêneos, a TRELIS também chegaria a uma solução para o problema, embora, nesse caso, algum desvio em relação à solução fosse possível devido ao conhecimento de apenas uma parte da infraestrutura. No entanto, espera-se que esse desvio seja pequeno devido às sucessivas iterações da TRELIS, refinando os resultados com base no parâmetro de ganho do Algoritmo 1.

6.2 Resultados da TRELIS

Conforme visto na seção anterior, os resultados da solução ótima para o posicionamento de SFCs curtas não são escaláveis. Isso levou ao desenvolvimento da heurística TRELIS para resolver o problema de posicionamento de SFCs com resiliência para os três experimentos propostos. O primeiro realiza o posicionamento de SFCs curtas, com 2

Tabela 3 - Tempos de execução da heurística em segundos ($\mathcal{S}_i = 1000$).

Topologia	Número de SFCs				
	1	2	4	8	16
RNP	0.2597	0.0318	0.0589	0.6282	24.3213
GEANT	0.0335	0.0557	0.0630	0.2722	22.2596
RENATER	0.0558	0.0671	0.0711	0.3070	15.4008

Tabela 4 - Tempos de execução da heurística em segundos ($\mathcal{S}_i = 4000$).

Topologia	Número de SFCs					
	1	2	4	8	16	32
RNP	0.3760	0.0699	0.0728	0.1120	0.1460	1.5536
GEANT	0.0587	0.0698	0.0742	0.1167	0.1535	2.9250
RENATER	0.0685	0.0790	0.0828	0.1205	0.1575	1.5581

VNFs em cada, o segundo posiciona SFCs mais longas, com 4 VNFs em cada, e o terceiro posiciona VNFs com diferentes valores de capacidade.

6.2.1 Posicionamento de SFCs curtas

No posicionamento de SFCs curtas são considerados dois cenários, um com servidores de capacidade $\mathcal{S}_i = 1000 \forall i \in \mathcal{I}$, utilizado para o resultado ótimo da Seção 6.1, e outro com servidores de capacidade $\mathcal{S}_i = 4000 \forall i \in \mathcal{I}$. Dois tipos de VNF são utilizados e as instâncias requerem $\mathcal{V}_{m,j} = 125 \forall m \in \mathcal{M}, \forall j \in \mathcal{J}_m$ unidades de capacidade de um servidor. As SFCs são compostas por duas VNFs e cada solicitação requer $\mathcal{F}_{m,q} = 100 \forall m \in \mathcal{M}, \forall q \in \mathcal{Q}$ unidades de recurso de uma VNF.

As Tabelas 3 e 4 ilustram os tempos de execução obtidos com a TRELIS para o experimento de posicionamento de SFCs curtas. De modo geral, o tempo aumenta para mais SFCs, devido ao aumento do número de servidores necessários. Comparando os tempos de execução da solução ótima e da heurística, mostrados, respectivamente, pelas Tabelas 2 e 3, nota-se que a TRELIS reduz consideravelmente o tempo. Além disso, resolve-se o problema para até 16 SFCs no caso de servidores com capacidade 1000.

Como a TRELIS utiliza uma quantidade reduzida de servidores da rede para entregar ao CPLEX, apenas a rede em linha formada a partir da topologia original é considerada. A quantidade inicial de servidores necessários é calculada com base no valor *minun*, que depende da quantidade de VNFs solicitadas. Por exemplo, se 4 servidores forem suficientes para atender a todas as SFCs, a rede em linha terá apenas 4 servidores,

independente do número original de servidores da rede. Com isso, as três topologias são reduzidas a uma rede em linha com a mesma quantidade de nós, pois a requisição de serviços é igual para todas. Por isso, a diferença entre os tempos de execução entre as três redes é muito pequena. Assim, a TRELIS é capaz de encontrar uma solução para o problema independente do tamanho da rede original, de modo que a maior quantidade de nós na rede da RENATER não impacta mais no tempo de execução da solução. Entretanto, quanto mais servidores forem necessários, isto é, quanto maior for o número de SFCs, maior será a rede em linha, o que impactará no tempo de execução e na escalabilidade da heurística.

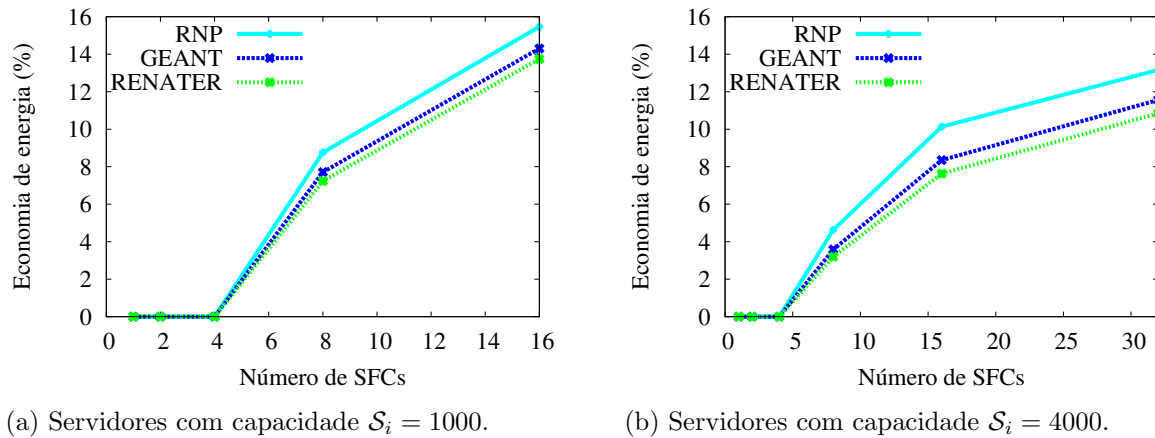
Observa-se que os valores da Tabela 3 para 16 SFCs são superiores aos da Tabela 4 até para 32 SFCs, pois os servidores possuem menos capacidade no primeiro caso, sendo assim necessários mais servidores na rede em linha para solucionar o problema. Ao aumentar a capacidade de um servidor, mais VNFs podem ser instanciadas nele e mais SFCs podem ser atendidas sem que haja a necessidade de se utilizar mais servidores. Como os servidores de capacidade 4000 possuem quatro vezes mais capacidade, são necessários menos servidores para alocar as SFCs. Como a quantidade de servidores é o principal fator de aumento da complexidade do problema, os tempos de execução do segundo cenário são menores.

As Figuras 15a e 15b mostram a redução no consumo de energia devido ao compartilhamento de VNFs por SFCs. A economia de energia é definida como o quanto o consumo de energia dos servidores da rede diminui devido ao compartilhamento de VNFs. A Equação 41 mostra como é calculada a economia de energia utilizada nos três experimentos. Os valores de $Consumo_{completo}$ e $Consumo_{compartilhamento}$ são ambos tirados da função objetivo, na Equação 2. A diferença entre esses dois valores está na quantidade de VNFs alocadas nos servidores, o que muda o valor da variável binária $y_{i,m,j}$ no somatório. Como há mais VNFs instanciadas quando não há compartilhamento, há mais variáveis $y_{i,m,j}$ com valor 1 do que com menos instâncias e é por isso que o $Consumo_{completo}$ é sempre maior que o $Consumo_{compartilhamento}$ para uma dada quantidade de SFCs a serem posicionadas.

Para o caso sem compartilhamento, o consumo de energia é completo, porque há uma VNF para cada solicitação de VNF na SFC. Sempre que uma SFC chega, são alocadas novas VNFs. Quando ocorre o compartilhamento, o consumo de energia é dado por $Consumo_{compartilhamento}$ e as SFCs buscam utilizar as VNFs que já estiverem alocadas nos servidores, ao invés de sempre alocar instâncias novas. Sendo assim, sempre que uma SFC chega, novas VNFs podem ou não ser alocadas. Com isso, a economia de energia é alcançada ao alocar menos VNFs e compartilhá-las entre as SFCs.

$$Economia_{rede} = \frac{Consumo_{completo} - Consumo_{compartilhamento}}{Consumo_{completo}} \cdot 100\% \quad (41)$$

Figura 15 - Economia de energia devido ao compartilhamento de VNFs por SFCs curtas.



Nas Figuras 15a e 15b, com até 4 SFCs, não há economia de energia devido aos valores de capacidade das VNFs serem muito próximos à capacidade requerida pelas SFCs, de modo que a alocação até esse ponto é feita de uma VNF para uma SFC, como no caso sem compartilhamento. Como a capacidade das VNFs é $\mathcal{V}_{m,j} = 125$ e a capacidade solicitada pelas SFCs é $\mathcal{F}_{m,q} = 100$, a relação é de 1 para 1 até 4 SFCs. Com 4 VNFs, a capacidade delas é de 500 e a capacidade solicitada é de 400. Com 5 SFCs, a capacidade solicitada é de 500, mas não é necessário alocar uma quinta VNF, pois as quatro têm juntas uma capacidade igual a 500, o que é suficiente para atender as 5 SFCs. É por isso que a economia de energia só aparece no gráfico a partir de 8 SFCs.

Para 16 SFCs, os servidores com capacidade 1000 obtêm melhores valores de economia de energia do que os servidores com capacidade 4000. Isso acontece porque a economia está relacionada à quantidade de recursos alocados e de recursos totais. Como a capacidade dos servidores aumenta e a alocação se mantém fixa, já que a requisição de 16 SFCs é a mesma, a proporção de recursos alocados diminui, diminuindo a porcentagem da economia de energia.

O posicionamento de 16 SFCs leva à alocação de 26 VNFs com o compartilhamento, ao invés das 32 VNFs quando não há compartilhamento. Como $\mathcal{V}_{m,j} = 125$, a economia de capacidade é de 750. Esse valor é mais significativo em uma rede com servidores de capacidade 1000, mas é menos impactante se comparado aos servidores de capacidade 4000. Por isso, a economia de energia ao posicionar 16 SFCs é maior ao usar servidores de capacidade 1000. No entanto, as maiores capacidades permitem que mais SFCs sejam posicionadas e a economia melhora gradualmente. Enquanto a economia chega a 16% para 16 SFCs com servidores de capacidade 1000, a economia de energia se aproxima desse valor para 32 SFCs com servidores de capacidade 4000, alcançando 14% de economia de energia.

Como a rede é considerada homogênea, com todos os servidores possuindo os mesmos valores de consumo de energia e recursos, o resultado da heurística coincide com o da

solução ótima, como visto na Seção 6.1. Assim, a comparação com os resultados ótimos foi omitida. Além disso, com uma rede homogênea, o valor de ϵ utilizado foi E_{max} para que o Algoritmo 1 retorne assim que achar uma solução. Da mesma forma, como os servidores são homogêneos, a diferença no consumo de energia entre as três topologias se deve à quantidade de servidores disponíveis. Como a RENATER possui mais servidores que as outras duas, são mais servidores em *idle* consumindo energia, o que faz com que ela tenha valores menores de economia. A partir de 4 SFCs, como mostram as Figuras 15a e 15b, há economia de energia com a heurística proposta. Além disso, quanto mais requisições de SFCs, mais energia é economizada, já que há mais VNFs sendo compartilhadas.

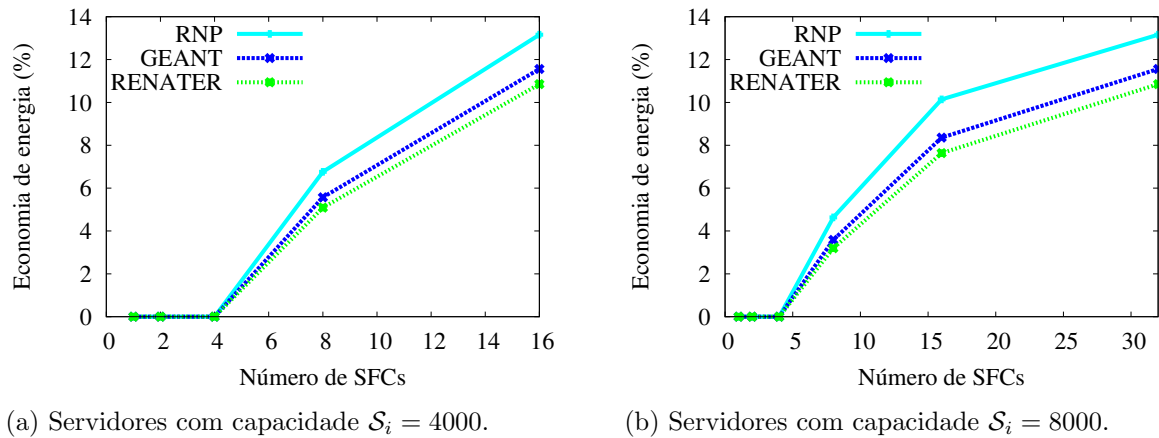
6.2.2 Posicionamento de SFCs longas

No posicionamento de SFCs longas, os servidores possuem capacidades $\mathcal{S}_i = 4000$ e $\mathcal{S}_i = 8000 \forall i \in \mathcal{I}$. São utilizados quatro tipos de VNF, cada SFC possui quatro VNFs e os demais parâmetros do primeiro experimento são mantidos. A capacidade dos servidores é aumentada em relação ao posicionamento de SFCs curtas porque as SFCs agora requerem mais VNFs e, conseqüentemente, mais capacidade. Esses parâmetros são variados para observar como o tamanho das SFCs influencia no compartilhamento de VNFs e na economia de energia. Embora também seja um valor viável, a capacidade $\mathcal{S}_i = 1000$ não é utilizada nesse experimento para observar a economia de energia com uma quantidade maior de SFCs posicionadas, o que requer mais recursos dos servidores.

Ao utilizar mais tipos de VNF em relação ao experimento anterior, as oportunidades de compartilhamento são reduzidas, pois o compartilhamento de VNFs por SFCs deve obedecer ao tipo da função solicitada. Por exemplo, uma instância de *proxy* não é capaz de atender às solicitações de um serviço de NAT. Como as SFCs passam a requerer mais VNFs, aumentou-se a capacidade dos servidores utilizados para 4000 e 8000. Com mais tipos de VNF sendo alocados, o compartilhamento só começa quando há uma quantidade maior de VNFs já alocadas. Por exemplo, o posicionamento de 16 SFCs curtas envolve a solicitação de 32 VNFs, mas o compartilhamento de 26 instâncias já atende a essas SFCs. Já o posicionamento de 8 SFCs longas também envolve a solicitação de 32 VNFs, mas são necessárias 28 VNFs para atender a essas SFCs.

A Figura 16 mostra a economia de energia para o experimento de SFCs longas. Como no experimento de SFCs curtas, é possível notar que o compartilhamento de VNFs também começa após o posicionamento de 4 SFCs. Isso é esperado, pois o que determina o início do compartilhamento é a razão entre a capacidade requerida de uma VNF pela SFC e a capacidade da instância da VNF. Como ambos os parâmetros são os mesmos do primeiro experimento, a energia começa a ser economizada apenas com 8 SFCs. No entanto, ao se considerar a quantidade de VNFs instanciadas, o compartilhamento no

Figura 16 - Economia de energia devido ao compartilhamento de VNFs por SFCs mais longas.



segundo experimento demora mais para acontecer, já que existem menos oportunidades de compartilhamento. Nas Figuras 15b e 16a, o compartilhamento começa a partir de 8 SFCs, o que significa que há 16 e 32 instâncias de VNFs, respectivamente, alocadas na rede. Dessa forma, a economia de energia só acontece depois de se ter mais VNFs alocadas nos servidores, mas ainda assim demonstra como a TRELIS é capaz de posicionar tanto SFCs curtas quanto longas e economizar energia em ambos os casos.

Uma vez que o compartilhamento se torna possível, o crescimento da economia de energia ocorre de modo semelhante para ambas as Figuras 15 e 16, demonstrando que o tamanho das SFCs não impacta significativamente na economia de energia, já que a economia de energia nas Figuras 15b e 16a atingem valores próximos, variando de 10% a 14%, quando possuem 64 VNFs operacionais instanciadas, nos valores de 32 e 16 SFCs posicionadas, respectivamente. Essa comparação é possível porque ambos possuem um valor comum de capacidade, que é 4000.

As Figuras 16a e 16b mostram que é possível atingir uma economia de energia equivalente para os dois valores de capacidade adotados no segundo experimento, de modo que em ambos os casos a rede da RNP alcança uma economia próxima de 14% para 16 e 32 SFCs, respectivamente. Esses números de SFCs posicionadas são razoáveis, pois representam o posicionamento de um lote de SFCs para uma execução da TRELIS. Nota-se que a economia cresce mais rapidamente para o cenário de servidores com capacidade 4000. Por exemplo, a rede da RNP já economiza quase 8% de energia para 8 SFCs posicionadas, enquanto economiza pouco mais de 4% de energia para a mesma quantidade de SFCs ao usar servidores com capacidade 8000. Isso ocorre porque a economia de energia é baseada na quantidade de SFCs que podem ser posicionadas na infraestrutura sem precisar alocar uma nova VNF em algum servidor. Como o valor das requisições de capacidade de VNF pelas SFCs é fixo ($\mathcal{F}_{m,q} = 100 \forall m \in \mathcal{M}, q \in \mathcal{Q}$), sua proporção em relação à capacidade dos servidores diminui conforme ele aumenta. Por exemplo, com

16 SFCs longas, há 64 requisições de VNFs, mas apenas 52 precisam ser instanciadas devido ao compartilhamento. Com isso, são economizadas 1500 unidades de capacidade de VNF, independente da capacidade dos servidores. Por essa economia ser fixa para um dado número de SFCs, a proporção entre capacidade economizada e capacidade total diminui. Economizar 1500 unidades de capacidade possui maior impacto em uma rede com servidores de capacidade 1000 se comparado a uma rede com servidores de capacidade 8000. Além disso, assim como no primeiro experimento, os valores de economia das redes maiores, como a RENATER, foram inferiores. Isso ocorre porque essas redes possuem mais servidores em suas infraestruturas, que consomem energia mesmo em *idle*. Da mesma forma, quanto mais SFCs são posicionadas, mais instâncias de VNFs são alocadas e podem ser compartilhadas.

6.2.3 Posicionamento de SFCs com diferentes capacidades de VNF

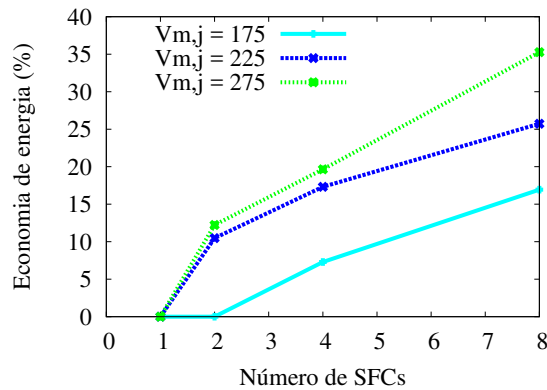
No posicionamento de SFCs com diferentes capacidades de VNF, são utilizados os servidores apenas com capacidade $\mathcal{S}_i = 8000 \forall i \in \mathcal{I}$ e a capacidade das instâncias de VNFs varia, requerendo $\mathcal{V}_{m,j} = 175$, $\mathcal{V}_{m,j} = 225$ ou $\mathcal{V}_{m,j} = 275 \forall m \in \mathcal{M}, \forall j \in \mathcal{J}_m$ unidades de capacidade de um servidor, mantendo os demais parâmetros do segundo experimento. Nesse experimento, o objetivo é variar apenas a capacidade das VNFs e observar como ocorre a utilização de VNFs para cada tamanho, observando o impacto da capacidade das VNFs nas oportunidades de compartilhamento, no desperdício de recursos e na economia de energia. Uma VNF com mais recursos pode ser compartilhada por mais SFCs, aumentando as oportunidades de compartilhamento. No entanto, usar uma VNF maior para posicionar uma SFC também aumenta o desperdício de recursos, já que há mais recursos alocados e não utilizados do que haveria se uma VNF menor fosse utilizada. Da mesma forma, ao reduzir o desperdício de recursos, menor é o consumo, melhorando a economia de energia.

Para o terceiro experimento, o objetivo é analisar como a variação da capacidade das instâncias de VNFs, expressa por $\mathcal{V}_{m,j}$, impacta tanto no consumo de energia quanto na economia. São utilizadas três capacidades de VNF diferentes para verificar se é mais eficiente utilizar muitas instâncias de baixa capacidade ou poucas instâncias com muita capacidade.

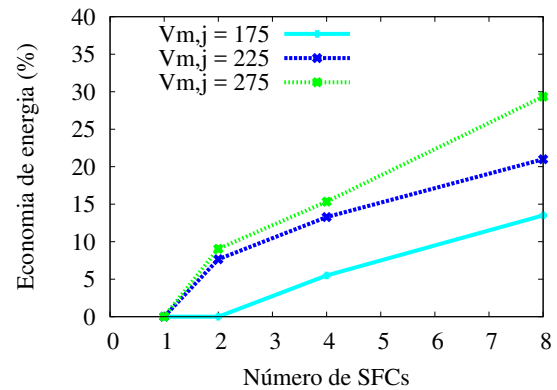
A Figura 17 mostra a economia de energia para cada tamanho proposto. Dessa vez, há um gráfico para cada rede, a fim de verificar a relação entre os diferentes valores de capacidades de VNFs ao invés de comparar as redes entre si.

Os resultados mostram que, ao utilizar instâncias de VNFs com maior capacidade, as oportunidades de compartilhamento já surgem para menos SFCs em relação ao experimento anterior. Isso acontece porque mais SFCs podem ser posicionadas sem que novas

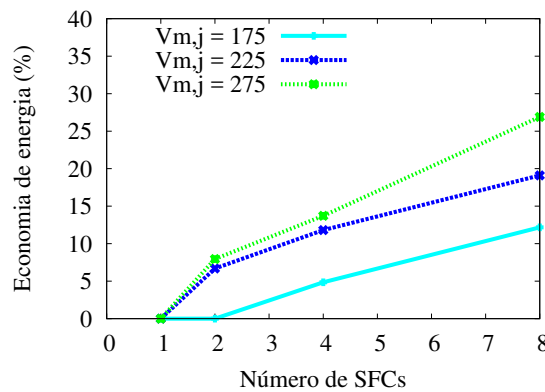
Figura 17 - Economia de energia com VNFs de diferentes capacidades ($\mathcal{S}_i = 8000$).



(a) RNP.



(b) GEANT.



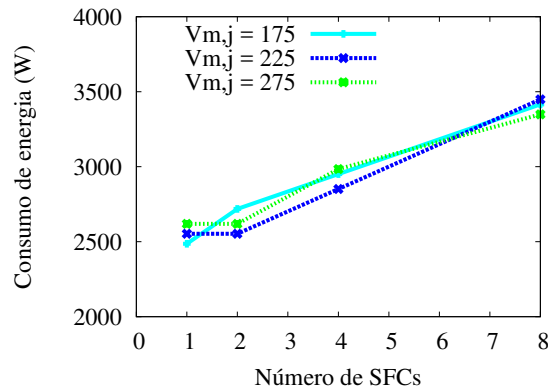
(c) RENATER.

instâncias sejam alocadas na infraestrutura. Nas Figuras 16a e 16b, o compartilhamento de VNFs tem início a partir de 8 SFCs, utilizando VNFs com capacidade $\mathcal{V}_{m,j} = 125 \forall m \in \mathcal{M}, j \in \mathcal{J}_m$. Na Figura 17, o compartilhamento começa a ocorrer a partir de 2 SFCs para as instâncias de tamanho 225 e 275. Assim, ao usar instâncias com pouco menos que o dobro da capacidade, o compartilhamento de VNFs já se iniciou a partir de um quarto da quantidade de SFCs necessárias no segundo experimento.

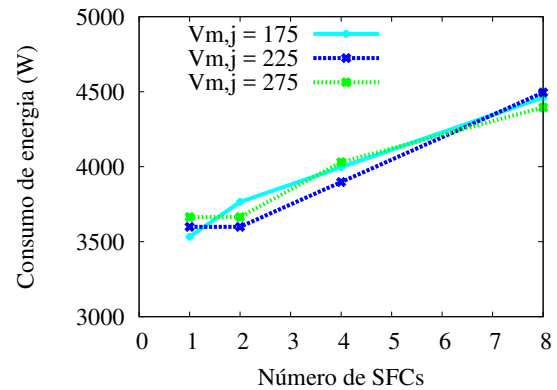
Os resultados também mostram que os valores de economia de energia crescem consideravelmente ao utilizar instâncias maiores, podendo atingir mais de 35% de economia com 8 SFCs e VNFs de capacidade 275. Isso se deve à própria natureza do compartilhamento de VNFs pelas SFCs. Como todas as requisições consomem apenas 100 unidades da capacidade de uma VNF, à medida que os recursos alocados a essa VNF aumentam, a parcela de recursos alocados e não consumidos também aumenta, gerando desperdício de capacidade. Uma VNF com maior capacidade está mais apta a ser compartilhada pelas SFCs justamente por possuir mais recursos excedentes e que podem ser consumidos por outra SFC.

Além de verificar a economia de energia devido ao compartilhamento de VNFs para

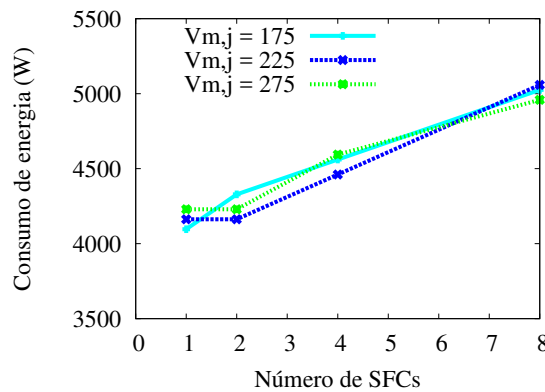
Figura 18 - Consumo de energia para instâncias de VNFs com diferentes capacidades.



(a) RNP.



(b) GEANT.



(c) RENATER.

cada tamanho utilizado, o consumo de energia das três redes também é medido, já que são possíveis diferentes combinações de consumo e VNFs instanciadas. Como o consumo de energia de um servidor é proporcional à capacidade das VNFs instanciadas nele, instâncias que utilizem mais recursos do equipamento consomem mais energia. Porém, quanto maior a capacidade das VNFs, menor é a quantidade de instâncias necessárias, o que reduz o consumo de energia.

A Figura 18 mostra os valores de consumo de energia para as três redes. Devido à utilização de servidores homogêneos, a diferença entre os valores de energia de cada rede se deve exclusivamente à quantidade de servidores. Como todos os servidores estão ativos na rede, a quantidade de servidores em *idle* aumenta, assim como o consumo da rede.

O eixo vertical das Figuras 18a, 18b e 18c foi deslocado em cada uma para melhorar a visualização dos resultados. A diferença entre as curvas torna-se sutil demais ao usar um eixo vertical comum para todas, dificultando a visualização. Isso ocorre porque as capacidades das VNFs variam de 50 a 100 unidades, enquanto o consumo de energia da rede é da ordem de milhares de unidades de capacidade.

As Figuras 18a, 18b e 18c possuem curvas muito semelhantes entre si porque o que

Tabela 5 - Razão entre a capacidade total requerida e a capacidade do total de VNFs.

Número de SFCs	Capacidades das VNFs		
	175	225	275
1	0.5714	0.4444	0.3636
2	0.5714	0.8889	0.7273
4	0.7619	0.8889	0.7273
8	0.9143	0.8889	0.9697

diferencia as redes da RNP, GEANT e RENATER é a quantidade de servidores na rede. Como todos possuem um valor de consumo em *idle* de 80.5 W, a quantidade de servidores funciona como um ajuste de *offset*. Deslocando as curvas da RNP para cima, obtém-se as da GEANT, e, deslocando um pouco mais, obtém-se as da RENATER.

Analisando as Figuras 18a, 18b e 18c, observa-se que o consumo de energia se mantém o mesmo para as capacidades 225 e 275 para as duas primeiras SFCs. Como essas instâncias possuem maior capacidade, posicionar tanto uma quanto duas SFCs tem exatamente o mesmo impacto no consumo de energia, já que esse só aumenta quando mais VNFs precisam ser instanciadas e não quando mais SFCs são posicionadas. Isso não ocorre para a capacidade 175, já que 2 SFCs requerem 200 unidades de capacidade de VNF, uma quantidade de recursos maior do que a instância de capacidade 175 é capaz de oferecer. Com relação ao número de VNFs instanciadas, quanto menor for a capacidade, mais frequentemente novas VNFs precisam ser alocadas nos servidores. Por outro lado, utilizar instâncias de grande capacidade diminui a quantidade de VNFs necessárias e mais SFCs podem ser posicionadas sem aumentar o número de VNFs na rede. Além disso, ter menos VNFs instanciadas na infraestrutura diminui o número de variáveis do problema e torna o seu gerenciamento mais fácil para o MANO. Com 8 SFCs posicionadas, foram necessárias 20, 16 e 12 instâncias com as capacidades 175, 225 e 275, respectivamente, o que indica como a capacidade de compartilhamento de VNFs pelas SFCs está diretamente relacionada à capacidade de cada VNF instanciada.

Ainda observando a Figura 18, é possível notar que não existe um padrão sobre qual tamanho de VNF é o ideal para reduzir o consumo de energia. Para alocar apenas uma SFC, utilizar a VNF de capacidade 175 é mais vantajoso, pois alocar uma VNF maior apenas aumentaria a parcela de recursos alocados e não consumidos, o que não apenas é um desperdício de recursos, mas também aumenta o consumo de energia. Para alocar 2 e 4 SFCs, a VNF de capacidade mediana se mostrou mais eficiente, enquanto que a VNF de capacidade 275 obtém melhores valores de consumo de energia para 8 SFCs posicionadas. Para entender essa não uniformidade com relação ao consumo de energia e o tamanho das VNFs, a Tabela 5 apresenta a relação entre a capacidade total requerida pelas SFCs e a capacidade de todas as VNFs instanciadas na infraestrutura. Quanto mais próximo de 1

é o valor, menor é o desperdício por capacidade alocada e não consumida.

Analisando a Tabela 5, percebe-se que o menor desperdício para cada quantidade de SFCs posicionadas coincide com os resultados obtidos pelos gráficos da Figura 18. Dessa forma, não é possível dizer qual tamanho de VNF é o mais indicado, já que isso varia com o número de SFCs e suas requisições de capacidade. Entretanto, é correto afirmar que o mais indicado é instanciar VNFs de modo que a relação entre recursos posicionados e recursos alocados seja sempre o mais próximo possível de 1, o que minimiza o desperdício de energia. Como a função objetivo da TRELIS é minimizar o consumo de energia, ela naturalmente aloca as VNFs de modo que essa relação sempre seja o mais próximo possível de 1.

A Tabela 5 mostra ainda que utilizar as VNFs de capacidade 275 obteve os melhores e os piores valores para desperdício de recursos, consumindo 3200 unidades de capacidade das 3300 alocadas para 8 SFCs e consumindo 400 das 1100 alocadas para 1 SFC, respectivamente. Por outro lado, usar as VNFs com menor capacidade leva a valores menores de desperdício de uma forma geral. O pior aproveitamento de recursos para as VNFs com menor capacidade é de 0.5714, sendo ainda assim melhor que os piores resultados para as capacidades 225 e 275, com taxas de 0.4444 e 0.3636, respectivamente. Embora cada capacidade de VNF possa consumir menos energia para um dado número de SFCs posicionadas, as VNFs de capacidade 225 são, de forma geral, as mais indicadas. Isso é razoável, pois se obtém um aproveitamento de recursos próximo de 90% para três dos quatro números de SFCs utilizados. Dessa forma, as VNFs de capacidade 225 podem ser inferiores às de capacidade 175 e 275 para 1 e 8 SFCs, respectivamente, mas desperdiçam menos recursos alocados em geral.

7 CONCLUSÃO

Esta dissertação verificou que é possível desenvolver mecanismos no contexto de NFV que permitem o posicionamento energeticamente eficiente de SFCs, através do compartilhamento de VNFs pelas SFC. Além disso, devido à virtualização das funções de rede, os *backups* posicionados permanecem desligados durante a operação normal da rede. Assim, é possível oferecer resiliência sem prejudicar a eficiência energética.

Conforme a literatura, este trabalho é um dos primeiros esforços a tratar do posicionamento ótimo de SFCs com resiliência em NFV. Dada a complexidade do problema formulado, a solução ótima não escala, então a heurística TRELIS foi desenvolvida para resolver o posicionamento ótimo de SFCs com resiliência. Os resultados mostraram que é possível reduzir em até 15% o consumo de energia nos servidores com a estratégia adotada, suportando até uma falha silenciosa de um dos servidores e contornando a interrupção do serviço fornecido pelas SFCs através de VNFs de *backup*. Além disso, para uma mesma quantidade de requisições, se uma VNF tiver uma capacidade maior, aumentando as chances de compartilhamento, a economia de energia pode chegar até a 35% em relação ao cenário sem compartilhamento, ainda que servidores com maiores capacidades sejam necessários. Dessa forma, a heurística TRELIS cumpre o que foi prometido e consegue minimizar o consumo de energia enquanto oferece resiliência para até uma falha de servidor. A resiliência é verificada através da própria formulação do problema MILP, pois é garantida pelas restrições. Se não for possível garantir a resiliência para até uma falha de servidor, então o problema simplesmente não encontra uma solução possível.

Utilizando a heurística TRELIS, foi verificado que a redução do número de servidores entregues ao CPLEX melhora consideravelmente os tempos de execução. Assim, a TRELIS permite alcançar cenários maiores de posicionamento de SFCs em relação à solução de otimização, que apresenta problemas de escalabilidade, típicos dos problemas MILP. Como o número de servidores está presente em todas as variáveis do problema de otimização, a utilização de apenas parte dos servidores disponíveis na infraestrutura reduz a quantidade de todas as variáveis e restrições consideradas. Com isso, a escalabilidade limitada da solução ótima levou ao desenvolvimento da heurística TRELIS.

Para analisar o funcionamento da TRELIS e solucionar o problema de posicionamento de SFCs com resiliência, foram realizados três experimentos, posicionando SFCs curtas, SFCs longas e variando a capacidade das VNFs. As SFCs curtas possuem 2 VNFs, as longas possuem 4 VNFs e as capacidades consideradas no terceiro experimento são 175, 225 e 275.

Para o experimento de posicionamento de SFCs curtas, realizado tanto para a solução ótima quanto para a heurística, a TRELIS foi capaz de alocar quatro vezes mais SFCs em um quarto do tempo. Enquanto a solução ótima posiciona 4 SFCs em um minuto

na rede da RENATER, a TRELIS posiciona 16 SFCs em 15 segundos. Como o número de servidores impacta na quantidade de todas as variáveis consideradas, reduzi-lo se mostrou eficiente para alocar mais SFCs e reduzir os tempos de execução em relação à solução de otimização. Ao quadruplicar a capacidade dos servidores no primeiro experimento, é possível alocar o dobro de SFCs em aproximadamente um décimo do tempo para a rede da RENATER em relação à capacidade 1000, o que mostra como o número de servidores utilizados impacta no tempo de execução para uma determinada quantidade de SFCs posicionadas.

Ao posicionar SFCs longas, a quantidade de instâncias utilizadas na formação das SFCs também possui impacto na economia de energia. Ao utilizar mais tipos de VNF para compor os serviços, as oportunidades de compartilhamento e, conseqüentemente, de economizar energia são reduzidas. Isso fica evidente ao comparar os experimentos de posicionamento de SFCs curtas e longas. Como as SFCs só podem compartilhar VNFs do mesmo tipo, somente com um número maior de instâncias alocadas é que começam a surgir os benefícios do compartilhamento. No segundo experimento, o compartilhamento começa para 8 SFCs posicionadas, assim como no primeiro experimento, mas a quantidade de VNFs alocadas na infraestrutura é o dobro, justamente porque as SFCs são formadas por quatro VNFs, ao invés de duas.

No experimento com diferentes capacidades de VNF, um aspecto observado foi a relação entre recursos alocados e recursos consumidos, que se mostrou fundamental para a economia de energia. Embora a alocação de VNFs com capacidade considerada média tenha obtido resultados melhores de forma geral, as VNFs pequenas e grandes também foram as mais indicadas em certos casos, dependendo da quantidade de SFCs posicionadas. Essa não uniformidade levou à observação da proporção entre recursos alocados e recursos consumidos e foi constatado que, quanto menor for a diferença entre os recursos alocados e consumidos, isto é, quanto menor for o desperdício de recursos, maior será a economia de energia. Dessa forma, além de o desperdício de recursos reduzir o número de SFCs a mais que podem ser posicionadas, já que fragmentos de recursos espalhados pelas VNFs instanciadas em vários servidores não podem ser aproveitados, esses recursos ociosos também aumentam o consumo de energia da rede.

Também foi visto que o compartilhamento de VNFs leva à aglutinação das instâncias nos mesmos servidores. Isso é esperado, pois, ao juntar VNFs do mesmo tipo na mesma máquina, suas capacidades são reunidas e a soma dos fragmentos de recursos que possam ser utilizados juntos aumenta as chances de aceitar uma nova SFC. Essa estratégia de compartilhamento é feita, por exemplo, em (BARI et al., 2016; MEHRAGHDAM; KELLER; KARL, 2014; MOENS; TURCK, 2014; LUIZELLI et al., 2015; ADDIS et al., 2015).

Embora o desperdício de recursos afete diretamente a economia de energia, ele não tem impacto sobre a resiliência para o modelo de consumo de energia considerado.

Como as VNFs de *backup* não precisam estar em funcionamento ao mesmo tempo que as operacionais, elas são postas em *cold standby* e não consomem os recursos do servidor. Dessa forma, elas não contribuem para aumentar o consumo de energia, já que um servidor sem VNFs ou apenas com *backups* tem o consumo relativo ao estado de *idle*.

A solução não busca espalhar as VNFs, o que pode ser considerado uma técnica de resiliência, já que minimiza as perdas decorrentes da queda de um servidor. Mesmo que muitas VNFs possam ser alocadas em um ponto único de falha, a estratégia de resiliência com enlaces e VNFs de *backup* contorna esse problema. Mesmo que um servidor contenha todas as VNFs operacionais instanciadas e ele cair, as VNFs de *backup* estarão em outros servidores e nenhuma SFC é comprometida.

A resiliência é alcançada através da replicação de enlaces e VNFs de *backup* de modo que esses *backups* nunca fiquem todos inoperantes ao mesmo tempo. Assim como as VNFs operacionais, as próprias VNFs de *backup* também são preferencialmente alocadas em conjunto para aproveitar os fragmentos de capacidade. Isso é feito para também minimizar o número de instâncias de *backup* alocadas, poupando os recursos dos servidores. Embora o consumo de energia dos *backups* não tenha sido considerado, pois os mesmos podem ficar desligados ou em *standby*, a TRELIS também é capaz de minimizar o consumo de energia dos *backups* caso seja necessário.

Ainda com relação à resiliência, a estratégia de replicação de enlaces e VNFs nas SFCs serviram ao seu propósito e as restrições de resiliência e SRG do problema de otimização conseguem criar caminhos de *backup* independentes do caminho principal. Além disso, mesmo que qualquer um dos servidores se torne inoperante, todos os serviços das SFCs posicionadas ainda encontram condições para serem concluídos. Em nenhum momento as restrições de resiliência limitaram os experimentos ou levaram a soluções impossíveis. Da mesma forma, a estratégia de resiliência também não interfere no uso da energia, já que os *backups* estão apenas alocados e não são utilizados enquanto o caminho principal ainda estiver disponível.

7.1 Direções futuras

Como direções futuras, destacam-se a extensão do problema para mais de uma falha de servidor e também para falhas específicas de enlace, além de testar configurações heterogêneas de servidores, verificando o quanto a heurística se distancia da solução ótima. É esperado que a solução da TRELIS seja ligeiramente diferente por ter conhecimento apenas de uma parte dos servidores da rede. No entanto, como a TRELIS itera para melhorar os resultados obtidos, aumentando a quantidade de servidores considerada em cada etapa, espera-se que seus resultados sejam suficientemente próximos dos da solução ótima. Com essas novas configurações, se torna possível analisar o desvio da solução

da TRELIS em relação à solução ótima. Além disso, é possível expandir o modelo de representação das SFCs para incluir pontos fixos de entrada e saída, possivelmente nas extremidades da rede, como identificação de onde as requisições são feitas e para onde se destinam.

Novos experimentos serão realizados para analisar se a alocação de SFCs em duas etapas, posicionando primeiro as VNFs operacionais e depois as VNFs de *backup*, traz benefícios à heurística TRELIS, verificando o impacto que as duas abordagens têm na economia de energia. Além disso, o problema formulado considera requisições de SFCs em lote, alocadas em uma única execução. Na prática, requisições individuais podem ser realizadas sob demanda. O problema de otimização e a heurística podem ser adaptados para alocar as requisições individuais ou em grupos de SFCs, atualizando os valores de capacidade de servidores, VNFs e enlaces à medida que seus recursos são consumidos pelas SFCs. Com isso, também podem ser simulados clientes sintéticos que enviem requisições de posicionamento de SFCs à TRELIS.

Os parâmetros de capacidade dos servidores, VNFs e requisições podem ser expandidos para considerar individualmente memória, número de processadores, espaço de armazenamento em disco e outros fatores. Além de utilizar valores reais de configurações de servidores e VNFs, é possível considerar a variação, na prática, das solicitações de capacidade das VNFs pelos clientes.

Com relação à economia de energia, cita-se a criação de um novo cenário com novos valores de consumo de energia, variando o consumo mínimo, em *idle*, e o consumo máximo, em que todos os recursos da máquina são utilizados, para analisar como esses valores interferem na economia de energia. Também é possível definir uma parcela de consumo de energia para as VNFs de *backup* e verificar como isso impacta na economia de energia.

Como os recursos desperdiçados consomem energia e não são aproveitados para prover serviços, a combinação entre diferentes capacidades de VNF, ou seja, o quanto ela consome dos recursos físicos do servidor, pode ser utilizada para alcançar um ponto ótimo. Isso permite verificar se há alguma combinação ótima de instâncias que leve aos melhores valores de consumo de energia para todas as SFCs ou, pelo menos, para a maior parte delas.

Em uma etapa seguinte, é possível simular um NFVO que realize o chaveamento do caminho principal da SFC para algum dos caminhos de *backup*. O problema de posicionamento de SFCs com resiliência e a TRELIS alocam os recursos necessários nos servidores e nos enlaces e fornecem o mapeamento das VNFs e das SFCs na infraestrutura física antes da operação. No decorrer do funcionamento do sistema, caso ocorra uma falha, é preciso orientar o encaminhamento do fluxo das SFCs atingidas para o caminho alternativo previamente estabelecido. Além disso, é possível testar opções de *cold standby*, que foram consideradas nesta dissertação, e de *hot standby*, avaliando o tempo de recuperação

e a economia de energia em cada caso. Estratégias de variação do *clock* das máquinas e períodos de hibernação também podem ser considerados para atuar em conjunto com a TRELIS para elevar os níveis de economia de energia.

REFERÊNCIAS

- ADDIS, B. et al. Virtual network functions placement and routing optimization. In: IEEE. *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*. [S.l.], 2015. p. 171–177.
- BARI, M. F. et al. Orchestrating virtualized network functions. *Aceito para publicação em IEEE Transactions on Network and Service Management*, IEEE, 2016.
- BOTERO, J. F. et al. Energy efficient virtual network embedding. *IEEE Communications Letters*, IEEE, v. 16, n. 5, p. 756–759, 2012.
- BOUTEN, N. et al. Semantic validation of affinity constrained service function chain requests. In: *2016 IEEE NetSoft*. [S.l.: s.n.], 2016. p. 202–210.
- BRUSCHI, R.; CARREGA, A.; DAVOLI, F. A game for energy-aware allocation of virtualized network functions. *Journal of Electrical and Computer Engineering*, Hindawi Publishing Corp., v. 2016, p. 2, 2016.
- CHOWDHURY, N. M. K.; BOUTABA, R. Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, IEEE, v. 47, n. 7, 2009.
- CLAYMAN, S. et al. The dynamic placement of virtual network functions. In: IEEE. *2014 IEEE Network Operations and Management Symposium (NOMS)*. [S.l.], 2014. p. 1–9.
- COSTA-PÉREZ, X. et al. Latest trends in telecommunication standards. *ACM SIGCOMM Computer Communication Review*, ACM, v. 43, n. 2, p. 64–71, 2013.
- COUTO, R. S. et al. Network design requirements for disaster resilience in IaaS clouds. *IEEE Communications Magazine*, v. 52, n. 10, p. 52–58, out. 2014.
- COUTO, R. S. et al. Otimização do posicionamento de servidores físicos em centros de dados resilientes a desastres. In: *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. [S.l.: s.n.], 2015. p. 417–430.
- ETSI. *ETSI GS NFV 002 V1.1.1: Network Functions Virtualization (NFV); Architectural Framework*. 2013.
- ETSI. *ETSI GS NFV-REL 001 V1.1.1: Network Functions Virtualization (NFV); Resiliency Requirements*. 2015.
- FISCHER, A. et al. Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 15, n. 4, p. 1888–1906, 2013.
- FLOODLIGHT: Openflow controller - project floodlight. 2013. Disponível em: <http://www.projectfloodlight.org/floodlight/>. Acesso em: 7 fev. 2017.
- FORNEY, G. D. The Viterbi algorithm. *Proceedings of the IEEE*, IEEE, v. 61, n. 3, p. 268–278, 1973.

- FRY, M. et al. Challenge identification for network resilience. In: IEEE. *Next Generation Internet (NGI), 2010 6th EURO-NF Conference on*. [S.l.], 2010. p. 1–8.
- GEANT: pan-european network. 2016. Disponível em: http://www.geant.org/Networks/Pan-European_network/Pages/Home.aspx. Acesso em: 8 fev. 2017.
- GUO, T. et al. Shared backup network provision for virtual network embedding. In: IEEE. *2011 IEEE international conference on communications (ICC)*. [S.l.], 2011. p. 1–5.
- HAIDER, A.; POTTER, R.; NAKAO, A. Challenges in resource allocation in network virtualization. In: *20th ITC Specialist Seminar*. [S.l.: s.n.], 2009. v. 18, p. 20.
- HAN, B. et al. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, IEEE, v. 53, n. 2, p. 90–97, 2015.
- HAWILO, H. et al. NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). *IEEE Network*, IEEE, v. 28, n. 6, p. 18–26, 2014.
- HERRERA, J. G.; BOTERO, J. F. Resource allocation in NFV: A comprehensive survey. *IEEE Transactions on Network and Service Management*, IEEE, v. 13, n. 3, p. 518–532, 2016.
- IBM: Cplex optimizer. 2013. Disponível em: <https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>. Acesso em: 7 fev. 2017.
- JAIN, R.; PAUL, S. Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, IEEE, v. 51, n. 11, p. 24–31, 2013.
- JOHN, W. et al. Research directions in network service chaining. In: IEEE. *IEEE SDN for Future Networks and Services (SDN4FNS)*. [S.l.], 2013. p. 1–7.
- KREUTZ, D. et al. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, 2015.
- LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop: rapid prototyping for software-defined networks. In: ACM. *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. [S.l.], 2010. p. 19.
- LEE, G. et al. Optimal flow distribution in service function chaining. In: ACM. *The 10th International Conference on Future Internet*. [S.l.], 2015. p. 17–20.
- LEVITIN, G.; XING, L.; DAI, Y. Cold vs. hot standby mission operation cost minimization for 1-out-of-n systems. *European Journal of Operational Research*, Elsevier, v. 234, n. 1, p. 155–162, 2014.
- LUIZELLI, M. C. et al. Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions. In: IEEE. *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. [S.l.], 2015. p. 98–106.
- MA, W.; MEDINA, C.; PAN, D. Traffic-aware placement of NFV middleboxes. In: *IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2015. p. 1–6.

- MEHRAGHDAM, S.; KELLER, M.; KARL, H. Specifying and placing chains of virtual network functions. In: *IEEE 3rd International Conference Cloud Networking (CloudNet)*. [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: IEEE. *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. [S.l.], 2015. p. 1–9.
- MIJUMBI, R. et al. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, IEEE, v. 18, n. 1, p. 236–262, 2015.
- MOENS, H.; TURCK, F. D. VNF-P: A model for efficient placement of virtualized network functions. In: *10th International Conference on Network and Service Management (CNSM) and Workshop*. [S.l.: s.n.], 2014. p. 418–423.
- NUNES, B. A. A. et al. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 3, p. 1617–1634, 2014.
- RAHMAN, M. R.; AIB, I.; BOUTABA, R. Survivable virtual network embedding. In: SPRINGER. *International Conference on Research in Networking*. [S.l.], 2010. p. 40–52.
- RAHMAN, M. R.; BOUTABA, R. SVNE: Survivable virtual network embedding algorithms for network virtualization. *IEEE Transactions on Network and Service Management*, IEEE, v. 10, n. 2, p. 105–118, 2013.
- RENATER: Connecting knowledge - network services. 2014. Disponível em: https://www.renater.fr/spip.php?page=categorie&id_rubrique=424. Acesso em: 8 fev. 2017.
- RIERA, J. F. et al. Modelling the NFV forwarding graph for an optimal network service deployment. In: IEEE. *2015 17th International Conference on Transparent Optical Networks (ICTON)*. [S.l.], 2015. p. 1–4.
- RNP: Rede ipê. 2014. Disponível em: <https://www.rnp.br/servicos/conectividade/rede-ipe>. Acesso em: 8 fev. 2017.
- ROSA, R. et al. Network function virtualization: Perspectivas, realidades e desafios. *Minicursos XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos Florianópolis*, 2014.
- SAHHAF, S. et al. Network service chaining with optimized network function embedding supporting service decompositions. *Computer Networks*, Elsevier, v. 93, p. 492–505, 2015.
- SEKAR, V. et al. The middlebox manifesto: enabling innovation in middlebox deployment. In: ACM. *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. [S.l.], 2011. p. 21.
- STERBENZ, J. P. et al. Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation. *Telecommunication systems*, Springer, v. 52, n. 2, p. 705–736, 2013.

STERBENZ, J. P. et al. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, Elsevier, v. 54, n. 8, p. 1245–1265, 2010.

SU, S. et al. Energy-aware virtual network embedding through consolidation. In: IEEE. *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. [S.l.], 2012. p. 127–132.

TANENBAUM, A. S.; STEEN, M. V. *Distributed systems*. [S.l.]: Prentice-Hall, 2007.

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