

Usuários de nuvens computacionais esperam executar suas aplicações em Infraestruturas Virtuais (IVs) eficientes e confiáveis, enquanto provedores de serviço (InPs) objetivam maximizar lucros e reduzir a fragmentação do substrato físico. Nesse contexto, é proposto um modelo e resultados experimentais para alocar recursos considerando a coordenação conjunta de confiabilidade e fragmentação.

Orientador: Rafael Rodrigues Obelheiro

Coorientador: Guilherme Piêgas  
Koslovski

Joinville, 2014

ANO  
2014

GUSTAVO ANDRIOLLI DE SIQUEIRA CAVALCANTI | ALOCAÇÃO ÓTIMA  
DE RECURSOS PARA INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS



UNIVERSIDADE DO ESTADO DE SANTA CATARINA -  
UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO  
APLICADA

DISSERTAÇÃO DE MESTRADO

ALOCÇÃO ÓTIMA DE  
RECURSOS PARA  
INFRAESTRUTURAS  
VIRTUAIS CONFIÁVEIS

GUSTAVO ANDRIOLLI DE SIQUEIRA  
CAVALCANTI

JOINVILLE, 2014



**GUSTAVO ANDRIOLLI DE SIQUEIRA CAVALCANTI**

**ALOCÇÃO ÓTIMA DE RECURSOS PARA  
INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS**

Dissertação apresentada ao Programa de Pós-Graduação *Stricto Sensu* em Computação Aplicada, do Centro de Ciências Tecnológicas, da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Computação.

Orientador: Prof. Dr. Rafael Rodrigues Obelheiro

**JOINVILLE, SC  
2014**

## FICHA CATALOGRÁFICA

C376a Cavalcanti, Gustavo Andriolli de Siqueira  
Alocação ótima de recursos para infraestruturas virtuais  
confiáveis / Gustavo Andriolli de Siqueira Cavalcanti. -  
2014.

86 p. : il. ; 21 cm

Orientador: Rafael Rodrigues Obelheiro  
Co-orientador: Guilherme Piegas Koslovski  
Bibliografia: p. 81-86

Dissertação (mestrado) - Universidade do Estado de Santa  
Catarina, Centro de Ciências Tecnológicas, Mestrado  
Acadêmico em Computação Aplicada, Joinville, 2014.

1. Computação aplicada 2. Sistemas de computação 3. Redes de  
computadores e sistemas distribuídos 4. Nuvens computacionais  
I. Obelheiro, Rafael Rodrigues II. Koslovski, Guilherme  
Piêgas III. Universidade do Estado de Santa Catarina.  
Programa de Pós-Graduação em Computação Aplicada. IV.  
Alocação ótima de recursos para infraestruturas virtuais  
confiáveis

CDD: 004 - 20.ed.

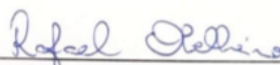
GUSTAVO ANDRIOLLI DE SIQUEIRA CAVALCANTI

ALOCÇÃO ÓTIMA DE RECURSOS PARA  
INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS

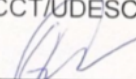
Dissertação apresentada ao Curso de Mestrado Acadêmico  
Computação Aplicada como requisito parcial para obtenção do  
título de Mestre em Computação Aplicada na área de  
concentração "Ciência da Computação".

**Banca Examinadora**

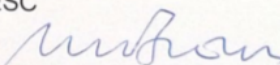
Orientador:

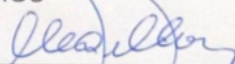
  
\_\_\_\_\_  
Prof. Dr. Rafael Rodrigues Obelheiro  
CCT/UDESC

Coorientador:

  
\_\_\_\_\_  
Prof. Dr. Guilherme Piêgas Koslovski  
CCT/UDESC

**Membros**

  
\_\_\_\_\_  
Prof. Dr. Antonio Marinho Pilla Barcellos  
UFRGS

  
\_\_\_\_\_  
Prof. Dr. Maurício Aronne Pilon  
CCT/UDESC

Joinville, SC, 11 de julho de 2014.



Dedico este trabalho à minha mãe, ao meu pai, aos meus dois irmãos e à minha companheira.





## AGRADECIMENTOS

Muitas pessoas, direta e/ou indiretamente, me apoiaram na realização deste trabalho, contudo algumas se tornaram notáveis, então eu as agradeço de coração:

Aos membros da banca da defesa final, professores Maurício Pillon e Marinho Barcellos, e da de projeto, Charles Miers, pelas valiosas contribuições. E um especial agradecimento aos meus orientadores, Rafael Obelheiro e Guilherme Koslovski, por serem modelos de inspiração e excelência acadêmica.

Aos meus professores do mestrado, pelos esforços dedicados: Rafael Obelheiro, Maurício Pillon, Marcelo Hounsell, Cristiano Vasconcellos e Marco Wehrmeister.

E, por fim, às minhas irmãs e aos irmãos, que são familiares e amigxs, pela compreensão e amorosidade. Principalmente, agradeço à Susana Huller.



Uma mulher sem um homem é como um  
peixe sem uma bicicleta.

(Gloria Steinem)



## RESUMO

CAVACLANTI, Gustavo Andriolli de Siqueira. **Alocação Ótima de Recursos para Infraestruturas Virtuais Confiáveis**. 2014. 86 f. Dissertação (Mestrado em Computação Aplicada - Área: Sistemas Computacionais) – Universidade do Estado de Santa Catarina. Programa de Pós-Graduação em Computação Aplicada, Joinville, 2014.

Com a consolidação do provisionamento dinâmico de Infraestruturas Virtuais (IVs) – conjuntos de máquinas virtuais interconectadas por enlaces e roteadores também virtuais –, provedores de serviço (InPs) enfrentam um desafio na escolha da melhor abordagem para alocação e reserva de recursos computacionais e de comunicação. É fato que a alocação de recursos (como CPU, disco, memória, largura de banda) é um problema complexo que precisa satisfazer diferentes objetivos: usuários esperam executar suas aplicações em IVs eficientes e confiáveis, enquanto InPs objetivam maximizar lucros, minimizar custos e reduzir a fragmentação do substrato físico. Sobretudo, há uma dicotomia entre maximizar a confiabilidade de IVs, alocando recursos esparsos para diminuir o impacto de falhas no substrato, e minimizar a fragmentação do substrato, co-allocando IVs. Nesse contexto, é proposto um modelo de programação inteira mista para alocar recursos considerando a coordenação conjunta de confiabilidade e fragmentação. Resultados experimentais mostram que é possível aprimorar a confiabilidade de IVs sem impactar significativamente na fragmentação do substrato.

**Palavras-chave:** Computação em Nuvem. Virtualização. Confiabilidade. Otimização. Cloud Networking.



## ABSTRACT

CAVACLANTI, Gustavo Andriolli de Siqueira. **Optimal Resource Allocation for Survivable Virtual Infrastructures**. 2014. 86 f. Dissertation (Master Degree in Applied Computing - Field: Computer Systems) – Santa Catarina State university. Graduate Program in Applied Computing, Joinville, 2014.

Virtual infrastructures (VIs) – sets of virtual machines interconnected by links and routers also virtual – consolidated the dynamic provisioning of computing and communication resources and their services providers (InPs) face a challenge in choose the better approach to allocating and reserve these resources. Resource allocation (such as CPU, disk, memory, bandwidth) is a complex problem that needs to satisfy different goals: users expect to run their applications on survivable VIs, while InPs aim to maximize profits, minimize costs and reduce substrate fragmentation. However, there is a dichotomy between maximizing VI survivability, by sparsely allocating resources to decrease the impact of substrate failures, and minimizing substrate fragmentation, by co-locating VIs. In this context, we propose a mixed integer programming model to allocate resources considering the joint coordination of survivability and fragmentation. Experimental results show that it is possible to enhance VI survivability without significantly impacting substrate fragmentation.

**Keywords:** Cloud Computing. Virtualization. Survivability. Optimization. Cloud Networking.





## LISTA DE FIGURAS

Figura 1	A reserva de uma rede virtual em uma rede física.....	32
Figura 2	Modelos de serviços de computação em nuvem. ....	36
Figura 3	Modelos de implementação de nuvens computacionais.	37
Figura 4	A reserva de uma infraestrutura virtual em uma infraes- trutura física.....	38
Figura 5	Relação entre entidades de uma IV.....	39
Figura 6	Hierarquia recursiva de papéis entre um PS e um InP. .	40
Figura 7	Um exemplo um substrato físico $s$ . ....	42
Figura 8	Um exemplo de duas IVs $v_a$ e $v_b$ .....	42
Figura 9	Uma alocação $m^f$ , exemplo de mapeamento entre o sub- trato $s$ (Figura 7) as IVs $v_a$ e $v_b$ (Figura 8). ....	43
Figura 10	Uma alocação $m^{fr}$ , exemplo de mapeamento entre o substrato $s$ (Figura 7) as IVs $v_a$ e $v_b$ (Figura 8). ....	43
Figura 11	O substrato $s$ (Fig. 7) com tipos de domínios de faltas distintos (circuito elétrico, <i>rack</i> , servidor e enlace). ....	47
Figura 12	Topologia da infraestrutura física. ....	66
Figura 13	Resultados experimentais de falhas locais de nó para os cenários BC e AC.....	69
Figura 14	Resultados experimentais de falhas globais de nó para os cenários BC e AC. ....	70
Figura 15	Resultados experimentais de falhas locais de enlace para os cenários BC e AC. ....	71
Figura 16	Resultados experimentais de falhas globais de enlace para os cenários BC e AC.....	72
Figura 17	Resultados experimentais de fragmentações de nó e en- lace para os cenários BC e AC.....	72
Figura 18	Resultados experimentais de taxas de aceitação para os cenários BC e AC.....	73



## LISTA DE TABELAS

Tabela 1	Métricas de falha (%) para as Figuras 9 e 10. ....	47
Tabela 2	Métricas de fragmentação (%) para as Figuras 9 e 10. ....	50
Tabela 3	Notação para o modelo de alocação de grafo.....	53
Tabela 4	Notação para o modelo de confiabilidade.....	57
Tabela 5	Notação para o modelo de fragmentação. ....	61
Tabela 6	Tamanho das infraestruturas físicas e virtuais. ....	66
Tabela 7	Cenários avaliados. ....	67
Tabela 8	Comparação entre os trabalhos relacionados. ....	78



## LISTA DE ABREVIATURAS E SIGLAS

AC	Alta Carga
API	<i>Application Programming Interface</i> – Interface de Programação de Aplicação
BC	Baixa Carga
CPU	<i>Central Processing Unit</i> – Unidade Central de Processamento
DC	<i>Data Center</i>
DF	Domínio de Falta
E/S	Entrada/Saída
EV	Enlace Vritual
EF	Enlace Físico
FG	Falha Global
FL	Falha Local
FrE	Fragmentação de Enlace
FrN	Fragmentação de Nó
IaaS	<i>Infrastructure as a Service</i> – Infraestrutura como um Serviço
IF	Infraestrutura Física
InP	<i>Infrastructure Provider</i> – Provedor de Infraestrutura
ISP	<i>Internet Service Provider</i> – Provedor de Internet
IV	Infraestrutura Virtual
MF	Máquina Física
MV	Máquina Virtual
PaaS	<i>Platform as a Service</i> – Plataforma como um Serviço
PLIM	Programação Linear Inteira Mista
PS	Provedor de Serviço
QoS	<i>Quality of Service</i> – Qualidade do Serviço
RV	Rede Virtual

SF	<i>Switch Físico</i>
SaaS	<i>Software as a Service – Software como um Serviço</i>
SO	Sistema Operacional
SV	<i>Switch Virtual</i>
SLA	<i>Service Level Agreement – Acordo de Nível de Serviço</i>
TI	Tecnologia da Informação
VLAN	<i>Virtual Local Area Network</i>
VPN	<i>Virtual Private Network</i>
WAN	<i>Wide Area Network</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>25</b>
1.1	OBJETIVOS .....	27
1.2	PRINCIPAIS CONTRIBUIÇÕES .....	27
1.3	ORGANIZAÇÃO DO TRABALHO .....	28
<b>2</b>	<b>ALOCAÇÃO DE REDES E INFRAESTRUTURAS VIRTUAIS .....</b>	<b>29</b>
2.1	REDES VIRTUAIS .....	29
<b>2.1.1</b>	<b>Princípios da Virtualização .....</b>	<b>30</b>
<b>2.1.2</b>	<b>Virtualização de Redes .....</b>	<b>32</b>
2.2	INFRAESTRUTURAS VIRTUAIS .....	34
<b>2.2.1</b>	<b>Computação em Nuvem .....</b>	<b>34</b>
<b>2.2.2</b>	<b>Virtualização de Infraestruturas Computacionais ....</b>	<b>37</b>
2.3	ALOCAÇÃO DE RECURSOS VIRTUAIS .....	40
2.4	CONSIDERAÇÕES PARCIAIS .....	44
<b>3</b>	<b>MODELO DE ALOCAÇÃO DE RECURSOS PARA INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS ...</b>	<b>45</b>
3.1	CONFIABILIDADE E FRAGMENTAÇÃO EM INFRAESTRUTURAS VIRTUAIS .....	45
<b>3.1.1</b>	<b>Confiabilidade .....</b>	<b>46</b>
<b>3.1.2</b>	<b>Fragmentação .....</b>	<b>49</b>
<b>3.1.3</b>	<b>Visão Geral do Modelo .....</b>	<b>50</b>
3.2	ALOCAÇÃO DE GRAFOS .....	51
3.3	CONFIABILIDADE .....	56
3.4	FRAGMENTAÇÃO .....	61
3.5	MODELO COMBINADO .....	62
3.6	CONSIDERAÇÕES PARCIAIS .....	63
<b>4</b>	<b>AVALIAÇÃO .....</b>	<b>65</b>
4.1	AMBIENTE DE TESTES .....	65
4.2	RESULTADOS .....	68
4.3	CONSIDERAÇÕES PARCIAIS .....	73
<b>5</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>75</b>
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>79</b>
	<b>REFERENCIAS .....</b>	<b>83</b>





## 1 INTRODUÇÃO

A computação em nuvem introduziu o provisionamento dinâmico de recursos e serviços virtualizados guiados pelos requisitos dos usuários. Esse conceito revolucionou o gerenciamento de Tecnologia da Informação (TI) ao entregar computação, rede e armazenamento como serviços. Neste cenário, usuários não necessitam investimentos iniciais significativos para compor o parque de TI, podendo apenas alugar um conjunto de serviços ofertado por provedores de infraestruturas (InPs, *infrastructure providers*) seguindo um modelo *pay-as-you-go* [Mell e Grance 2011]. Atualmente, InPs podem oferecer infraestruturas completamente virtualizadas, possibilitando que usuários executem suas aplicações em conjuntos confinados e privados de recursos. Estas *infraestruturas virtuais* (IVs) podem ser definidas como um conjunto de máquinas virtuais (MVs) interconectadas por recursos virtuais de rede (enlaces, *switches* e roteadores) [Anhalt, Koslovski e Primet 2010].

Interrupções não planejadas no funcionamento de *data centers* (DCs) são comuns. Um estudo recente [Ponemon 2013] indica que empresas americanas sofreram, nos 24 meses anteriores, uma média de 2,04 interrupções completas e 10,16 parciais, cada uma durando uma média de 107 minutos. Outro estudo [Ponemon 2011] estimou que cada minuto de interrupção no funcionamento de um DC custa US\$ 5.600 em média. É sabido que DCs de nuvens também são afetados por interrupções, as quais em alguns casos podem durar várias horas [McCarthy 2012, Raphael 2013]. Provedores populares de nuvens divulgam dados sobre a disponibilidade e/ou confiabilidade do substrato<sup>1</sup> (e.g., 99,95% e 99,9%), mas, na prática, quando há ocorrência de indisponibilidade, os usuários somente recebem créditos para reexecutarem suas aplicações.

Um desafio para os InPs é alocar recursos físicos para hospedar IVs. Da perspectiva do usuário, uma única falha pode comprometer toda a execução de uma aplicação. Para evitar essa situação, algumas aplicações são tolerantes a faltas, contudo os mecanismos associados

---

<sup>1</sup>Componentes de computação e de rede que compõem a infraestrutura física.

a tolerância a falta normalmente afeta seu tempo de execução e dificulta seu desenvolvimento e manutenção. Sobretudo, uma falha pode resultar em uma violação de SLA (*Service Level Agreement*, Acordo de Nível de Serviço). InPs estão cientes deste fato e podem mitigar o impacto em algumas aplicações levando em consideração falhas de recursos (ou requisitos de disponibilidade) durante a alocação no substrato [Bodík et al. 2012]. Geralmente, isso requer a alocação em recursos físicos esparsos [Rahman, Aib e Boutaba 2010, Barla, Schupke e Carle 2012, Koslovski et al. 2010, Yu et al. 2010, Xu et al. 2012], o qual aumenta o custo de provisionamento. De fato, conforme o InP adiciona redundâncias (servidores, roteadores, *switches* e enlaces de *backup*), aumentam tanto o consumo de energia quanto a complexidade do gerenciamento. Apesar dos recursos de *backup* poderem ser compartilhados entre diversos usuários (até um certo ponto) [Yeow, Westphal e Kozat 2010], os custos associados com confiabilidade são, em última instância, transferidos para os usuários finais.

Usualmente, *frameworks* de gerenciamento consideram os objetivos dos InPs, orientados pelas suas perspectivas financeiras. InPs objetivam maximizar suas receitas alocando o maior número possível de IVs, usando para isso o mínimo de infraestrutura computacional que for viável. Por exemplo, alocando todas as IVs em um mesmo *rack*, economiza-se energia desativando os *racks* ociosos. De fato, trabalhos anteriores tentaram minimizar a fragmentação [Greenberg et al. 2008] do substrato físico (i.e. a quantidade de recursos físicos necessários) para diminuir os custos de provisionamento. Este objetivo pode também beneficiar os usuários, uma vez que, normalmente, também é diminuída a latência da comunicação entre os recursos internos da IV [Koslovski et al. 2011]. No entanto, a fragmentação é minimizada pelo agrupamento dos recursos provisionados, o que pode prejudicar a disponibilidade das IVs e, portanto, colocar usuários e provedores em conflito: uma única falha física pode tornar indisponíveis ou inacessíveis várias IVs e as aplicações que elas hospedam. Conforme o exemplo anterior, se um único *rack* hospeda todas as IVs alocadas, então basta uma única falha neste *rack* para comprometer todas as IVs.

Nesse contexto, este trabalho propõe um novo modelo de alo-

cação ótima para fornecer IVs confiáveis. O principal objetivo desse modelo é selecionar os recursos físicos que devem hospedar recursos virtuais de forma a minimizar tanto o número de recursos virtuais afetados por uma falha no substrato quanto a fragmentação do substrato. Para alcançar esse objetivo é apresentada uma formulação de programação linear inteira mista que incorpora as perspectivas dos InPs (redução da fragmentação) e as dos usuários (aumento da confiabilidade). Resultados experimentais discutem o *trade-off* entre confiabilidade e fragmentação.

## 1.1 OBJETIVOS

O objetivo geral deste trabalho é estabelecer um bom *trade-off* entre a confiabilidade de IVs e a fragmentação do substrato físico.

Para alcançar o objetivo geral, colocam-se os seguintes objetivos específicos:

- propor e desenvolver um modelo de otimização para a alocação de infraestruturas virtuais;
- definir métricas de fragmentação e confiabilidade e incorporá-las ao modelo de otimização; e
- realizar experimentos que possibilitem avaliar a correção do modelo e a sua eficiência quanto as métricas de fragmentação e confiabilidade, avaliando os *trade-offs* entre estas métricas.

## 1.2 PRINCIPAIS CONTRIBUIÇÕES

São três as principais contribuições deste trabalho:

- um modelo de alocação ótima que considera tanto a confiabilidade de IVs quanto a fragmentação do substrato físico;
- a introdução de métricas de falhas locais e globais, as quais fornecem uma medida de confiabilidade da perspectiva dos usuários e das InPs, respectivamente;

- resultados experimentais que fornecem uma linha de base ótima do *trade-off* entre fragmentação e confiabilidade e servem como referência para análises futuras.

### 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte forma: o Capítulo 2 motiva o problema, enquanto o Capítulo 3 apresenta a formulação do modelo de programação linear inteira mista. O Capítulo 4 discute os resultados experimentais. O Capítulo 5 revisa os trabalhos relacionados. E o Capítulo 6 conclui o trabalho e indica as perspectivas futuras.

## 2 ALOCAÇÃO DE REDES E INFRAESTRUTURAS VIRTUAIS

A virtualização compreende desde a virtualização de recursos computacionais (e.g., CPU – *Central Processing Unit*, Unidade Central de Processamento –, memórias) até a de equipamentos de rede e de comunicação (e.g., roteadores, *switches* e enlaces). Este capítulo revisa os conceitos, técnicas e modelos relacionados ao provisionamento dinâmico de redes e infraestruturas virtuais. A Seção 2.1 introduz os principais conceitos e objetivos de virtualização em geral e específicos para redes. A Seção 2.2 discute computação em nuvem, *cloud networking* e infraestruturas virtuais, os papéis, as entidades e os modelos de negócios envolvidos. A Seção 2.3 aborda a alocação e provisionamento de recursos computacionais e de rede, levantando as diferentes estratégias para alocar uma infraestrutura virtual de acordo com os interesses dos provedores de serviço e de infraestrutura. A Seção 2.4 encerra com as considerações do capítulo.

### 2.1 REDES VIRTUAIS

Redes Virtuais (RVs) são a abstração de recursos físicos como enlaces e equipamentos de redes e de comunicação (e.g., roteadores e *switches*) [Chowdhury e Boutaba 2010]. As RVs surgiram com a separação do papel dos provedores de Internet (ISPs) em duas entidades independentes: o provedor de infraestrutura (InP, *Infrastructure Provider*), que gerencia o substrato físico, e o provedor de serviços (PS), que cria e mantém as RVs em um agregado de recursos de múltiplos InPs e fornece serviços fim a fim. Uma RV é alocada em uma rede física, compartilhando os equipamentos de rede e os enlaces com outras RVs. Podendo estar em diferentes camadas da Internet, a virtualização da rede pode se caracterizar de diversas formas, como VLANs (*Virtual Local Area Networks*), VPNs (*Virtual Private Networks*), redes sobrepostas e redes programáveis, com a proposta de superar as limitações da atual implementação da Internet (e.g., ossificação da In-

ternet [Handley 2006]) ou de prover novos serviços e garantias (e.g., flexibilidade administrativa, segurança, resiliência, engenharia de tráfego).

### 2.1.1 Princípios da Virtualização

Virtualização é a representação de uma ou mais entidades reais como uma ou mais entidades virtuais [Popek e Goldberg 1974]. Ou seja, uma entidade virtual pode ser a abstração de uma ou mais entidades reais e um entidade real pode ser usada por uma ou mais entidades virtuais. Essa abstração consiste em representar virtualmente as capacidades e funções da sua contraparte real, como servidores, aplicações, sistemas, armazenamento e redes.

Com isso, é possível ter uma visão dos objetivos da virtualização [Chowdhury e Boutaba 2009]:

**Flexibilidade:** visa o uso arbitrário de qualquer função ou capacidade, independente do substrato físico ou de outras entidades virtuais coexistentes.

**Escalabilidade:** deve ser garantida para que o substrato físico seja capaz de hospedar tantos recursos virtuais quanto possível sem afetar o desempenho de nenhum desses recursos.

**Manutenibilidade:** visa modularizar as tarefas de administração, separando a função do provedor de serviço da do InP. Assim, o InP operaria o substrato físico e proveria acesso aos recursos, e o provedor de serviços alugaria recursos de diferentes InPs.

**Isolamento, segurança e privacidade:** uma falha ou brecha de segurança em um recurso virtual não deve afetar os outros que coexistem no mesmo recurso físico. Por exemplo, um serviço ou protocolo não confiável, experimental ou mal configurado executando em um recurso virtual não deve afetar recursos virtuais coexistentes.

**Programabilidade:** visa expor interfaces de programação dos recursos para permitir a flexibilidade e manutenibilidade na virtua-

lização. Faz com que um provedor de serviço possa fornecer protocolos e funções personalizadas e diversas. Apesar disso, questões como “o quanto de programabilidade deve ser permitida” e “como deve ser exposta” devem ser tratadas para tentar garantir a simplicidade, eficácia e segurança dos recursos.

**Heterogeneidade:** há duas perspectivas de heterogeneidade: dos recursos virtuais e dos substratos físicos. Cada recurso virtual, criado sobre uma combinação heterogênea de recursos físicos (e.g., redes ópticas, sem fio e de sensores), também deve ser heterogêneo. Um provedor de serviço pode compor e executar recursos virtuais sem nenhuma restrição e através de múltiplos domínios sem a necessidade de soluções e tecnologias específicas, assim como o substrato físico deve ser capaz de hospedar protocolos e algoritmos heterogêneos implementados por diferentes provedores de serviços.

**Mecanismo de teste:** deve permitir que um serviço seja testado em um ambiente controlado antes de ser implantado em um ambiente de produção. Mas deve ser considerado que é caro reproduzir o ambiente de produção e podem ser necessárias simplificações da representação.

**Suporte a aplicações, tecnologias e serviços legados:** aquilo que está surgindo ou vai algum dia surgir deve ser compatível com o que atualmente já existe.

Um outro objetivo de virtualização que pode ser adicionado é permitir a redução dos custos, como administrativos, de energia, de refrigeração e de manutenção, por alocar diferentes demandas de serviços em um único elemento ou plataforma computacional que compartilha a capacidade dos recursos físicos.

### 2.1.2 Virtualização de Redes

Uma RV é uma coleção de nós virtuais, que são tipicamente roteadores e *switches*<sup>1</sup>. Ao longo do texto, *switches* virtuais serão denotados SVs. Os SVs (representados graficamente por círculos) são interconectados por enlaces virtuais (EVs, representados por linhas pontilhadas entre os nós). A Figura 1 ilustra uma rede virtual mapeada e hospedada em uma rede física. O processo de mapeamento é, tipicamente, um processo de reserva de recursos físicos ou funcionalidades.

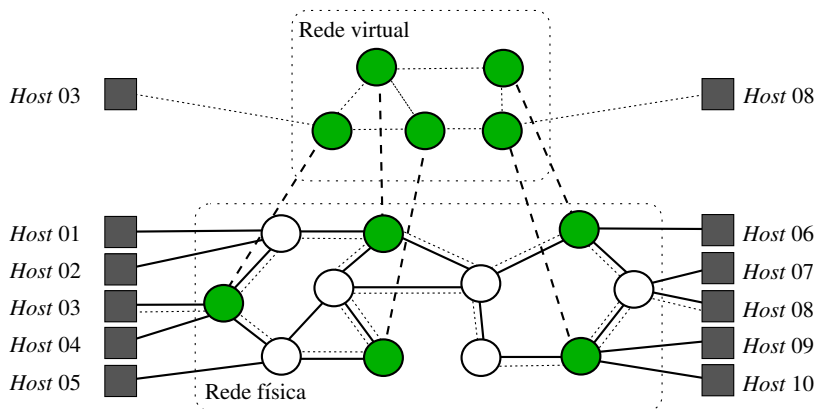


Figura 1 – A reserva de uma rede virtual em uma rede física.

Fonte: produção do próprio autor.

Na parte superior da figura, é especificada a comunicação lógica entre os *hosts* 03 e 08, definindo a rede virtual requisitada como o canal de comunicação virtual e lógico entre estes *hosts*. Na parte inferior, é realizado o mapeamento entre a rede virtual e a rede física (representado por linhas tracejadas). Os *switches* físicos (SFs) podem, depois do processo de mapeamento, se encontrar ociosos (representados pelos círculos em branco) ou ocupados (representados pelos círculos preenchidos). Os *hosts* 03 e 08, apesar de estarem conectados à RV, não fazem parte do processo de mapeamento e alocação. Um EV, conectando diretamente dois SVs, pode ser composto por/hospe-

<sup>1</sup>Os termos *switch* e *roteador* são usados indistintamente no restante do trabalho.



dado em um ou mais enlaces físicos (EFs, representados por linhas sólidas), formando um caminho físico. Assim, as linhas pontilhadas na rede física representam os EFs que hospedam um determinado EV. Na figura, o restante dos *hosts* continuam a se comunicar diretamente através dos roteadores e enlaces físicos. Múltiplas RVs, que compartilham do mesmo espaço físico, poderiam ser co-alocadas para estabelecer canais de comunicação virtuais e independentes entre subconjuntos distintos destes *hosts*.

Os canais virtuais de comunicação abstraem o caminho físico entre os *hosts*. Esta camada adicional proporciona um uso mais eficiente do substrato físico, permitindo que as redes virtualizadas possam ser realocadas, reconfiguradas independentemente, armazenadas e recuperadas de um estado qualquer. Assim, provedores de serviços de RVs com soluções para camadas distintas (e.g., camadas 1 e 3) podem ser contratados para melhorar o serviço prestado por uma outra empresa, sem que os clientes desta última percebam a diferença.

Usualmente, a virtualização de redes pode ser implementada como (uma combinação de) [Chowdhury e Boutaba 2009]:

**LANs virtuais:** (*Virtual LANs, VLANs*) são redes locais logicamente isoladas (tipicamente na camada 2) mas que compartilham um mesmo *switch* ou roteador físico, diminuindo o domínio de colisão e até permitindo priorização de parte do tráfego.

**Redes virtuais privadas:** (*Virtual Private Networks, VPNs*) são redes de uma mesma (Intranet) ou diferentes empresas (Extranet), geograficamente distribuídas em uma WAN (*Wide Area Network*). São interconectadas (tipicamente nas camadas 1, 2 ou 3) por uma infraestrutura pública e se comunicam por canais lógicos/virtuais privados, geralmente criptografados.

**Redes sobrepostas:** são redes construídas na camada de aplicação, normalmente, onde os enlaces são caminhos ou rotas na Internet que interconectam os nós que fazem parte da rede, sendo utilizadas para prover resiliência, *multicasting*, garantias de QoS (*Quality of Service*) e distribuição de conteúdo.

**Redes programáveis:** são redes construídas através de APIs (*Appli-*

*cation Programming Interfaces*) e abstraem ou separam aspectos da rede como em: *Active Networks*, onde a rede é reconfigurada dinamicamente baseada nas informações de tráfego obtidas dos roteadores e *switches*; e *Open Signaling*, onde há a divisão dos planos de dados (tráfego de dados do usuário), de controle (protocolos de roteamento e sinalização) e de gerenciamento (tráfego de monitoramento e configuração dos planos de dados e de controle) [Greenberg et al. 2005], como no OpenFlow [McKeown et al. 2008].

## 2.2 INFRAESTRUTURAS VIRTUAIS

Infraestruturas virtuais são a combinação de redes virtuais (enlaces e dispositivos de rede) com recursos computacionais virtualizados (CPU, memória e dispositivos de entrada/saída – E/S), e se caracterizam como uma evolução e conjunção destes dois conceitos [Sotomayor et al. 2009]. Tanto infraestruturas virtuais quanto recursos computacionais virtualizados compartilham dos mesmos princípios de virtualização que as redes virtuais (Seção 2.1.1).

IVs normalmente são implementadas em *data centers* como um meio de se prover a computação em nuvem [Schoo et al. 2011]. A comunicação entre IVs visa interconectar diferentes nuvens, *data centers* virtualizados e usuários finais, sendo conhecida também, neste contexto, como *cloud networking*.

### 2.2.1 Computação em Nuvem

Antes de dar sequência para a discussão de infraestruturas virtuais, é necessário entender o que é computação em nuvem para, assim, ser possível estabelecer um ponto de comparação entre IVs e computação em nuvem. Segundo [Mell e Grance 2011], computação em nuvem é um modelo de serviço que permite acesso sob demanda e ubíquo a um conjunto compartilhado de recursos computacionais configuráveis (e.g., redes, servidores, armazenamento, aplicações e serviços) que pode ser provisionado com um mínimo esforço do admi-

nistrador de nuvem ou do provedor de serviço. Existem basicamente três modelos de serviços de nuvem [Mell e Grance 2011], ilustrados pela Figura 2:

**Software como um Serviço** (*Software as a Service*, SaaS): nesse modelo, um consumidor usa aplicações do provedor de serviços que executam em uma infraestrutura de nuvem. As aplicações são acessíveis a partir de vários *front-ends* e os seus clientes não administram e nem controlam a infraestrutura subjacente, como a rede, os servidores, os sistemas operacionais, o sistema de armazenamento e certas configurações da aplicação.

**Plataforma como um Serviço** (*Platform as a Service*, PaaS): nesse modelo, o consumidor desenvolve sobre a infraestrutura da nuvem aplicações usando linguagens de programação, bibliotecas, serviços e ferramentas disponibilizadas pelo provedor. O cliente não administra e nem controla a infraestrutura subjacente, como a rede, os servidores, os SOs (Sistemas Operacionais) ou o sistema de armazenamento, mas tem controle sobre as aplicações desenvolvidas e a possibilidade de configurar o ambiente de desenvolvimento.

**Infraestrutura como um Serviço** (*Infrastructure as a Service*, IaaS): nesse modelo, a capacidade fornecida ao consumidor é provisionar processamento, armazenamento, redes ou qualquer outro recurso básico computacional ou de comunicação, nos quais o cliente pode desenvolver e executar sistemas de software arbitrários, como SOs e aplicações. O cliente não administra e nem controla o hardware da infraestrutura da nuvem, mas tem controle sobre os SOs, os sistemas de armazenamento, as aplicações desenvolvidas e os canais de comunicação.

A forma com que esses serviços são acessados, utilizados e mantidos define o modelo de implementação de uma determinada nuvem. Ou seja, é possível classificar uma nuvem computacional de acordo com quem são os seus usuários e a quem ela pertence. Quatro classificações são dadas por [Mell e Grance 2011], ilustradas pela Figura 3:

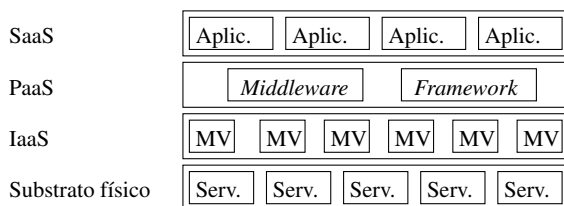


Figura 2 – Modelos de serviços de computação em nuvem.

Fonte: produção do próprio autor.

**Nuvens Privadas:** a infraestrutura de nuvem é provisionada para uso exclusivo de uma única organização com vários clientes (e.g., unidades de negócios). Pode pertencer, ser administrada e operada pela própria organização, um terceiro ou uma combinação desses.

**Nuvens Comunitárias:** a infraestrutura da nuvem é provisionada para o uso exclusivo de consumidores de organizações que compartilham alguns objetivos. Pode pertencer, ser administrada e operada por uma ou mais organizações na comunidade, um terceiro ou uma combinação desses.

**Nuvens Públicas:** a infraestrutura de nuvem é provisionada para um uso aberto ao público geral. Pode pertencer, ser administrada e operada por organizações governamentais, acadêmicas, empresariais ou uma combinação dessas.

**Nuvens Híbridas:** a infraestrutura da nuvem é uma composição de duas ou mais infraestruturas distintas (privada ou pública) que continuam entidades únicas, mas são combinadas por protocolos e tecnologias padronizadas ou proprietárias que permitem a portabilidade de dados e aplicações (e.g., balanceamento de carga entre nuvens).

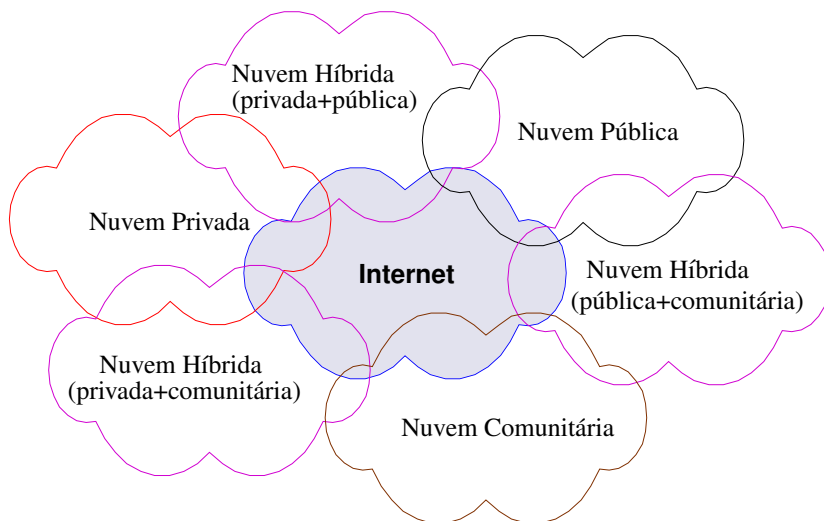


Figura 3 – Modelos de implementação de nuvens computacionais.  
Fonte: produção do próprio autor.

### 2.2.2 Virtualização de Infraestruturas Computacionais

Infraestruturas virtuais são uma extensão do modelo de serviço de nuvem IaaS. Com essa extensão, os recursos computacionais e de redes podem ser abstraídos da infraestrutura física e utilizadas como serviços que são dinamicamente reservados e provisionados [Laganier e Primet 2005, Rochwerger et al. 2009]. A Figura 4 ilustra os principais conceitos de infraestruturas físicas e virtuais. Analogamente a uma RV, uma infraestrutura virtual (IV) é uma coleção de nós virtuais. Contudo, além de serem especificados os SVs, também são especificadas as máquinas virtuais (MVs, representadas com quadrados) a serem alocadas.

Dessa forma, os *hosts* na borda de uma IV são máquinas virtuais, e não máquinas físicas, como acontece nas RVs. E assim como o de uma rede física, o substrato de uma infraestrutura física é composto por nós (mas de dois tipos: máquinas e *switches* físicos) interconectados por enlaces. As MFs agora podem ser utilizadas para alocar uma

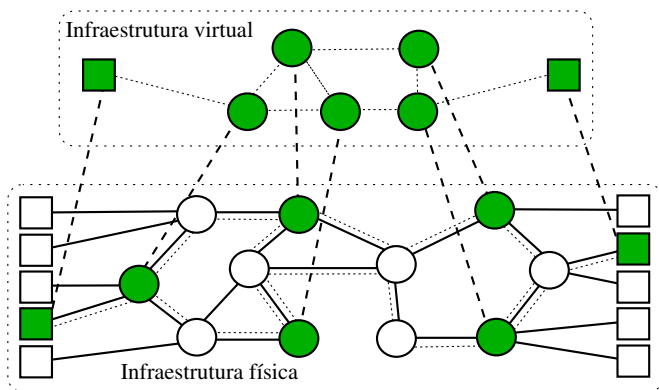


Figura 4 – A reserva de uma infraestrutura virtual em uma infraestrutura física.

Fonte: produção do próprio autor.

ou mais MVs.

Segundo [Chowdhury e Boutaba 2010], as entidades envolvidas na especificação, alocação e uso de redes virtuais podem seguir um modelo de interação entre si. Aqui, este modelo é estendido para IVs, como ilustram as Figuras 5 e 6, da forma que as redes virtuais discutidas na Seção 2.1.2 também compartilham por herança deste modelo.

Os papéis envolvidos em um ambiente de virtualização são diferentes daqueles envolvidos em um ambiente tradicional de computação e comunicação. Se no modelo tradicional existe o papel do ISP (*Internet Service Provider*), no ambiente de virtualização esse papel é dividido em dois, o do InP e do PS (Provedor de Serviço). Além disso, uma entidade, como será visto mais adiante, pode assumir mais de um papel.

**Provedor de Infraestrutura (InP):** o InP implementa e gerencia os recursos da infraestrutura física (IF) subjacente à IV. É o encarregado de operar e manter a IF e fornecer seus recursos através de APIs a diferentes provedores de serviço (PSs). Eles não fornecem serviços diretamente ao usuário final. InPs se diferenciam pela qualidade dos recursos fornecidos, a liberdade delegada aos seus clientes (i.e., os provedores de serviço) e as ferramentas

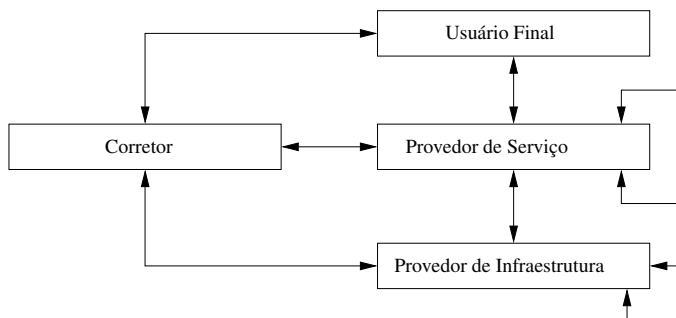


Figura 5 – Relação entre entidades de uma IV.  
 Fonte: Adaptada de [Chowdhury e Boutaba 2010].

fornecidas para explorar essa liberdade.

**Provedor de Serviço:** um PS aluga recursos de múltiplos InPs para criar IVs e implementar protocolos customizados, fornecendo serviços de infraestrutura ao usuário final. Um PS pode criar uma IV filha, particionando os seus recursos, e a alugando a outros PSs, recursivamente, como ilustra a Figura 6.

**Usuário Final:** usuários finais de uma IV são similares aos usuários finais da Internet: a existência de múltiplas IVs de PSs concorrentes permite que os usuários finais escolham entre diversos serviços e especifiquem QoS (*Quality-of-Service*). Qualquer usuário final pode se conectar a múltiplos PSs para obter diferentes serviços.

**Corretor:** o corretor atua como mediador entre os InPs, os PSs e os usuários finais. PSs compram ou alugam recursos dos InPs para criarem IVs e venderem serviços implementados nelas para usuários finais através dos corretores. A sua presença simplifica o processo de conciliar os requisitos dos PSs por recursos disponíveis agregando os ofertas de múltiplos InPs. Similarmente, também permite os usuários finais selecionarem serviços que desejam a partir de diversos PSs.

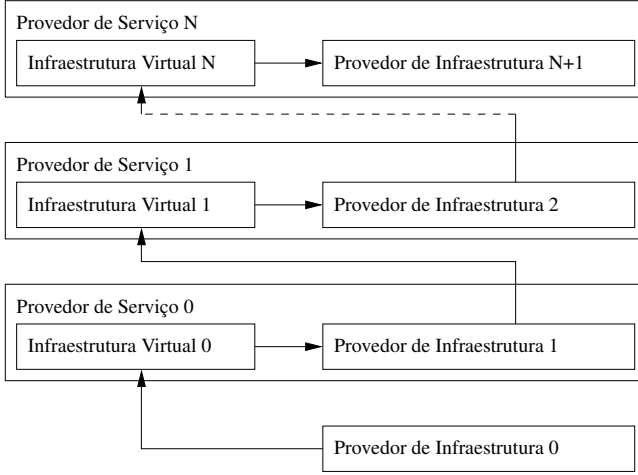


Figura 6 – Hierarquia recursiva de papéis entre um PS e um InP.  
 Fonte: Adaptada de [Chowdhury e Boutaba 2009].

## 2.3 ALOCAÇÃO DE RECURSOS VIRTUAIS

Alocar recursos computacionais e de comunicação é reservar, por um determinado tempo, frações ou partes desses recursos a entidades virtuais. Após ser esgotado o tempo concedido a essas entidades virtuais, elas são desalocadas do substrato, que volta a ter aquela fração disponível para uma outra alocação. Normalmente, a alocação ocorre em dois passos: o mapeamento e a reserva de recursos. No mapeamento, um conjunto de recursos físicos são designados para hospedar um recurso virtual. Assim, o grafo descrevendo um infraestrutura virtual,  $G^v(N^v, E^v)$ , deve ser mapeado para o grafo do substrato físico,  $G^f(N^f, E^f)$ , onde  $N^v$  e  $N^f$  são os conjuntos de nós virtuais e de físicos, respectivamente, e  $E^v$  e  $E^f$  os conjuntos de enlaces virtuais e físicos, respectivamente. A função de mapeamento  $M : G^v \uparrow G^f$  determina as tuplas  $(n^v, n^f)$ , para nós, onde cada nó virtual  $n^v \in N^v$  é mapeado para somente um nó físico  $n^f \in N^f$ , e as tuplas  $(e^v, C^f)$ , para enlaces, onde cada enlace virtual  $e^v \in E^v$  é mapeado para um caminho físico  $e^f \in C^f \subseteq E^f$  (um caminho físico é formado por um ou múltiplos enlaces físicos conexos).



No segundo passo, a reserva de recursos, tenta-se garantir que cada recurso virtual disponha de uma quantidade mínima das capacidades dos recursos físicos nos quais foi mapeado. Assim, para cada tupla  $(n^v, n^f)$  são garantidas uma fração mínima das capacidades de  $n^f$  de acordo com a demanda de  $n^v$ , e, analogamente, para cada tupla  $(e^v, C^f)$  são garantidas uma fração mínima das capacidades de cada enlace físico  $e^f$  em  $C^f$  de acordo com a demanda de  $e^v$ .

O InP é o responsável por realizar a alocação e pode expor uma interface e parâmetros da alocação aos PSs e aos corretores. Com a decisão sobre quais IVs serão alocadas e quais recursos serão reservados, o InP pode determinar quais tipos de alocações estão de acordo com o seu modelo de negócios. Um InP pode priorizar a alocação da IV de um cliente em detrimento da IV de um outro cliente, ou alocar IVs em recursos dedicados para clientes que tenham requisitos de acesso e violação dos seus dados sensíveis, ou, ainda, alocar IVs de acordo com modelos de custos, segurança, confiabilidade, tolerância a faltas etc [Bari et al. 2013]. Essa decisão do InP sobre a alocação normalmente é traduzida como uma função objetivo, onde um algoritmo de alocação minimiza ou maximiza essa função.

Com isso, uma alocação pode ocorrer de três maneiras distintas: otimização, heurística ou uma combinação destas. Uma otimização tenta encontrar um valor ótimo para a função objetivo definida pelo InP, e o tempo para encontrar esse valor é exponencial [Andersen 2002, Yu et al. 2008]. Programação linear, inteira ou mista, e não-linear são alguns exemplos de algoritmos que encontram valores ótimos. Uma heurística permite ao InP encontrar um valor sub-ótimo a favor de um tempo linear para encontrar esse valor. Para que a alocação seja efetuada, é necessário que o InP especifique as restrições e os recursos de sua infraestrutura onde as restrições serão aplicadas. Por exemplo, uma restrição pode impedir que um recurso físico hospede mais recursos virtuais além da sua capacidade, ou garantir que um caminho virtual conecte dois nós virtuais e que seja conexo, ou outras restrições que imponham condições de segurança ou outras previstas no seu modelo de negócio. As Figuras 7 e 8 ilustram os principais componentes de infraestruturas físicas e virtuais para um cenário de alocação. Inicialmente, a Figura 7 define  $s$  como um *substrato físico*

composto por máquinas físicas  $m^p$  e *switches*  $s^p$ , interconectados por enlaces unidirecionais  $e^p$ .

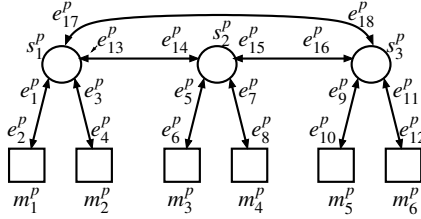


Figura 7 – Um exemplo um substrato físico  $s$ .  
Fonte: produção do próprio autor.

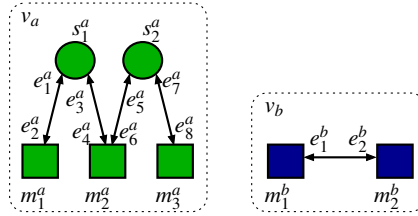


Figura 8 – Um exemplo de duas IVs  $v_a$  e  $v_b$ .  
Fonte: produção do próprio autor.

Em seguida, a Figura 8 ilustra duas *infraestruturas virtuais*,  $v_a$  e  $v_b$ , ambas compostas por recursos virtuais: máquinas ( $m^a$  e  $m^b$ ), *switches* ( $s^a$  e  $s^b$ ), e enlaces ( $e^a$  e  $e^b$ ). Tomando as Figuras 7 e 8 como grafos, os pesos das arestas denotam as capacidades de enlace no substrato e as demandas por largura de banda em uma IV. Similarmente, as capacidades/demandas de CPU e memória são associadas com os vértices nos grafos. O objetivo é que as demandas por recursos de uma IV sejam reservadas pelo InP no substrato físico, como exemplificado pelas Figuras 9 e 10. Particularmente, a Figura 9 mostra que um enlace virtual (e.g.,  $e_1^b$ ) pode ser alocado em um caminho físico com múltiplos saltos ( $e_9^p, e_{15}^p, e_8^p$ ).

As Figuras 9 e 10 ilustram como as IVs  $v_a$  e  $v_b$  podem ser alocadas<sup>2</sup>. Em particular, a Figura 9 ilustra uma variação na alocação,

<sup>2</sup>Para adequar o trabalho ao estado-da-arte, os termos alocação e mapeamento pos-

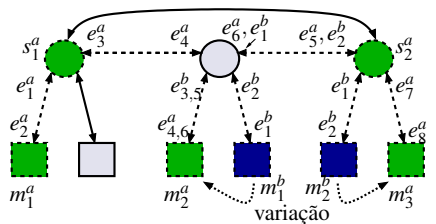


Figura 9 – Uma alocação  $m^f$ , exemplo de mapeamento entre o substrato  $s$  (Figura 7) as IVs  $v_a$  e  $v_b$  (Figura 8).

Fonte: produção do próprio autor.

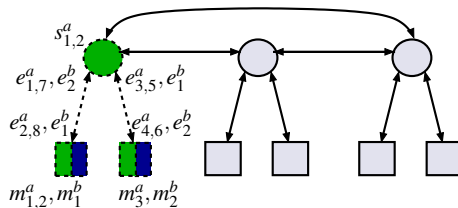


Figura 10 – Uma alocação  $m^{fr}$ , exemplo de mapeamento entre o substrato  $s$  (Figura 7) as IVs  $v_a$  e  $v_b$  (Figura 8).

Fonte: produção do próprio autor.

indicando uma terceira possibilidade, frente às duas primeiras. Os recursos físicos são alocados conforme as estratégias de cada InP: minimizar a fragmentação do substrato, utilizando poucos recursos físicos (Figura 10) ou maximizar a confiabilidade das IVs, alocando-as esparsamente (Figura 9). Na Figura 9, somente alguns EVs das duas IVs compartilham os mesmos recursos físicos, ao passo que, na Figura 10, as MVs e os EVs das IVs  $v_a$  e  $v_b$  compartilham recursos das mesmas MFs e EFs. Assim, cada alocação pode ter um impacto diferente, ou um serviço diferente prestado pelo InP, como será discutido na Capítulo 3.

## 2.4 CONSIDERAÇÕES PARCIAIS

As tecnologias de virtualização de redes e recursos computacionais impulsionaram o crescimento da computação em nuvem. O advento das nuvens permite que provedores de serviços ofereçam infraestruturas virtuais de forma desacoplada da operação e do gerenciamento de servidores e equipamentos de rede, funções assumidas por provedores de infraestrutura.

Para que as características que definem as nuvens computacionais e infraestruturas virtuais – como isolamento, qualidade de serviço e escalabilidade – possam ser oferecidas, um problema chave a ser resolvido é a alocação de recursos virtuais no substrato físico. Essa alocação envolve não apenas mapear recursos virtuais nos recursos físicos que os hospedam, mas também reservar a capacidade demandada para garantir o desempenho desejado da infraestrutura virtual.

A alocação de infraestruturas virtuais pode ser resolvida de várias maneiras, podendo levar em consideração diferentes objetivos, seja do ponto de vista do provedor de infraestrutura ou do usuário. No próximo capítulo é apresentado um modelo de otimização para a alocação de recursos que considera simultaneamente os aspectos conflitantes de minimização de confiabilidade das infraestruturas virtuais alocadas do substrato físico e da capacidade ociosa nesse substrato.

### 3 MODELO DE ALOCAÇÃO DE RECURSOS PARA INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS

Este capítulo introduz um modelo de programação linear inteira mista (PLIM) que proporciona confiabilidade de IVs sem aumentar a fragmentação no substrato físico. As definições e as métricas gerais de confiabilidade e de fragmentação são dadas na Seção 3.1. O modelo é apresentado em quatro partes: i) um problema tradicional de alocação de grafos (Seção 3.2); ii) uma formulação para alcançar as confiabilidades local e global (Seção 3.3); iii) uma formulação para alcançar o uso ótimo do substrato considerando fragmentação (Seção 3.4); e iv) o modelo combinado para minimizar tanto as falhas de IVs (Seção 3.5) quanto a fragmentação do substrato. E as considerações finais (Seção 3.6) encerram o capítulo.

#### 3.1 CONFIABILIDADE E FRAGMENTAÇÃO EM INFRAESTRUTURAS VIRTUAIS

Um InP pode definir um conjunto de objetivos para determinar o mapeamento e a reserva de recursos no seu substrato. Esses objetivos são guiados, geralmente, por perspectivas financeiras, ou o que afeta direta ou indiretamente a sua receita, como diminuir o uso da largura de banda, o número de servidores em uso, a latência entre MVs de uma IV ou o número de recursos físicos afetados por uma falha (esses dois últimos afetam os lucros por quebras de SLAs). Dentre esse objetivos, alguns podem ser conflitantes. Diminuir a latência entre MVs de uma IV vai, invariavelmente, alocá-las em servidores próximos, que podem estar sob o mesmo *rack*, ou em um mesmo servidor. Contudo, se o InP também tem como objetivo diminuir o impacto de falhas físicas sobre as IVs, uma única falha, seja naquele *rack* ou servidor, vai comprometer toda ou grande parte daquela IV. InPs, algumas vezes, têm que lidar com objetivos conflitantes e definir os limites que julguem aceitáveis dos *trade-offs* entre esses objetivos. Ao InP que tenha dentre os seus objetivos aumentar a confiabilidade das IVs e diminuir a

fragmentação do substrato físico, este trabalho fornece uma ferramenta que explore os limites entre os *trade-offs* destes objetivos. As Figuras 9 e 10 (página 43) ilustram como as IVs  $v_a$  e  $v_b$  (Figura 8) podem ser alocadas no substrato  $s$  (Figura 7) de acordo com diferentes *trade-offs* entre confiabilidade e fragmentação, e serão discutidas nas próximas seções, auxiliando na definição dos respectivos conceitos (métricas).

### 3.1.1 Confiabilidade

Neste trabalho, consideram-se que os recursos físicos estão sujeitos a falhas de parada (*crash*); uma falha pode afetar um único recurso (e.g., um servidor) ou um grupo de recursos (e.g., um *rack* ou um DC inteiro). Para quantificar o impacto destas falhas, [Bodík et al. 2012] definiu um *domínio de faltas* (DF) como um conjunto de recursos físicos que compartilham um único ponto de falha, como, por exemplo, todos os servidores em um dado *rack* compartilham um único ponto de falha: o próprio *rack*. O mesmo nó pode pertencer a múltiplos DFs. Em um ambiente de nuvem, uma IV pode ser distribuída sobre diversos recursos físicos pertencentes a diferentes DFs. Para quantificar o impacto de falhas no substrato, definimos uma *métrica de falha* como a fração de recursos virtuais (nós e enlases) afetada pelo pior caso de falha em recursos físicos (i.e., a falha física que afeta mais recursos virtuais). Essa métrica é calculada para todos os *tipos de domínios de faltas* (ou simplesmente *tipos*) definidos pelo InP. Por exemplo, o substrato físico  $s$  (Figura 11) tem quatro tipos de domínio de faltas: *servidor*, *rack*, *circuito elétrico* e *enlace*. Cada DF é denotado por  $d$  e cada tipo é denotado por  $t$ .

Esta métrica é exemplificada analisando o impacto de uma falha em um *rack* na Figura 9. Neste cenário, cada *rack*  $d_n^r$  contém duas MFs e um *switch*. Se um *rack* falha, todos os recursos virtuais hospedados nas máquinas ou no *switch* deste *rack* são perdidos. Considerando três DFs para o tipo *rack* ( $d_1^r$ ,  $d_2^r$  e  $d_3^r$ ), o pior caso na alocação  $m^{fr}$  (Figura 10, página 43) é a falha do *rack*  $d_1^r$ , a qual induz a perda de todos os recursos virtuais. Considerando as alocações  $m^f$  e  $m^{f'}$  (Figura 9, página 43), o pior caso é dado pela falha de  $d_3^r$ , a qual afeta 40% de  $v_a$

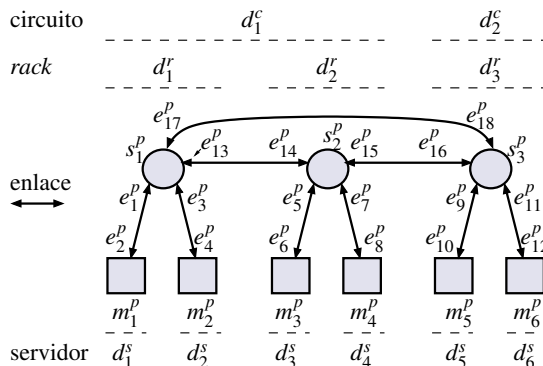


Figura 11 – O substrato  $s$  (Fig. 7) com tipos de domínios de falhas distintos (circuito elétrico, *rack*, servidor e enlace).

Fonte: produção do próprio autor.

e 50% e  $v_b$ , como resumido na Tabela 1. Observa-se, portanto, que a minimização das métricas de falha induz maior confiabilidade das IVs.

Tabela 1 – Métricas de falha (%) para as Figuras 9 e 10.

$m$	circuito ( $c$ )		<i>rack</i> ( $r$ )		servidor ( $s$ )		enlace ( $e$ )	
	FLN	FGN	FLN	FGN	FLN	FGN	FLE	FGE
	$v_a/v_b$		$v_a/v_b$		$v_a/v_b$		$v_a/v_b$	
$m^f$	60/50	57	40/50	43	33/50	20	25/50	33
$m^{f'}$	60/50	57	40/50	43	33/50	40	25/50	50
$m^{fr}$	100/100	100	100/100	100	67/50	60	50/50	50

Fonte: produção do próprio autor.

Embora o *objetivo geral de confiabilidade* seja minimizar as métricas de falha de uma alocação, usuários e InPs têm diferentes perspectivas de confiabilidade. Um usuário tipicamente deseja maximizar a confiabilidade de suas próprias IVs sem se preocupar com a confiabilidade das outras IVs, ao passo que um InP tentará proporcionar uma boa confiabilidade global (a qual pode ser sub-ótima para certos usuários). *De fato, InPs querem fornecer IVs confiáveis como um serviço.* Essas perspectivas distintas são traduzidas em métricas de *falha local* e de *falha global*, respectivamente. A métrica de falha local (FL) in-

corpora a perspectiva do usuário, considerando ambas as métricas de falha de nós e de enlaces ( $FLN_{v,t}$  e  $FLE_{v,t}$ , respectivamente). Para uma dada IV  $v$  e um tipo de DF  $t$ , a métrica de falha local de nós ( $FLN_{v,t}$ ) é dada pela razão de nós falhos ( $|\mathcal{N}_{v,t}^f|$ ) e o total de nós ( $|\mathcal{N}_{v,t}^t|$ ) de  $v$  que estão alocados em recursos físicos do tipo  $t$ , como mostrado na Equação 3.1; uma definição análoga para  $FLE_{v,t}$  é dada pela Equação 3.2. Ao diminuir  $FLN_{v,t}$  e  $FLE_{v,t}$ , a confiabilidade de  $v$  aumenta; logo, idealmente  $FLN_{v,t}$  e  $FLE_{v,t}$  devem ser minimizadas. A Tabela 1 ilustra essa métrica para as alocações  $m^{fr}$ ,  $m^f$  e  $m^{f'}$ . Em  $m^{fr}$ ,  $v_a$  tem uma alta FLN, visto que todos os nós são perdidos quando falha um circuito elétrico ou *rack*. Mesmo se um servidor físico falhar, somente um único nó virtual sobrevive. Nas alocações  $m^f$  e  $m^{f'}$ , as  $FLN_{v,t}$  para  $v_a$  e  $v_b$  diminuem em todos os tipos, exceto para falhas de servidor afetando  $v_b$ .

$$FLN_{v,t} = \frac{|\mathcal{N}_{v,t}^f|}{|\mathcal{N}_{v,t}^t|} \quad (3.1)$$

$$FLE_{v,t} = \frac{|\mathcal{E}_{v,t}^f|}{|\mathcal{E}_{v,t}^t|} \quad (3.2)$$

A métrica de falha global (FG) segue o mesmo princípio da FL, com a diferença que todas as IVs hospedadas são consideradas. A Equação 3.3 define a métrica de falha global de nós como  $FGN_t$ , a razão entre os recursos falhos ( $|\mathcal{N}_t^f|$ ) e o total de recursos ( $|\mathcal{N}_t^t|$ ) alocados no tipo de DF  $t$ ; a Equação 3.4 dá uma definição análoga para a métrica de falha global de enlace ( $FGE_t$ ). Assim como as métricas de falha local, os InPs querem minimizar  $FGN_t$  e  $FGE_t$  para fornecer um serviço com confiabilidade. A alocação  $m^{fr}$  ilustra um cenário no qual  $FGN_r = 100\%$  quando ocorre uma falha de circuito ou *rack*. Por outro lado, ela oferece um valor aceitável para falhas de servidor (ao menos 60% dos nós são capazes de sobreviver no pior caso). As métricas de falha global para as alocações  $m^f$  e  $m^{f'}$  são menores, indicando um maior nível de confiabilidade para falhas de circuito e de *rack*. Considerando o tipo servidor, para a alocação  $m^f$ , a  $FGN_s$  é 1/3 da  $FGN_s$  para  $m^{fr}$ , e



a migração de  $v_b$  em  $m^{fr}$  traz a  $FGN_s$  para  $2/3$  da de  $m^{fr}$ . Atualmente, o modelo concede o mesmo peso para as métricas global e local, contudo esses pesos podem ser redefinidos por um InP para priorizar tanto IVs individuais quanto o seu conjunto e para agregar valor aos serviços. O ajuste dos pesos e as estratégias de preços que podem ser explorados são temas para pesquisas futuras.

$$FGN_t = \frac{|\mathcal{N}_t^f|}{|\mathcal{N}_t^g|} \quad (3.3)$$

$$FGE_t = \frac{|\mathcal{E}_t^f|}{|\mathcal{E}_t^g|} \quad (3.4)$$

### 3.1.2 Fragmentação

Um InP que tenha como objetivo diminuir a fragmentação do substrato físico busca utilizar o mínimo possível de recursos físicos alocando o máximo possível de IVs. Com isso, tenta-se evitar o aumento do consumo de energia elétrica ao ligar mais equipamentos do que o necessário, caso os que já estejam ativos tenham capacidade o suficiente para hospedar mais IVs. A alocação  $m^{fr}$  (Figura 10) reduz a *fragmentação do substrato físico* [Greenberg et al. 2008], definida como a razão entre o número de recursos físicos ativos (i.e., aqueles que hospedam recursos virtuais) e o número total de recursos físicos. Considerando a alocação de IVs, ou seja, a alocação combinada de recursos de comunicação e computadores, são definidas duas métricas de fragmentação: de nó (FrN) e de enlace (FrE). A fragmentação de nó é dada pela Equação 3.5, onde  $|\mathcal{N}_a^p|$  é o número de nós ativos (tanto máquinas físicas  $m^p$  quanto *switches*  $s^p$ ) e  $|\mathcal{N}_t^p|$  é o número total de nós. Similarmente, a fragmentação de enlace é dada pela Equação 3.6, onde  $|\mathcal{E}_a^p|$  é o número de enlaces ativos e  $|\mathcal{E}_t^p|$  é o número total de enlaces.

$$\text{FrN} = \frac{|\mathcal{N}_a^p|}{|\mathcal{N}_t^p|} \quad (3.5)$$

$$\text{FrE} = \frac{|\mathcal{E}_a^p|}{|\mathcal{E}_t^p|} \quad (3.6)$$

Diminuindo a fragmentação (minimizando a soma das Equações 3.5 e 3.6), os InPs podem reduzir os custos desativando os recursos ociosos. Por exemplo, na alocação  $m^{fr}$  (Figura 10), somente um *rack* ( $d_1^r$ ) está hospedando ambas as IVs  $v_a$  e  $v_b$ , e portanto os *racks*  $d_2^r$  e  $d_3^r$  estão ociosos.

Tabela 2 – Métricas de fragmentação (%) para as Figuras 9 e 10.

$m$	FrN	FrE
$m^f$	78	78
$m^{f'}$	56	56
$m^{fr}$	33	22

Fonte: produção do próprio autor.

A Tabela 2 mostra os valores de fragmentação para as alocações nas Figuras 9 e 10. Para a alocação  $m^{fr}$  observa-se 33% de fragmentação de nó e 22% de fragmentação de enlace, enquanto a alocação  $m^f$  – a qual favorece a confiabilidade, como foi explicado na Seção 3.1.1 – tem 78% de fragmentação de nó/enlace (2,3 e 3,5 vezes os valores para  $m^{fr}$ ). No entanto, uma pequena diferença na alocação  $m^{f'}$  (a variação na Figura 9) resulta em um menor aumento nas fragmentações de nó e de enlace (1,7 vezes maior que em  $m^{fr}$ ).

### 3.1.3 Visão Geral do Modelo

Encontrar um bom *trade-off* entre confiabilidade e fragmentação não é uma tarefa trivial. Por exemplo, comparada com  $m^f$ , a alocação  $m^{f'}$  na Figura 9 diminui a fragmentação de nó (FrN) em quase 29%, sem impactar nas métricas global e local para os tipos circuito e *rack*. De fato, os valores das métricas de falha local para  $m^f$  e  $m^{f'}$  são idênticos. Em contrapartida, para o tipo servidor, a alocação  $m^f$

é significativamente mais confiável do que  $m^{f'}$ , uma vez que metade dos servidores falham no pior caso. Ademais, em  $m^f$ , qualquer falha isolada afetará no máximo uma IV, enquanto em  $m^{f'}$  ambas as IVs podem ser afetadas.

A dificuldade reside no fato de que melhorias na confiabilidade levam a uma piora da fragmentação, e vice-versa. Diante disso, a estratégia adotada neste trabalho é encontrar um bom *trade-off* entre esses dois aspectos. Matematicamente, isso pode ser expresso por uma função que pondere a contribuição das métricas de confiabilidade e de fragmentação; devido à natureza das métricas, o desejável é que o valor dessa função seja o menor possível. Esse raciocínio é capturado pela função objetivo mostrada na Equação 3.7, na qual  $\beta$  e  $(1 - \beta)$  são s pesos atribuídos pelo InP a confiabilidade e fragmentação, respectivamente ( $0 \leq \beta \leq 1$ ).

$$\text{minimizar: } \beta \cdot (\text{confiabilidade}) + (1 - \beta) \cdot (\text{fragmentacao}) \quad (3.7)$$

As próximas seções detalham o modelo, iniciando por um conjunto de restrições para a alocação do grafo, passando pelas restrições referentes a confiabilidade e fragmentação e finalizando com a combinação das duas perspectivas em uma visão refinada da função objetivo.

### 3.2 ALOCAÇÃO DE GRAFOS

O modelo proposto neste trabalho busca a alocação ótima de recursos físicos para hospedar IVs confiáveis. Este problema envolve conjuntos de variáveis, parâmetros e funções para representar os objetivos dos usuários e dos InPs. Antes dos modelos de fragmentação e confiabilidade serem apresentados, será introduzida uma formulação tradicional para a alocação de recursos de IVs. Esse problema tenta resolver a alocação de recursos em um substrato físico (ou grafo de uma infraestrutura física) para hospedar recursos virtuais (um grafo de IV), o qual é sabidamente um problema NP-difícil [Chowdhury, Rahman e Boutaba 2009].

A Tabela 3 sumariza a notação usada na formulação. O substrato físico é modelado como um conjunto de máquinas ( $\mathcal{M}^p$ ) que se comunicam através de *switches* ( $\mathcal{S}^p$ ). O conjunto de nós  $\mathcal{N}^p$  agrupa as máquinas físicas e os *switches*. Um conjunto de enlaces  $\mathcal{E}^p$  define todas as interconexões físicas; um enlace  $e^p$  entre dois nós  $n_1^p$  e  $n_2^p$  é denotado pela tupla  $(n_1^p, n_2^p)$ . Similarmente, uma IV  $v$  é composta por conjuntos de máquinas ( $\mathcal{M}^v$ ), *switches* ( $\mathcal{S}^v$ ) e enlaces ( $\mathcal{E}^v$ ). O conjunto  $\mathcal{N}^v$  denota a união de  $\mathcal{M}^v$  e  $\mathcal{S}^v$ . Uma máquina  $m$  (de  $\mathcal{M}^v$  ou  $\mathcal{M}^p$ ) tem capacidades  $\text{CM}_{m,r}$ , onde  $r \in \mathcal{R}^m$  indexa atributos individuais (e.g., CPU, memória, armazenamento) dessa máquina. Para nós físicos, CM representa a capacidade total disponível, e para nós virtuais indica a demanda de recurso. Do mesmo modo, um *switch*  $s$  (de  $\mathcal{S}^v$  ou  $\mathcal{S}^p$ ) tem capacidades  $\text{CS}_{s,r}$ , onde, por exemplo,  $r$ , um recurso pertencente a  $\mathcal{R}^s$ , é a memória disponível (ou qualquer outra informação pertinente para representar um comutador). Finalmente, um enlace  $e$  entre dois nós tem capacidades  $\text{CE}_{e,r}$ , onde  $r \in \mathcal{R}^e$  é um de seus atributos (e.g., latência, largura de banda). Optou-se por modelar capacidade de CPU para as máquinas, capacidade de memória para os *switches* e largura de banda para os enlaces. Como discutido, o modelo indica a presença de conjuntos para modelar esses atributos, podendo ser facilmente estendido para cobrir mais atributos funcionais e não funcionais, oriundos da definição de SLAs.

O objetivo da alocação de IV é encontrar um mapeamento combinado para alocar nós virtuais em nós físicos ( $mn_{n^v, n^p}$ ) e enlaces virtuais em caminhos físicos ( $me_{e^v, e^p}$ ). Um nó virtual  $n^v$  é mapeado em um único nó físico  $n^p$ , o qual pode hospedar múltiplos nós virtuais ( $mn: \mathcal{N}^v \uparrow \mathcal{N}^p$ ); se  $n^p$  hospeda  $n^v$ , então  $mn_{n^v, n^p}$  assume 1, senão 0. Um enlace virtual  $e^v$  é mapeado em um caminho físico composto por um ou mais enlaces físicos  $e^p$ . Um enlace físico pode hospedar múltiplos enlaces virtuais ( $me: \mathcal{E}^v \uparrow \mathcal{E}^p$ ); se  $e^p$  hospeda  $e^v$ , então  $me_{e^v, e^p}$  assume 1, senão 0.

Na alocação *online* de IVs, usuários podem submeter pedidos de IVs em momentos diferentes. Nesse cenário, quando uma IV  $v_a$  já está alocada e outra IV  $v_b$  é solicitada, o InP deve escolher uma de duas abordagens: (i) realocar todas as IVs, o que pode melhorar as métricas de fragmentação do substrato e confiabilidade de IVs mas au-

Tabela 3 – Notação para o modelo de alocação de grafo.

Notação	Descrição
<b>Conjuntos</b>	
$m^v \in \mathcal{M}^v$	uma máquina virtual (MV)
$m^p \in \mathcal{M}^p$	uma máquina física (MF)
$r \in \mathcal{R}^m$	uma capacidade de máquina (e.g., CPU ou memória)
$s^v \in \mathcal{S}^v$	um <i>switch</i> virtual
$s^p \in \mathcal{S}^p$	um <i>switch</i> físico
$r \in \mathcal{R}^s$	uma capacidade de <i>switch</i> (e.g., memória)
$n^v \in \mathcal{N}^v = \mathcal{M}^v \cup \mathcal{S}^v$	um nó virtual (máquina ou <i>switch</i> )
$n^p \in \mathcal{N}^p = \mathcal{M}^p \cup \mathcal{S}^p$	um nó física (máquina ou <i>switch</i> )
$(n^v, n^p) \in \mathcal{A}^n \subseteq \mathcal{N}^v \times \mathcal{N}^p$	um nó virtual pré-alocado em um nó físico
$e^v := (n_1^v, n_2^v) \in \mathcal{E}^v \subseteq \mathcal{N}^v \times \mathcal{N}^v$	um enlace virtual unidirecional
$e^p := (n_1^p, n_2^p) \in \mathcal{E}^p \subseteq \mathcal{N}^p \times \mathcal{N}^p$	um enlace físico unidirecional
$r \in \mathcal{R}^e$	uma capacidade de enlace (e.g., largura de banda)
$(e^v, e^p) \in \mathcal{A}^e \subseteq \mathcal{E}^v \times \mathcal{E}^p$	um enlace virtual pré-alocado em um físico
<b>Parâmetros</b>	
$\text{CM}\{m \in \mathcal{M}^v \cup \mathcal{M}^p, r \in \mathcal{R}^m\}$ , real	as capacidades de nós
$\text{CS}\{s \in \mathcal{S}^v \cup \mathcal{S}^p, r \in \mathcal{R}^s\}$ , real	as capacidades de <i>switch</i>
$\text{CE}\{e \in \mathcal{E}^v \cup \mathcal{E}^p, r \in \mathcal{R}^e\}$ , real	as capacidades de enlace
<b>Variáveis</b>	
$mn\{n^v \in \mathcal{N}^v, n^p \in \mathcal{N}^p\}$ , binário	o mapeamento entre os nós virtuais e físicos
$me\{e^v \in \mathcal{E}^v, e^p \in \mathcal{E}^p\}$ , binário	o mapeamento entre os enlaces virtuais e físicos

Fonte: produção do próprio autor.

menta o tempo de alocação e introduz problemas de migração de IVs; ou (ii) manter os nós e enlaces pré-alocados em seus respectivos recursos físicos, o que evita descumprir SLAs existentes. É importante ressaltar que em ambos os casos as alocações podem ser ótimas. Optou-se pela estratégia (ii), atualmente adotada por provedores populares de nuvens, o que envolve o gerenciamento de dois novos conjuntos:  $\mathcal{A}^n$  para um nó virtual  $n^v$  pré-alocado em um nó físico  $n^p$ , denotado pela tupla  $(n^v, n^p)$ ; e  $\mathcal{A}^e$  para um enlace  $e^v$  pré-alocado em um enlace físico  $e^p$ , denotado pela tupla  $(e^v, e^p)$ .

As restrições para o modelo de alocação são especificadas pelas Equações 3.8–3.16. Inicialmente, a Equação 3.8 garante que uma MP não hospedará mais MVs do que as suas capacidades permitem;

restrições similares são aplicadas aos *switches* (Equação 3.9) e enlaces (Equação 3.10).

$$\sum_{m^v \in \mathcal{M}^v} \text{CM}_{m^v, r} \cdot mn_{m^v, m^p} \leq \text{CM}_{m^p, r}, \quad \forall r \in \mathcal{R}^m, \forall m^p \in \mathcal{M}^p \quad (3.8)$$

$$\sum_{s^v \in \mathcal{S}^v} \text{CS}_{s^v, r} \cdot mn_{s^v, s^p} \leq \text{CS}_{s^p, r}, \quad \forall r \in \mathcal{R}^s, \forall s^p \in \mathcal{S}^p \quad (3.9)$$

$$\sum_{e^v \in \mathcal{E}^v} \text{CE}_{e^v, r} \cdot me_{e^v, e^p} \leq \text{CE}_{e^p, r}, \quad \forall r \in \mathcal{R}^e, \forall e^p \in \mathcal{E}^p \quad (3.10)$$

As Equações 3.11 e 3.12 garantem que um nó virtual será alocado em somente um nó físico e esses nós serão do mesmo tipo (i.e., MVs devem ser hospedadas em MPs, não em *switches*, e vice-versa). Para satisfazer essas condições, são definidos subconjuntos de  $\mathcal{N}^p$ , na Equação 3.11 como

$$\mathcal{N}^{p'} := \forall n^p \in \mathcal{N}^p: (n^v \in \mathcal{M}^v \wedge n^p \in \mathcal{M}^p) \vee (n^v \in \mathcal{S}^v \wedge n^p \in \mathcal{S}^p)$$

e na Equação 3.12 como

$$\mathcal{N}^{p''} := \forall n^p \in \mathcal{N}^p: (n^v \in \mathcal{M}^v \wedge n^p \in \mathcal{S}^p) \vee (n^v \in \mathcal{S}^v \wedge n^p \in \mathcal{M}^p)$$

Portanto, se  $n^v$  é uma MV, então  $\mathcal{N}^{p'}$  conterá MPs e  $\mathcal{N}^{p''}$  conterá *switches* (e vice-versa).

$$\sum_{n^{p'} \in \mathcal{N}^{p'}} mn_{n^v, n^{p'}} = 1, \quad \forall n^v \in \mathcal{N}^v \quad (3.11)$$

$$\sum_{n^{p''} \in \mathcal{N}^{p''}} mn_{n^v, n^{p''}} = 0, \quad \forall n^v \in \mathcal{N}^v \quad (3.12)$$

Seguindo a formulação proposta por [Bays et al. 2012], a Equação 3.13 garante que um enlace virtual  $e^v$  entre quaisquer dois nós virtuais,  $n_1^v$  e  $n_2^v$ , será alocado em um ou mais enlaces físicos que formam um caminho conexo entre os nós físicos  $n_1^p$  e  $n_2^p$  que hospedam  $n_1^v$  e  $n_2^v$ . Para satisfazer essas condições, são definidos subconjuntos de  $\mathcal{N}^p$ , na primeira soma como

$$\mathcal{N}^{p'''} := \forall n_2^p \in \mathcal{N}^p: (n_1^p, n_2^p) \in \mathcal{E}^p$$

e na segunda soma como

$$\mathcal{N}^{p''''} := \forall n_2^p \in \mathcal{N}^p: (n_2^p, n_1^p) \in \mathcal{E}^p$$

Em ambos os casos,  $\mathcal{N}^{p''''}$  e  $\mathcal{N}^{p''''}$  terão somente nós físicos  $n_2^p$  conectados com  $n_1^p$ .

$$\sum_{(n_2^{p''''}) \in \mathcal{N}^{p''''}} me_{n_1^v, n_2^v, n_1^p, n_2^{p''''}} + \sum_{(n_2^{p''''}) \in \mathcal{N}^{p''''}} me_{n_1^v, n_2^v, n_2^{p''''}, n_1^p} = mn_{n_1^v, n_1^p} + mn_{n_2^v, n_1^p}, \quad (3.13)$$

$$\forall (n_1^v, n_2^v) \in \mathcal{E}^v, \forall n_1^p \in \mathcal{N}^p$$

A Equação 3.14 determina que dois nós virtuais diretamente conectados não podem estar alocados em um mesmo nó físico. Isso é realizado para que a capacidade reservada seja garantida e para que os canais de comunicação formem um grafo conexo com esses nós.

$$mn_{n_1^v, n^p} + mn_{n_2^v, n^p} \leq 1, \quad \forall n^p \in \mathcal{N}^p, \forall (n_1^v, n_2^v) \in \mathcal{E}^v \quad (3.14)$$

Aspectos temporais não são considerados no modelo. A abordagem de solução é *on-line*: a cada requisição de IV, uma nova alocação é obtida usando o modelo. Quando uma IV é retirada do substrato,

as capacidades alocadas para essa IV são restituídas no modelo. Por razões de eficiência, uma implementação prática desse modelo pode realizar a alocação periodicamente, considerando o conjunto de requisições de IV que foram submetidas desde a última alocação efetuada. Essa ainda seria considerada uma solução *on-line*, que tem como principal vantagem em relação à solução *off-line* o fato de não precisar conhecer todas as requisições de IV de antemão, sendo portanto mais flexível. Para garantir que IVs pré-alocadas permanecerão hospedadas em seus recursos físicos, as Equações 3.15 e 3.16 garantem que nós e enlaces virtuais permanecerão nos mesmos nós e caminhos físicos, respectivamente, ou seja, não ocorre a migração de recursos.

$$mn_{n^v, n^p} = 1, \quad \forall n^v \in \mathcal{N}^v, \forall n^p \in \mathcal{N}^p: (n^v, n^p) \in \mathcal{A}^n \quad (3.15)$$

$$me_{e^v, e^p} = 1, \quad \forall e^v \in \mathcal{E}^v, \forall e^p \in \mathcal{E}^p: (e^v, e^p) \in \mathcal{A}^e \quad (3.16)$$

### 3.3 CONFIABILIDADE

A Tabela 4 sumariza as notações complementares usadas no modelo de confiabilidade. Todos os nós virtuais de uma dada IV têm um identificador único  $v$ , para serem distinguíveis das outras IVs (i.e.,  $\text{REQ}: \mathcal{N}^v \uparrow \mathcal{V}$ ; se  $n^v \in \text{IV } v$ , então  $\text{REQ}_{n^v, v}$  assume 1, senão 0). Como discutido na Seção 3.1.1, as *métricas de falha local* medem o impacto de uma falha física em uma única IV, e as *métricas de falha global* medem o impacto em todos os nós e enlaces, indiferente a qual IV eles pertencem. Para alcançar um valor ótimo para essas métricas, os recursos virtuais devem ser distribuídos pelos domínios de faltas, levando em conta todos os tipos de DF (i.e., circuitos, *racks*, servidores, enlaces). No entanto, distribuir por DFs os nós e enlaces de *uma* IV (métrica de falha local) não é o mesmo que distribuir por DFs *todos* os nós e enlaces (métrica de falha global). Por exemplo, a primeira métrica pode distribuir uma IV  $v_a$  em dois *racks* e uma IV  $v_b$  nos *mesmos* dois *racks*, mesmo se houver um terceiro *rack* desocupado; o pior



caso de falha de *rack* afeta uma fração de cada IV, mas *sempre* afeta ambas. A segunda métrica pode distribuir todos os nós nos três *racks*, ocupando todos eles, mas deixar os nós de  $v_a$  concentrados em um *rack* e os de  $v_b$  em outro; o pior caso de falha de *rack* afetaria uma porção menor de nós, mas seria *fatal* para uma IV.

Tabela 4 – Notação para o modelo de confiabilidade.

Notação	Descrição
<b>Conjuntos</b>	
$v \in \mathcal{V}$	um identificador de IV
$d \in \mathcal{D}^n$	um DF para <i>nó</i>
$t \in \mathcal{T}^n$	um tipo de DF para <i>nós</i>
$d \in \mathcal{D}^e$	um DF para <i>enlace</i>
$t \in \mathcal{T}^e$	um tipo de DF para <i>enlace</i>
<b>Parâmetros</b>	
$\text{REQ}\{n^v \in \mathcal{N}^v, v \in \mathcal{V}\}$ , binário	o mapeamento de nós virtuais e IVs
$\text{MND}\{n^p \in \mathcal{N}^p, d \in \mathcal{D}^n\}$ , binário	o mapeamento de <i>nós</i> físicos e DFs
$\text{MDTN}\{d \in \mathcal{D}^n, t \in \mathcal{T}^n\}$ , binário	o mapeamento de DFs e tipos para <i>nós</i>
$\text{MED}\{e^p \in \mathcal{E}^p, d \in \mathcal{D}^e\}$ , binário	o mapeamento de <i>enlaces</i> físicos e DFs
$\text{MDTE}\{d \in \mathcal{D}^e, t \in \mathcal{T}^e\}$ , binário	o mapeamento de DFs e tipos para <i>enlaces</i>
$\text{EFLN}\{v \in \mathcal{V}, t \in \mathcal{T}^n\}$ , inteiro	os <i>fln</i> máximos estimados
$\text{EFGN}\{t \in \mathcal{T}^n\}$ , inteiro	os <i>fgn</i> máximos estimados
$\text{EFLE}\{v \in \mathcal{V}, t \in \mathcal{T}^e\}$ , inteiro	os <i>fle</i> máximos estimados
$\text{EFGE}\{t \in \mathcal{T}^e\}$ , inteiro	os <i>fge</i> máximos estimados
<b>Variáveis</b>	
$\text{fln}\{v \in \mathcal{V}, t \in \mathcal{T}^n\}$ , inteiro	o número máximo de <i>nós de uma IV</i> em um DF de um tipo
$\text{fgn}\{t \in \mathcal{T}^n\}$ , inteiro	o número máximo de <i>todos os nós</i> em um DF de um tipo
$\text{fle}\{v \in \mathcal{V}, t \in \mathcal{T}^e\}$ , inteiro	o número máximo de <i>enlaces de uma IV</i> em um DF de um tipo
$\text{fge}\{t \in \mathcal{T}^e\}$ , inteiro	o número máximo de <i>todos os enlaces</i> em um DF de um tipo

Fonte: produção do próprio autor.

Um nó ou enlace físico pertence a um ou mais tipos de domínios de falta. Por exemplo, uma máquina  $m^p$  é um *servidor* dentro de um *rack* conectado a um *circuito elétrico*, e portanto  $m^p$  pertence a três tipos. Adicionalmente, um *switch*  $s^p$  está dentro de um *rack* e também conectado a um *circuito*, mas não é hospedado em um *servidor* (con-

sequentemente,  $s^p$  pertence a dois tipos). O *mapeamento entre DFs e tipos* é denotado como MDTN (Mapeamento entre DFs e Tipos para Nós), para nós (i.e.,  $\text{MDTN}: \mathcal{D}^n \uparrow \mathcal{T}^n$ ), e como MDTE (Mapeamento entre DFs e Tipos para Enlaces), para enlaces (i.e.,  $\text{MDTE}: \mathcal{D}^e \uparrow \mathcal{T}^e$ ). Se um DF  $d$  pertence a um tipo  $t$ , então  $\text{MDTN}_{d,t}$  assume 1, senão 0. O *mapeamento entre recursos físicos e DFs* é denotado como MND (Mapeamento entre Nós e DFs), para nós (i.e.,  $\text{MND}: \mathcal{N}^p \uparrow \mathcal{D}^n$ ), e como MED (Mapeamento entre Enlaces e DFs), para enlaces (i.e.,  $\text{MED}: \mathcal{E}^p \uparrow \mathcal{D}^e$ ); se  $n^p$  está no tipo  $d$ , então  $\text{MND}_{n^p,d}$  assume 1, senão 0 (o mesmo se aplica aos enlaces).

A função objetivo de confiabilidade visa minimizar simultaneamente as métricas de falha local e global. As Equações 3.17–3.20 especificam as restrições para alcançar a função objetivo ótima garantindo as confiabilidades local e global. Para simplificar a formulação, a Equação 3.17 define um subconjunto de  $\mathcal{N}^v$ , para considerar somente nós de uma dada IV  $v$ , como

$$\mathcal{N}^{v'} := \forall n^v \in \mathcal{N}^v: \text{REQ}_{n^v,v}$$

e a Equação 3.19 define um subconjunto de  $\mathcal{N}^p$ , para considerar somente nós físicos no DF  $d$ , como

$$\mathcal{N}^{p'} := \forall n^p \in \mathcal{N}^p: \text{MND}_{n^p,d}$$

As mesmas abordagens são válidas para enlaces: na Equação 3.18, um subconjunto de  $\mathcal{E}^v$ , para considerar somente enlaces de uma IV  $v$ , é definido como

$$\mathcal{E}^{v'} := \forall (n_1^v, n_2^v) \in \mathcal{E}^v: \text{REQ}_{n_1^v,v}$$

e, na Equação 3.20, um subconjunto de  $\mathcal{E}^p$ , para incluir somente enlaces no DF  $d$ , é definido como

$$\mathcal{E}^{p'} := \forall e^p \in \mathcal{E}^p: \text{MED}_{e^p, d}$$

As Equações 3.17 e 3.18 garantem que, para cada tipo de DF  $t$ , o número de nós (ou enlaces) virtuais de uma dada IV afetados por uma falha deste tipo (no pior caso) é no máximo  $fln$  (ou  $fle$ ).

$$\sum_{n^{p'} \in \mathcal{N}^{p'}} \sum_{n^{v'} \in \mathcal{N}^{v'}} mn_{n^{v'}, n^{p'}} \leq fln_{v, t}, \quad \forall v \in \mathcal{V}, \forall d \in \mathcal{D}^n, \forall t \in \mathcal{T}^n: \text{MDTN}_{d, t} \quad (3.17)$$

$$\sum_{e^{p'} \in \mathcal{E}^{p'}} \sum_{e^{v'} \in \mathcal{E}^{v'}} me_{e^{v'}, e^{p'}} \leq fle_{v, t}, \quad \forall v \in \mathcal{V}, \forall d \in \mathcal{D}^e, \forall t \in \mathcal{T}^e: \text{MDTE}_{d, t} \quad (3.18)$$

As Equações 3.19 e 3.20 fornecem restrições análogas para as métricas de falha global.

$$\sum_{n^{p'} \in \mathcal{N}^{p'}} \sum_{n^v \in \mathcal{N}^v} mn_{n^v, n^{p'}} \leq fgn_t, \quad \forall d \in \mathcal{D}^n, \forall t \in \mathcal{T}^n: \text{MDTN}_{d, t} \quad (3.19)$$

$$\sum_{e^{p'} \in \mathcal{E}^{p'}} \sum_{e^v \in \mathcal{E}^v} me_{e^v, e^{p'}} \leq fge_t, \quad \forall d \in \mathcal{D}^e, \forall t \in \mathcal{T}^e: \text{MDTE}_{d, t} \quad (3.20)$$

As Equações 3.21–3.24 são as componentes de falha da função objetivo (Equação 3.29). Elas minimizam o número de falhas de nós e enlaces virtuais, tendendo a distribuir os recursos de IVs pelo substrato físico. Conforme as definições da Seção 3.1.1, a Equação 3.21 (e Equação 3.22 para enlaces) denotam a razão entre a soma das falhas locais (e.g., soma das falhas de circuito, *rack* e servidor para cada IV) e o produto entre o número de tipos e o número de IVs solicitadas.

$$\text{FLN} := \frac{\sum_{t \in \mathcal{T}^n} \sum_{v \in \mathcal{V}} \frac{fln_{v,t}}{\text{EFLN}_{v,t}}}{|\mathcal{T}^n| \cdot |\mathcal{V}|} \quad (3.21)$$

$$\text{FLE} := \frac{\sum_{t \in \mathcal{T}^e} \sum_{v \in \mathcal{V}} \frac{fle_{v,t}}{\text{EFLE}_{v,t}}}{|\mathcal{T}^e| \cdot |\mathcal{V}|} \quad (3.22)$$

Similarmente, na Equação 3.23 (Equação 3.24), esses valores denotam a razão entre as falhas globais e o número de tipos. Essas componentes são normalizadas pelos valores de pior caso estimados para as falhas globais de nós e enlaces: EFLN (Estimativa de Falha Local de Nó), EFGN (Estimativa de Falha Global de Nó); e para as falhas locais de nós e enlace: EFLE (Estimativa de Falha Local de Enlace), EFGE (Estimativa de Falha Global de Enlace).

$$\text{FGN} := \frac{\sum_{t \in \mathcal{T}^n} \frac{fgn_t}{\text{EFGN}_t}}{|\mathcal{T}^n|} \quad (3.23)$$

$$\text{FGE} := \frac{\sum_{t \in \mathcal{T}^e} \frac{fge_t}{\text{EFGE}_t}}{|\mathcal{T}^e|} \quad (3.24)$$

Essas estimativas representam o número de recursos em cada tipo de DF. Por exemplo, na Figura 9,  $v_a$  tem três MVs e dois *switches* (cinco nós) e  $v_b$  somente duas MVs; já que ambas podem ser alocadas no mesmo *rack* (como exemplificado pela alocação  $m^{fr}$ ),  $\text{EFLN}_{v_a,r} = 5$ ,  $\text{EFLN}_{v_b,r} = 2$  e  $\text{EFGN}_r = 7$ . Cada IV tem um número de nós e enlaces comprometidos por um pior caso de falha, denotados como  $fln$  e  $fle$ , respectivamente. Esses valores representam as métricas de falha local (nós e enlaces). Similarmente, considerando todos os recursos virtuais pertencendo a uma única grande IV, as métricas de falha global são

denotadas como  $fgn$  e  $fge$  para nós e enlaces, respectivamente. As métricas locais e globais têm diferentes valores para cada tipo de DF.

### 3.4 FRAGMENTAÇÃO

A Tabela 5 sumariza as notações complementares usadas no modelo de fragmentação. A fragmentação pode ser analisada em diferentes níveis de granularidade, tais como capacidades de recursos ou máquinas físicas. Foi considerada a fragmentação no nível de nós e enlaces físicos, como adotado em trabalhos anteriores [Greenberg et al. 2008, Koslovski et al. 2011]. Como discutido na Seção 3.1.2, se uma alocação  $a$  requer menos nós/enlaces físicos para hospedar uma dada IV do que uma alocação  $b$ , então  $a$  induz uma menor fragmentação no substrato do que  $b$ . A fragmentação de nós é formalmente denotada por  $frn$ , e a fragmentação de enlaces por  $fre$ .

Tabela 5 – Notação para o modelo de fragmentação.

Notação	Descrição
$frn\{n^p \in \mathcal{N}^p\}$ , binário	a fragmentação dos <i>nós</i> físicos
$fre\{e^p \in \mathcal{E}^p\}$ , binário	a fragmentação dos <i>enlaces</i> físicos

Fonte: produção do próprio autor.

As Equações 3.25 e 3.26 especificam as restrições de fragmentação. A Equação 3.25 verifica se um nó físico hospeda ao menos um nó virtual. O primeiro membro define a razão entre o número de nós virtuais alocados em um recurso físico e o total de nós virtuais. O segundo membro controla o estado de fragmentação do nó físico  $\mathcal{N}^p$ : em uso, se  $frn$  assume 1, ou ocioso, se assume 0. Deve-se ressaltar que, aqui, o número de nós virtuais hospedados é irrelevante, o que importa é se um recurso físico está *ocupado* ou *ocioso*. Analogamente, a Equação 3.26 verifica o estado dos enlaces físicos (i.e., se hospedam ao menos um enlace virtual), seguindo a mesma abordagem.

$$\frac{\sum_{n^v \in \mathcal{N}^v} mn_{n^v, n^p}}{|\mathcal{N}^v|} \leq \lceil fm_{n^p} \rceil, \quad \forall n^p \in \mathcal{N}^p \quad (3.25)$$

$$\frac{\sum_{e^v \in \mathcal{E}^v} me_{e^v, e^p}}{|\mathcal{E}^v|} \leq \lceil fre_{e^p} \rceil, \quad \forall e^p \in \mathcal{E}^p \quad (3.26)$$

As Equações 3.27 e 3.28 definem as métricas de fragmentação usadas na função objetivo (Equação 3.29, discutida na Seção 3.5). Ambas as métricas devem ser minimizadas para diminuir o número de recursos físicos em uso. Elas denotam a razão entre os recursos físicos em uso e o número total de recursos (Equação 3.27 para nós e Equação 3.28 para enlaces).

$$\text{FrN} = \frac{\sum_{n^p \in \mathcal{N}^p} fm_{n^p}}{|\mathcal{N}^p|} \quad (3.27)$$

$$\text{FrE} = \frac{\sum_{e^p \in \mathcal{E}^p} fre_{e^p}}{|\mathcal{E}^p|} \quad (3.28)$$

### 3.5 MODELO COMBINADO

A função objetivo (Equação 3.29) para o modelo combinado minimiza as métricas de falhas (Equações 3.21–3.24) e fragmentação (Equações 3.27 e 3.28). O peso  $\beta$  pode ser ajustado pelo InP para obter o equilíbrio desejado entre confiabilidade e fragmentação. Por exemplo, um provedor pode querer aumentar  $\beta$  para oferecer melhor confiabilidade por uma tarifa *premium*. O peso  $\beta$  é explorado mais

adiante no Capítulo 4.

$$\begin{aligned} \text{minimizar:} \quad & \beta \cdot \left( \frac{\text{FLN} + \text{FGN} + \text{FLE} + \text{FGE}}{4} \right) + (1 - \beta) \cdot \left( \frac{\text{FrN} + \text{FrE}}{2} \right) \\ \text{sujeito a:} \quad & \text{Equações 3.8–3.20, 3.25 e 3.26} \end{aligned} \tag{3.29}$$

### 3.6 CONSIDERAÇÕES PARCIAIS

Este capítulo discutiu dois objetivos conflitantes na alocação de infraestruturas virtuais: confiabilidade de IVs e fragmentação do substrato físico. Métricas foram estabelecidas para cada um desses objetivos: a confiabilidade é dada pelo pior caso de falha de nó ou enlace físico, que afeta a maior quantidade de recursos virtuais de uma IV (perspectiva local) ou de todas as IVs (perspectiva global); e a fragmentação é dada pela quantidade de nós ou enlaces ativos. Esses objetivos, quando otimizados, beneficiam tanto o InP quanto o usuário. Também foi apresentado um modelo tradicional de alocação de grafos.

Um modelo de otimização foi introduzido para tratar o problema. A função objetivo (Equação 3.29) foi modelada como um problema de programação linear inteira mista (PLIM). Ela permite que, dado um conjunto de recursos físicos e um de IVs, sejam satisfeitos os objetivos tanto do InP quanto do usuário.

Os trabalhos relacionados não incorporam simultaneamente a otimização de ambos os objetivos, maximizar confiabilidade de IVs e minimizar fragmentação. O InP que desejava otimizar um desses objetivos acabava, invariavelmente, prejudicando o outro, afetando a sua receita tanto por aumento no consumo de energia elétrica tanto por quebra de SLAs. Fez-se necessária uma ferramenta que permitisse explorar os *trade-offs* da fragmentação e confiabilidade de IVs. No próximo capítulo, diferentes aspectos do modelo proposto serão avaliados experimentalmente.





## 4 AVALIAÇÃO

Neste capítulo, o modelo proposto no capítulo anterior é avaliado, focando no *trade-off* entre confiabilidade e fragmentação (Equação 3.29 na página 63). O ambiente de testes é descrito na Seção 4.1, os resultados são discutidos na Seção 4.2 e, por fim, a Seção 4.3 encerra o capítulo.

### 4.1 AMBIENTE DE TESTES

O modelo foi implementado em CPLEX<sup>1</sup> e executado em um computador AMD Phenom II X4 com 4GB de RAM e Linux Ubuntu 12.04.02. As topologias das infraestruturas físicas e virtuais foram geradas usando um simulador escrito em Python. Conforme a Figura 12, a topologia da infraestrutura física é invariável, com um projeto de DC de três camadas baseado na arquitetura de referência da Cisco [Cisco Data Center Infrastructure: 2.5 Design Guide 2007], a qual é normalmente implementada em grandes DCs de alta-resiliência. Na camada de acesso há enlaces de 1 Gbps e 8 *racks*, com 12 servidores e um *switch top-of-rack* em cada um. Nas camadas de agregação e núcleo há enlaces de 10 Gbps e 8 e 4 *switches*, respectivamente. A Tabela 6 resume o número de nós para cada camada e DC. São considerados cinco tipos de domínio de faltas: *data center*, circuito elétrico, *rack*, servidor e enlace. Os tipos *data center*, circuito, *rack* e servidor já foram utilizados anteriormente em trabalhos como [Bodík et al. 2012]; já o tipo enlace, no contexto de falhas dos enlaces das infraestruturas físicas para fornecimento de IVs confiáveis, é apresentado primeiramente neste trabalho. Os *switches* de agregação e de núcleo não foram considerados como domínios de faltas separados para simplificar o modelo, mas qualquer outro agrupamento pode ser adotado no modelo, sendo a critério do provedor.

O número de recursos virtuais em uma IV (Tabela 6) e suas

---

<sup>1</sup>Versão 12.5.1, <http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/>

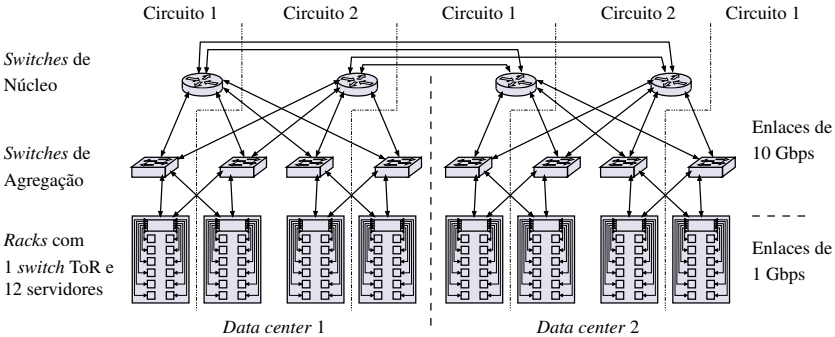


Figura 12 – Topologia da infraestrutura física.  
Fonte: produção do próprio autor.

capacidades foram gerados aleatoriamente de acordo com uma distribuição uniforme. IVs têm entre três e cinco nós. As demandas dos recursos de nós virtuais são dimensionadas como uma fração das capacidades físicas; assim, uma MV demanda de 25 a 50% da capacidade de CPU de uma MF, um *switch* virtual demanda de 15 a 25% da memória de um *switch* físico e um enlace virtual demanda de 1 a 3% da largura de banda de um enlace de acesso (i.e., de 10 a 30Mbps). Esses parâmetros são similares àqueles usados em trabalhos relacionados [Bays et al. 2012, Oliveira et al. 2013]. As topologias de IVs seguem uma regra: uma MV está sempre conectada a outro nó e um *switch* está sempre conectado a dois outros nós. Basicamente, essa regra gera um conjunto de grafos conexos direcionados, que representam cenários reais. Ou seja, *hosts* são interconectados por *switches* para trocar dados e, por sua vez, *switches* interconectam pelo menos dois *hosts* distintos.

Tabela 6 – Tamanho das infraestruturas físicas e virtuais.						
Tipo	Tamanho	Servidores	Switches			DC
			Acesso	Agregação	Núcleo	
físico	fixo	96	8	8	4	2
virtual	variável	2-3		1-2		-

Fonte: produção do próprio autor.

O modelo é avaliado em dois cenários de carga (Tabela 7): um substrato com baixa carga de uso (BC) e outro com alta carga de uso (AC). Em cada cenário são geradas 150 IVs a serem alocadas no substrato. Uma IV nova chega a cada iteração discreta e é aceita (e consequentemente alocada) se o substrato não estiver saturado, permanecendo ativa por um tempo de vida definido. Caso a requisição de IV não possa ser alocada, ela é rejeitada; um dos objetivos de um provedor é rejeitar o mínimo de requisições possível. O tempo de vida de uma IV é de poucas (para o cenário BC) ou muitas (para o cenário AC) iterações. Em média, o modelo leva 8,9 segundos para alocar uma IV, com tempo mínimo inferior a um segundo e tempo máximo de 160 segundos (a distribuição dos tempos de execução do modelo exibe cauda longa: 90% das requisições foram atendidas em 25,6 segundos, e 95% em 40,3 segundos). O cenário BC tenta mostrar o comportamento do modelo frente a um baixo uso do substrato, quando todas as IVs são desalocadas em poucas iterações e têm a taxa de aceitação em 100%. Já o cenário AC tenta estressar o substrato, mostrando o comportamento do modelo quando as IVs levam mais tempo para serem desalocadas e têm a taxa de aceitação inferior a 100%. Ou seja, em BC, as requisições não devem ser rejeitadas, ao passo que em AC haverá rejeição de requisições e, por outro lado, deve-se analisar em AC a taxa de rejeição para cada valor de  $\beta$  e a diferença entre elas (e.g., valores de taxa de rejeição muito altos podem significar um valor ruim de  $\beta$ ).

Tabela 7 – Cenários avaliados.

Cenário	Descrição	Requisições de IV	Duração de uma IV
BC	Baixa Carga	150	5 iterações
AC	Alta Carga	150	85 iterações

Fonte: produção do próprio autor.

## 4.2 RESULTADOS

As Figuras 13–18 ilustram os resultados para cada cenário utilizando diferentes pesos de falha ( $\beta$ ) e fragmentação ( $1 - \beta$ ) na função objetivo (Equação 3.29 na página 63). Em cada uma, o eixo x representa o número de uma iteração e a chegada de uma nova requisição de IV, e o eixo y representa os valores médios das métricas de falha ou fragmentação ou da taxa de aceitação das requisições desde a iteração inicial até uma dada iteração. Os valores de  $\beta$  menores que 0,5 foram omitidos por motivos de clareza, uma vez que resultavam em uma confiabilidade pior sem diminuir significativamente a fragmentação. Para fins de comparações, também são mostradas as curvas de fragmentação “pura” ( $\beta = 0$ ) e confiabilidade “pura” ( $\beta = 1$ ), as quais fornecem linhas de base de fragmentação e falha; o principal objetivo é avaliar os cenários com  $\beta \in [0,5; 0,9]$ .

As Figuras 13(a)–16(b) ilustram as falhas local e global de nó e enlace (FLN, FGN, FLE e FGE na Equação 3.29) para cada cenário. Nesses cenários, a linha de base de fragmentação representa o *limite superior de falha* (i.e., os piores valores de falha) e a linha de base de falha representa o *limite inferior de falha* (i.e., os valores ótimos de falha). Os resultados para a falha local de nós (FLN) são fornecidos pelas Figuras 13(a) e 13(b) para os cenários BC e AC, respectivamente. Na Figura 13(a), para  $\beta = 0$ , as primeiras alocações conseguem valores melhores de FLN do que as subseqüentes, até uma estabilização. Pode ser observado que  $\beta \in [0,5; 0,9]$  resultam em efeitos similares entre si nessa métrica, com as curvas praticamente sobrepostas às da linha de base de falha. No cenário AC (Figura 13(b)), quando  $\beta = 0$  o percentual de nós falhos de uma mesma IV diminui conforme mais IVs vão sendo alocadas. Isso se deve ao fato de, mesmo no caso de todas as IVs alocadas tiverem requisitado a capacidade mínima de cada recurso, sempre ser utilizado mais de um servidor para hospedá-las, implicando a distribuição de algumas IVs e um limite superior no seu valor de FLN. Contudo, esses valores iniciais refletem o limite inferior alcançado no espalhamento de uma IV pelo substrato, que é a falha de um DF hospedando somente um nó dessa IV. A piora nos valores de FLN, em relação ao cenário BC, conforme as iterações se

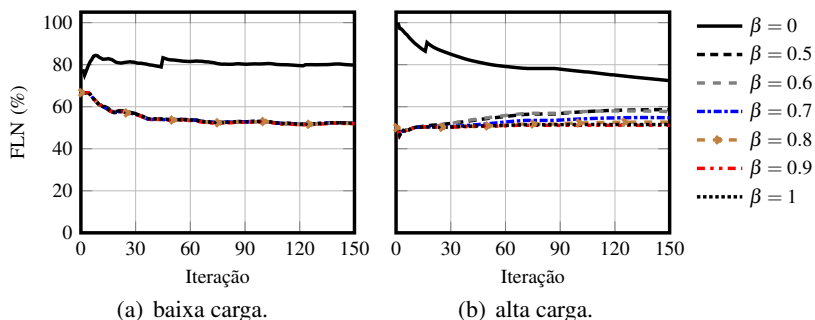


Figura 13 – Resultados experimentais de falhas locais de nó para os cenários BC e AC.

Fonte: produção do próprio autor.

sucedem é resultado da alocação de pelo menos dois nós de algumas das IVs em tipos de DFs menos diversificados, como circuito e DCs. Para valores de  $\beta \in [0,5;0,6]$ , o peso que é conferido à fragmentação acaba degradando levemente a confiabilidade das IVs: 7,08 pontos percentuais (pp) acima da linha de base de falha. No entanto, só a linha de base de fragmentação é visivelmente pior em termos de falhas, estando 27,65 pp e 13,81 pp acima das outras curvas, respectivamente, nos cenários BC e AC.

Os resultados para a falha global de nós (FGN) são fornecidos pelas Figuras 14(a) e 14(b) para os cenários BC e AC, respectivamente. A tendência para ambos os cenários é que os valores de FGN melhorem desde as primeiras iterações. Isso acontece porque tende a ser maior a fração de nós virtuais afetados no melhor caso de falha (i.e., o melhor caso de sobrevivência, quando é afetado somente um nó virtual dentre todos os alocados) quando somente uma IV está alocada do que quando se tem duas ou mais IVs alocadas, onde o melhor caso de falha afeta um nó virtual dentre muitos outros mais. Assim, aqui os valores iniciais refletem os limites superiores nos valores de FGN. Para valores de  $\beta \in [0,5;0,7]$  a confiabilidade das IVs é um pouco afetada, provavelmente devido ao peso atribuído a  $\beta$ , ficando 5,06 pp acima de linha de base de falha. O cenário AC (Figura 14(b)) mostra valores menos estáveis de FGN. Em relação ao cenário BC, todas as curvas

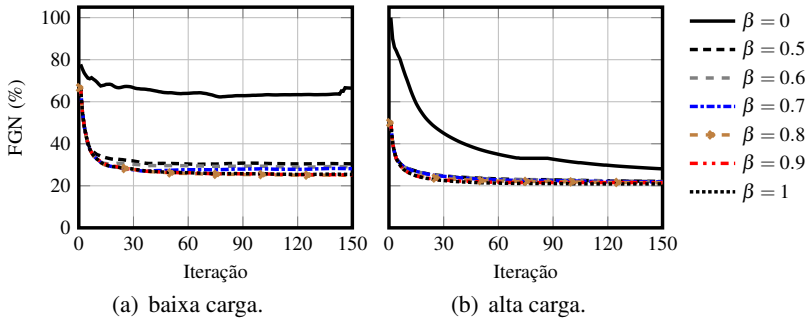


Figura 14 – Resultados experimentais de falhas globais de nó para os cenários BC e AC.

Fonte: produção do próprio autor.

conseguem melhores valores de FGN, possivelmente ocasionados pela situação apresentada anteriormente: mais nós virtuais estão hospedados fora dos DFs considerados os de pior caso de falha. Aqui, a linha de base de fragmentação converge para as outras curvas conforme o substrato vai sendo saturado e requisições de IVs vão sendo rejeitadas. No entanto, só a linha de base de fragmentação é visivelmente pior em termos de falhas, estando 35,99pp e 5,97pp acima das outras curvas, respectivamente, nos cenários BC e AC.

Os resultados para a falha local de enlaces (FLE) são fornecidos pelas Figuras 15(a) e 15(b) para os cenários BC e AC, respectivamente. Basicamente, as tendências são as mesmas que as métrica de falha para nós, só que aplicadas a enlaces. Assim, conforme avançam as iterações, os valores de FLE melhoram para a linha de base de fragmentação e pioram para as outras curvas. No cenário AC (Figura 15(b)), os valores de  $\beta \in [0,5; 0,7]$  implicam uma pequena degradação na confiabilidade das IVs, ficando 2,42pp acima de linha de base de falha. Isso ocorre pelo valor de  $\beta$  de cada curva. Contudo, só a linha de base de fragmentação é visivelmente pior em termos de falhas, estando 7,72pp e 6,27pp acima das outras curvas, respectivamente, nos cenários BC e AC.

Os resultados para a falha global de enlaces (FGE) são fornecidos pelas Figuras 16(a) e 16(b) para os cenários BC e AC, res-

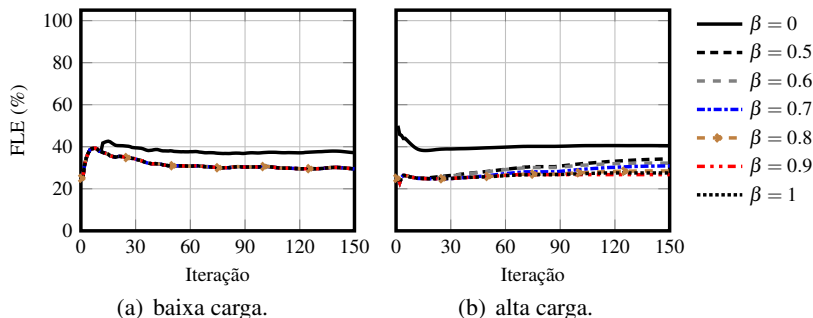


Figura 15 – Resultados experimentais de falhas locais de enlace para os cenários BC e AC.

Fonte: produção do próprio autor.

pectivamente. A tendência para ambos os cenários é que, conforme as iterações progredirem, os valores de FGE melhorem para todas as curvas. No cenário BC (Figura 16(a)), os valores de  $\beta \in [0,5;0,7]$  implicam uma pequena degradação na confiabilidade das IVs, ficando 5,68pp acima de linha de base de falha. Também no cenário AC essa degradação ocorre, com 5,22pp acima de  $\beta = 1$ , seguindo a mesma linearidade no aumento do valor de  $\beta$  e das métricas de falha. Novamente, só a linha de base de fragmentação é visivelmente pior em termos de falhas, estando 9,52pp e 4,15pp acima das outras curvas, respectivamente, nos cenários BC e AC.

A fragmentação resultante dos diferentes valores de  $\beta$  também deve ser avaliada, para ser possível observar de que forma os pesos afetam o *trade-off* entre fragmentação e confiabilidade. As Figuras 17(a) e 17(b) ilustram a fragmentação combinada de nó e enlace (FrN e FrE na Equação 3.29) para cada cenário. Nesses cenários, a linha de base de falha representa o *limite superior de fragmentação* (i.e., os piores valores de fragmentação) e a linha de base de fragmentação representa o *limite inferior de fragmentação* (i.e., os valores ótimos de fragmentação). No cenário BC (Figura 17(a)), as métricas resultam em valores muito próximos, com  $\beta \in [0,5;0,9]$  estando somente 7,15pp acima da linha de base de fragmentação e 8,75pp abaixo da de falha, em média. No cenário AC (Figura 17(b)),  $\beta \in [0,5;0,9]$  está

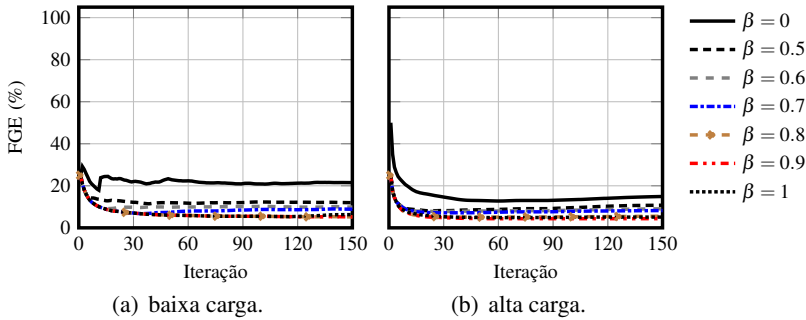


Figura 16 – Resultados experimentais de falhas globais de enlace para os cenários BC e AC.

Fonte: produção do próprio autor.

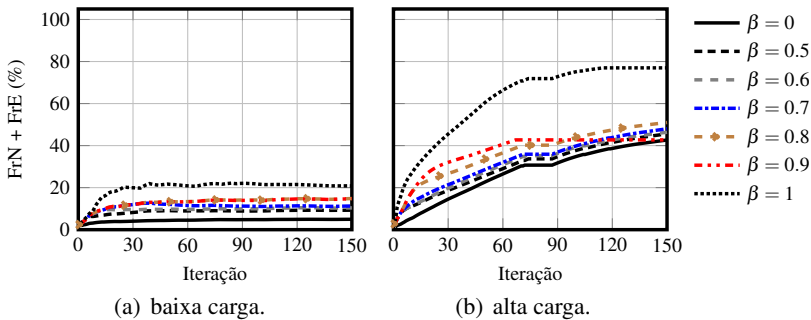


Figura 17 – Resultados experimentais de fragmentações de nó e enlace para os cenários BC e AC.

Fonte: produção do próprio autor.

somente 0,83pp acima da linha de base de fragmentação e 30,26pp menos fragmentada que a de falha, em média. Os valores de FrN e FrE aumentam quase linearmente conforme  $\beta$  aumenta, sendo diretamente proporcionais, no cenário BC. Entretanto, no cenário AC, conforme o substrato vai sendo saturado e as requisições de IVs vão sendo rejeitadas, mesmo o caso de  $\beta = 0$  acaba induzindo a uma fragmentação igual ou muito próxima do que quando  $\beta \in [0,6; 0,7]$ .

As Figuras 18(a) e 18(b) mostram a taxa de aceitação para



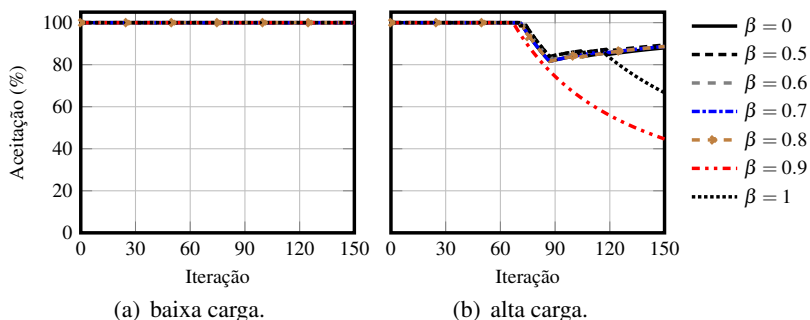


Figura 18 – Resultados experimentais de taxas de aceitação para os cenários BC e AC.

Fonte: produção do próprio autor.

os cenários BC e AC, respectivamente. Como no cenário BC não há saturação do substrato, todas as requisições de IVs são aceitas. No cenário AC, o modelo com valores de  $\beta \in [0,5;0,8]$  alcança as melhores taxas de aceitação, com curvas praticamente sobrepostas. Essa não linearidade na taxa de aceitação mostra que, para cada substrato físico e conjunto de requisições de IVs, somente determinados valores de  $\beta$  (não necessariamente contíguos) resultarão em um bom *trade-off* entre confiabilidade e fragmentação.

É possível perceber que todos os valores de  $\beta \in [0,5;0,9]$  resultam em boas métricas. Contudo, levando em consideração a taxa de aceitação para o cenário AC (Figura 18(b)), de alta carga, os valores para  $\beta \in [0,9;1]$  tiveram alta taxa de rejeição de IVs, de 44,67% a 66,67%, respectivamente. É importante notar, ainda, que para  $\beta \in [0,5;0,8]$  a taxa de aceitação foi 0,83 pp maior que a da fragmentação “pura” ( $\beta = 0$ ).

#### 4.3 CONSIDERAÇÕES PARCIAIS

Com base nos cenários avaliados e levando em consideração ambas as perspectivas (confiabilidade e fragmentação) e a taxa de aceitação, é possível dizer que, nesses cenários, os valores de  $\beta$  de 0,5

até 0,8 fornecem os melhores *trade-offs*. Um InP realizando alocações com esse modelo e usando esses pesos poderá aprimorar a confiabilidade das IVs a um pequeno aumento no custo, balanceando estes objetivos conflitantes. Enquanto outros cenários possam se beneficiar mais de outros valores para  $\beta$ , os resultados evidenciam que o modelo proposto pode ser usado para encontrar um bom *trade-off* entre fragmentação e confiabilidade na alocação de IVs. Eles destacam que InPs podem oferecer IVs confiáveis sem comprometer os custos de gerenciamento, uma vez que é possível aprimorar a confiabilidade enquanto se mantêm as métricas de fragmentação próximas às suas linhas de base ótimas.

## 5 TRABALHOS RELACIONADOS

Os trabalhos correlatos encontrados na literatura podem ser categorizados em dois grupos, os que tratam de confiabilidade em redes virtuais e os que tratam de confiabilidade em IVs. Esses trabalhos foram analisados e comparados com o presente trabalho a partir de outras três características: 1) o que é considerado como confiabilidade; 2) o que é considerado como fragmentação; e 3) uso de recursos de *backup* e/ou migração de recursos.

O primeiro grupo de trabalhos discutidos compreende os que tratam requisitos de confiabilidade na alocação de redes virtuais. Em [Bodík et al. 2012] é proposto um modelo para mapeamento entre MVs e recursos de *data centers* levando em conta requisitos de disponibilidade. Os nós no substrato são agrupados em DFs distintos, e a função objetivo busca maximizar a confiabilidade no pior caso de falha e minimizar a utilização da largura de banda (isso favorece, e.g., a hospedagem de uma rede virtual em um único *rack* ou sob um mesmo *switch* de agregação). Contudo, como cada rede virtual é somente considerada isoladamente (abordagem é análoga à métrica de falha local apresentada neste trabalho), isto pode levar a uma situação indesejável da perspectiva das falhas globais. Em [Rahman, Aib e Boutaba 2010] são introduzidos um modelo e uma heurística de três fases para a alocação de redes virtuais confiáveis. São minimizadas métricas de penalidades por violações de SLAs e de utilização de largura de banda, que é reservada também para recursos de *backup*. A heurística aplica uma estratégia de migração de nós para tentar resolver o problema de falha de nós físicos no substrato, induzindo a um *overhead* na alocação.

Em [Barla, Schupke e Carle 2012] é proposto um modelo para a minimização da latência em redes virtuais, que são espalhadas por diferentes *data centers* para aumentar a confiabilidade. Nessa abordagem, o mapeamento entre nós virtuais e *data centers* pode ser controlado tanto pelo usuário (que define um DC primário e um de *backup* para a sua rede virtual) quanto pelo provedor (que determina quais DCs devem ser usados). No entanto, o modelo não considera a capacidade de recursos virtuais, e o esquema de confiabilidade não somente

usa uma maior granularidade de DFs (e.g., regiões geográficas), mas também delega ao usuário tarefas com objetivos conflitantes (distribuir nós para melhorar a confiabilidade enquanto tenta minimizar a latência). Esse trabalho foi estendido para fornecer resiliência fim a fim nas camadas física e virtual [Barla et al. 2013]. De forma similar ao presente trabalho, os autores consideram que um InP pode oferecer para o usuário um serviço de resiliência ou confiabilidade. Essa discussão é estendida considerando uma camada intermediária entre os usuários finais e os InPs, os operadores, que podem gerenciar os serviços dos usuários finais. Ambas as abordagens podem ser combinadas para oferecer um roteamento com resiliência ótima e uma qualidade de experiência melhor.

A alocação de IVs considerando requisitos de confiabilidade é discutida em [Koslovski et al. 2010, Marcon et al. 2013, Yu et al. 2010]. Em [Koslovski et al. 2010], a confiabilidade de IVs é alcançada reservando recursos de *backup*, os quais podem ser compartilhados entre vários usuários. Em [Marcon et al. 2013] é proposto um modelo PLIM e uma heurística de alocação ótima que agrupa IVs com confiança mútua para mitigar ataques onde aplicações maliciosas tentam saturar a largura de banda da rede, além de um modelo de custo para uso de MVs e recursos de rede. A largura de banda é minimizada e é discutida a fragmentação interna de recursos sob um conceito e contexto diferentes da forma com que é tratada fragmentação neste trabalho. Em [Yu et al. 2010] é proposta uma solução para mapeamento e realocação de IVs confiáveis em um substrato federado. Essa abordagem parte de uma solução ótima de mapeamento, refinada por uma heurística que adiciona recursos de *backup* considerando uma sequência de possíveis cenários de falha, o que consequentemente aumenta os custos de provisionamento.

Modelos ótimos para alocação de redes virtuais com suporte a requisitos tanto de resiliência de enlace [Oliveira et al. 2013] quanto de confiabilidade [Bays et al. 2012] também foram propostos; nenhum desses trabalhos considera a fragmentação de recursos, o que pode levar a uma utilização ineficaz do substrato. Com a exceção de [Koslovski et al. 2010], os trabalhos citados não incorporam simultaneamente recursos de computação e de comunicação: mesmo os que li-

dam com IVs consideram apenas um tipo de recurso, como canais de comunicação. Todas as propostas resolvem problemas de confiabilidade reservando recursos de contingência no substrato ou distribuindo os recursos virtuais pelo substrato, o que aumenta a fragmentação e, conseqüentemente, os custos para o InP.

A Tabela 8 resume e compara os trabalhos relacionados acima, também considerando as características do presente trabalho. São identificados: 1) os tipos de virtualização utilizados (somente de rede e canais de comunicação ou de toda a infraestrutura); 2) o que é considerado ou o que mais se aproxima de confiabilidade (e.g., se considera domínios de falta, se são locais e/ou globais e a granularidade, se considera confiabilidade de enlaces, MVs etc.); 3) o que é considerado ou o que mais se aproxima de fragmentação (e.g., noções de estratégias que, indiretamente, induzem a uma minimização da fragmentação do substrato físico, como largura de banda – BW, *bandwidth* –, e inclusão de quebra de SLAs e precificação na função objetivo); e 4) se o trabalho aborda alguma estratégia de uso de recursos sobressalentes, de *backup*, ou de migração. Dentre essas características, as linhas que aparecem destacadas em negrito são as mais desejáveis e que acabam induzindo a um efeito semelhante ao do modelo desenvolvido no presente trabalho.

Tabela 8 – Comparação entre os trabalhos relacionados.

Trabalho	Virtualiz.	Confiabil.	Fragment.	Backup/Migração
[Bodík et al. 2012]	redes	<b>DFs; global</b>	BW	<b>não aplica</b>
[Rahman, Aib e Boutaba 2010]	redes	enlaces	BW/SLAs	<i>backup/migração</i>
[Barla et al. 2013]	redes	<b>DFs reg. geográf.;</b> usuár./InP	BW	<i>backup</i>
[Oliveira et al. 2013]	redes	resiliência en- lace	BW	<i>backup</i>
[Bays et al. 2012]	redes	confiabilidade	BW	<b>não aplica</b>
[Koslovski et al. 2010]	<b>infraest.</b>	MTBF	<b>preço &amp; BW</b>	<i>backup</i>
[Marcon et al. 2013]	<b>infraest.</b>	confiança mútua	<b>preço &amp; BW</b>	<b>não aplica</b>
[Yu et al. 2010]	<b>infraest.</b>	federado	não aplica	<i>backup</i>
[Cavalcanti, Obelheiro e Koslovski 2014a]	infraest.	DFs; local & global	BW/CPU/etc.	<b>não aplica</b>

Fonte: produção do próprio autor.

## 6 CONCLUSÃO

Provedores de infraestruturas oferecem, como serviço, infraestruturas virtuais, que são uma conjunção dos conceitos de recursos virtuais de rede e de máquinas virtuais. Esse serviço de computação em nuvem, no modelo *pay-as-you-go*, possibilita aos usuários executarem as suas aplicações sem a necessidade de um investimento significativo em TI.

Nuvens de *data centers* estão sujeitas a falhas aleatórias. Por si só, essas falhas introduzem um custo de manutenção e de novos equipamentos. Além do mais, na falha de um recurso de DC também devem ser consideradas as aplicações e as IVs hospedadas no recurso afetado. Para o usuário, a interrupção da execução de parte ou de toda a sua aplicação pode ter altos custos, como tempo de execução e dados perdidos, sem que, muitas vezes, o InP tenha como recompensá-lo adequadamente sem ser com créditos para reexecutar a aplicação. Para tentar evitar que uma grande porção das IVs alocadas sejam afetadas por uma única falha, ocasionando quebra de SLAs e perdas na receita, o InP pode alocar os recursos virtuais de forma esparsa. Contudo, a simples distribuição de IVs pelo substrato físico introduz problemas de fragmentação, utilizando mais recursos do que seria necessário e aumentando os gastos do InP para manter um mesmo conjunto de IVs.

Este trabalho apresentou um modelo de alocação ótima de recursos para IVs confiáveis. Desenvolvido em programação linear inteira mista, o modelo tenta selecionar os recursos físicos que devem hospedar as IVs de forma a minimizar tanto o número de recursos virtuais afetados por uma falha no substrato físico quanto a fragmentação deste substrato. Assim, as contribuições propostas foram alcançadas. Resultados experimentais demonstraram que o modelo é capaz de otimizar simultaneamente: i) a confiabilidade, beneficiando os usuários executando suas aplicações em IVs garantidamente confiáveis e, ao mesmo tempo, garantindo ao InP que o maior número de SLAs sejam cumpridas; e ii) a fragmentação, minimizando a utilização do substrato e diminuindo os custos do provedor. Essa otimização fornece um ganho expressivo na confiabilidade de IVs, apesar da pequena degradação

da fragmentação do substrato físico, em comparação à confiabilidade alcançada pela minimização somente da fragmentação. Este é o primeiro trabalho a considerar conjuntamente a confiabilidade de IVs e a fragmentação do substrato. Além disso, as métricas de falha no substrato tentam satisfazer tanto os objetivos do usuário (métrica de falha local) quanto os do InP (métrica de falha global), abraçando ambas as perspectivas.

Destaca-se que este trabalho de pesquisa proporcionou alguns resultados importantes em termos de publicação:

- *Optimal resource allocation for survivable virtual infrastructures*, publicado no *International Conference on Design of Reliable Communication Networks (DRCN)*, Ghent, Bélgica, na edição de 2014 [Cavalcanti, Obelheiro e Koslovski 2014a]; e,
- Alocação ótima de recursos para infraestruturas virtuais confiáveis, publicado no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Florianópolis, Brasil, na edição de 2014 [Cavalcanti, Obelheiro e Koslovski 2014b].

Como perspectivas de trabalhos futuros, podem ser investigadas heurísticas para soluções em tempo polinomial. Apesar dos modelos de programação linear fornecerem resultados ótimos, o tempo para a obtenção destes é exponencial, sendo pouco prática a implementação dos modelos de PLIM em ambientes de produção. Contudo, o modelo ótimo desenvolvido neste trabalho proporciona uma linha de base de comparação com as eventuais heurísticas que tentem resolver o problema otimização simultânea de IVs confiáveis e de fragmentação do substrato.

Outro trabalho seria aumentar a expressividade do modelo, permitindo aos usuários especificarem quais componentes de suas IVs devem ser confiáveis (e, então, alocá-las em recursos que falham de forma independente). Por exemplo, dado um conjunto de nós *slaves* de uma aplicação, onde um deles pode estar executando uma parte crítica e o seu resultado só é obtido após o término da execução e o restante fornece resultados *online* ou não executa tarefas críticas. Nesse cenário, somente um dos nós precisa de um recurso virtual confiável, e



pode ser um custo muito alto para o usuário colocar os outros nós não críticos em recursos virtuais confiáveis e mais caros, sendo que estas tarefas pode ser reexecutadas mais facilmente.

Uma outra perspectiva seria a avaliação deste modelo frente a outras topologias de *data centers* e de IVs, ou até diferentes topologias de DCs interconectadas, e diferentes distribuições e quantidade (saturações) de requisições de IVs, como uma distribuição normal.

E, por fim, garantir a exclusão mútua de IVs pela categoria do usuário: empresas concorrentes (bancos, por exemplo), com requisitos de confidencialidade e integridade, podem desejar executar as suas IVs em recursos físicos distintos, devido a preocupações de violação do isolamento de IVs ou roubo e/ou alteração de dados sensíveis. Nesse cenário, tanto máquinas virtuais quanto *switches* e enlaces de IVs conflitantes teriam que ser alocadas em recursos físicos distintos, para satisfazer o requisito de exclusão mútua.



## REFERÊNCIAS

ANDERSEN, D. G. **Theoretical Approaches to Node Assignment**. Massachusetts Institute of Technology, 2002.  
<http://repository.cmu.edu/cgi/viewcontent.cgi?article=1079&context=compsci>.

ANHALT, F.; KOSLOVSKI, G.; PRIMET, P. V.-B. Specifying and Provisioning Virtual Infrastructures with HIPerNET. **International Journal of Network Management**, John Wiley & Sons, Inc., v. 20, n. 3, p. 129–148, 2010.

BARI, M. F. et al. Data Center Network Virtualization: A Survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 15, n. 2, p. 909–928, 2013.

BARLA, I. et al. Optimal Design of Virtual Networks for Resilient Cloud Services. In: **Proceedings of Design of Reliable Communication Networks (DRCN)**. [S.l.: s.n.], 2013. p. 218–225.

BARLA, I. B.; SCHUPKE, D. A.; CARLE, G. Resilient Virtual Network Design for End-to-End Cloud Services. In: **Proceedings of IFIP Networking**. [S.l.: s.n.], 2012. p. 161–174.

BAYS, L. R. et al. Security-Aware Optimal Resource Allocation for Virtual Network Embedding. In: **Proceedings of ACM/IEEE/IFIP CNSM**. [S.l.: s.n.], 2012. p. 378–384.

BODÍK, P. et al. Surviving Failures in Bandwidth-Constrained Datacenters. In: **Proceedings of ACM SIGCOMM**. [S.l.: s.n.], 2012. p. 431–442.

CAVALCANTI, G. A. d. S.; OBELHEIRO, R. R.; KOSLOVSKI, G. Optimal Resource Allocation for Survivable Virtual Infrastructures. In: IEEE. **Proceedings of Design of Reliable Communication Networks (DRCN)**. [S.l.], 2014a. p. 1–8.

CAVALCANTI, G. A. d. S.; OBELHEIRO, R. R.; KOSLOVSKI, G. Alocação ótima de Recursos para Infraestruturas Virtuais Confiáveis. In: **Simp Brasileiro de Redes de Computadores e Sistemas Distribus (SBRC)**. [S.l.: s.n.], 2014b. p. 3–16.

CHOWDHURY, N.; BOUTABA, R. A Survey of Network Virtualization. **Computer Networks**, Elsevier, v. 54, n. 5, p. 862–876, 2010.

CHOWDHURY, N. M. K.; BOUTABA, R. Network Virtualization: State of the Art and Research Challenges. **IEEE Communications Magazine**, v. 47, n. 7, p. 20–26, 2009.

CHOWDHURY, N. M. K.; RAHMAN, M. R.; BOUTABA, R. Virtual Network Embedding With Coordinated Node and Link Mapping. In: **Proceedings of IEEE INFOCOM**. [S.l.: s.n.], 2009. p. 783–791.

CISCO Data Center Infrastructure: 2.5 Design Guide. 2007.

GREENBERG, A. et al. The Cost of a Cloud: Research Problems in Datacenter Networks. **ACM SIGCOMM Computer Communication Review**, v. 39, n. 1, p. 68–73, 2008.

GREENBERG, A. et al. A Clean Slate 4D Approach to Network Control and Management. **ACM SIGCOMM Computer Communication Review**, ACM, v. 35, n. 5, p. 41–54, 2005.

HANDLEY, M. Why the Internet Only Just Works. **BT Technology**, Springer, v. 24, n. 3, p. 119–129, 2006.

KOSLOVSKI, G. et al. Locating Virtual Infrastructures: Users and InP Perspectives. In: **Proceedings of IFIP/IEEE IM**. [S.l.: s.n.], 2011. p. 153–160.

KOSLOVSKI, G. et al. Reliability Support in Virtual Infrastructures. In: **Proceedings of IEEE CloudCom**. [S.l.: s.n.], 2010. p. 49–58.

LAGANIER, J.; PRIMET, P.-B. HIPernet: a Decentralized Security Infrastructure for Large Scale Grid Environments. In: **IEEE/ACM International Workshop on Grid Computing**. [S.l.], 2005. p. 8–pp.

MARCON, D. S. et al. Trust-based Grouping for Cloud Datacenters: Improving Security in Shared Infrastructure. In: **Proceedings of IFIP Networking**. [S.l.: s.n.], 2013. p. 1–9.

MCCARTHY, J. The 10 Biggest Cloud Outages of 2012. **The Channel Company Sites CRN**, 2012.

<http://www.crn.com/slide-shows/cloud/240144284/the-10-biggest-cloud-outages-of-2012.htm>.

MCKEOWN, N. et al. OpenFlow: Enabling Innovation in Campus Networks. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 2, p. 69–74, 2008.

MELL, P.; GRANCE, T. **The NIST Definition of Cloud Computing**. [S.l.], 2011. <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>.

OLIVEIRA, R. R. et al. DoS-Resilient Virtual Networks through Multipath Embedding and Opportunistic Recovery. In: **Proceedings of ACM SAC**. [S.l.: s.n.], 2013. p. 597–602.

PONEMON. **Understanding the Cost of Data Center Downtime: An Analysis of the Financial Impact on Infrastructure Vulnerability**. Emerson Network Power, 2011.

[http://emersonnetworkpower.com/en-US/Brands/Liebert/Documents/White%20Papers/data-center-uptime\\_24661-R05-11.pdf](http://emersonnetworkpower.com/en-US/Brands/Liebert/Documents/White%20Papers/data-center-uptime_24661-R05-11.pdf).

PONEMON. **2013 Study on Data Center Outages**. Emerson Network Power, 2013. [http://emersonnetworkpower.com/en-US/Brands/Liebert/Documents/White%20Papers/2013\\_emerson\\_data\\_center\\_outages\\_sl-24679.pdf](http://emersonnetworkpower.com/en-US/Brands/Liebert/Documents/White%20Papers/2013_emerson_data_center_outages_sl-24679.pdf).

POPEK, G. J.; GOLDBERG, R. P. Formal Requirements for Virtualizable Third Generation Architectures. **Communications of the ACM**, v. 17, n. 7, p. 412–421, 1974.

RAHMAN, M. R.; AIB, I.; BOUTABA, R. Survivable Virtual Network Embedding. In: **Proceedings of IFIP Networking**. [S.l.: s.n.], 2010. p. 40–52.

RAPHAEL, J. R. The Worst Cloud Outages of 2013 (So Far). **Infoworld**, 2013. <http://www.infoworld.com/slideshow/107783/the-worst-cloud-outages-of-2013-so-far-221831>.

ROCHWERGER, B. et al. The RESERVOIR Model and Architecture for Open Federated Cloud Computing. **IBM Journal of Research and Development**, v. 53, n. 4, 2009.

SCHOO, P. et al. Challenges for Cloud Networking Security. In: **Mobile Networks and Management**. [S.l.]: Springer, 2011. p. 298–313.

SOTOMAYOR, B. et al. Virtual Infrastructure Management in Private and Hybrid Clouds. **IEEE Internet Computing**, v. 13, n. 5, p. 14–22, 2009.

XU, J. et al. Survivable Virtual Infrastructure Mapping in Virtualized Data Centers. In: **Proceedings of IEEE GLOBECOM**. [S.l.: s.n.], 2012. p. 196–203. ISSN 2159-6182.

YEOW, W.-L.; WESTPHAL, C.; KOZAT, U. Designing and Embedding Reliable Virtual Infrastructures. In: **Proceedings of ACM SIGCOMM VISA**. [S.l.: s.n.], 2010. p. 33–40.

YU, H. et al. Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures. In: **Proceedings of IEEE GLOBECOM**. [S.l.: s.n.], 2010. p. 1–6.

YU, M. et al. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 2, p. 17–29, 2008.