



Pós-Graduação em Ciência da Computação

Victor Antônio Ribeiro de Lira Cavalcanti

**Uma Estratégia Integrada para Alocação de Redes Virtuais de Computadores
Considerando Disponibilidade e Consumo de Energia**



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
<http://cin.ufpe.br/~posgraduacao>

Recife
2018

Victor Antônio Ribeiro de Lira Cavalcanti

**Uma Estratégia Integrada para Alocação de Redes Virtuais de Computadores
Considerando Disponibilidade e Consumo de Energia**

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: redes de computadores e sistemas distribuídos

Orientador: Eduardo Antônio Guimarães Tavares

Recife
2018

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

C376e Cavalcanti, Victor Antônio Ribeiro de Lira
Uma estratégia integrada para alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia / Victor Antônio Ribeiro de Lira Cavalcanti. – 2018.
106 f.: il., fig., tab.

Orientador: Eduardo Antônio Guimarães Tavares.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2018.
Inclui referências.

1. Redes de computadores. 2. Virtualização de redes. I. Tavares, Eduardo Antônio Guimarães (orientador). II. Título.

004.6

CDD (23. ed.)

UFPE- MEI 2019-006

Tese de Doutorado apresentada por **Victor Antônio Ribeiro de Lira Cavalcanti** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Uma Estratégia Integrada para Alocação de Redes Virtuais de Computadores Considerando Disponibilidade e Consumo de Energia**” Orientador: **Eduardo Antônio Guimarães Tavares** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Dr. Paulo Roberto Freire Cunha
Centro de Informática / UFPE

Prof. Dr. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Prof. Dr. Nelson Souto Rosa
Centro de Informática / UFPE

Prof. Dr. José Neuman de Souza
Departamento de Computação / UFC

Profa. Dra. Erica Teixeira Gomes de Sousa
Departamento de Computação / UFRPE

Visto e permitida a impressão.
Recife, 30 de Novembro de 2018.

Prof. Aluizio Fausto Ribeiro Araújo
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Decido este trabalho à minha avó, Luíza Perônico de Andrade.

AGRADECIMENTOS

Acima de tudo, agradeço à Deus por todas as conquistas obtidas em minha vida.

Agradeço ao meu pai, Ricardo Antônio de Lira Cavalcanti (*in memoriam*), por tudo.

Ao meu orientador e amigo, Eduardo Tavares, pela paciência, ensinamentos e por ter acreditado no meu potencial desde o início da minha vida profissional.

Ao meu irmão Ricardo, pela paciência de realizar a revisão gramatical desta tese.

Aos meus irmãos e a minha avó Luíza Perônico de Andrade (*in memoriam*), pelo suporte durante toda minha existência.

À minha esposa Julia Santos Barbosa de Lira Cavalcanti, por estar comigo ao longo de toda essa jornada, compartilhando os momentos alegres e me apoiando nos momentos difíceis.

Aos meus amigos e familiares, por toda paciência nos dias em que estive ausente para a realização do doutorado.

Ao Prof. Paulo Maciel, pelos diversos ensinamentos e oportunidades desde o período de iniciação científica.

RESUMO

A *Internet* possui um papel vital na sociedade atual, possibilitando a existência de aplicações e serviços em diferentes áreas, como saúde, entretenimento, educação e segurança. Entretanto, apesar da *Internet* ter evoluído consideravelmente em aspectos como velocidade e capacidade, inovações arquiteturais necessárias, por exemplo, para melhor suporte à mobilidade, sofrem resistência em virtude de sua estrutura atual. Nesse contexto, a virtualização de redes de computadores é uma abordagem promissora para resolver o problema atual de ossificação da *Internet*. Um grande desafio associado a esse conceito é o mapeamento das redes virtuais (RV), devido à natureza *NP*-difícil do problema. Com isso, diversas heurísticas têm sido propostas com o objetivo de alcançar alocações eficientes. Contudo, apesar de as abordagens existentes aperfeiçoarem métricas de desempenho, como o custo e a taxa de aceitação das requisições, questões de dependabilidade e consumo energético são usualmente negligenciadas no mapeamento. Nos últimos anos, o crescente custo da energia associado à cada vez maior consciência ecológica têm estimulado o interesse dos provedores de serviço de Internet (ISP) na redução do consumo de energia. Além disso, dependabilidade é um importante requisito, já que envolve métricas como confiabilidade e disponibilidade, as quais impactam diretamente a qualidade do serviço (QoS). Logo, tais métricas são de grande importância e devem ser consideradas na formulação de problemas de otimização para alocação de redes virtuais. Este trabalho propõe a alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia. A abordagem adotada considera uma formulação de problema que concomitantemente considera o consumo de energia e restrições de disponibilidade no problema de mapeamento de redes virtuais. Além disso, um algoritmo baseado na metaheurística GRASP é adotado. O algoritmo proposto utiliza análise de sensibilidade baseada em índices de importância de disponibilidade para atingir o nível de QoS requerido por cada rede virtual. Modelos baseados em diagramas de bloco de confiabilidade (RBD) e redes de Petri estocásticas (SPN) são utilizados para estimar a disponibilidade das redes virtuais. Estudos de caso são propostos para demonstrar a aplicabilidade dos modelos e algoritmo adotados. Os resultados experimentais demonstram o impacto da adoção de questões de dependabilidade e energia na alocação, bem como a relação entre a disponibilidade, o consumo energético, o custo e a receita associada às redes virtuais.

Palavras-chaves: Virtualização de Redes. GRASP. Avaliação de Desempenho de Sistemas. Dependabilidade. Redes de Petri Estocásticas. Diagramas de Bloco de Confiabilidade.

ABSTRACT

The Internet has a vital role in today's society, enabling the existence of applications and services in different areas such as health, entertainment, education and security. However, although the Internet has evolved considerably in aspects as such as speed and capacity, necessary architectural innovations, such as for better support for mobility, suffer particular resistance because of its current structure. Virtualized networks are a promising approach to solving the current ossification problem of Internet. A major challenge is related to virtual network (VN) mapping, due to NP-hard nature problem. Thus, several heuristics have been proposed aiming to achieve efficient allocation. However, while existing approaches consider performance issues, such as the cost and the acceptance rate of requests, dependability and energy consumption issues are neglected in the mapping. Over recent years, growing energy costs and increased ecological awareness have stimulated the interest in reducing energy consumption by Internet service providers (ISP). Dependability is also an important requirement, as it involves metrics such as reliability and availability, which directly impact quality of service (QoS). Therefore, such metrics are of great importance and should be considered in the formulation of optimization problems to allocate virtual networks. This paper presents an energy-efficient mapping of dependable virtual networks. The approach considers a problem formulation that concomitantly takes into account energy consumption and availability constraints for VNE problem, and an algorithm based on GRASP metaheuristic is adopted. The algorithm utilizes a sensitivity analysis based on availability importance to achieve QoS required by each VN. Models based on reliability block diagrams (RBD) and stochastic Petri nets (SPN) are utilized to estimate availability. Results demonstrate the feasibility of the proposed approach, and they show the impact of the adoption of dependability and energy consumption issues in the allocation as well as the trade-off between the availability, energy consumption, cost and revenue of the virtual networks.

Keywords: Network Virtualization. GRASP. Dependability. Stochastic Petri Nets. Reliability Block Diagrams.

LISTA DE ILUSTRAÇÕES

Figura 1 – Principais papéis de negócio em virtualização de redes	23
Figura 2 – Árvore de Dependabilidade. Adaptação de (LAPRIE, 1995)	25
Figura 3 – Redundância <i>Cold Standby</i>	26
Figura 4 – Redundância <i>Hot Standby</i>	27
Figura 5 – Arranjo RBD em série	30
Figura 6 – Arranjo RBD em paralelo	31
Figura 7 – Elementos de uma rede de Petri	33
Figura 8 – Estados de uma lâmpada	34
Figura 9 – Exemplo de grafo de alcançabilidade	35
Figura 10 – Modelo SPN e a respectiva CTMC	37
Figura 11 – Pseudo-código GRASP	38
Figura 12 – Pseudo-código da fase de construção - GRASP	39
Figura 13 – Pseudo-código da fase de busca local - GRASP	39
Figura 14 – Método proposto.	48
Figura 15 – Um ambiente de virtualização de redes.	51
Figura 16 – Exemplo de economia de energia	55
Figura 17 – Modelo de Componente Simples	59
Figura 18 – Modelo <i>Cold Standby</i>	60
Figura 19 – Modelo do componente base de comunicação de enlaces	60
Figura 20 – Exemplo de componente de comunicação de enlaces	61
Figura 21 – Componentes de HW e SW	62
Figura 22 – Sem Redundância - Exemplo de ambiente	62
Figura 23 – Sem Redundância - Exemplo de modelo RBD	62
Figura 24 – <i>Hot standby</i> - Exemplo de ambiente	63
Figura 25 – Geração de modelo <i>hot standby</i>	64
Figura 26 – <i>Cold standby</i> - Exemplo de ambiente	65
Figura 27 – <i>Cold standby</i> - Exemplo de modelo	65
Figura 28 – <i>Cold standby</i> - Exemplo de expressão de disponibilidade	66
Figura 29 – Cenário de alocação sem redundância	66
Figura 30 – Modelo de vazão orientada à disponibilidade sem redundância	67
Figura 31 – Cenário de alocação com <i>Hot standby</i>	68
Figura 32 – Modelo de vazão orientada à disponibilidade <i>hot standby</i>	69
Figura 33 – Análise de sensibilidade - Exemplo	70
Figura 34 – Análise de sensibilidade - Primeiro modelo	71
Figura 35 – Análise de sensibilidade - Segundo modelo	71
Figura 36 – Diagrama de Sequência UML - Comunicação remota com a Mercury	73

Figura 37 – Pseudo-código da metaheurística GRASP	74
Figura 38 – Pseudo-código da fase de construção	75
Figura 39 – Pseudo-código da fase de busca local	75
Figura 40 – Tradução de rede virtual em modelo de dependabilidade	76
Figura 41 – Tradução de rede virtual sem redundância em modelo RBD	77
Figura 42 – Tradução de rede virtual com hot standby em modelo RBD	78
Figura 43 – Tradução de rede virtual com <i>cold standby</i> em modelo SPN	78
Figura 44 – Arquitetura geral da ferramenta	79
Figura 45 – Captura de tela ferramenta MODERADOR	80
Figura 46 – Arquitetura de validação do modelo de consumo de energia	83
Figura 47 – Resultados de disponibilidade	88
Figura 48 – Resultados de downtime	89
Figura 49 – Resultados do preço da eletricidade	90
Figura 50 – Resultados de consumo de energia	90
Figura 51 – Resultados do custo de alocação de recursos	91
Figura 52 – Resultados de receita - SLA 90-95%	91
Figura 53 – Resultados de receita - SLA 95-99%	91
Figura 54 – Resultados de receita - SLA 99-99,99999%	92
Figura 55 – Resultados da taxa de aceitação - SLA 90-95%	92
Figura 56 – Resultados da taxa de aceitação - SLA 95-99%	93
Figura 57 – Resultados da taxa de aceitação - SLA 99-99,99999%	93
Figura 58 – Resultados da vazão	94
Figura 59 – Resultados de perda de pacotes	95
Figura 60 – Resultados dos custos de alocação dos enlaces virtuais	95
Figura 61 – Resultados de disponibilidade	96
Figura 62 – Resultados de confiabilidade	96

LISTA DE TABELAS

Tabela 2 – Trabalhos relacionados	46
Tabela 3 – Resumo da Notação Adotada	50
Tabela 4 – Atributos das transições do componente simples	59
Tabela 5 – Atributos das transições do componente cold standby	60
Tabela 6 – Exemplo de Alocação com Hot Standby	63
Tabela 7 – Exemplo de Alocação com Cold Standby	65
Tabela 8 – Exemplo de Alocação com SA - 1 ^a Tentativa	70
Tabela 9 – Exemplo de Alocação com SA - 2 ^a Tentativa	71
Tabela 10 – Caracterização dos nós físicos	83
Tabela 11 – Caracterização dos nós virtuais	84
Tabela 12 – Resultados da medição - Cenários 1 e 2	84
Tabela 13 – RESULTADO DO TESTES ANOVA - Cenários 1 e 2	84
Tabela 14 – Resultados da medição - Cenário 3	85
Tabela 15 – MTTFS E MTTRS DOS NÓS E ENLACES FÍSICOS	86
Tabela 16 – RESULTADOS DO TESTE ANOVA	87

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface (Interface de Programação de Aplicação)</i>
CMF	<i>Common Mode Failure (Modo de Falha Comum)</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure (Procedimento de Busca Adaptativa Aleatória Gulosa)</i>
HD	<i>Hard Disk (Disco Rígido)</i>
ISP	<i>Internet Service Provider (Provedor de Serviços de Internet)</i>
MIP	<i>Mixed Integer Programming (Programação Inteira Mista)</i>
MTActivate	<i>Mean Time to Activate (Tempo médio de Ativação)</i>
MTBF	<i>Mean Time Between Failures (Tempo Médio Entre Falhas)</i>
MTTF	<i>Mean Time to Failure (Tempo médio de Falha)</i>
MTTR	<i>Mean Time to Repair (Tempo médio de Recuperação)</i>
P2P	<i>Peer-to-Peer (Par-a-Par)</i>
PIP	<i>Physical Infrastructure Providers (Provedores de Infraestrutura Física)</i>
PN	<i>Petri Nets (Redes de Petri)</i>
QoS	<i>Quality of Service (Qualidade de Serviço)</i>
RBD	<i>Reliability Block Diagrams (Diagramas de Bloco de Confiabilidade)</i>
RCL	<i>Restricted Candidate List (Lista de Candidatos Restrita)</i>
SP	<i>Service Providers (Provedores de Serviços)</i>
SPD	<i>Sum of Disjoint Products (Soma dos Produtos Disjuntos)</i>
SPN	<i>Stochastic Petri Nets (Redes de Petri Estocásticas)</i>
VLAN	<i>Virtual Local Area Network (Rede Local Virtual)</i>
VM	<i>Virtual Machine (Máquina Virtual)</i>
VMM	<i>Virtual Machine Monitor (Gerenciador de Máquinas Virtuais)</i>
VN	<i>Virtual Networks (Redes Virtuais)</i>
VNO	<i>Virtual Network Operator (Operadores de Redes Virtuais)</i>
VNP	<i>Virtual Network Providers (Provedores de Redes Virtuais)</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	18
1.2	CONTRIBUIÇÕES	19
1.3	ESTRUTURA DO DOCUMENTO	20
2	REFERENCIAL TEÓRICO	21
2.1	VIRTUALIZAÇÃO DE REDES	21
2.2	DEPENDABILIDADE	24
2.2.1	Técnicas de Tolerância a Falhas	25
2.2.2	Modelos de Dependabilidade	26
2.3	SISTEMAS COERENTES	28
2.3.1	Funções Lógicas	29
2.4	RBD	29
2.4.1	Arranjo em série	30
2.4.2	Arranjo em Paralelo	30
2.4.3	Soma de produtos disjuntos	30
2.5	REDES DE PETRI	33
2.5.1	Propriedades das Redes de Petri	35
2.5.2	Redes de Petri Estocásticas	36
2.6	GRASP	38
2.7	CONSIDERAÇÕES FINAIS	39
3	TRABALHOS RELACIONADOS	40
3.1	MODELAGEM DE DEPENDABILIDADE	40
3.2	ALOCAÇÃO DE REDES VIRTUAIS	42
3.3	CONSIDERAÇÕES FINAIS	45
4	MÉTODO PROPOSTO E FORMULAÇÃO DO PROBLEMA	47
4.1	MÉTODO PROPOSTO	47
4.2	FORMULAÇÃO DO PROBLEMA	48
4.2.1	Rede física	49
4.2.2	Rede Virtual	50
4.2.3	Medição dos Recursos da Rede Física	51
4.2.4	Alocação de Redes Virtuais	52
4.2.4.1	Alocação de Nós	52
4.2.4.2	Alocação de Enlaces	53

4.2.5	Distância do Caminho físico dos Enlaces Virtuais	53
4.2.6	Modelo de Custo de Alocação de Enlace Virtual	54
4.2.7	Modelo de Custo de Alocação de Rede Virtual	54
4.2.8	Receita	54
4.2.9	Modelo de Consumo de Energia	55
4.2.10	Modelo de Precificação de Energia	56
4.2.11	Função Objetivo	57
4.3	CONSIDERAÇÕES FINAIS	57
5	MODELOS E FERRAMENTA PARA ALOCAÇÃO DE REDES VIRTUAIS DE COMPUTADORES CONSIDERANDO DISPONIBILIDADE E CONSUMO DE ENERGIA	58
5.1	MODELOS	58
5.1.1	Componente Simples	58
5.1.2	Cold Standby	59
5.1.3	Componente de Comunicação de Enlaces	59
5.1.4	Composição de Modelos para Representação de Redes Virtuais	61
5.1.4.1	Sem Redundância	62
5.1.4.2	Hot Standby	63
5.1.4.3	Cold Standby	64
5.1.5	Modelagem de Vazão Orientada a Disponibilidade	66
5.1.5.1	Sem redundância	66
5.1.5.2	Hot standby	67
5.2	ALOCAÇÃO DE REDES VIRTUAIS BASEADO EM ANÁLISE DE SENSIBILIDADE	69
5.2.1	Importância de Disponibilidade	70
5.2.2	Exemplo	70
5.3	MERCURY E A API DE COMUNICAÇÃO	72
5.4	ALGORITMO DE ALOCAÇÃO	73
5.5	CONVERSÃO DE REDES VIRTUAIS EM MODELOS DE DISPONIBILIDADE	76
5.5.1	Sem Redundância	76
5.5.2	Hot Standby	77
5.5.3	Cold Standby	77
5.6	VISÃO GERAL DA FERRAMENTA	79
5.6.1	Arquitetura da Ferramenta	79
5.6.2	Ambiente de Alocação de Redes Virtuais	80
5.7	CONSIDERAÇÕES FINAIS	81
6	RESULTADOS EXPERIMENTAIS	82

6.1	VALIDAÇÃO DO MODELO DE CONSUMO ENERGÉTICO	82
6.1.1	Framework de Medição	82
6.1.2	Resultados da Medição	83
6.2	COMPARAÇÃO DAS POLÍTICAS DE ALOCAÇÃO	85
6.2.1	Configurações do Experimento	85
6.2.2	Resultados	87
6.3	AVALIAÇÃO DA VAZÃO ORIENTADA À DISPONIBILIDADE EM ENLA- CES VIRTUAIS	94
6.3.1	Resultados	94
6.4	CONSIDERAÇÕES FINAIS	96
7	CONCLUSÃO E TRABALHOS FUTUROS	98
7.1	LIMITAÇÕES	99
7.2	CONTRIBUIÇÕES	99
7.3	TRABALHOS FUTUROS	100
	REFERÊNCIAS	101

1 INTRODUÇÃO

Notavelmente, a *Internet*, rede mundial de computadores, possui um papel vital na sociedade atual, uma vez que possibilita a comunicação na era da informação, permitindo a existência de uma grande variedade de aplicações de negócios em diferentes áreas, como entretenimento, educação e saúde.

Contudo, apesar desta rede ter evoluído significativamente em termos de capacidade e velocidade, inovações importantes e necessárias em sua arquitetura, por exemplo, para melhor suporte à mobilidade, sofrem particular resistência devido à sua estrutura atual. Além disso, seus principais envolvidos, como os *Internet Service Provider (Provedor de Serviços de Internet)* (ISP) precisam estar convencidos de que essas mudanças são de seu interesse, para que elas de fato ocorram. Logo, essa situação leva a criação de barreiras à introdução de inovações tecnológicas e melhorias, o que resulta no processo atual de ossificação da *Internet* (SCHAFFRATH et al., 2009), (CLARK et al., 2002), (PAPASTERGIOU et al., 2017).

O projeto fim-a-fim da arquitetura de IPs adotada requer acordos e coordenação global para implantação das mudanças necessárias à *Internet*. Contudo, observa-se que, no ambiente atual, mesmo mudanças básicas e necessárias são penosas e lentas, por exemplo, o uso de IPv6 (VAUGHAN-NICHOLS, 2004). A introdução dessas inovações tecnológicas se tornou visivelmente importante nos últimos anos devido à expansão do uso da *Internet* mundialmente em termos de escala e escopo, o que expôs diversas deficiências fundamentais em sua arquitetura.

Logo, enquanto um serviço de conectividade para a entrega de pacotes de rede, a *Internet* cumpre adequadamente o seu papel, mas não foi projetada de forma a atender certas necessidades de comunicação futuras. Exemplos dessas necessidades são: melhor suporte a mobilidade, flexibilidade no roteamento, maior confiabilidade, garantia de serviços robustos, e assim por diante.

Nesse contexto, o conceito de *Virtual Networks (Redes Virtuais)* (VN) tem recebido especial atenção da comunidade científica, já que possibilita a coexistência de múltiplas instâncias de redes virtuais alocadas em uma mesma infraestrutura física compartilhada (CHOWDHURY et al., 2017), (NGUYEN et al., 2017), (ARAÚJO et al., 2018), (HOU et al., 2018). Os aspectos dinâmicos intrínsecos das redes virtuais permitem diversas possibilidades, como por exemplo, sua instanciação de acordo com diferentes critérios de projeto - com controle administrativo customizado, e negociação sob demanda de uma variedade de serviços com usuários em alto-nível (como Operadores de Redes Virtuais e Provedores de Redes Virtuais) (SCHAFFRATH et al., 2009). Além disso, as redes virtualizadas possuem propriedades bastante proeminentes, como o particionamento de recursos, isolamento de tráfego, abstração e uso concorrente de recursos (FERNANDES

et al., 2012).

Em geral, em um ambiente de virtualização de redes, há a dissociação dos papéis dos ISPs tradicionais em duas entidades diferentes e independentes (TURNER; TAYLOR, 2005), (FEAMSTER et al., 2007). Essas entidades são (i) os provedores de infraestrutura, responsáveis pelo gerenciamento da infraestrutura física e (ii) os provedores de serviços, responsáveis por criar redes virtuais alocadas em recursos de diversos provedores de infraestrutura, e provê-las para os clientes finais. Com isso, esse tipo de ambiente permite a implantação de diversas arquiteturas de redes heterogêneas coexistentes, livres das limitações inerentes da *Internet* atual.

Nesse contexto, a comunidade científica têm desenvolvido diversas técnicas e heurísticas para a alocação eficiente dos elementos da infraestrutura física, considerando questões de desempenho relacionadas ao balanceamento de carga e utilização dos recursos físicos (KHAN et al., 2016), (SHAHRIAR et al., 2017), (HOU et al., 2018), (ARAÚJO et al., 2018), (YU et al., 2018). Uma vez que diversas redes virtuais podem compartilhar os mesmos recursos de infraestrutura física, é inegável a importância de que a alocação de cada requisição de rede virtual seja realizado de forma eficiente e efetiva. Logo, ao melhorar as questões de desempenho associadas às alocações, diminui-se a utilização dos recursos da rede física, e, conseqüentemente, incrementa-se os rendimentos dos provedores de infraestrutura física.

Apesar da eficiência dessas abordagens, questões de dependabilidade e consumo energético não são usualmente levadas em conta na alocação. A avaliação de dependabilidade em redes virtualizadas é especialmente importante, uma vez que a infraestrutura física contempla componentes suscetíveis a falhas, as quais podem afetar as redes virtuais, e, conseqüentemente, os clientes que utilizam os serviços providos pela mesma.

Dependabilidade envolve métricas e atributos que impactam diretamente a *Quality of Service (Qualidade de Serviço) (QoS)*, por exemplo, a disponibilidade e a confiabilidade. Portanto, a avaliação de dependabilidade é de grande importância na formulação de mecanismos de alocação de recursos para redes virtualizadas (FERNANDES et al., 2012). Disponibilidade e confiabilidade têm sido cada vez mais vistos como atributos importantes por empresas que proveem serviços de forma contínua e interrupta.

A indisponibilidade de um sistema o impossibilita de prover o serviço para o qual foi projetado. Uma vez que os usuários finais que estejam utilizando o sistema são instantaneamente afetados pelo evento de indisponibilidade, este tem que ser solucionado rapidamente pelos administradores de sistema responsáveis (WAN et al., 2008). A fim de solucionar esse problema, e conseguir prover serviços pelo maior tempo possível, diversas empresas buscam utilizar mecanismos de segurança e técnicas de tolerância a falhas (como o uso de redundância) como forma de evitar prejuízos.

Uma técnica proeminente de tolerância a falhas é o uso de replicação, o qual permite lidar com a indisponibilidade de componentes de *hardware* e *software* individualmente, de

forma que não resulte na indisponibilidade do sistema como um todo. Devido ao fato de que as redes virtuais podem necessitar de alta disponibilidade para prover seu serviço de forma adequada, torna-se particularmente importante a investigação do impacto do uso de diferentes mecanismos de redundância, como *hot* e *cold standby* na disponibilidade de um ambiente de virtualização de redes. A abordagem proposta neste trabalho considera ambas as falhas de *hardware* e *software*.

Nesse contexto, é importante mencionar que, o uso de técnicas de modelagem formal, como Redes de Petri Estocásticas (SPN) (NATKIN, 1990) e Diagramas de Bloco de Confiabilidade (RBD) (RAUSAND; HOYLAND, 2004) podem ser adotadas para a avaliação de métricas de dependabilidade, como a confiabilidade e disponibilidade em diferentes tipos de sistemas e infraestruturas. Contudo, apesar de observar-se a utilização desses modelos (SPN e RBD) para a avaliação de dependabilidade de diversos tipos de sistema (CALLOU et al., 2011), (DANTAS et al., 2012), não se observa na literatura modelos para redes virtuais.

Adicionalmente, o grau de sofisticação dos sistemas e serviços de rede tem conduzido a um maior grau de complexidade na avaliação da qualidade dos serviços. Tal avaliação é especialmente crucial para a virtualização de redes, uma vez que determinados fluxos de dados podem ser negativamente impactados caso o caminho físico utilizado na alocação não satisfaça requisitos de desempenho. Um exemplo desse cenário é um fluxo de dados de áudio, o qual necessita ser entregue através de caminho de enlaces físicos que minimize o atraso na comunicação. Logo, a avaliação de métricas de QoS em enlaces virtuais é imperativa, uma vez que é demonstrado na literatura que o uso de técnicas de virtualização podem resultar em métricas de desempenho de rede instáveis devido a questões relacionadas à infraestruturas física (WANG, 2010).

Além disso, a maioria das técnicas existentes não consideram questões energéticas na alocação das redes virtuais. A eficiência energética na alocação de redes virtuais é especialmente importante devido ao crescente custo da energia associado à cada vez maior consciência ecológica (BOTERO et al., 2012). Por exemplo, o Akamai (um provedor de serviços de redes) possui o custo anual com eletricidade em cerca de US\$ 10 milhões. Um outro exemplo, é a Telecom Italia, a qual consome mais de 2TWh de energia por ano, o que é equivalente ao consumo de cerca de 660.000 famílias (SU et al., 2014). Além disso, a *Internet* contempla 10% do consumo mundial de energia, e é esperado que esse número cresça significativamente num futuro próximo (PICKAVET et al., 2008). Além disto, 80% da emissão de carbono oriunda das tecnologias de informação e comunicação deriva do uso de equipamentos de hardware (CHIARAVIGLIO; MELLIA; NERI, 2012). Nesse contexto, provedores de redes, pesquisadores e fabricantes têm investido esforços objetivando a redução do consumo de energia em diversos componentes (ex: software). Como consequência, alguns estudos têm se dedicado a investigar o consumo de energia no problema de alocação de redes virtuais. Contudo, tais trabalhos não coletaram os

dados de consumo energético em um ambiente de redes real, além de assumirem que os componentes físicos da rede não falham.

1.1 OBJETIVOS

Este trabalho propõe a alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia. A abordagem adotada considera uma formulação de problema que concomitantemente considera o consumo de energia e restrições de disponibilidade no problema de mapeamento. Além disso, a formulação do problema adotada considera a possibilidade da adoção de técnicas de redundância na alocação. Um algoritmo baseado na metaheurística GRASP é adotado, o qual utiliza análise de sensibilidade baseada em índices de importância de disponibilidade para atingir o QoS desejado por cada rede virtual. Outrossim, é apresentada uma ferramenta chamada Emeter, a qual permite medir o consumo de energia em dispositivos computacionais. Tal ferramenta é adotada para coletar os valores de consumo de energia e auxiliar a validação do modelo de energia proposto. Os resultados experimentais demonstram o impacto da adoção de questões de dependabilidade e energia na alocação, bem como a relação entre a disponibilidade, o consumo energético, o custo, a receita e a vazão associada às redes virtuais.

Finalmente, foi desenvolvida a ferramenta MODERADOR, a qual engloba os modelos, técnicas e algoritmos desenvolvidos para permitir a alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia. Através da MODERADOR, gestores e administradores de rede podem se beneficiar das técnicas propostas para obter alocações de redes virtuais confiáveis de computadores considerando consumo de energia.

Mais especificamente, os objetivos desta tese são:

- **Avaliar o consumo energético em redes virtuais.** É proposto um modelo energético para a avaliação do consumo energia de redes virtuais.
- **Avaliar a disponibilidade em redes virtuais.** São propostos modelos formais para a modelagem e avaliação de disponibilidade de redes virtuais, através do uso de redes de Petri estocásticas (SPN) e diagramas de bloco de confiabilidade (RBD).
- **Avaliar a vazão orientada à disponibilidade em enlaces virtuais.** São propostos modelos formais para a modelagem e avaliação da vazão orientada à disponibilidade em enlaces virtuais, através do uso de redes de Petri estocásticas (SPN).
- **Alocar redes virtuais considerando disponibilidade e consumo de energia.** É proposto um algoritmo baseado na metaheurística GRASP para realizar a alocação de redes virtuais em infraestruturas físicas. A abordagem adotada utiliza análise de sensibilidade baseada em índices de importância de disponibilidade para atender às restrições de disponibilidade definidas pelo cliente, minimizando os recursos físicos utilizados.

- **Verificar o impacto da adoção de técnicas de redundância na alocação de redes virtuais.** São propostas diferentes políticas de alocação de redes virtuais, utilizando ou não técnicas de redundância nos recursos físicos adotados na alocação das requisições dos clientes.
- **Validar o modelo energético.** É desenvolvida a ferramenta Emeter, a qual é utilizada na condução de experimentos para validação dos resultados obtidos utilizando o modelo energético proposto.
- **Prover mecanismos para avaliação de disponibilidade e consumo energético de redes virtuais de forma automática.** A metodologia adotada neste trabalho permite que redes virtuais sejam convertidas nos modelos de energia e disponibilidade correspondentes e avaliadas automaticamente através do uso de ferramentas. Mais especificamente, a ferramenta Mercury (SILVA et al., 2012) (CALLOU et al., 2014), (FERREIRA et al., 2013), (MARWAH et al., 2010) é adotada para a avaliação da disponibilidade das redes virtuais.
- **Desenvolver uma ferramenta para a alocação de redes virtuais.** É apresentada uma ferramenta de *software*, chamada MODERADOR, a qual permite a alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia, através do uso dos modelos, técnicas e algoritmos propostos neste trabalho.

1.2 CONTRIBUIÇÕES

Como resultado do trabalho desta tese, podem-se destacar as seguintes contribuições:

- Proposição de um modelo energético validado em uma infraestrutura física real para estimar o consumo de energia de redes virtuais.
- Proposição de uma técnica de modelagem híbrida, a qual adota modelos combinatoriais e baseados em estados para representação de redes virtualizadas. Além disso, pode-se destacar a metodologia hierárquica adotada, que, em conjunto com a técnica de modelagem proposta, permite mitigar a complexidade de representação de redes virtuais. Logo, tal técnica de modelagem permite estimar a disponibilidade em redes virtuais de computadores.
- Elaboração de uma técnica de modelagem, baseada no uso de redes de Petri estocásticas (SPN), para modelar e estimar a vazão orientada a disponibilidade em enlaces virtuais.
- Um algoritmo, baseado na metaheurística GRASP, para alocação de redes virtuais considerando disponibilidade e consumo de energia na alocação. Através da técnica

de alocação proposta, a qual utiliza análise de sensibilidade baseada em índices de importância de disponibilidade, é possível realizar a alocação das redes virtuais utilizando diferentes políticas de redundância, a depender das necessidades específicas de cada requisição.

- A ferramenta de *software* MODERADOR, a qual integra as técnicas e modelos desenvolvidos, bem como os algoritmos propostos para a realização de alocações de redes virtuais considerando as métricas de desempenho e dependabilidade exploradas neste trabalho.

A ferramenta MODERADOR permite que administradores e gestores de redes possam prover redes virtuais capazes de satisfazer clientes com diferentes necessidades de restrições de disponibilidade, de forma a minimizar o consumo de energia, bem como o custo advindo do uso dos recursos da infraestrutura física.

1.3 ESTRUTURA DO DOCUMENTO

Esta tese é organizada como segue. O Capítulo 2 introduz os conceitos básicos necessários para um melhor entendimento do trabalho. O Capítulo 3 apresenta os trabalhos relacionados. O Capítulo 4 demonstra o método proposto para o desenvolvimento do trabalho, bem como a formulação matemática do problema de otimização considerado. O Capítulo 5 apresenta os modelos, o algoritmo de alocação e a ferramenta propostos. O Capítulo 6 apresenta os estudos de caso e o Capítulo 7 apresenta a conclusão desta dese.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos fundamentais para o melhor entendimento desta tese. Primeiramente, é abordado o conceito de virtualização de redes, suas características e benefícios. Posteriormente, são apresentados os conceitos de dependabilidade. Então, introduz-se os formalismos Redes de Petri Estocásticas e Diagramas de Bloco de Confiabilidade. Por fim, apresentam-se os conceitos relativos à metaheurística GRASP.

2.1 VIRTUALIZAÇÃO DE REDES

Virtualização de redes é uma abordagem promissora para lidar com o problema atual de ossificação da *Internet* (CHOWDHURY et al., 2017), (NGUYEN et al., 2017), (ARAÚJO et al., 2018), (HOU et al., 2018), (CAO et al., 2018), (YU et al., 2018). Esse conceito permite a coexistência de múltiplas instâncias de redes virtuais em uma única infraestrutura física compartilhada. Além disso, possibilita o isolamento de tráfego de rede, o total controle administrativo e a customização das instâncias de redes virtuais de acordo com as necessidades específicas de cada cliente (SCHAFFRATH et al., 2009). A virtualização de redes consiste do desacoplamento das responsabilidades dos Provedores de Serviços de *Internet* (ISPs) tradicionais em duas entidades distintas: os *Physical Infrastructure Providers* (*Provedores de Infraestrutura Física*) (PIP) e os Provedores de Serviços (SPs). Nesse contexto, os PIPs são responsáveis pela administração da infraestrutura física que é fornecida aos SPs, os quais contratam os serviços dos PIPs para criar instâncias de redes virtuais e prover seus próprios serviços. Logo, uma rede virtual pode estar alocada em recursos físicos de diferentes PIPs (CHOWDHURY; BOUTABA, 2009).

A separação de uma rede física em diferentes redes lógicas já é um conceito bastante difundido. Contudo, a virtualização de redes difere-se das demais abordagens existentes em termos de como tais redes lógicas são construídas. CHOWDHURY; BOUTABA (2009) apresentam as seguintes definições das técnicas para divisão de uma rede física em redes lógicas:

- **VLAN - Rede Local Virtual:** Consiste da existência de diversas estações dentro de um mesmo *broadcast*, sendo possível obter várias VLANs através da divisão de uma rede local. Em uma *Virtual Local Area Network* (*Rede Local Virtual*) (VLAN), cada estação só recebe quadros que são destinados a ela, através do uso de rotulação dos quadros da VLAN. Então, é possível que o comutador envie os quadros apenas para as portas associadas à VLAN para a qual o quadro está rotulado;
- **VPN - Rede Privada Virtual:** Comumente utilizada em instituições para construir uma rede dedicada entre suas estações. Em uma VPN, as estações não precisam

estar, necessariamente, em uma mesma rede local. Uma das maneiras para a implementação de VPNs é a criação de canais seguros entre cada estação da VPN. Deste modo, os pacotes são encapsulados, e apenas as estações da VPN podem desencapsular e acessar os quadros da rede (CHOWDHRY; RAHMAN; BOUTABA, 2010).

- **Rede Programável Ativa:** As redes programáveis ativas possuem a característica de serem modificáveis em tempo de execução, permitindo que demandas de novos serviços sejam satisfeitas dinamicamente. Além disso, o conceito de ambiente isolado utilizado nesse tipo de rede permite que usuários realizem modificações no comportamento da rede sem interferir nas redes uns dos outros.
- **Rede Sobreposta:** São redes lógicas construídas usualmente em nível de aplicação. Tal conceito permite que as aplicações da rede construam redes lógicas com diferentes estações. *Peer-to-Peer (Par-a-Par) (P2P)* é um exemplo de redes sobrepostas criadas por aplicações na *Internet*, onde os usuários fazem parte de redes lógicas criadas para troca de informações de forma bilateral (Lua & et al. 2005). Contudo, é importante ressaltar que cada rede é construída com os objetivos específicos de cada aplicação. Além disso, tal abordagem não permite a inserção de inovações tecnológicas nas camadas inferiores da *Internet*, uma vez que tais redes são usualmente criadas em cima das camadas já existentes da *Internet*. Um exemplo disso pode ser observado na forma de endereçamento da *Internet* atual, o IPv4 (*Internet Protocol version 4*). O uso de IPv4 possui algumas restrições, como a quantidade de nós suportados. Uma abordagem para resolver algumas dessas limitações é o uso do protocolo IPv6 (*Internet Protocol version 6*), que, dentre outras coisas, permite o endereçamento de uma quantidade maior de nós. Contudo, o uso de redes sobrepostas não se mostram úteis para a adoção do IPv6.

Diferentemente das abordagens descritas, um ambiente de redes virtualizadas é composto por diferentes arquiteturas, fornecidas por diversos Provedores de Infraestrutura Física (PIPs). Logo, os Provedores de Serviço (SPs) contratam diferentes PIPs para construção das redes virtuais. Uma rede virtual, por sua vez, é um conjunto de nós virtuais conectados por enlaces virtuais, de forma a compor uma topologia virtual. Tais nós virtuais são alocados em nós físicos da infraestrutura física, e cada enlace virtual é alocado em um conjunto de recursos físicos os quais promovem a comunicação entre dois nós virtuais. É importante mencionar que, uma rede virtual deve ser flexível o suficiente para permitir que cada SP utilize seu formato próprio de pacotes, protocolos e mecanismos de roteamento, bem como planos de gerenciamento. Os benefícios com o uso de redes virtuais incluem:

- **Flexibilidade** na topologia, protocolos de rotina (ex: realização de *backups*) e controle, uma vez que redes virtuais são independentes da rede física adotada e das outras redes virtuais alocadas;
- **Gerenciabilidade**, uma vez que as redes virtuais são gerenciadas independentemente e de forma totalmente customizada;
- **Escalabilidade**, já que os provedores de infraestrutura podem facilmente aumentar o número de redes virtuais coexistentes em sua rede física;
- **Isolamento** entre redes virtuais de uma mesma infraestrutura física, garantindo assim segurança, privacidade e melhorando as questões de tolerância a falhas; e
- **Heterogeneidade**, considerando que ambas as redes físicas e virtuais podem ser compostas por diferentes tecnologias heterogêneas (como rede óptica e *wireless*, por exemplo).

Nesse contexto, os seguintes papéis de negócio (Figura 1) são usualmente considerados: PIP, *Virtual Network Providers (Provedores de Redes Virtuais)* (VNP), *Virtual Network Operator (Operadores de Redes Virtuais)* (VNO) e *Service Providers (Provedores de Serviços)* (SP). O PIP é quem possui e gerencia a infraestrutura física, e adicionalmente provê serviços que possibilitam a virtualização de redes. VNP é o responsável por alocar os recursos de um ou mais PIPs em uma topologia virtual. VNOs são responsáveis pela instanciação e operação de cada topologia virtual disponibilizada pelo provedor de redes virtuais. Finalmente, tem-se o provedor de serviços que adota a rede virtual alocada para oferecer serviços específicos.

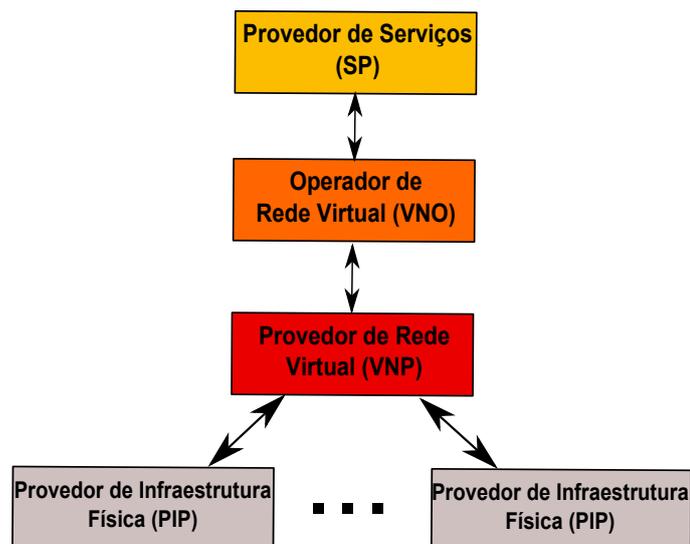


Figura 1 – Principais papéis de negócio em virtualização de redes

2.2 DEPENDABILIDADE

Devido ao crescimento da prestação de serviços pela *Internet*, atributos de dependabilidade têm ganhado grande importância no desenvolvimento, implantação e operação de *hardware* e *software* (MACIEL et al., 2010). Dependabilidade pode ser vista como a capacidade de um sistema em entregar um determinado serviço de forma atestadamente confiável (MACIEL et al., 2010), (LAPRIE et al., 1992), (FIGUERÊDO, 2011). O serviço disponibilizado num sistema é o comportamento do mesmo perceptível pelo seu usuário (pessoa ou máquina) (LAPRIE, 1995). Os conceitos relacionados à dependabilidade podem ser agrupados basicamente em três grupos: (i) os atributos, (ii) os meios para alcançá-los e (iii) as ameaças. A Figura 2 ilustra esta organização.

Neste trabalho, os atributos de dependabilidade de interesse são:

- **Confiabilidade:** probabilidade de um sistema realizar sua função predefinida sem falhas para um período de tempo específico (MACIEL et al., 2010): $R(t) = P\{T \geq t\}$, onde T é a variável aleatória representando o tempo de falha do sistema (ou de um componente único). Além disso, $R(t) = 1 - F(t)$, de tal forma que $F(t) = P\{T < t\}$ é a função de distribuição cumulativa, a qual representa que uma falha ocorreu antes do tempo t .
- **Disponibilidade:** probabilidade do sistema estar em uma condição de funcionamento. Considera a alternância entre o estado operacional e de falha (MACIEL et al., 2010). A disponibilidade instantânea (A) é comumente adotada, e as Equações 2.1 e 2.2 são consideradas.

$$A = \frac{uptime}{uptime + downtime} \quad (2.1)$$

$$A = \frac{MTTF}{MTTF + MTTR} \quad (2.2)$$

MTTF é o tempo médio de falha e MTTR é o tempo médio de reparo, de tal forma que:

$$MTTF = \int_0^{\infty} R(t)dt \quad (2.3)$$

$$MTTR = \int_0^{\infty} (1 - M(t))dt \quad (2.4)$$

$M(t)$ é a função de distribuição cumulativa do tempo de reparo. $R(t)$ é a função de confiabilidade apresentada anteriormente.

As funções lógicas são utilizadas para representar os modos de funcionamento e falha dos sistemas (MACIEL et al., 2010), (KUO; ZUO, 2003), através do uso de

expressões booleanas para definir o estado do sistema através dos estados de seus componentes. Ou seja, de acordo com os estados individuais dos componentes do sistema, utilizam-se expressões *booleanas* para inferir o estado do sistema como um todo. Além disso, se um dado sistema é capaz de prover suas funcionalidades mesmo na presença de falhas, diz-se que esse sistema é tolerante à falhas.

Existem basicamente quatro técnicas para obtenção dos atributos de dependabilidade em um sistema. O primeiro é a prevenção de falhas, que se refere a prevenir a ocorrência ou a introdução de falhas no sistema. Outro meio é a tolerância a falhas (causa de um erro), que se baseia em que o sistema mantenha algum nível de funcionamento mesmo na presença de falhas. Já a remoção de falhas, é referente a como reduzir a presença e severidade das mesmas. Por fim, tem-se o meio da previsão de falhas, que é a realização de estimativas (número e consequência) de falhas futuras (LAPRIE, 1995).

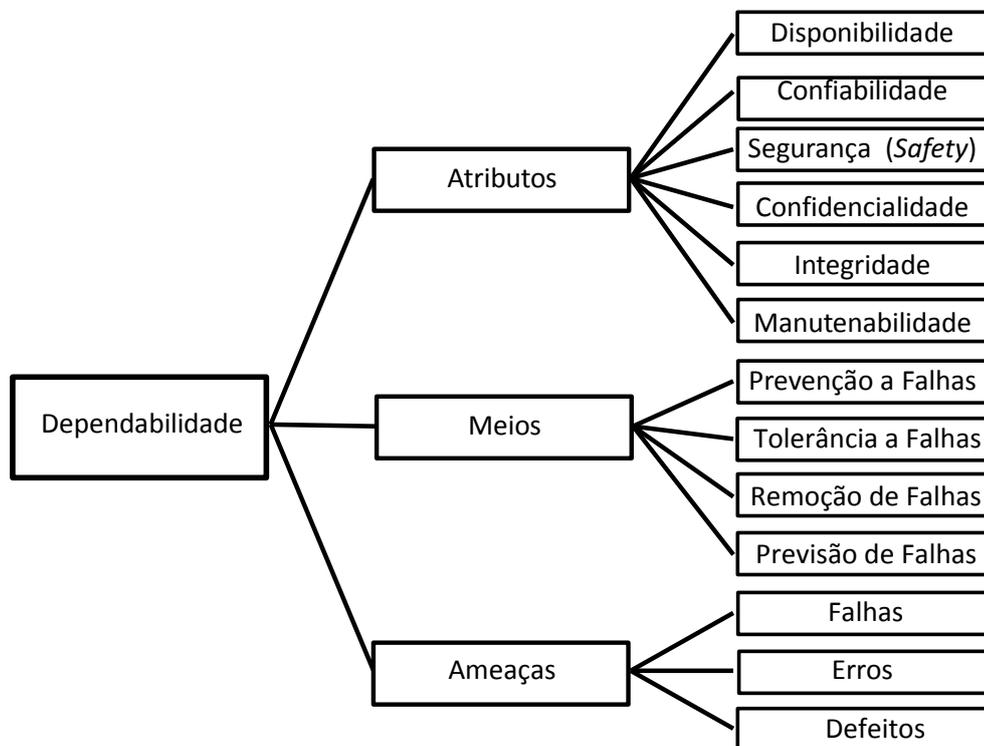


Figura 2 – Árvore de Dependabilidade. Adaptação de (LAPRIE, 1995)

2.2.1 Técnicas de Tolerância a Falhas

Redundância é uma técnica bastante importante para a implementação de sistemas tolerantes a falhas, com objetivo de melhorar a disponibilidade e confiabilidade (MACIEL et al., 2010). Em geral, essa técnica consiste na adição de componentes extras no sistema, de tal forma que se um componente falhar, o componente redundante assume o seu lugar. Redundância dinâmica é uma técnica representativa que considera componentes redun-

dantes para substituir os componentes principais sempre que estes estão em estados de falha. *Hot* e *cold standby* são técnicas representativas para redundância.

A Figura 3 apresenta o esquema de redundância *cold standby*. Nessa técnica, o componente de *backup* inicialmente está inativo, e permanece nesse estado até que ocorra o evento de falha do componente principal. Então, o componente redundante é ativado e substitui o componente que falhou. É importante mencionar que, como o componente redundante inicialmente está desligado, existe um atraso entre o momento em que o componente principal falha e o componente redundante é ativado.

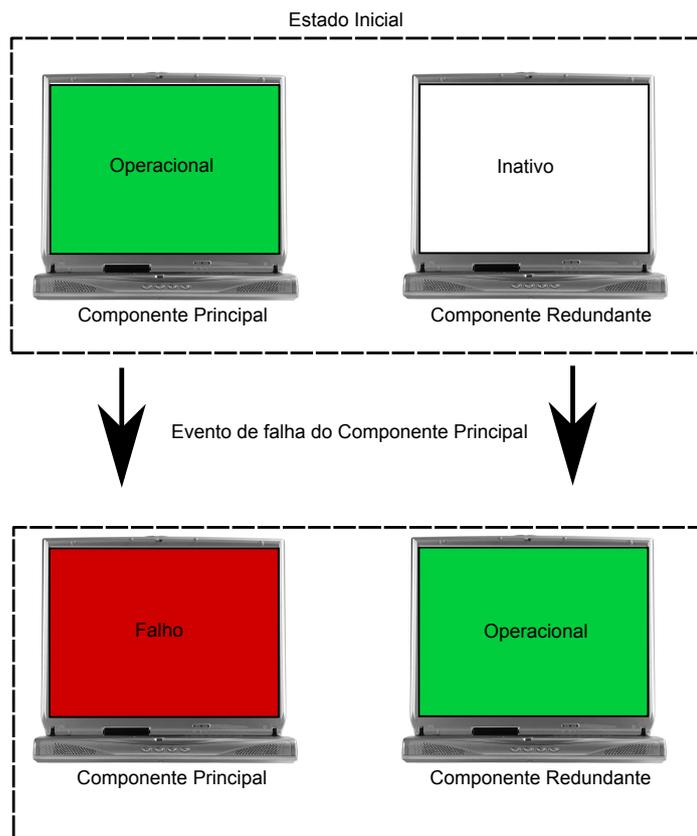


Figura 3 – Redundância *Cold Standby*

Por outro lado, em *hot standby* o componente redundante já é ativado inicialmente, e funciona simultaneamente com o primeiro componente. Logo, sempre que este falha, o redundante o substitui imediatamente, sem o atraso que afeta a técnica de *cold standby*. A Figura 3 apresenta uma visão geral da técnica de *hot standby*.

2.2.2 Modelos de Dependabilidade

Modelos para estimar métricas de dependabilidade são geralmente classificados em combinatoriais e baseados em estados (MACIEL et al., 2010). Modelos combinatoriais consideram as condições que mantêm o sistema operacional ou falho, através da relação estrutural entre seus componentes. Contudo, esses modelos possuem limitações para representação

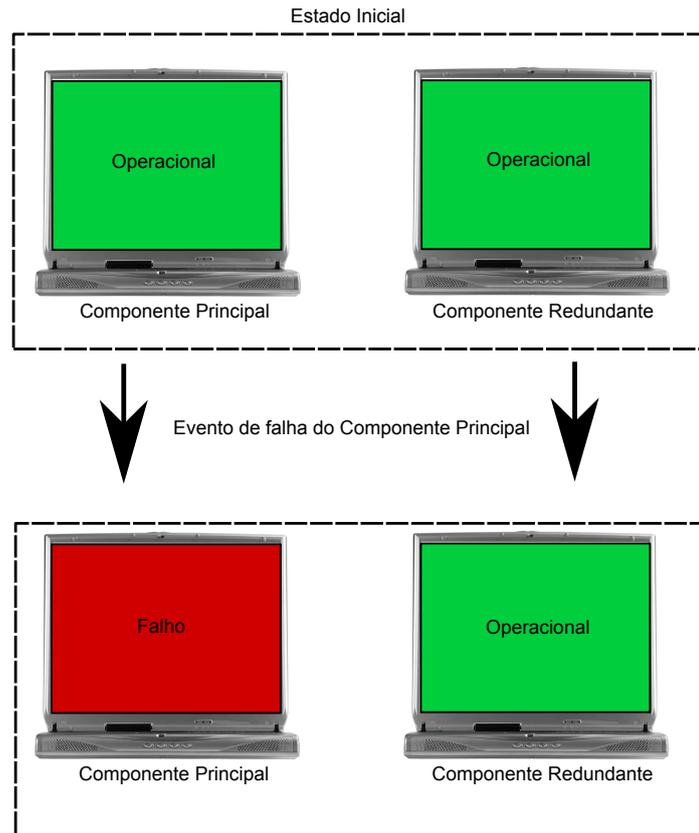


Figura 4 – Redundância Hot Standby

de interações complexas entre os componentes do sistema, bem como políticas de manutenção sofisticadas.

Por outro lado, os modelos baseados em estados representam o comportamento dinâmico dos sistemas através de seus estados e das ocorrências de eventos (MACIEL et al., 2010). Esses modelos são mais adequados para a modelagem de interações complexas entre componentes, assim como mecanismos baseados em redundância dinâmica. Contudo, modelos baseados em estados sofrem do problema de explosão de espaço de estados.

Reliability Block Diagrams (Diagramas de Bloco de Confiabilidade) (RBD), *Árvore de Falhas* e *Grafos de Confiabilidade* são modelos combinatoriais representativos. Por outro lado, *Cadeias de Markov*, bem como *Stochastic Petri Nets (Redes de Petri Estocásticas)* (SPN) são modelos proeminentes, baseados em estados. Devido ao seu poder de modelagem, bem como do acesso à ferramenta ASTRO/Mercury (SILVA et. al, 2012) para avaliação de tais modelos, os formalismos SPN e RBD são adotados neste trabalho para modelagem e avaliação de dependabilidade em redes virtuais, e são apresentados detalhadamente nas próximas seções.

2.3 SISTEMAS COERENTES

As funções de estrutura de um sistema definem o arranjo em que os componentes do sistema estão dispostos. Tais arranjos podem ser definidos por conjuntos de caminhos e cortes. Por exemplo, considere um sistema formado por diversos componentes. Caso seja necessário o funcionamento de todos os componentes para que o sistema esteja operacional, pode-se afirmar que tais componentes definem o fluxo que permite o funcionamento do sistema. Ou seja, há apenas uma forma do sistema funcionar, que é com todos os componentes funcionando.

Contudo, em diversos sistemas, observa-se que o funcionamento do sistema é dado pela operação de diferentes combinações de componentes. Nesses casos, pode-se dizer que existem diferentes fluxos que permitem o funcionamento do sistema. Esses fluxos representam os caminhos. Analogamente, são chamadas de cortes, as combinações de componentes que, quando falham, levam o sistema a interromper sua operação. Logo, é dito um caminho, um conjunto de componentes que, quando funcionam simultaneamente, garantem o funcionamento do sistema. Por outro lado, um corte é um conjunto de componentes que, quando falham simultaneamente, levam o sistema ao estado de falha. É importante mencionar que um caminho mínimo é o conjunto mínimo de componentes em que o seu funcionamento simultâneo garante o funcionamento do sistema.

Pode-se definir um sistema coerente como um sistema com função estrutural $\phi(x)$ se, e somente se, para cada x_i em $i(1 \leq i \leq n)$, $\phi(x)$ é não-decrescente, e todos os componentes são relevantes.

Um dado componente é dito relevante caso exista, no mínimo, um vetor de estados x em que o estado do sistema seja definido pelo estado do componente i . Ou seja, quando os outros componentes do sistema encontram-se em um estado, dado por $(x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$, o estado de $\phi(x_1, x_2, \dots, x_n)$ é igual ao estado do componente (x_i) . Logo, quando o componente i está operacional, o sistema funciona, caso contrário, o sistema falha (KUO; ZUO, 2003).

Além disso, é importante mencionar que melhorar o desempenho de um dado componente geralmente não piora o desempenho do sistema como um todo (KUO; ZUO, 2003). Usualmente, assume-se uma função não-decrescente do estado de todos os componentes para representar a função de estrutura de todo o sistema. Para um sistema coerente, considere dois vetores, x e y , com n elementos cada: diz-se que $x < y$ se, para cada $i(1 \leq i \leq n)$, $x_i < y_i$. Ou seja, assume-se que o vetor x é menor que o vetor y (KUO; ZUO, 2003). Então, um sistema coerente deve satisfazer as seguintes condições (KUO; ZUO, 2003):

1. $\phi(0) = 0$: O sistema é falho quando todos os componentes são falhos.
2. $\phi(1) = 1$: O sistema funciona quando todos os componentes funcionam.

3. Caso $x < y$, então $\phi(x) < \phi(y)$: Melhorar o desempenho de qualquer componente não piora o desempenho do sistema.
4. Existe um vetor de estados de componentes no qual o estado do sistema é dado pelo estado do componente i , para todos os componentes do sistema.

2.3.1 Funções Lógicas

O objetivo das funções lógicas é indicar o relacionamento do estado do sistema com os estados dos componentes que o compõem. Para funções lógicas, utiliza-se técnicas de minimização booleana para simplificar ou minimizar as funções obtidas.

Componentes em série

Considere o vetor $x = (x_1, x_2, \dots, x_n)$, que representa os estados dos n componentes que compõem um dado sistema. Considerando que os n componentes estão em série, a função lógica (*Sserial*) do sistema, que representa o funcionamento do mesmo, é dada por:

$$Sserial(x) = (x_1 \wedge x_2 \wedge \dots \wedge x_n) \quad (2.5)$$

Componentes em paralelo

Considere o vetor $x = (x_1, x_2, \dots, x_n)$, que representa os estados dos n componentes que compõem um dado sistema. Considerando que os n componentes estão em paralelo, a função lógica (*Sparalelo*) do sistema, que representa o funcionamento do mesmo, é dada por:

$$Sparalelo(x) = (x_1 \vee x_2 \vee \dots \vee x_n) \quad (2.6)$$

Uma discussão detalhada sobre funções lógicas pode ser encontrada em (KUROKI; ZUO, 2003).

2.4 RBD

Os modelos de diagrama de bloco de confiabilidade (RBD) consistem de componentes e suas relações lógicas. Os componentes são graficamente representados por retângulos, enquanto as relações lógicas são representadas por arcos. O estado de operação do sistema é dado pelos seus componentes em funcionamento (MACIEL et al., 2010).

O uso de RBD é bastante adotado em sistemas com múltiplos módulos independentes, onde cada módulo pode ser representado por um bloco RBD. Diversas métricas de dependabilidade podem ser obtidas através do uso de RBD, como manutenibilidade, disponibilidade e confiabilidade (métrica para a qual esse formalismo foi inicialmente proposto).

Em um modelo RBD se, e somente se, houver um caminho ligando os blocos de início e fim, diz-se que o sistema está em estado operacional. É importante enfatizar que, conforme citado anteriormente, RBD é um modelo combinatorial, logo possui limitações para representação de mecanismos redundantes dinâmicos, bem como políticas de manutenção preventivas.

Diversos tipos de arranjo podem ser utilizados para modelagem com RBD.

2.4.1 Arranjo em série

No arranjo em série, o sistema está em estado operacional apenas se todos os componentes que compõem o arranjo estão em estado operacional, ou seja, caso não haja falhas em nenhum dos componente do bloco (Figura 5).

Assumindo um modelo RBD com n componentes, a confiabilidade ou disponibilidade (P_s) pode ser estimada da seguinte forma:

$$P_s = \prod_{i=1}^n p_i \quad (2.7)$$

onde p_i denota a disponibilidade ou confiabilidade do componente i .

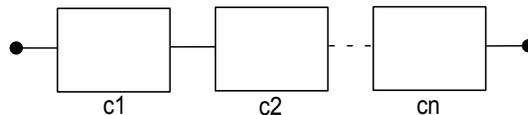


Figura 5 – Arranjo RBD em série

2.4.2 Arranjo em Paralelo

Nesse tipo de arranjo, para que o sistema esteja no estado operacional basta que um dos seus componentes esteja funcionando (Figura 6).

Em um sistema RBD em paralelo com n componentes, a confiabilidade ou disponibilidade (P_s) do sistema é calculada como segue:

$$P_s = 1 - \prod_{i=1}^n [1 - p_i] \quad (2.8)$$

Onde p_i denota a disponibilidade ou confiabilidade do componente i . Arranjos em paralelo são usualmente considerados para representação de sistemas com redundância *hot standby*.

2.4.3 Soma de produtos disjuntos

Sum of Disjoint Products (Soma dos Produtos Disjuntos) (SPD) é um método que utiliza os caminhos mínimos (ou cortes mínimos) para verificar se o sistema está em estado de funcionamento ou falha, analisando à probabilidade de união dos cortes (ou caminhos).

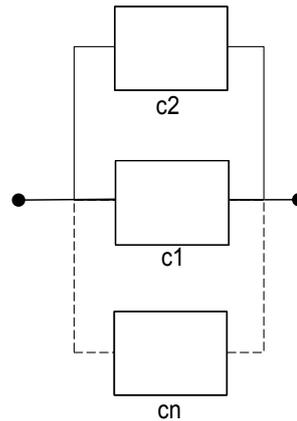


Figura 6 – Arranjo RBD em paralelo

A função lógica do sistema pode ser utilizada para verificar a probabilidade de união dos cortes mínimos. Além disso, esse método é particularmente útil para a avaliação de sistemas RBD que possuem componentes repetidos.

Uma vez que a função lógica pode ser expressa através da união de vários termos, caso estes (os termos) sejam disjuntos, significa que existe uma relação de um-para-um entre a métrica de confiabilidade do sistema e sua função lógica. O método de soma dos produtos disjuntos visa expressar a função lógica do sistema através da união dos termos disjuntos, onde os termos disjuntos são produto dos eventos de falha e reparo dos componentes individuais (FIGUERÊDO, 2011), (KUO; ZUO, 2003), (MACIEL et al., 2010).

O método de soma dos produtos disjuntos tem sua justificativa na lei adicional de probabilidade. Segundo essa lei, se a intersecção de dois ou mais eventos é vazia, a soma das probabilidades dos eventos individuais representa a probabilidade de que pelo menos um destes eventos irá ocorrer (MACIEL et al., 2010).

Se a intersecção de dois eventos X e Y não é vazia, a união destes dois elementos $X \cup Y$ pode ser expressa como a união do evento X com o evento $\bar{X}Y$, onde \bar{X} é o complemento de X . Logo, as probabilidades de $X \cup Y$ podem ser avaliadas através da seguinte equação:

$$Pr(X \cup Y) = Pr(X) + Pr(\bar{X}Y) \quad (2.9)$$

De forma análoga, para 3 eventos X, Y e Z , utiliza-se a seguinte equação:

$$Pr(X \cup Y \cup Z) = Pr(X) + Pr(\bar{X}Y) + Pr(\bar{X}\bar{Y}Z) \quad (2.10)$$

Pode-se generalizar a equação para n eventos X_1, X_2, \dots, X_n , da seguinte forma:

$$Pr(X_1 \cup \dots \cup X_n) = Pr(X_1) + Pr(\bar{X}_1 X_2) + \dots + Pr(\bar{X}_1 \dots \bar{X}_{n-1} X_n) \quad (2.11)$$

A probabilidade de união dos n eventos é expressa na Equação 2.11 como a soma de n termos prováveis. Os termos adicionais expressam a contribuição realizada por um evento adicional à probabilidade de união. Logo, o termo $Pr(X_1)$ representa a contribuição para o

primeiro evento (X_1), e $Pr(\overline{X_1}X_2)$ representa a contribuição adicional do segundo evento (X_2) não considerada pelo primeiro evento em relação à probabilidade de união. Contudo, a tarefa escolhida é como analisar a contribuição de eventos adicionais não considerados por nenhum evento anterior.

Pode-se dividir a lei de adição em duas partes, (i) o laço interno e (ii) o externo.

O laço externo baseia-se na Equação 2.11. Considere a existência de n caminhos mínimos no sistema, representados por CM_1, CM_2, \dots, CM_n . Assumindo que S_k representa a contribuição do k -ésimo caminho mínimo na união de todos os caminhos mínimos entre 1 e k , têm-se as seguintes equações para avaliar S_k .

A probabilidade do primeiro caminho funcionar será equivalente a sua expressão de contribuição:

$$S_1 = CM_1 \quad (2.12)$$

A probabilidade do segundo caminho funcionar, dado que o primeiro falhou, é equivalente a sua expressão de contribuição:

$$S_2 = \overline{CM_1}CM_2 \quad (2.13)$$

Generalizando, a Equação 2.14 é dada para obter a expressão de contribuição do k -ésimo termo.

$$S_k = \overline{CM_1}CM_2 \dots \overline{CM_{k-1}}CM_k \quad (2.14)$$

Logo, o valor da contribuição do k -ésimo caminho é obtido por:

$$U_k = Pr(S_k), 1 \leq k \leq l \quad (2.15)$$

U_k é a probabilidade de que apenas o k -ésimo caminho do sistema funciona. Ou seja, o caminho k funciona, mas os primeiros $k - 1$ caminhos estejam falhos. Para calcular a confiabilidade do sistema, têm-se a equação 2.16.

$$R_s = U_1 + U_2 + \dots + U_l \quad (2.16)$$

Podem-se encontrar diversos limites inferiores na confiabilidade do sistema ao longo das iterações do laço externo. Uma vez que tenham sido incluídos todos os caminhos mínimos, a confiabilidade exata do sistema é obtida (KUO; ZUO, 2003).

O laço interno, por sua vez, utiliza em cada iteração um ou mais caminhos críticos. Quando o k -ésimo caminho é considerado, o laço interno busca a expressão do evento que inclui exclusivamente o k -ésimo caminho.

2.5 REDES DE PETRI

Petri Nets (Redes de Petri) (PN) é uma técnica formal de modelagem que permite a representação de sistemas não determinísticos, paralelos, concorrentes e assíncronos (MACIEL; LINS; CUNHA, 1996). A aplicabilidade da modelagem de Redes de Petri pode ser observada em diversas áreas, como sistemas operacionais, controle de tráfego, administração de empresas, entre outras.

A representação de uma rede de Petri dar-se-á através de um grafo bipartido o qual é composto por um conjunto finito e não vazio de *lugares* P (os quais representam as variáveis de estado), um conjunto finito e não vazio de *transições* T (os quais representam as ações), e um conjunto de *arcos direcionados* A . Além disso, uma rede de Petri é dita *marcada* se, e somente se, ela possuir um conjunto não vazio de *tokens* (ou marcas) (MACIEL; LINS; CUNHA, 1996).

A Figura 7 apresenta os elementos de uma rede de Petri representados graficamente, onde os lugares (Figura 7 (a)) são representados através de círculos, as transições (Figura 7 (b)) por retângulos, os arcos direcionados (Figura 7 (c)) por setas (os quais conectam lugares e transições) e os *tokens* (Figura 7 (d)) por pontos pretos.

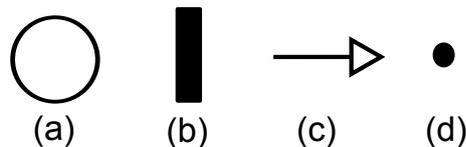


Figura 7 – Elementos de uma rede de Petri

Diz-se que um lugar é *entrada* para uma transição se, e somente se, este lugar possuir um arco direcionado partindo de si em direção à transição. Além disso, diz-se que um lugar é *saída* para uma transição no caso de existir um arco direcionado da transição para o lugar em questão (MACIEL; LINS; CUNHA, 1996). Os *tokens*, por sua vez, encontram-se nos lugares, podendo variar em quantidade e posições durante o funcionamento da rede de Petri (CHANIN et al., 2005).

A semântica de uma rede de Petri está no disparo de transições. Na ocorrência de um disparo, são consumidos *tokens* dos lugares de *entrada* da transição que disparou, e depositados *tokens* nos lugares de *saída* da mesma transição (CHANIN et al., 2005).

Entretanto, para que ocorra o disparo de uma transição, a mesma precisa estar *habilitada*. Uma transição é dita *habilitada* para disparo se, e somente se, todos os seus lugares de entrada possuírem *tokens* (MACIEL; LINS; CUNHA, 1996). Pode haver várias transições disparáveis no modelo, e a seleção de qual transição irá disparar é não determinística. Além disso, o disparo de uma transição pode levar a um novo estado de marcação no modelo, desabilitando e habilitando as transições para disparo (CHANIN et al., 2005).

A Figura 8 apresenta o processo de funcionamento de uma lâmpada modelado utilizando o formalismo das redes de Petri. Os estados possíveis de uma lâmpada são acesa

ou apagada. Além disso, observam-se duas ações possíveis neste modelo, quais sejam a de acender a lâmpada e apagá-la. Logo, para modelar esse sistema são necessários dois lugares, representando os possíveis estados, e duas transições, representando as ações. A marcação inicial deste modelo é com um *token* no lugar Apagada (ver Figura 8 (a)). A partir da marcação inicial, a única ação possível é o disparo da transição Acender. Uma vez realizado esse disparo, é removido um *token* do lugar Apagada e adicionado um *token* no lugar Acesa (ver Figura 8 (b)). Uma vez no estado Acesa, o único evento possível é o disparo da transição Apagar, levando o modelo novamente ao estado inicial.

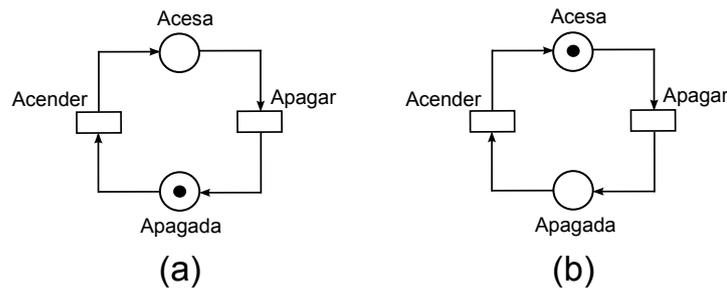


Figura 8 – Estados de uma lâmpada

As redes de Petri possuem dois conjuntos de propriedades, as comportamentais e as estruturais. As comportamentais dependem apenas da marcação inicial do modelo, enquanto as estruturais dependem apenas da estrutura da rede.

Uma apresentação formal do formalismo de redes de Petri é apresentado a seguir (MARSAN et al., 1998), (MACIEL; LINS; CUNHA, 1996):

(Redes de Petri) Uma rede de Petri é uma 5-tupla $PN = (P, T, F, M, M_0)$, na qual:

- P é o conjunto finito de lugares;
- T é o conjunto finito de transições, tal que $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$;
- $F \subseteq (P \times T) \cup (T \times P)$ é o conjunto finito de arcos;
- $M : F \rightarrow \mathbb{N}$ é a função que atribui peso aos arcos;
- $M_0 : P \rightarrow \mathbb{N}$ é a marcação inicial.

O espaço de estados de uma rede de Petri é o seu grafo de alcançabilidade. Trata-se de um grafo rotulado e direcionado utilizado para a representação das marcações que a rede de Petri pode alcançar. Formalmente, um grafo de alcançabilidade pode ser definido como uma tupla (V, E) , onde V representa o conjunto dos vértices representados pelas marcações possíveis, e E é o conjunto de arestas rotuladas. Por exemplo, considere o conjunto $M = \{m_0 = \{1, 0\}, m_1 = \{0, 1\}\}$ de marcações alcançáveis da rede de Petri representada na Figura 8. O respectivo grafo de alcançabilidade da rede de Petri é representado na Figura 9.

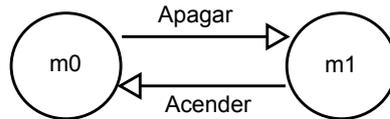


Figura 9 – Exemplo de grafo de alcançabilidade

2.5.1 Propriedades das Redes de Petri

As propriedades das redes de Petri podem ser divididas em duas categorias: comportamentais e estruturais. Essas propriedades permitem a observação de diversas características do sistema modelado. As propriedades das redes de Petri independentes de marcação são denominadas propriedades estruturais, uma vez que dependem exclusivamente da topologia (estrutura) da rede. Segue a definição das principais propriedades estruturais das redes de Petri (MURATA, 1989).

- **Limitação Estrutural:** Caso o número de *tokens* em uma rede de Petri seja limitado para qualquer marcação inicial, diz-se que essa rede é limitada.
- **Conservação:** Essa propriedade permite verificar se não há destruição de recursos (*tokens*) em uma rede.
- **Repetitividade:** Uma rede possui essa propriedade se, para alguma marcação do modelo e uma sequência de transições disparáveis, um conjunto de transições da rede possam ser disparadas de forma ilimitada.
- **Consistência:** Para que uma rede seja considerada consistente, é necessário que, dada uma marcação inicial M_0 e uma sequência de disparo de transições, seja possível retornar à marcação M_0 . Além disso, é necessário que todas as transições da rede sejam disparadas nessa sequência.

Em relação às propriedades comportamentais, são aquelas que dependem da marcação do modelo. As principais propriedades comportamentais das redes de Petri (MURATA, 1989) são:

- **Limitação:** Se para qualquer marcação alcançável a partir da marcação inicial, o número de *tokens* em cada lugar de uma rede não ultrapassar um número finito k .
- **Vivacidade:** A propriedade de *liveness* (ou vivacidade) está fortemente associada ao conceito de *deadlock*. Caso um sistema seja *liveness*, significa que o mesmo está livre de *deadlocks*. Contudo, o fato de um sistema ser livre de *deadlock* não leva necessariamente à conclusão de que o sistema é *liveness*. Para que uma rede seja *live*, é necessário que, para quaisquer marcações alcançáveis a partir de uma marcação inicial m_0 , todas as transições da rede possam ser disparadas após o disparo de

alguma sequência de transições. Tal propriedade permite analisar se foram incluídos eventos no modelo que nunca serão disparados.

- **Reversibilidade e Home State:** Uma rede é dita reversível se, e somente se, a marcação inicial é alcançável a partir de quaisquer marcações do modelo. Considera-se ainda que, caso uma marcação diferente da marcação inicial seja sempre alcançável, essa marcação é considerada *Home State*.

2.5.2 Redes de Petri Estocásticas

Apesar do conceito de redes de Petri prover um mecanismo poderoso para modelagem, tal conceito permite apenas a representação lógica dos sistemas, uma vez que não possui meios para a representação de comportamentos temporais, frequentemente necessários para representar o comportamento dinâmico de sistemas (CHANIN, et al., 2005). O formalismo SPN (*Stochastic Petri Nets*) ou Redes de Petri Estocásticas (MARSAN et al., 1998, BALBO, 2001) é uma extensão ao conceito inicial das redes de Petri, adotada para esse propósito.

SPN é uma ferramenta para modelagem e avaliação de desempenho e dependabilidade de sistemas, na qual se assume que todas as transições *temporizadas* têm seu atraso de disparo associado à uma variável aleatória distribuída exponencialmente. Além disso, as transições ditas *imediatas* possuem tempo de disparo associado igual a zero. Os formalismos de redes de Petri estocásticas e cadeias de Markov de tempo contínuo (CTMC) (TRIVEDI, 2002) possuem o mesmo escopo, o que significa que o espaço de estados de qualquer SPN pode ser convertido para uma CTMC, através do grafo de alcançabilidade da SPN. Além disso, as SPNs permitem a adoção de técnicas de simulação como uma alternativa à geração de cadeias de Markov.

Em SPN, usualmente transições imediatas são representadas por um retângulo fino e preto, enquanto transições temporizadas são representadas por um retângulo mais esparsos e branco. A Figura 10 apresenta um modelo SPN e a respectiva CTMC. Segue a definição formal do conceito de SPN, baseada em BALBO (2001) e ARAÚJO et al. (2009).

(Redes de Petri Estocásticas) Uma rede de Petri estocástica (SPN) é uma 9-tupla $SPN = (P, T, I, O, H, \Pi, G, M_0, Atts)$, na qual:

- $P = \{p_1, p_2, \dots, p_n\}$ é o conjunto de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é o conjunto de transições imediatas e temporizadas $P \cap T = \emptyset$;
- $I \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n \times m}$ é a matriz que representa os arcos de entrada (que podem ser dependente de marcações);
- $O \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n \times m}$ é a matriz que representa os arcos de saída (que podem ser dependente de marcações);

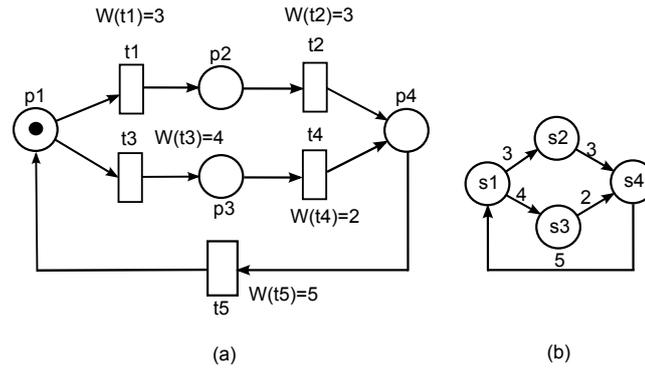


Figura 10 – Modelo SPN e a respectiva CTMC

- $H \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n \times m}$ é a matriz que representa os arcos inibidores (que podem ser dependente de marcações);
- $\Pi \in \mathbb{N}^n$ é um vetor que associa o nível de prioridade a cada transição;
- $G \in (\mathbb{N}^n \rightarrow \{true, false\})^m$ é o vetor que associa uma condição de guarda relacionada à marcação do lugar a cada transição;
- $M_0 \in \mathbb{N}^n$ é o vetor que associa uma marcação inicial de cada lugar (estado inicial);
- $Atts = (Dist, Markdep, Concurrency, W)^m$ compreende o conjunto de atributos associados às transições, onde:
 - $Dist \in \mathbb{N}^n \rightarrow F$ é uma possível função de distribuição de probabilidade associada ao tempo de uma transição (esta distribuição pode ser dependente de marcação) (o domínio de F é $[0, \infty]$);
 - $Markdep \in \{constante, enabdep\}$ onde a distribuição de probabilidade associada ao tempo de uma transição pode ser independente (*constante*) ou dependente de marcação (*enabdep* - a distribuição depende da condição de habilitação atual);
 - $Concurrency \in \{ss, is\}$ é o grau de concorrência das transições, onde *ss* representa a semântica *single server* e *is* representa a semântica *infinity server*;
 - $W : T \rightarrow IR^+ \cup \{0\}$ é a função de peso, que representa o peso (w_t) de transições imediatas e a taxa λ_t de transições temporizadas, onde:

$$\pi(t) = \begin{cases} \geq 1, & \text{se } t \text{ é uma transição imediata} \\ 0, & \text{caso contrário.} \end{cases}$$

Se t é uma transição temporizada, então λ_t será o valor do parâmetro da função de densidade de probabilidade exponencial.

Se t é uma transição imediata, então W_t será um peso, que é usado para o cálculo das probabilidades de disparo das transições imediatas em conflitos;

Os arcos inibidores são usados para prevenir as transições de serem habilitadas quando certas condição é verdadeira.

É importante mencionar que, $\#p$ denota o número de *tokens* no lugar p ; e $P\{exp\}$ estima a probabilidade da expressão exp ser verdadeira. Como modelos SPN são avaliadas usando a geração da cadeia de Markov de tempo contínuo (CTMC) equivalente, neste trabalho, $P\{exp\}$ é estimado pela soma das probabilidades de estado de cada CTMC que satisfaz a expressão exp .

2.6 GRASP

Greedy Randomized Adaptive Search Procedure (Procedimento de Busca Adaptativa Aleatória Gulosa) (GRASP) é uma metaheurística baseada em um algoritmo aleatório iterativo (OSMAN, 1996), e é usualmente utilizada para resolução de problemas da área de otimização combinatória. A Figura 11 apresenta o pseudo-algoritmo GRASP. É composto comumente por iterações que possuem duas fases: construção e busca local. A fase de construção é responsável pela construção de uma solução aleatória, que atenda a todas as restrições. A busca local investiga a vizinhança da solução obtida na fase de construção, substituindo a solução atual por uma melhor, sucessivamente, até que a solução local ótima seja atingida.

```

procedure GRASP(Max_Iteracoes, Param)
1   for  $k=1, \dots, \text{Max\_Iteracoes}$  do
2       Solucao  $\leftarrow$  Construir_Solucao_Randomica(Param);
3       Solucao  $\leftarrow$  Busca_Local(Solucao);
4       Atualizar_Solucao(Solucao, Melhor_Solucao);
5   end;
6   return Melhor_Solucao;
end procedure

```

Figura 11 – Pseudo-código GRASP

A Figura 12 apresenta o pseudo-algoritmo da fase de construção. A cada iteração, encontra-se uma possível solução para o problema (que atenda a todas as restrições estabelecidas), adotando uma abordagem gulosa (guiando-se pelos elementos mais valiosos no momento). Na fase de construção, é criada uma *Restricted Candidate List (Lista de Candidatos Restrita)* (RCL), onde o critério para construção dessa lista baseia-se exatamente na função gulosa pré-definida. Ao longo das iterações, escolhe-se aleatoriamente um elemento da RCL e, caso este seja melhor que os elementos selecionados em iterações passadas, armazena-se o mesmo. Após a execução de todas as iterações, a melhor solução da RCL é retornada (RESENDE; RIBEIRO, 2004).

```

procedure Construir_Solucao_Randomica(Param)
1   Solucao  $\leftarrow \emptyset$ ;
2   Calcular custo incremental dos elementos candidatos;
3   while Solucao não está completa do
4       Construir RCL;
5       Selecionar aleatoriamente um elemento de RCL;
6       Solucao  $\leftarrow$  Solucao  $\cup$  ( $s$ );
7       Recalcular Custo;
8   end for
9   return Solucao;
end procedure

```

Figura 12 – Pseudo-código da fase de construção - GRASP

Na fase de busca local (Figura 13), responsável por otimizar a solução encontrada na fase de construção, existem diversas estratégias possíveis.

```

procedure Busca_Local (Solucao)
1   while Solucao não é a melhor solução local do
2       Encontre  $s' \in N(Solucao)$  com  $f(s') < f(Solucao)$ 
3       Seleccione aleatoriamente um elemento da RCL;
4       Solucao  $\leftarrow s'$ ;
5   end while
6   return Solucao;
end procedure

```

Figura 13 – Pseudo-código da fase de busca local - GRASP

A metaheurística GRASP permite de encontrar resultados próximos ao ótimo a um baixo custo de processamento (RANGEL et al., 2000), e o seu uso tem sido realizado para a solução de diversos tipos de problemas, como de localização (KLINCEWICZ, 1992), partição em grafos (LAGUNA et al., 1993), problema de cobertura (FEO; RESENDE, 1995) e planejamento e escalonamento de produção (LAGUNA; GONZÁLEZ-VELARDE, 1991).

2.7 CONSIDERAÇÕES FINAIS

Este capítulo introduziu os principais conceitos necessários para melhor entendimento deste trabalho. Inicialmente, foi abordado o conceito de virtualização de redes. Então, a definição de dependabilidade, técnicas de tolerância a falhas e modelos de dependabilidade foram apresentados. Posteriormente, os formalismos RBD e SPN foram introduzidos. Por fim, a metaheurística GRASP foi apresentada.

3 TRABALHOS RELACIONADOS

Este capítulo descreve os principais trabalhos relacionados a esta tese. Inicialmente, são apresentados trabalhos representativos na área de modelagem e avaliação de dependabilidade de sistemas. Posteriormente, abordam-se trabalhos com foco em redes virtualizadas, bem como técnicas de alocação eficiente dos recursos da infraestrutura física.

3.1 MODELAGEM DE DEPENDABILIDADE

Modelagem de dependabilidade é uma tarefa importante, e diversos estudos têm sido propostos objetivando a avaliação de atributos de dependabilidade em sistemas computacionais.

HU et al. (2010) adota modelos de redes de Petri estocásticas para avaliar essas métricas em máquinas virtuais e sistemas operacionais. Esse trabalho adota o modelo de filas M/M/n/k e técnicas de rejuvenescimento de sistemas de computadores virtuais. São analisadas a estrutura das redes, bem como os atributos de dependabilidade do sistema, considerando a possibilidade de recuperação de falhas no *Virtual Machine Monitor (Gerenciador de Máquinas Virtuais)* (VMM). Os resultados obtidos demonstram que o uso de reparos no VMM aumenta significativamente os valores de dependabilidade do sistema. Contudo, esse trabalho não considera dependabilidade em ambientes de virtualização de redes, com nós e enlaces suscetíveis à falhas.

SUN et al. (2010) apresentam modelos de dependabilidade para computação em nuvem, chamado *Cloud Dependability*, objetivando aumentar a segurança de ambientes de computação em nuvem heterogêneos, utilizando virtualização a nível de sistema (CDSV). Apesar desse trabalho considerar diversas métricas quantitativas para avaliação de dependabilidade, o mesmo foca apenas em segurança de computação em nuvem, e avalia dependabilidade apenas considerando os componentes de virtualização a nível de sistema, por exemplo, *Virtual Machine (Máquina Virtual)* (VM) e *hypervisor*.

Similarmente, SARIPALLI; WALTERS (2010) propõem uma metodologia para avaliação de riscos em ambientes de computação em nuvem, focando em riscos de segurança. Basicamente, é apresentado um *framework* (chamado QUIRC) de avaliação quantitativa de riscos, baseado na NIST-FIPS-199 (NIST, 2014), para avaliar os riscos de segurança (por exemplo, a confidencialidade, integridade e disponibilidade) em uma plataforma de computação em nuvem. Além disso, o trabalho descrito propõe uma definição quantitativa de riscos, baseada na probabilidade de comprometimento da segurança do ambiente de computação em nuvem.

WEI; LIN; KONG (2011) propõem a análise da confiabilidade e disponibilidade em centros de dados em nuvens. Nesse trabalho, utiliza-se de uma abordagem de modelagem

hierárquica e híbrida, através dos formalismos Redes de Petri Estocásticas (SPN) (NATKIN, 1990) e Diagramas de Bloco de Confiabilidade (RBD) (RAUSAND; HOYLAND, 2004) para verificar a relação entre a carga de trabalho e a disponibilidade em centros de dados virtuais em nuvem.

Cadeias de Markov de tempo contínuo (*Continuous Time Markov Chains - CTMC*) são utilizadas em HONG et al. (2012) para a avaliação de disponibilidade em sistemas em *cluster* de múltiplos nós, verificando o impacto de considerar ou não *Common Mode Failure (Modo de Falha Comum)* (CMF) nos nós. Nesse trabalho, foram obtidos e comparados resultados numéricos a partir de modelos de cadeias de Markov de tempo contínuo (CTMC). Primeiramente, foi considerado um sistema com um único nó. Posteriormente, os resultados obtidos foram contrapostos com os de sistemas com dois e três nós.

TRIVEDI et al. (2006) propõem modelos de diagramas de bloco de confiabilidade e cadeias de Markov para avaliar disponibilidade em sistemas de transmissão de dados e voz. Além disso, é realizada uma análise de sensibilidade, verificando os impactos dos parâmetros críticos da plataforma.

KIM et al. (2009) apresentam modelos de dependabilidade para sistemas virtualizados e não virtualizados. Em linhas gerais, este trabalho analisa a disponibilidade estacionária, tempo de inatividade anual e disponibilidade orientada à capacidade de cada sistema analisado. Além disso, é adotada uma abordagem hierárquica e analítica para modelagem dos sistemas, considerando cadeias de Markov de tempo contínuo (CTMC) para representação dos submodelos em baixo nível. Para os modelos de alto nível, utilizam-se modelos em árvore de falha, e são incorporadas falhas de *hardware* (CPU, memória, etc) e *software* (VMM, VM e aplicação) aos modelos de baixo nível iniciais.

DANTAS et al. (2012) propõem modelos hierárquicos, utilizando RBD e CTMC para avaliação de dependabilidade em infraestruturas de computação em nuvem. Os modelos propostos nesse trabalho permitem avaliar o impacto da adoção do mecanismo de replicação *warm standby* em partes críticas de sistemas de computação em nuvem. Além disso, esse trabalho adota os recursos de infraestrutura de nuvem baseando-se no *software Eucalyptus* (EUCALYPTUS, 2012). Os resultados obtidos demonstram o impacto do uso ou não de redundância na disponibilidade do sistema, onde observa-se que a abordagem com redundância resulta em um aumento significativo na disponibilidade, bem como diminuição no *downtime* anual do sistema.

Apesar dos trabalhos apresentados possibilitarem a modelagem e avaliação de atributos de dependabilidade em diversos tipos de sistemas, tais trabalhos não consideram dependabilidade em ambientes de redes virtualizadas. Neste trabalho, diferentemente dos anteriores, são consideradas nos modelos as particularidades das redes virtuais, por exemplo, a condição de funcionamento de uma rede virtual e as falhas dos nós e enlaces físicos e virtuais. Os modelos aqui propostos permitem a avaliação do impacto na disponibilidade do uso dos diferentes componentes da infraestrutura física na alocação das redes virtuais.

3.2 ALOCAÇÃO DE REDES VIRTUAIS

Ao longo dos últimos anos, diversas técnicas têm sido propostas para lidar com a alocação de redes virtuais em infraestruturas compartilhadas, objetivando melhorar métricas importantes, tais como a taxa de aceitação (requisições aceitas sobre o total de requisições), utilização dos recursos de rede (a soma total dos recursos reservados sobre a capacidade total da rede) e o custo (CHOWDHURY et al., 2017), (NGUYEN et al., 2017), (ARAÚJO et al., 2018), (HOU et al., 2018), (CAO et al., 2018), (YU et al., 2018). Contudo, poucos trabalhos lidam com questões de dependabilidade na alocação das redes virtuais.

É importante mencionar que o problema de alocação de redes virtuais é similar ao de alocação de Redes Virtuais Privadas (VPN) em uma topologia de rede compartilhada. Contudo, uma requisição de VPN típica considera apenas restrições de largura de banda nos enlaces, sem quaisquer restrições associadas aos nós. Logo, a maior parte dos algoritmos para alocação de VPNs consistem na busca de caminhos entre pares de nós origem e destino (CHOWDHURY et al., 2009). Entretanto, em um ambiente de virtualização de redes, observam-se restrições em ambos nós e enlaces da rede. Além disso, os nós e enlaces físicos podem ser compartilhados por múltiplas redes virtuais. A fim de reduzir a complexidade do problema de alocação de RVs, alguns trabalhos existentes têm simplificado-o de diferentes formas, as quais incluem:

1. Modelagem da versão *offline* do problema, onde todas as requisições de redes virtuais são conhecidas previamente (ZHU; AMMAR, 2006), (LU; TURNER, 2006);
2. Abstração das restrições de nós ou enlaces (LU; TURNER, 2006);
3. Foco em topologias de redes virtuais específicas (ZHU; AMMAR, 2006); e,
4. Assume-se capacidade infinita para os nós e enlaces físicos (LU; TURNER, 2006).

LISCHKA; KARL (2009) propõem um algoritmo que realiza a alocação dos nós e enlaces ao mesmo tempo. A principal vantagem dessa abordagem é que as restrições de alocação dos enlaces são consideradas durante cada etapa da alocação. Ou seja, caso uma decisão de alocação ineficiente seja realizada, o processo volta para o estágio anterior e uma nova decisão de alocação é realizada. Estudos de caso realizados demonstram que esse método realiza alocações eficientes e mais rápidos do que a abordagem baseada em duas fases (uma para a alocação de nós e outra posterior para a alocação dos enlaces), especialmente para redes virtuais grandes, com grande utilização de recursos físicos.

SZETO; IRAQI; BOUTABA (2003) apresenta uma técnica para alocação de redes virtuais com objetivo de maximizar o número de redes virtuais que podem ser alocadas simultaneamente em uma infraestrutura física compartilhada. Ou seja, o objetivo de tal abordagem é maximizar a taxa de aceitação das requisições. Além disso, é adotada uma

estratégia de pré-alocação dos nós utilizados na alocação. Resultados experimentais indicam que o uso dessa técnica resulta em melhor utilização dos recursos físicos, permitindo que mais redes virtuais coexistam em uma mesma rede física compartilhada. Contudo, esse trabalho simplifica as restrições dos recursos físicos, assumindo capacidade infinita nos nós físicos da rede.

Os autores YU et al. (2008) consideram a alocação das redes virtuais em duas fases: (i) alocação dos nós e (ii) dos enlaces. Na primeira fase, adota-se uma abordagem gulosa para alocar os nós virtuais em nós físicos, onde os nós físicos selecionados são os que possuem mais recursos disponíveis. Posteriormente, os enlaces virtuais são alocados em caminhos físicos, através de um algoritmo de geração de caminhos mínimos. Os resultados obtidos demonstram os benefícios de tal abordagem, que torna o problema computacionalmente mais simples e resulta em alocações eficientes. Contudo, tal abordagem não considera métricas como confiabilidade e disponibilidade na alocação.

MELO et al. (2013) apresentam uma abordagem baseada em programação inteira linear (ILP) (WOLSEY, 1998) para formular e resolver o problema *online* de alocação de redes virtuais. A função objetiva adotada visa minimizar a utilização dos recursos físicos, bem como realizar balanceamento de carga. Os experimentos realizados demonstram resultados proeminentes para métricas de desempenho, como taxa de aceitação e utilização dos recursos físicos. Entretanto, apenas métricas de desempenho são adotadas.

Os autores CHOWDHURY; RAHMAN; BOUTABA (2009) propõem dois algoritmos para alocação de redes virtuais: determinístico (D-ViNE) e aleatório (R-ViNE). Esses algoritmos realizam a alocação dos nós virtuais de forma a facilitar a fase posterior, de alocação de enlaces virtuais em caminhos físicos. Além disso, é adotada uma formulação baseada em programação inteira mista (MIP) (SCHRIJVER, 1996) para resolver o problema de alocação. Os resultados experimentais desse trabalho demonstram que o algoritmo que utiliza randomização (R-ViNE) pode apresentar melhores resultados de desempenho quando comparado com a abordagem determinística (D-ViNE), bem como com outros algoritmos representativos existentes. Novamente, questões de dependabilidade não são adotadas na formulação do problema de otimização.

KOSLOVSKI et al. (2010) apresentam um *framework* para alocação de redes virtualizadas, considerando o *Mean Time Between Failures (Tempo Médio Entre Falhas)* (MTBF) associado aos nós e enlaces da rede. É apresentada uma linguagem descritiva, chamada ViPXi, a qual permite que requisições de redes virtuais, bem como suas restrições, sejam especificadas, incluindo restrições de confiabilidade. Entretanto, tal abordagem não apresenta o mecanismo adotado para estimar os valores de MTBFs. Além disso, a confiabilidade é informada pelo usuário em termos da confiabilidade dos componentes individuais. Logo, não é apresentada uma abordagem para estimar a confiabilidade ou disponibilidade da rede virtual como um todo.

RAHMAN; AIB; BOUTABA (2010) incorporam o conceito de sobrevivência (o qual

envolve falhas na virtualização de redes) para lidar com o problema de sobrevivência de redes virtuais, bem como as heurísticas propostas consideram roteamentos redundantes (enlaces) para a comunicação entre os nós. Nesse trabalho, há a formulação do problema de sobrevivência na alocação de redes virtuais (SVNE). Para resolução do SVNE, uma política híbrida é adotada, a qual consiste de três fases.

Primeiramente, e antes de quaisquer requisições de RVs serem recebidas, são computados candidatos a *backup* para cada enlace físico da rede. Ou seja, cada enlace físico l é alocado em um conjunto de enlaces físicos que possam servir de *backup* de l . A segunda fase ocorre no momento da chegada de uma requisição de RV. Nessa fase, é realizada a alocação dos nós utilizando heurísticas propostas em trabalhos anteriores. Por fim, a terceira fase ocorre no evento de falha de um enlace físico da rede, onde um *backup* (pré-definido na primeira fase) é selecionado para substituir o enlace físico com falha.

Contudo, esse trabalho não considera eventos de falha nos nós físicos. Além disso, falhas nos componentes de *software* (como sistema operacional, máquina virtual, e assim por diante) não são consideradas. Por fim, a abordagem proposta não permite que o cliente especifique um nível mínimo desejado para a disponibilidade ou confiabilidade da rede virtual. Logo, essa técnica não garante que níveis de SLAs baseados em tais métricas de dependabilidade sejam satisfeitos.

KHAN et al. (2016) proveem uma estratégia de alocação chamada SiMPLE (*Survivability in Multi-Path Link Embedding*), para promover tolerância a falhas em casos de falhas individuais de enlaces em redes virtuais. Entretanto, tal abordagem não considera falhas dos nós, além de não possibilitar que os clientes especifiquem um nível mínimo de disponibilidade que deve ser atingido pela rede virtual. Por fim, questões de consumo energético não são levadas em consideração.

BOTERO et al. (2012) propõem uma extensão do problema de alocação de redes virtuais para lidar com questões energéticas. Contudo, o trabalho não provê um modelo para estimar o consumo de energia em redes virtuais. SU et al. (2014) apresentam uma técnica para a alocação de redes virtuais considerando consumo de energia. O trabalho assume uma formulação de problema baseada em programação inteira linear, mas nada é considerado em relação à dependabilidade nas alocações. WANG et al. (2012) propõem um modelo de consumo energético para a alocação de redes virtuais. Entretanto, os autores não validam o modelo em uma infraestrutura real de redes virtuais. Além disso, novamente, questões de dependabilidade não são consideradas na formulação do problema de otimização.

GONG et al. (2016) apresentam um modelo de consumo energético para redes heterogêneas e formulam o problema de alocação de redes virtuais considerando consumo de energia, utilizando programação inteira linear. Além disso, os autores apresentam um algoritmo de alocação de redes virtuais, o qual explora o consumo energético distinto dos nós físicos para obter economia de energia. Contudo, não é demonstrada a aplicabilidade

do modelo energético proposto, uma vez que tal modelo não é validado em um ambiente real. Além disso, os autores não apresentam o impacto dos resultados em termos de custo de energia e disponibilidade.

Diferentemente dos trabalhos citados, esta tese propõe uma abordagem de alocação para redes virtualizadas que considera questões de dependabilidade e energia, bem como o uso ou não de diferentes políticas de redundância na alocação. A formulação do problema de otimização proposto, inclui nos enlaces e nós físicos da rede atributos de dependabilidade, como o *Mean Time to Failure (Tempo médio de Falha)* (MTTF) e o *Mean Time to Repair (Tempo médio de Recuperação)* (MTTR) de cada componente. Adicionalmente, é apresentada uma técnica de modelagem que permite estimar a vazão orientada à disponibilidade em enlaces virtuais. Além disso, é proposto um modelo energético, o qual permite estimar o custo de energia bem como o consumo energético associados à alocação de uma rede virtual. Não obstante, o modelo energético apresentado é validado através de medição em um ambiente real de virtualização de redes. Além disso, são propostos mecanismos para modelagem e avaliação de dependabilidade nas redes virtuais, através do uso de modelos formais como redes de Petri estocásticas (SPN) e diagramas de bloco de confiabilidade (RBD). Não obstante, são propostos mecanismos para a tradução automática de redes virtuais nos modelos de dependabilidade equivalentes. Isso possibilita que, durante a alocação, as soluções candidatas encontradas sejam avaliadas em termos de dependabilidade, e as soluções que não satisfaçam as necessidades dos clientes sejam identificadas e descartadas. Em seguida, é apresentado um algoritmo de alocação, baseado na metaheurística GRASP, para alocação de redes virtuais considerando restrições de dependabilidade e consumo de energia. O algoritmo apresentado adota análise de sensibilidade baseada em índices de importância de disponibilidade para incluir redundâncias na alocação até o ponto em que o nível de disponibilidade requerido pelo cliente é atingido. Finalmente, as técnicas de modelagem e alocação desenvolvidas são integradas na ferramenta de *software* MODERADOR, permitindo que administradores e gestores de redes possam projetar redes virtuais considerando métricas de dependabilidade, consumo de energia e desempenho.

3.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais trabalhos relacionados à área de pesquisa desta tese. Inicialmente, apresentou-se trabalhos representativos na área de modelagem e avaliação de dependabilidade de sistemas. Posteriormente, foram apresentados trabalhos na área de virtualização de redes, focando nas técnicas atuais mais proeminentes de alocação de requisições de redes virtuais em infraestruturas físicas compartilhadas. A Tabela 2 apresenta um resumo das características dos principais trabalhos relacionados à alocação de redes virtuais de computadores. As características de cada trabalho foram agrupadas quanto à avaliação de métricas de desempenho na alocação de redes virtuais, avaliação de

métricas de QoS (por exemplo, a disponibilidade), aplicação ou não de estratégias para medição do consumo de energia e quanto à validação da estratégia adotada para estimar o consumo de energia em redes virtuais. Como pode-se observar, nenhum dos trabalhos relacionados propõe a alocação de redes virtuais de computadores considerando de forma integrada ambas as métricas de desempenho, QoS e consumo de energia.

Tabela 2 – Trabalhos relacionados

	Avaliação de Desempenho	QoS	Consumo de Energia	Validação do modelo energético
CAO et al. (2018)	sim	não	não	não
YU et al. (2018)	sim	não	não	não
HOU et al. (2018)	sim	sim	não	não
NGUYEN et al. (2017)	sim	não	não	não
CHOWDHURY et al. (2017)	sim	sim	não	não
SHAHRIAR et al. (2017)	sim	sim	não	não
GONG et al. (2016)	sim	não	não	não
KHAN et al. (2016)	sim	sim	não	não
SU et al. (2014)	sim	não	sim	não
MELO et al. (2013)	sim	não	não	não
BOTERO et al. (2012)	sim	não	sim	não
WANG et al. (2012)	sim	não	sim	não
KOSLOVSKI et al. (2010)	sim	sim	não	não
RAHMAN; AIB; BOUTABA (2010)	sim	sim	não	não
CHOWDHURY et al. (2009)	sim	não	não	não
LISCHKA; KARL (2009)	sim	não	não	não
YU et al. (2008)	sim	não	não	não
ZHU; AMMAR (2006)	sim	não	não	não
LU; TURNER (2006)	sim	não	não	não
SZETO; IRAQI; BOUTABA (2003)	sim	não	não	não

4 MÉTODO PROPOSTO E FORMULAÇÃO DO PROBLEMA

Este capítulo apresenta o método proposto e formula o problema de otimização considerado. Primeiramente, a técnica adotada é apresentada para a alocação e avaliação de dependabilidade das redes virtualizadas. Posteriormente, o capítulo trata a formulação do problema de alocação utilizado neste trabalho.

4.1 MÉTODO PROPOSTO

A Figura 14 apresenta o método adotado para a alocação e avaliação de disponibilidade de redes virtualizadas considerando consumo de energia e disponibilidade. Ou seja, trata-se do conjunto de entradas, atividades e saídas esperadas para a alocação de requisições de redes virtuais de computadores na infraestrutura física, considerando disponibilidade e consumo de energia.

Inicialmente, o cliente especifica as características da rede virtual desejada. Essa especificação contempla os nós virtuais (com suas respectivas capacidades de CPU), os enlaces virtuais (com suas respectivas larguras de banda) e a métrica de restrição de dependabilidade, a qual indica o valor mínimo de disponibilidade que a rede virtual alocada deve possuir.

Posteriormente, a especificação realizada pelo cliente é enviada ao Provedor de Redes Virtuais, o qual se encarrega de utilizar o algoritmo de alocação adotado para alocar a requisição recebida na infraestrutura de rede física existente. Este processo pode ser visualizado no retângulo vermelho da Figura 14. O método de alocação utiliza como entrada:

1. O estado atual da rede física (considerando as capacidades de nós e largura de banda dos enlaces).
2. A requisição de rede virtual G^V .
3. A lista de nós N_R que serão alocados utilizando redundância, que é inicialmente vazia.

Tão logo o processo de alocação não obtenha sucesso em encontrar uma solução que satisfaça a restrição de disponibilidade, a análise de sensibilidade baseada em índices de importância de disponibilidade é adotada para verificar qual o nó n que mais impacta a disponibilidade da rede virtual. Em seguida, o nó n é adicionado à lista N_R de nós que devem ser alocados com redundância. Então, o algoritmo de alocação tenta novamente encontrar uma solução para atender à requisição. Esse processo se repete até que uma solução viável (que atende a todas as restrições) seja encontrada ou o critério de parada

seja alcançado (ex: tempo máximo de execução). A função objetivo que guia o algoritmo é a minimização do consumo de energia relacionado à alocação de G^V .

Devido à natureza iterativa da metaheurística GRASP, várias soluções candidatas são consideradas e avaliadas ao longo do processo de alocação, até que o critério de parada seja satisfeito. O critério de parada adotado é o tempo máximo de execução do algoritmo de alocação, já que o cliente não pode esperar indefinidamente que uma solução seja encontrada.

Caso uma solução viável seja encontrada, que satisfaça todas as restrições requeridas pelo usuário, como as capacidades de CPU dos nós, as largura de banda dos enlaces e a disponibilidade mínima da rede, os recursos físicos associados à alocação são reservados. Tais recursos se mantêm indisponíveis para outras requisições até que a rede virtual G^V expire. Caso contrário, o processo de alocação da rede virtual é encerrado.

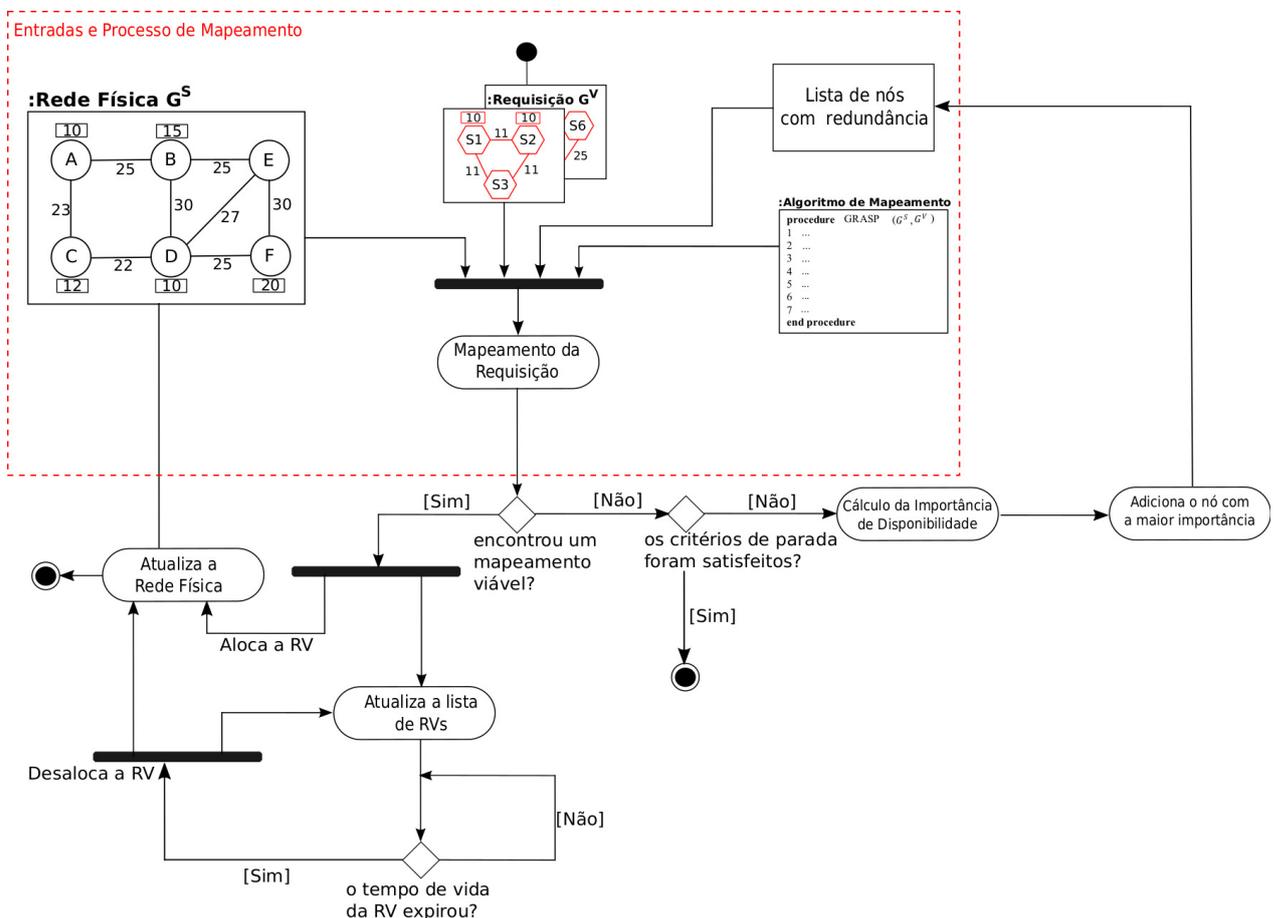


Figura 14 – Método proposto.

4.2 FORMULAÇÃO DO PROBLEMA

Esta seção apresenta a formulação do problema de alocação adotado, o qual é baseado nas abordagens utilizadas em (CHOWDHURY et al., 2009), (WANG et al., 2012) e

(BOTERO et al., 2012). Contudo, é importante observar que tais abordagens utilizam a formulação do problema como *Mixed Integer Programming (Programação Inteira Mista)* (MIP) (SCHRIJVER, 1996) para resolver o problema de alocação. Todavia, tal abordagem não é aplicável neste caso devido às questões de dependabilidade aqui consideradas. Além disso, esses trabalhos consideram a alocação de cada nó virtual em um único nó físico da rede. Analogamente, cada enlace virtual é alocado em apenas um único caminho físico. Por fim, este trabalho difere-se destes mencionados uma vez que eles não consideram concomitantemente disponibilidade e consumo de energia na formulação do problema de otimização.

Este trabalho considera ambas as falhas de hardware (ex: CPU) e software (ex: sistema operacional) na rede física (nós e enlaces), as quais podem ser geradas, por exemplo, por mal funcionamento. Cada recurso físico contém dois atributos adicionais, o tempo médio de falha (MTTF) e o tempo médio de reparo (MTTR). Esses atributos são adotados para estimar a disponibilidade de uma requisição de rede virtual, e abrangem todas as questões aqui consideradas relacionadas às falhas e reparos de cada recurso.

Além disso, a abordagem proposta considera três diferentes políticas de alocação, a qual pode considerar redundância (*hot e cold standby*) ou não. É importante mencionar que, algumas restrições a serem apresentadas não se aplicam a todos os casos. Por exemplo, a política sem redundância não considera restrições relacionadas aos nós e enlaces redundantes.

Mais especificamente, se uma rede virtual é criada considerando redundância, cada nó virtual é alocado em dois nós físicos, um principal e um redundante. Além disso, cada enlace virtual é alocado em quatro caminhos físicos, os quais correspondem aos caminhos do produto cartesiano dos nós físicos (o principal e o redundante) utilizados nas alocações dos nós e enlaces virtuais.

É importante mencionar que a alocação dos enlaces virtuais é considerado no cenário de fluxo indivisível. Ou seja, cada enlace deve ser alocado em apenas um caminho, sem violar as capacidades de cada enlace do caminho físico. A Tabela 3 apresenta a nomenclatura adotada na formulação do problema apresentado.

4.2.1 Rede física

A rede física é representada por um grafo não dirigido $G^S = (N^S, E^S)$, no qual N^S representa o conjunto dos nós físicos da rede e E^S representa o conjunto dos enlaces físicos. Cada nó físico $n^S \in N^S$ é associado a uma capacidade de CPU $c(n^S)$, um tempo médio para falha $mttf(n^S)$ e um tempo médio para reparo $mttr(n^S)$. Analogamente, cada enlace físico $e^S(i, j) \in E^S$ entre dois nós físicos i e j é associado a uma largura de banda $b(e^S)$, um tempo médio de falha $mttf(e^S)$ e um tempo médio de reparo $mttr(e^S)$.

O conjunto de todos os caminhos físicos da rede é representado por P^S , enquanto o conjunto de caminhos de um nó fonte s e um nó destino t é representado por $P^S(s, t)$.

Tabela 3 – Resumo da Notação Adotada

Notação	Descrição
G^S	Rede física
N^S	Conjunto de nós físicos
E^S	Conjunto de enlaces físicos
n^S	Nó físico
$c(x)$	Capacidade do nó x
$e^S(i, j)$	Enlace físico entre os nós i e j
$b(x)$	Largura de banda do enlace x
$mttf(x)$	Tempo médio de falha do componente x
$mttr(x)$	Tempo médio de reparo do componente x
P^S	Conjunto de todos os caminhos físicos da rede
$P^S(s, t)$	Conjunto de caminhos entre os nós s e t
G^V	Rede virtual
N^V	Conjunto de nós virtuais
E^V	Conjunto de enlaces virtuais
n^V	Nó virtual
$e^V(i, j)$	Enlace virtual entre os nós i e j
$D(G^V)$	Disponibilidade mínima requerida pela rede virtual G^V
$Av(G^V)$	Disponibilidade da rede virtual G^V
$R_N(n^S)$	Capacidade disponível no nó físico n^S
$R_E(e^S)$	Capacidade disponível no enlace físico e^S
$M_S(n^V)$	Nó físico no qual o nó virtual n^V está alocado
$M_{SN}(n^V)$	Nó físico redundante no qual o nó virtual n^V está alocado
$M_{ME}(e^V)$	Caminho físico em que o enlace virtual e^V está alocado
$dp(e^V)$	Distância entre os nós físicos em que o enlace virtual e^V está alocado
$c(e^V)$	Custo de alocação do enlace virtual e^V
$c(G^V)$	Custo de alocação da rede virtual G^V
$f_{e^S}^{e^V}$	Largura de banda alocada no enlace físico e^S para o enlace virtual e^V
$c_{n^S}^{e^V}$	Capacidade alocada no nó físico n^S para o nó virtual n^V
$\Pi(G^V)$	Receita associada à aceitação da requisição da rede virtual G^V
$Pw_u(G^V)$	Consumo de energia associado à rede virtual G^V
$Pr_{G^V}(t)$	Preço da eletricidade para a rede virtual G^V no instante t
ΔECS_{G^V}	Preço da eletricidade para a rede virtual G^V
ΔECN_{G^V}	Consumo de energia da rede virtual G^V

A Figura 15 representa uma rede física, na qual os números em volta dos enlaces representam a largura de banda disponível, e os números nos retângulos representam a capacidade de CPU disponível. Para melhor visualização, os tempos médios de falha e de reparo não estão representados na figura.

4.2.2 Rede Virtual

Similarmente à rede física, uma requisição de rede virtual é denotada por um grafo não dirigido, representado por $G^V = (N^V, E^V)$. N^V e E^V denotam, respectivamente, o con-

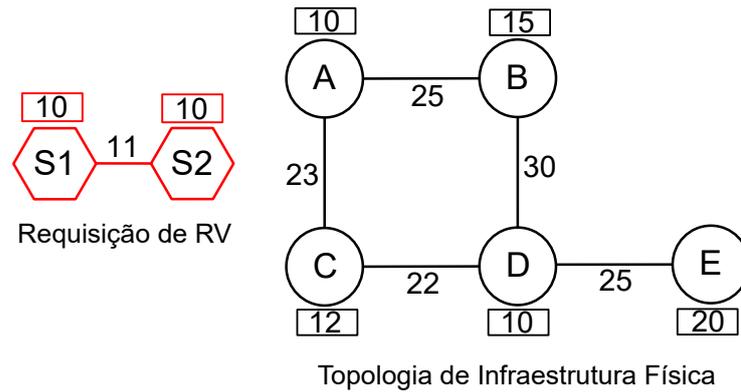


Figura 15 – Um ambiente de virtualização de redes.

junto dos nós e enlaces virtuais da rede. As restrições de capacidades dos nós virtuais são expressas através do atributo $c(n^V)$ (o qual representa a quantidade de CPU necessária durante a alocação do nó virtual), para cada $n^V \in N^V$. Além disso, as restrições de largura de banda dos enlaces virtuais são especificadas pelo atributo $b(e^V)$ (o qual representa a quantidade de largura de banda necessária durante a alocação do enlace virtual), associado a cada nó virtual $e^V(i, j) \in E^V$ entre dois nós virtuais i e j .

Adicionalmente, cada requisição possui uma restrição indicada por $D(G^V)$, a qual representa a disponibilidade mínima requerida pela rede virtual. Ou seja, cada requisição G^V está sujeita a:

$$D(G^V) \leq Av(G^V) \quad (4.1)$$

no qual $Av(G^V)$ representa a disponibilidade da rede virtual G^V .

A Figura 15 ilustra uma requisição de rede virtual, bem como as restrições de nós e enlaces consideradas.

4.2.3 Medição dos Recursos da Rede Física

Na medida em que as requisições das redes virtuais são alocadas e desalocadas na infraestrutura física existente, os recursos físicos disponíveis sofrem variação, diminuindo quando reservados para uma requisição, e aumentando quando os recursos alocados são liberados.

A capacidade restante ou disponível em um nó físico $R_N(n^S)$ corresponde à quantidade total de CPU que não está alocada para nós virtuais, e é definida como a capacidade de CPU disponível no nó físico $n^S \in N^S$.

$$R_N(n^S) = c(n^S) - \sum_{\forall n^V \uparrow n^S} c(n^V) \quad (4.2)$$

no qual $x \uparrow y$ denota que o nó virtual x é alocado no nó físico y .

Analogamente, a capacidade restante ou disponível em um enlace físico $R_E(e^S)$ corresponde à quantidade total de largura de banda que não está alocada para enlaces virtuais, e é definida como o total de largura de banda disponível no enlace físico $e^S \in E^S$.

$$R_E(e^S) = b(e^S) - \sum_{\forall e^V \uparrow e^S} b(e^V) \quad (4.3)$$

no qual $x \uparrow y$ denota que o enlace virtual x está alocado no enlace físico y .

Além disso, a largura de banda total disponível em um caminho $p \in P^S$ é dada por

$$R_E(p) = \min_{e^S \in p} R_E(e^S) \quad (4.4)$$

4.2.4 Alocação de Redes Virtuais

Para cada requisição de rede virtual recebida, o provedor de redes virtuais aceita ou rejeita a requisição, de acordo com os recursos disponíveis e as restrições do cliente. Em caso de aceitação, a alocação da rede virtual na rede física é realizada, reservando os recursos de rede necessários. Os recursos permanecem alocados até o momento que a rede virtual expira, e então, os recursos alocados são liberados para requisições futuras.

A alocação de uma rede virtual é dividido em duas atividades principais: (i) alocação de nós; e (ii) alocação de enlaces. Além disso, todas as requisições devem satisfazer a restrição $D(G^V)$.

4.2.4.1 Alocação de Nós

Cada nó virtual de uma requisição é alocado em um nó físico através de uma alocação $M_N : N^V \rightarrow N^S$, de tal forma que, $\forall n^V \in N^V$ está sujeito a:

$$c(n^V) \leq R_N(M_N(n^V)) \quad (4.5)$$

Sempre que um mecanismo de redundância é adotado (*cold* ou *hot standby*), uma alocação adicional $M_{SN} : N^V \rightarrow N^S$ se faz necessário, de tal forma que, $\forall n^V \in N^V$, $M_{SN}(n^V) \neq M_N(n^V)$ está sujeito a:

$$c(n^V) \leq R_N(M_{SN}(n^V)) \quad (4.6)$$

Adicionalmente, considerando o tipo de redundância *cold standby*, $\forall n^V, m^V \in N^V$ temos:

$$M_N(n^V) \neq M_{SN}(m^V) \quad (4.7)$$

$$M_N(n^V) = M_N(m^V), \text{ se e somente se } (n^V = m^V) \quad (4.8)$$

$$M_{SN}(n^V) = M_{SN}(m^V), \text{ se e somente se } (n^V = m^V) \quad (4.9)$$

Em outras palavras, assume-se que as alocações baseadas nas técnicas *cold* e *hot standby* consideram que o nó redundante deve ser alocado em um nó físico diferente do recurso onde o nó principal foi alocado. Ao mesmo tempo, ao utilizar a técnica de *cold standby* especificamente, assume-se que cada nó redundante deve ser alocado em um nó físico exclusivo na requisição, ou seja, que não esteja sendo utilizado por nenhuma outra alocação da mesma requisição, seja principal ou redundante.

4.2.4.2 Alocação de Enlaces

Cada enlace virtual é alocado em um caminho físico entre os nós físicos utilizados. A alocação dos enlaces virtuais em caminhos físicos é definido por $M_{ME} : E^V \rightarrow P^S(M_N(m^V), M_N(n^V))$, de tal forma que, $\forall e^V(m^V, n^V) \in E^V$ está sujeito a:

$$R_E(p) \geq b(e^V), \forall p \in M_{ME}(e^V) \quad (4.10)$$

Além disso, em requisições de redes virtuais que utilizam redundância, três enlaces virtuais adicionais são requeridos para contemplar os enlaces entre os nós redundantes alocados:

1. $M_{SP} = E^V \rightarrow P^S(M_{SN}(m^V), M_N(n^V))$, entre nó redundante e principal.
2. $M_{PS} = E^V \rightarrow P^S(M_N(m^V), M_{SN}(n^V))$, entre nó principal e redundante.
3. $M_{SS} = E^V \rightarrow P^S(M_{SN}(m^V), M_{SN}(n^V))$, entre nós redundantes.

Estas três alocações adicionais são realizados de tal forma que $\forall e^V(m^V, n^V) \in E^V$:

$$R_E(p) \geq b(e^V), \forall p \in M_{SP}(e^V) \quad (4.11)$$

$$R_E(p) \geq b(e^V), \forall p \in M_{PS}(e^V) \quad (4.12)$$

$$R_E(p) \geq b(e^V), \forall p \in M_{SS}(e^V) \quad (4.13)$$

4.2.5 Distância do Caminho físico dos Enlaces Virtuais

A distância associada ao caminho físico de um enlace virtual é definida como a quantidade de enlaces físicos necessários para conectar os nós físicos utilizados na alocação, e é calculada através da seguinte equação:

$$dp(e^V) = \sum_{e^S \in E^S} \beta_{e^S}^{e^V} \quad (4.14)$$

onde $\beta_{e^S}^{e^V}$ possui valor igual a '1' caso o enlace físico e^S seja utilizado na alocação do enlace virtual e^V . Caso contrário, o valor é igual a '0'. Logo, quanto mais enlaces físicos são utilizados na alocação, maior a distância do caminho físico.

4.2.6 Modelo de Custo de Alocação de Enlace Virtual

O custo de alocação $c(e^V)$ associado ao enlace virtual e^V é calculado através da seguinte equação:

$$c(e^V) = \sum_{e^S \in E^S} f_{e^S}^{e^V} + \sum_{n^S \in N^S} c_{n^S}^{e^V} \quad (4.15)$$

onde $f_{e^S}^{e^V}$ representa a quantidade total de largura de banda alocada no enlace físico e^S para o enlace virtual e^V . Além disso, $c_{n^S}^{e^V}$ representa a quantidade total de capacidade alocada no nó físico n^S para os nós virtuais conectados pelo enlace virtual e^V .

4.2.7 Modelo de Custo de Alocação de Rede Virtual

O custo de alocação de recursos $c(G^V)$ associado com a aceitação de uma requisição de rede virtual G^V é calculado pela seguinte equação:

$$c(G^V) = \sum_{e^V \in E^V} \sum_{e^S \in E^S} f_{e^S}^{e^V} + \sum_{n^V \in N^V} c(n^V) \times \epsilon \quad (4.16)$$

onde $f_{e^S}^{e^V}$ representa o total de largura de banda alocada no enlace físico e^S para o enlace virtual e^V . ϵ é uma variável inteira, a qual é igual a '2' sempre que é utilizado redundância na alocação do nó virtual. Caso contrário, o valor é igual a '1'.

A função de custo de alocação de recursos é a mesma adotada em (CHOWDHURY et al., 2009), com a diferença de que foi adicionada a possibilidade da alocação de recursos redundantes devido às políticas *hot* e *cold standby*.

4.2.8 Receita

O custo de alocação de recursos representa a quantidade total de recursos físicos de infraestrutura que o provedor de redes virtuais está adotando para uma determinada requisição. A receita, por outro lado, denota a quantidade total de recursos virtuais que um provedor de redes virtuais está provendo para uma requisição. Em outras palavras, mais recursos de infraestrutura alocados geram um custo mais elevado, o qual afeta negativamente o lucro do provedor. No entanto, uma maior receita significa que mais recursos virtuais estão sendo providos, o que possivelmente indica um maior lucro para o provedor.

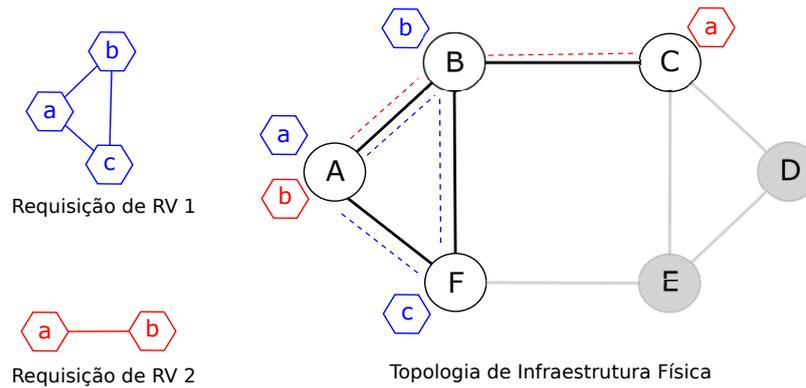


Figura 16 – Exemplo de economia de energia

Neste trabalho, a recompensa $\Pi(G^V)$ é definida como o benefício gerado ao provedor por aceitar a requisição de rede virtual G^V , expressa de acordo com os nós e enlaces da rede virtual.

$$\Pi(G^V) = T(G^V)[F_1 \sum_{e^v \in E^V} b(e^V) + F_2 \sum_{n^v \in N^V} c(n^V)] \quad (4.17)$$

onde F_1 e F_2 são fatores de peso, os quais representam, respectivamente, a importância relativa da largura de banda e CPU para a recompensa gerada. Por exemplo, pode-se adotar os valores $F_1 = 0,4$ e $F_2 = 0,6$, caso o projetista de rede considere que a recompensa gerada ao alocar uma unidade de CPU de um nó virtual é 50% maior do que para alocar uma unidade de largura de banda de um enlace virtual. Além disso, $T(G^V)$ denota o tempo de vida da rede virtual representada por G^V .

4.2.9 Modelo de Consumo de Energia

Neste trabalho, o modelo de consumo de energia considera as seguintes assertivas (WANG et al., 2012):

1. Se um enlace está desligado, economiza-se energia nas suas interfaces e nas portas de suas extremidades, as quais também estão desligadas;
2. Se um nó físico está desligado, todos os enlaces conectados a esse nó também estão desligados;
3. Sempre que um nó está inutilizado (ou seja, quando não possui nós virtuais alocados), o mesmo é desligado, logo, ele não consome energia.

Por exemplo, considere a rede física, as duas requisições de redes virtuais e a alocação apresentado na Figura 16. Todos os nós e enlaces físicos inutilizados podem ser desligados, o que resulta em economia de energia. Nesse caso particular, os nós inativos são D e E . Os enlaces inativos são (C, D) , (C, E) , (D, E) , (E, F) .

Haja vista tais considerações, o consumo de energia $Pw_u(G^V)$ resultante da aceitação da requisição G^V é calculado da seguinte forma:

$$Pw_u(G^V) = \sum_{n \in N^S} \alpha_n (\gamma_n^k - \lambda_n^k) + \sum_{e \in E^S} \beta_e (x_e^k - \eta_e^k) + \sum_{n \in N^S} u_{PP}^n c(n) \quad (4.18)$$

onde as variáveis seguintes são binárias (1 quando ativa, 0 caso contrário). γ_n^k denota se um nó físico está ativo ou não após a alocação de G^V . λ_n^k indica se um nó físico está ativo ou não antes da alocação de G^V . Após alocar G^V , o valor de x_e^k é 1 caso o enlace físico e esteja ativo. Finalmente, η_e^k é 1 caso o enlace físico estivesse ativo antes de alocar G^V .

Na equação anterior, o primeiro termo representa o consumo de energia fixo relativo aos nós físicos inativos e equipamentos de rede ativados após a alocação de G^V . α_n representa o consumo de energia base de um nó ativo n , bem como o consumo de energia dos equipamentos de rede relacionados ao nó físico n . A segunda parte em $Pw_u(G^V)$ denota o consumo de energia variável dos equipamentos de rede (relacionados aos enlaces ativos). Estudos anteriores demonstraram que o valor de β é aproximadamente igual a $1W$. Finalmente, na terceira parte da equação, u_{PP}^n representa o consumo de energia variável devido ao uso de memória e CPU (WANG et al., 2012).

É importante mencionar que o modelo energético adotado considera o consumo de energia elétrica de cada nó e enlace individualmente, além de permitir a análise de cada nova alocação no consumo energético final do provedor de infraestrutura física. Isso é particularmente importante, pois permite que o provedor visualize se uma determinada requisição, ou ainda, se um determinado padrão de requisições (ex: pela topologia da rede) conduz a um consumo energético muito acima da média, por exemplo.

4.2.10 Modelo de Precificação de Energia

Localidades diferentes podem adotar operadores de sistemas independentes para gerenciar questões relativas à eletricidade. Como consequência, cada localização possui um preço de eletricidade diferente das demais. Adicionalmente, mesmo em uma única localidade o preço da eletricidade pode variar significativamente ao longo do tempo (SU et al., 2014). Diante disto, de forma a representar o dinamismo inerente aos preços da eletricidade, este trabalho adota um modelo de tempo discreto. $Pr_{G^V}(t)$ denota o preço da eletricidade para a rede virtual G^V no instante de tempo t , t_a representa o tempo de chegada de G^V e t_d denota o tempo de vida da rede virtual. O tempo de expiração t_e de G^V é $t_e = t_a + t_d$, logo, podemos quantificar o preço da eletricidade ΔECS_{G^V} da seguinte forma:

$$ECS_{G^V} = \sum_{t=t_a}^{t_e} [Pw_u(G^V) \times Pr_s(t)] \quad (4.19)$$

4.2.11 Função Objetivo

O algoritmo realiza a alocação de requisições de redes virtuais satisfazendo às restrições especificadas (ex: disponibilidade), minimizando o consumo de energia ΔECN_{G^V} oriundo da alocação de G^V :

$$\Delta ECN(G^V) = Pw_u(G^V) \times T(G^V) \quad (4.20)$$

4.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou o método e a formulação do problema de otimização adotados. Primeiramente, foi apresentado o método proposto para a alocação e avaliação das redes virtuais. Posteriormente, foi realizada a formulação matemática do problema, no qual as restrições adotadas e a função objetiva da heurística foram apresentadas.

5 MODELOS E FERRAMENTA PARA ALOCAÇÃO DE REDES VIRTUAIS DE COMPUTADORES CONSIDERANDO DISPONIBILIDADE E CONSUMO DE ENERGIA

O uso de técnicas, estratégias e modelos que permitam a avaliação de dependabilidade em redes virtualizadas é de grande importância para as empresas que utilizarão essa infraestrutura para prover serviços, uma vez que equipamentos físicos (*hardware*) e programas (*software*) são suscetíveis a falhas. Logo, essas falhas podem impactar ou inviabilizar o uso do sistema por parte dos usuários finais.

Este capítulo apresenta os modelos de diagramas de bloco de confiabilidade (RBD) e redes de Petri estocásticas (SPN) utilizados para a representação e avaliação de disponibilidade e vazão orientada à disponibilidade de redes virtuais. Além disso, o algoritmo de alocação adotado é detalhado para a alocação das requisições na infraestrutura física. Finalmente, este capítulo apresenta a ferramenta implementada MODERADOR - uma ferramenta para alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia.

5.1 MODELOS

Esta seção apresenta os modelos RBD e SPN adotados para quantificar a disponibilidade e a vazão orientada à disponibilidade das redes virtuais. Inicialmente, são apresentados individualmente os modelos SPN adotados. Posteriormente, são apresentados os modelos finais utilizados para representação de uma rede virtual sem redundância, com redundância *cold standby* e *hot standby*, respectivamente.

É importante lembrar que, para os modelos SPN, $\#p$ representa o número de *tokens* em p , e $P\{exp\}$ representa a probabilidade da expressão *exp* ser verdadeira.

5.1.1 Componente Simples

O primeiro modelo apresentado é o que não utiliza redundância, onde os estados possíveis para cada componente são operacional ou falho. Esse componente (Figura 17) (Silva & et al. 2012) possui dois parâmetros, X_MTTF e X_MTTR , os quais representam os tempos associados as transições $X_Failure$ e X_Repair , respectivamente. Além disso, as transições adotam a política de concorrência servidor único (ss). O rótulo X refere-se ao nó ou enlace representado (exemplo: A_Repair e $A_Failure$, para o nó A). Os lugares X_ON e X_OFF representam os estados de atividade e inatividade do componente. A expressão $\#X_ON > 0$ é adotada para indicar se o componente está operacional.

A Tabela 4 apresenta os atributos das transições do componente simples. Por fim, é importante mencionar que, o componente simples apresentado é limitado (possui apenas dois estados), reversível e vivo (MURATA, 1989).

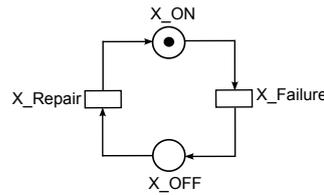


Figura 17 – Modelo de Componente Simples

Tabela 4 – Atributos das transições do componente simples

<i>Transição</i>	<i>Tipo</i>	<i>Atraso</i>	<i>Marcação</i>	<i>Concorrência</i>
X_Failure	exp	X_MTTF(_i)	constante/dependente	SS
X_Repair	exp	X_MTTR(_i)	constante/dependente	SS

5.1.2 Cold Standby

No modelo *cold standby* (Figura 18) (SILVA et al., 2012), cada nó virtual é alocado em dois nós físicos, um principal e um redundante. Inicialmente, o nó principal está ativado, enquanto o nó redundante permanece desativado até que ocorra uma falha no nó principal. Quando a falha ocorre, há um atraso, chamado *Mean Time to Activate (Tempo médio de Ativação)* (MTActivate), entre o momento em que o componente principal falha e o momento em que o redundante é, de fato, ativado.

Os nós principal e redundante possuem seus estados de atividade e falha representados pelos lugares X_ON , X_OFF , X_Spare_ON e X_Spare_OFF , respectivamente. Quando o nó principal falha (ocorre o disparo da transição $X_Failure$), então a transição $Activate_Spare_X$ é habilitada e pode disparar. Quando ocorre esse disparo, o nó redundante é ativado. Uma vez que o nó principal retorne para o estado operacional, ou seja, $\#X_ON > 0$, a transição $Deactivate_Spare_X$ dispara, desabilitando o nó redundante. Uma vez que para o componente funcionar, apenas um dos nós (principal ou redundante) precisa estar operacional, o estado de funcionamento é dado por $\#X_ON + \#X_Spare_ON > 0$. A Tabela 5 apresenta os atributos relacionados à cada transição do modelo descrito. Uma vez que cada distribuição de disparo das transições é independente de marcação, as transições dependentes de marcação são abstraídas. Em relação às propriedades estruturais e comportamentais, é importante mencionar que o modelo *cold standby* apresentado é alcançável, limitado, seguro e não possui *deadlocks*.

5.1.3 Componente de Comunicação de Enlaces

A Figura 19 apresenta o componente de comunicação de enlaces, o qual é composto por um bloco de origem, blocos intermediários e o bloco de destino. Primeiramente, o bloco de origem representa a transmissão de pacotes através do nó físico, o qual representa a origem da transmissão de dados. O bloco de destino, por sua vez, denota a chegada com sucesso ao nó físico de destino do pacote enviado. Além disso, para cada nó intermediário

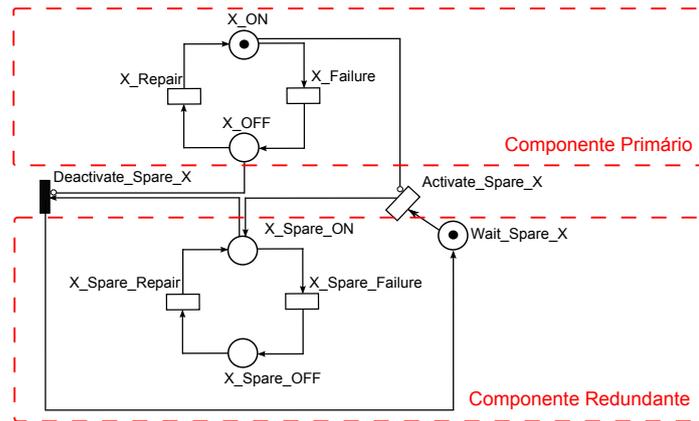


Figura 18 – Modelo Cold Standby

Tabela 5 – Atributos das transições do componente cold standby

<i>Transição</i>	<i>Tipo</i>	<i>Atraso</i>	<i>Pesos</i>	<i>Concorrência</i>
X_Failure	exp	X_MTTF	-	SS
X_Repair	exp	X_MTTR	-	SS
Activate_Spare_X	exp	MTActivate	-	SS
X_Spare_Failure	exp	X_MTTF_Spare1	-	SS
X_Spare_Repair	exp	X_MTTR_Spare1	-	SS
Deactivate_Spare_X	im	-	1	-

necessário para conectar os nós de origem e destino do enlace, um bloco intermediário é criado.

A transição *PACKET_TRANSMISSION* representa o atraso no envio com sucesso de um pacote entre dois nós físicos. Por outro lado, a transição *PACKET_LOSS* denota uma perda de pacote devido a uma falha nos respectivos enlace ou nós físicos. Logo, cada transição *PACKET_LOSS* possui uma expressão de guarda, de forma a garantir que a transição seja disparada apenas caso algum nó ou enlace adotado na comunicação falhe. Além disso, é importante mencionar que as transições *TARGET* e *PACKET_TRANSMISSION* possuem uma taxa de disparo maior (ou um atraso menor) quando comparadas com a transição *SOURCE*. Tal restrição é necessária para a avaliação do modelo através de simulação estacionária (MARSAN et al., 1998).

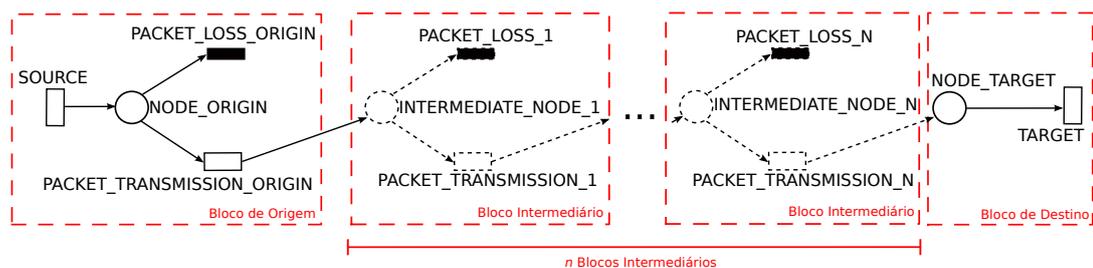


Figura 19 – Modelo do componente base de comunicação de enlaces

A Figura 20 apresenta um exemplo da construção deste modelo, representando a

comunicação entre dois nós: A e D . Como é possível observar no modelo resultante, uma vez que os pacotes necessitam ser transmitidos através de dois nós intermediários (B e C), o modelo SPN é composto pelo bloco de origem, dois blocos intermediários e o bloco de destino.

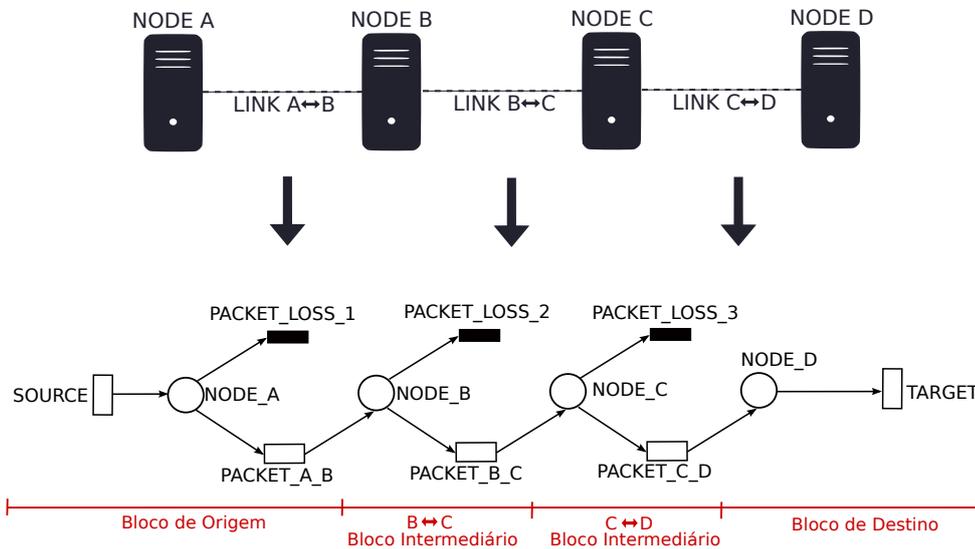


Figura 20 – Exemplo de componente de comunicação de enlaces

5.1.4 Composição de Modelos para Representação de Redes Virtuais

Esta seção apresenta (através de exemplos) a abordagem adotada para combinar modelos e estimar a disponibilidade de redes virtualizadas ($Av(G^V)$), Seção 4.2.2. A abordagem utilizada adota uma técnica de modelagem híbrida, uma vez que contempla redes de Petri estocásticas, ou diagramas de bloco de confiabilidade, ou mesmo ambos para mitigar a complexidade de representação das redes virtuais.

Inicialmente, cada nó e enlace físico é representado por um diagrama RBD distinto, no qual todos os componentes relacionados, tanto de HW (*hardware*) quanto de SW (*software*) estão em um arranjo em série. Ou seja, cada nó e enlace é modelado pela composição em série de todos os componentes suscetíveis à falhas que o compõem. Logo, a falha de quaisquer elementos da infraestrutura acarreta em uma falha no componente (por exemplo, uma falha de CPU, memória, ou interface de rede resulta na falha do nó a que estes elementos estão associados). Após essa representação inicial, os respectivos MTTFs e MTTRs são obtidos e utilizados em um único componente RBD (Figura 21). O MTTF e MTTR do componente final pode ser obtido através da Equação 2.7. Esse componente resultante é representado em uma das abordagens de composição apresentadas a seguir, redes de Petri estocásticas e diagramas de bloco de confiabilidade

Uma vez computados os valores finais dos MTTFs e MTTRs de cada nó e enlace do modelo, é possível estimar a disponibilidade dos nós e enlaces individualmente. Para estimar a disponibilidade da rede virtual como um todo, os nós e enlaces são combinados

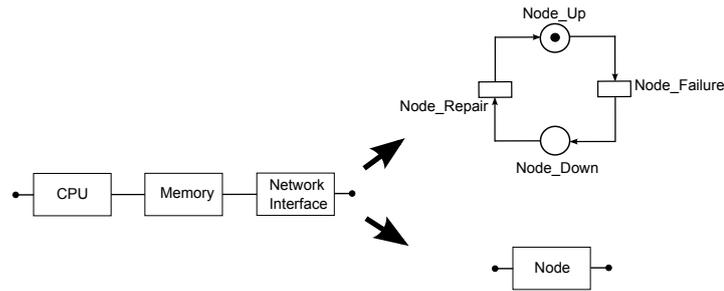


Figura 21 – Componentes de HW e SW

em modelos finais RBD ou SPN. A depender do tipo de política de redundância adotada pela rede virtual (sem redundância, *hot* ou *cold standby*), os modelos finais são gerados de diferentes formas. As seções seguintes apresentam os modelos referentes a cada uma dessas políticas.

5.1.4.1 Sem Redundância

Nesta política, a rede virtual é representada por um modelo RBD através de um arranjo em série. A adoção de RBD deve-se ao fato de que este formalismo permite aplicar equações fechadas para obter a disponibilidade do modelo. Os blocos contemplam os nós e enlaces físicos utilizados na alocação. Por exemplo, considere a infraestrutura, requisição e alocação de rede virtual ilustrados na Figura 22. Nesse caso, a seguinte alocação é adotada: $M_N(S1) = A$, $M_N(S2) = B$ e $M_{ME}(S1, S2) = (A, B)$. A Figura 23 apresenta o modelo RBD da rede virtual em questão.

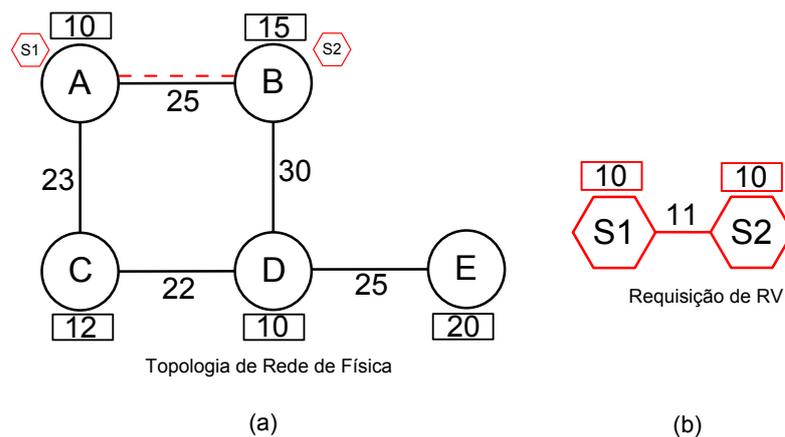


Figura 22 – Sem Redundância - Exemplo de ambiente

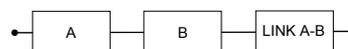


Figura 23 – Sem Redundância - Exemplo de modelo RBD

5.1.4.2 Hot Standby

Na técnica de *hot standby* (Seção 2.2.1), cada nó virtual da requisição é alocado em dois nós físicos da infraestrutura, um principal e um redundante (réplica). Em relação aos enlaces virtuais, cada um é alocado em até quatro caminhos físicos. Isso ocorre porque é necessário haver um caminho entre todas as combinações dos nós principais e redundantes de um enlace virtual. Ambos os nós principal e redundante estão ativos inicialmente. Nessa política, a abordagem proposta adota a combinação de arranjos RBD em série e paralelo. Cada enlace virtual considera uma composição paralela, contemplando os enlaces físicos, bem como os nós físicos utilizados na comunicação. Assim como na política sem redundância, o RBD foi adotado por permitir a adoção de equações fechadas para calcular a disponibilidade das redes virtuais.

Considere a rede física, a requisição de rede virtual e a alocação apresentados na Figura 24. A alocação considerada é:

Tabela 6 – Exemplo de Alocação com Hot Standby

<i>Componente Virtual</i>	<i>Notação</i>	<i>Alocação</i>
Nó $S1$	$M_N(S1)$	C
Réplica do nó $S1$	$M_{SN}(S1)$	E
Nó $S2$	$M_N(S2)$	D
Réplica do nó $S2$	$M_{SN}(S2)$	E
Caminho entre $S1$ principal e $S2$ principal	$M_{ME}(S1, S2)$	$\{(C, D)\}$
Caminho entre $S1$ réplica e $S2$ principal	$M_{SP}(S1, S2)$	$\{(E, D)\}$
Caminho entre $S1$ principal e $S2$ réplica	$M_{PS}(S1, S2)$	$\{(C, D), (D, E)\}$
Caminho entre $S1$ réplica e $S2$ réplica	$M_{SS}(S1, S2)$	$\{(\emptyset)\}$

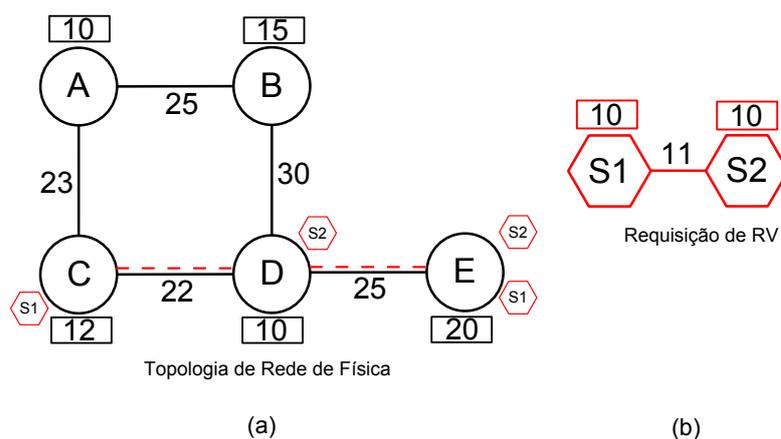


Figura 24 – Hot standby - Exemplo de ambiente

É importante mencionar que, a escolha dos nós e caminhos físicos adotados é realizada durante a alocação da requisição. A Figura 25 ilustra o processo de tradução da rede virtual para o modelo RBD. É importante mencionar que os modelos intermediários

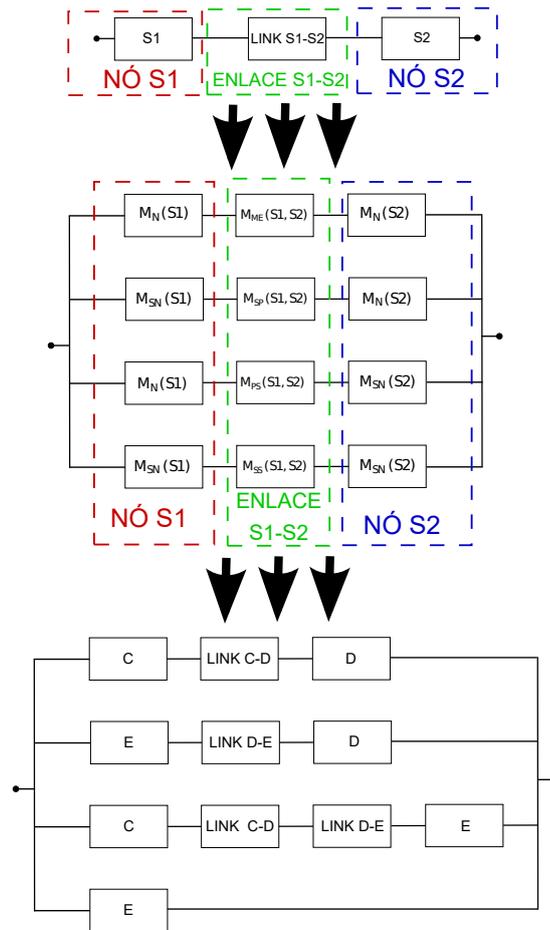


Figura 25 – Geração de modelo hot standby

são apresentados aqui apenas para um melhor entendimento dos modelos, sendo apenas o modelo final considerado para estimar métricas de dependabilidade. Na Figura 25, observa-se primeiramente uma visão de alto nível do modelo, formada pelos componentes da rede virtual (nós e enlaces). Então, cada componente virtual é dividido nos componentes (por exemplo, $S1$ é agora representado por ambos $M_N(S1)$ e $M_{S_N}(S1)$). Então, há um modelo RBD final, no qual os componentes físicos utilizados na alocação são de fato representados. É importante mencionar que, uma vez que o modelo RBD final possui componentes repetidos, o método de soma dos produtos disjuntos (KUO; ZUO, 2003) (Seção 2.4.3) é adotado na avaliação.

5.1.4.3 Cold Standby

Uma vez que um sistema *cold standby* possui estrutura dinâmica, o uso de RBD apresenta limitações para representar esse tipo de situação. Logo, adota-se SPN (Figura 27), assumindo que cada nó virtual é representado por um modelo *cold standby* e os enlaces virtuais utilizando componentes simples, um para cada enlace físico alocado. Considere a infraestrutura física, requisição e alocação apresentados na Figura 26. A alocação adotada é a apresentada na Tabela 7:

Tabela 7 – Exemplo de Alocação com Cold Standby

<i>Componente Virtual</i>	<i>Notação</i>	<i>Alocação</i>
Nó $S1$	$M_N(S1)$	A
Réplica do nó $S1$	$M_{SN}(S1)$	B
Nó $S2$	$M_N(S2)$	C
Réplica do nó $S2$	$M_{SN}(S2)$	D
Caminho entre $S1$ principal e $S2$ principal	$M_{ME}(S1, S2)$	$\{(A, C)\}$
Caminho entre $S1$ réplica e $S2$ principal	$M_{SP}(S1, S2)$	$\{(B, D), (D, C)\}$
Caminho entre $S1$ principal e $S2$ réplica	$M_{PS}(S1, S2)$	$\{(A, C), (C, D)\}$
Caminho entre $S1$ réplica e $S2$ réplica	$M_{SS}(S1, S2)$	$\{(B, D)\}$

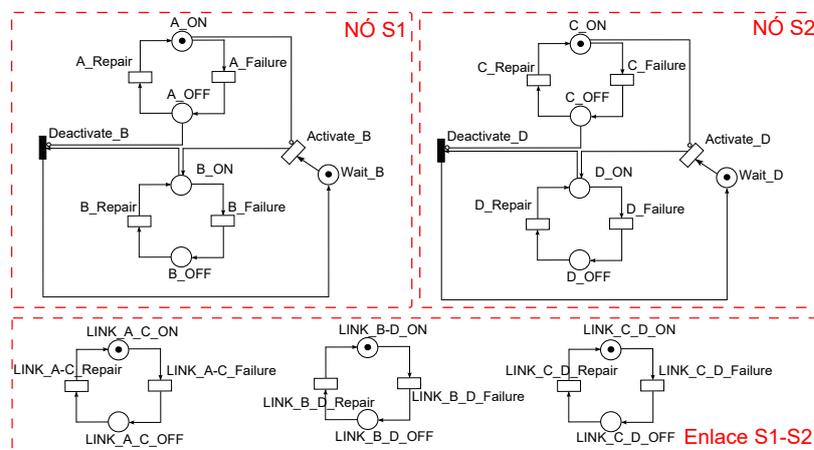


Figura 26 – Cold standby - Exemplo de ambiente

Uma vez que a falha do sistema é dada pela falha de algum nó ou enlace virtual da rede, a disponibilidade da rede virtual é dada pela expressão apresentada na Figura 28. Neste trabalho, utiliza-se a técnica de simulação estacionária para calcular a disponibilidade do modelo SPN gerado.

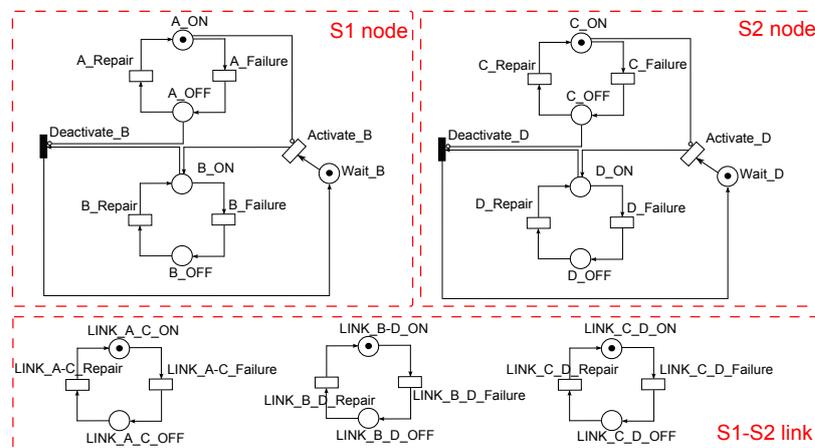


Figura 27 – Cold standby - Exemplo de modelo

```

P{(
  ((#A_ON > 0)AND(#LINK_A_C_ON > 0)AND(#C_ON > 0))
  OR
  ((#B_ON > 0)AND(#LINK_B_D_ON > 0)AND(#LINK_C_D_ON > 0)AND(#C_ON > 0))
  OR
  ((#A_ON > 0)AND(#LINK_A_C_ON > 0)AND(#LINK_C_D_ON > 0)AND(#D_ON > 0))
  OR
  ((#B_ON > 0)AND(#LINK_B_D_ON > 0)AND(#D_ON > 0))
})

```

Figura 28 – Cold standby - Exemplo de expressão de disponibilidade

5.1.5 Modelagem de Vazão Orientada a Disponibilidade

Esta seção apresenta a técnica de modelagem adotada para estimar a vazão orientada a disponibilidade em enlaces virtuais. A construção dos modelos consiste em duas etapas: (i) modelagem de dependabilidade e (ii) modelagem do fluxo de comunicação.

A etapa (i) se baseia na criação de um componente simples SPN para cada nó e enlace físico adotado na alocação do enlace virtual. Tais blocos são utilizados para garantir que a transmissão de dados ocorra apenas caso os componentes físicos necessários para o correto envio e recebimento dos dados estejam operacionais. Em seguida, a etapa (ii) é executada, a qual representa a transmissão de dados entre os nós de origem e destino do enlace virtual.

A depender da política de alocação adotada (sem redundância ou *hot standby*), o modelo final terá um fluxo de comunicação diferente. A subseção a seguir apresenta através de exemplos a modelagem de redes virtuais utilizando as políticas de alocação sem redundância e *hot standby*.

5.1.5.1 Sem redundância

Na política de alocação sem redundância, cada nó virtual n^V é alocado em um único nó físico n^S , e cada enlace virtual $e_V(n^{V1}, n^{V2})$ é alocado em um único caminho físico $p^S(M_N(n^{V1}), M_N(n^{V2}))$ correspondente.

Por exemplo, considere a requisição de rede virtual apresentada na Figura 29, bem como a respectiva alocação (ex: gerado por um algoritmo): $M_N(S1) = A$, $M_N(S2) = D$ e $M_{ME}(S1, S2) = (A, B, D)$. A Figura 30 apresenta o modelo final SPN.

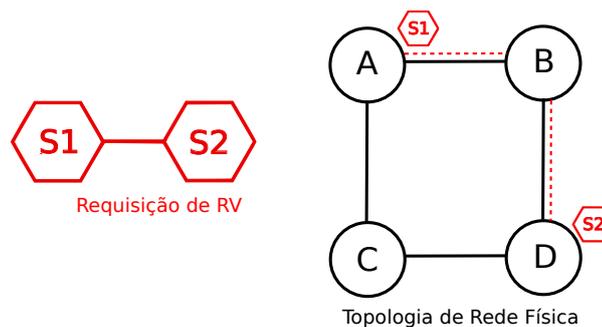


Figura 29 – Cenário de alocação sem redundância

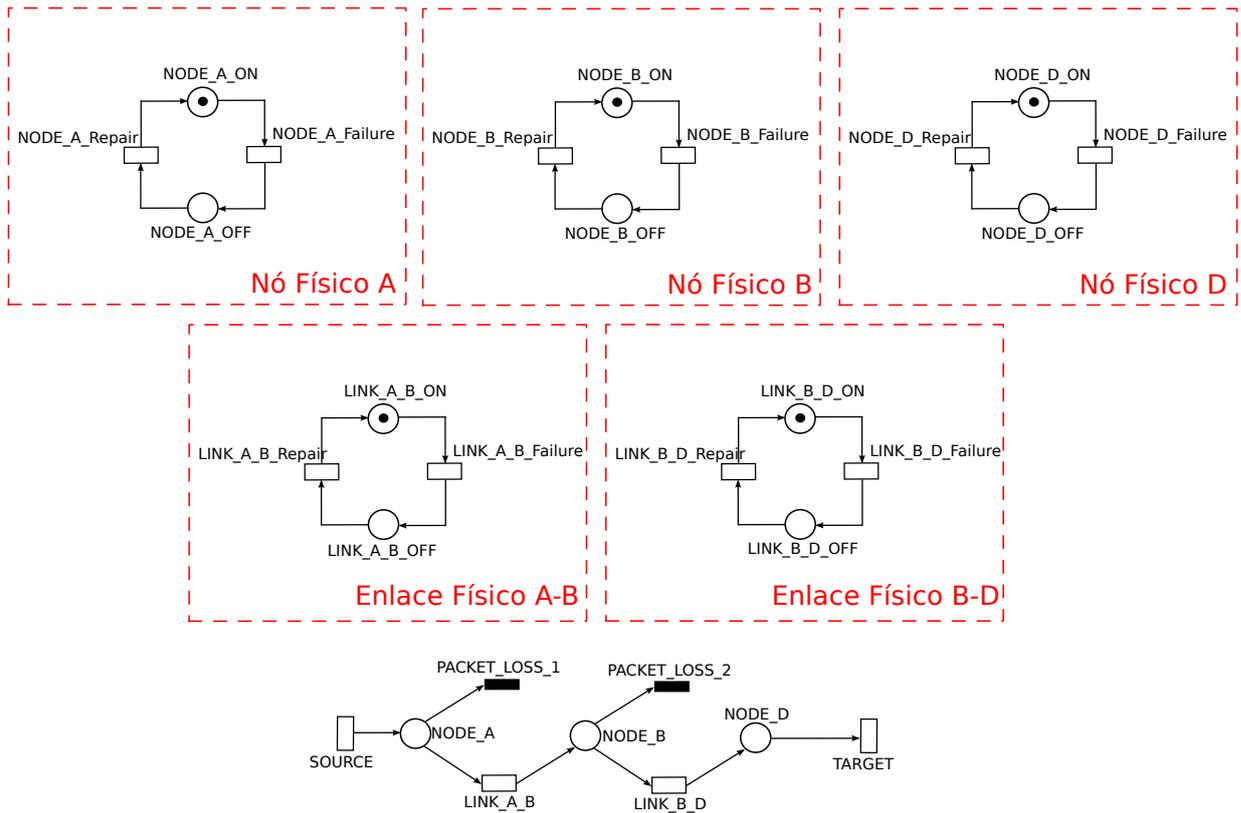


Figura 30 – Modelo de vazão orientada à disponibilidade sem redundância

O modelo final contém um componente simples para cada componente físico, ou seja: três nós físicos (A , B e D) e dois enlaces físicos (conectando A para B e B para D). Para o enlace virtual, o componente de comunicação de enlaces é adotado, no qual a transição $SOURCE$ representa a entrada de dados e a transição $TARGET$ é o destino da transmissão de dados. Nesse cenário, assumamos que há uma marcação nos lugares $NODE_A_ON$, $NODE_B_ON$, $NODE_D_ON$, $LINK_A_B_ON$ e $LINK_B_D_ON$. Caso os nós A e B , bem como o enlace de comunicação entre eles estejam todos operacionais, então a transição $LINK_A_B$ é habilitada para disparo, e, após o seu disparo, o pacote é transmitido para o $NODE_B$. Caso contrário, a transição $PACKET_LOSS_1$ dispara e o pacote é considerado perdido. Um comportamento similar ocorre entre os nós B e D . Nesse modelo, a vazão do enlace é estimada através de $P\{\#NODE_D > 0\} \times \lambda$, onde λ é a taxa de disparo da transição $TARGET$.

5.1.5.2 Hot standby

Como explicado anteriormente, cada nó virtual n^V é alocado em dois nós físicos, um principal (primário) $M_N(n^V)$ e um redundante $M_{SN}(n^V)$. Além disso, cada enlace virtual $e_V(n^{V1}, n^{V2})$ é alocado em quatro caminhos físicos: (i) primário-primário $P^S(M_N(m^V), M_N(n^V))$; (ii) primário-redundante $P^S(M_N(m^V), M_{SN}(n^V))$; (iii) redundante-primário $P^S(M_{SN}(m^V), M_N(n^V))$

e (iv) redundante-redundante $P^S(M_{SN}(m^V), M_{SN}(n^V))$. Por exemplo, a Figura 31 mostra um cenário de alocação utilizando a política *hot standby*, e a Figura 32 apresenta o respectivo modelo para estimar a vazão orientada à disponibilidade.

O modelo resultante é similar ao obtido com a política sem redundância. Entretanto, a diferença fundamental entre os dois modelos está no fluxo de comunicação dos enlaces virtuais. Ao utilizar *hot standby*, quatro caminhos físicos estão disponíveis para a comunicação. Logo, cada caminho possível de envio de pacotes é explicitamente representado no modelo. Considerando λ a taxa de disparo de cada transição X_TARGET , a vazão orientada à disponibilidade é calculada através de $P\{(\#PP_NODE_B > 0) OR (\#PS_NODE_D > 0) OR (\#SP_NODE_B > 0) OR (\#SS_NODE_D > 0)\} \times \lambda$.

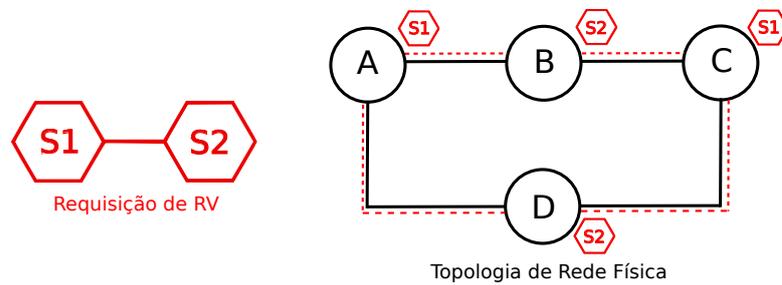


Figura 31 – Cenário de alocação com Hot standby

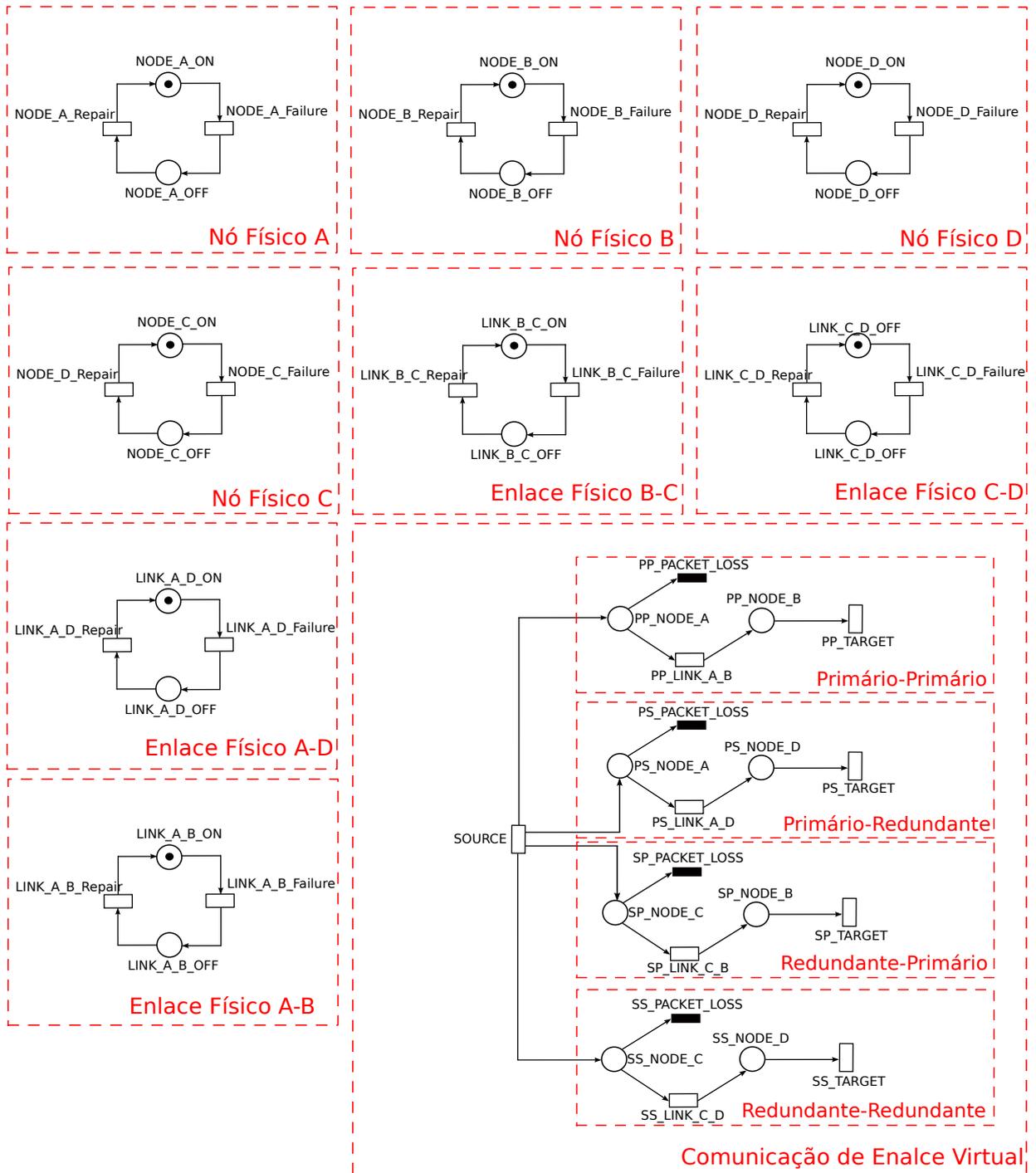


Figura 32 – Modelo de vazão orientada à disponibilidade hot standby

5.2 ALOCAÇÃO DE REDES VIRTUAIS BASEADO EM ANÁLISE DE SENSIBILIDADE

Esta seção apresenta o método de análise de sensibilidade adotado neste trabalho, chamado de Importância de Disponibilidade. Além disso, um exemplo é apresentado para um melhor entendimento.

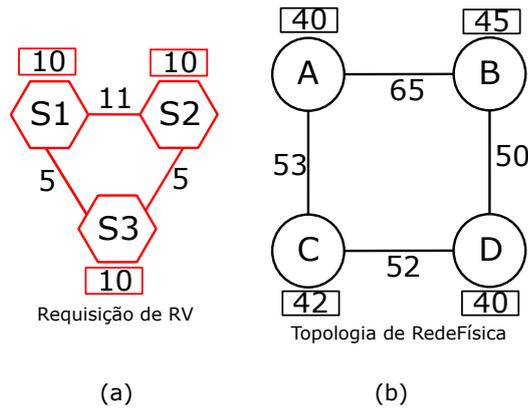


Figura 33 – Análise de sensibilidade - Exemplo

5.2.1 Importância de Disponibilidade

A importância dos componentes representa o impacto de cada componente na dependabilidade geral do sistema. Diversas técnicas são adotadas para estimar tal métrica (KUU; ZUO, 2003), e este trabalho adota a importância da disponibilidade para identificar o nó mais crítico para a disponibilidade da rede virtual. A importância da disponibilidade (AI) de um dado componente i (I_i^A) é a contribuição individual de i para a disponibilidade geral do sistema, e pode ser calculada da seguinte forma:

$$I_i^A = \frac{\partial A_s}{\partial a_i} = A_s(1_i, p) - A_s(0_i, p) \quad (5.1)$$

onde A_s denota a disponibilidade do sistema; a_i é a disponibilidade do componente i ; p é um vetor composto pelos valores de disponibilidade de cada componente, com exceção de i ; e 0_i e 1_i denotam o valor (0 ou 1) do componente i .

5.2.2 Exemplo

Considere a infraestrutura física e a requisição de rede virtual apresentados na Figura 33, onde a restrição de disponibilidade é igual a 0,99 (99%). Inicialmente, o algoritmo procura alocar a requisição utilizando a abordagem sem redundância, e a alocação resultante é:

Tabela 8 – Exemplo de Alocação com SA - 1ª Tentativa

<i>Componente Virtual</i>	<i>Notação</i>	<i>Alocação</i>
Nó $S1$	$M_N(S1)$	A
Nó $S2$	$M_N(S2)$	B
Nó $S3$	$M_N(S3)$	D
Caminho entre $S1$ principal e $S2$ principal	$M_{ME}(S1, S2)$	$\{(A, B)\}$
Caminho entre $S2$ principal e $S3$ principal	$M_{ME}(S2, S3)$	$\{(B, D)\}$
Caminho entre $S1$ principal e $S3$ principal	$M_{ME}(S1, S3)$	$\{(A, B), (B, D)\}$

Devido à uma combinação de valores de MTTF e MTTR adotados (não apresentados), a disponibilidade da rede é 98,77%. O respectivo modelo de dependabilidade é apresentado

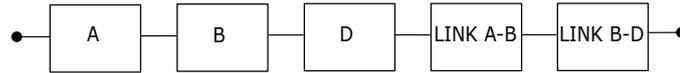


Figura 34 – Análise de sensibilidade - Primeiro modelo

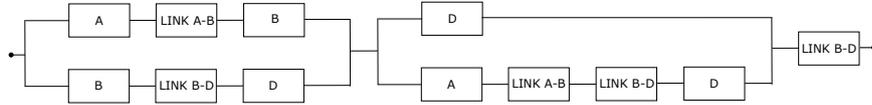


Figura 35 – Análise de sensibilidade - Segundo modelo

na Figura 34. Uma vez que a restrição de disponibilidade não foi satisfeita, a técnica de importância de disponibilidade é adotada para verificar qual é o nó com o maior impacto na disponibilidade do sistema (ou seja, o maior valor de AI). Nesse caso, *A* é o nó físico encontrado (no qual o nó virtual *S1* está alocado). Logo, o algoritmo adiciona *S1* à lista de nós redundantes, e em seguida, tenta alocar a requisição novamente. A alocação resultante é a apresentada na Tabela 9:

Tabela 9 – Exemplo de Alocação com SA - 2ª Tentativa

<i>Componente Virtual</i>	<i>Notação</i>	<i>Alocação</i>
Nó <i>S1</i>	$M_N(S1)$	<i>A</i>
Réplica do nó <i>S1</i>	$M_{SN}(S1)$	<i>D</i>
Nó <i>S2</i>	$M_N(S2)$	<i>B</i>
Nó <i>S3</i>	$M_N(S3)$	<i>D</i>
Caminho entre <i>S1</i> principal e <i>S2</i> principal	$M_{ME}(S1, S2)$	$\{(A, B)\}$
Caminho entre <i>S1</i> réplica e <i>S2</i> principal	$M_{SP}(S1, S2)$	$\{(B, D)\}$
Caminho entre <i>S2</i> principal e <i>S3</i> principal	$M_{ME}(S2, S3)$	$\{(B, D)\}$
Caminho entre <i>S1</i> principal e <i>S3</i> principal	$M_{ME}(S1, S3)$	$\{(A, B), (B, D)\}$
Caminho entre <i>S1</i> réplica e <i>S3</i> principal	$M_{SP}(S1, S3)$	$\{(\emptyset)\}$

Ao adotar a alocação supracitada, a disponibilidade da rede virtual é de 99,24%, e a restrição de disponibilidade é respeitada. Logo, o processo de alocação encerra. O modelo resultante é apresentado na Figura 35.

O método baseado em análise de sensibilidade inicia procurando por soluções com o menor custo energético possível, e, gradualmente, aplica redundância nos nós virtuais, até o momento que a disponibilidade requerida pelo cliente é atingida. Tal método permite a aplicação de redundância estrategicamente nos componentes, ao invés de aplicar ou não redundância em todos os nós desde o começo, o que poderia conduzir a custos desnecessariamente altos.

Como apresentado, a técnica proposta permite que seja aplicada a redundância nas redes virtuais somente quando necessário. Isso possibilita que o provedor de infraestrutura diminua os seus custos, uma vez que não serão alocados componentes redundantes desnecessários para o cliente, como aconteceria com as técnicas *hot* e *cold standby*. Por

outro lado, caso os clientes necessitem de uma rede com alta disponibilidade, a requisição não será rejeitada, como aconteceria com a técnica sem redundância. Logo, tal estratégia de alocação permite obter as vantagens de ambas as técnicas.

5.3 MERCURY E A API DE COMUNICAÇÃO

A ferramenta *Mercury* (SILVA et al., 2012) é o núcleo da ferramenta ASTRO, desenvolvida para permitir a avaliação de métricas de dependabilidade através do uso de modelos RBD e SPN (MARWAH et al., 2010). Além disso, a *Mercury* também suporta o uso de modelos de cadeias de Markov - MC para avaliação de desempenho e dependabilidade de sistemas. Contudo, apesar de a *Mercury* prover diversas funcionalidades, como simulações e análises, seu projeto inicial considera que os sistemas devem ser modelados por um usuário humano, através do uso dos editores gráficos da ferramenta.

Neste trabalho, foram implementadas extensões à ferramenta *Mercury*, para permitir que o processo de modelagem e avaliação de sistemas pela ferramenta seja automatizado. Basicamente, foi utilizada a arquitetura Cliente-Servidor (ECKERSON, 1995) para a construção de uma *Application Programming Interface (Interface de Programação de Aplicação)* (API) de comunicação com a *Mercury*. Na funcionalidade de comunicação implementada, a *Mercury* funciona como servidor, disponibilizando seus serviços através de *sockets*, utilizando o protocolo TCP-IP.

Para esta funcionalidade especificamente, a ferramenta funciona via linha de comando, sem interação via interface gráfica. Nesse caso, a *Mercury* deve ser executada com a passagem do argumento "remote" em sua execução. A Figura 36 apresenta o diagrama de sequência UML (*Unified Modeling Language*) (BOOCH; RUMBAUGH; JACOBSON, 1998) desta funcionalidade da ferramenta. Uma vez iniciada a *Mercury* em modo servidor, qualquer aplicação cliente com suporte à *sockets* TCP-IP pode solicitar uma conexão com a ferramenta. Após estabelecida a conexão entre a *Mercury* e o cliente, é iniciado o fluxo de comunicação via socket descrito a seguir:

1. Cliente envia à *Mercury* um número inteiro, que representa o tamanho em bytes do arquivo contendo a especificação do modelo a ser avaliado.
2. Cliente envia à *Mercury* a sequência de bytes correspondente ao arquivo o qual especifica o modelo em questão.
3. Cliente envia à *Mercury* um número inteiro, que representa o tamanho em bytes do arquivo contendo os parâmetros da avaliação.
4. Cliente envia à *Mercury* a sequência de bytes correspondente ao arquivo contendo os parâmetros.

5. *Mercury* processa internamente os arquivos contendo o modelo e os parâmetros, e avalia o modelo.
6. *Mercury* envia ao Cliente um número inteiro, que representa o tamanho em bytes do arquivo contendo os resultados.
7. *Mercury* envia ao Cliente a sequência de bytes correspondente ao arquivo contendo os resultados computados.

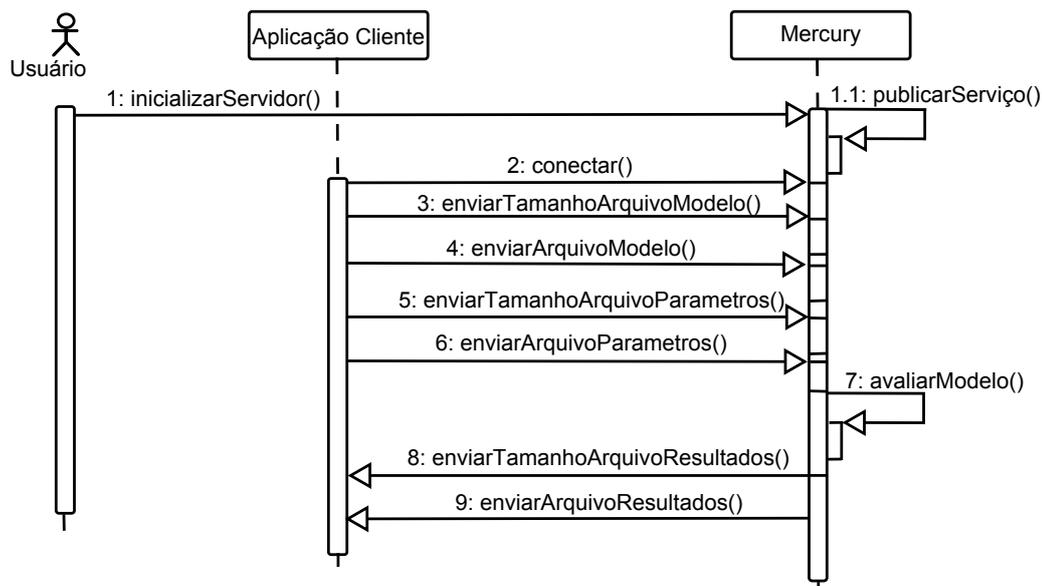


Figura 36 – Diagrama de Sequência UML - Comunicação remota com a Mercury

Conforme citado, a aplicação cliente deve basicamente enviar dois arquivos para que a *Mercury* possa avaliar o modelo, um com a especificação do modelo propriamente dito (SPN ou RBD), e o outro com os parâmetros necessários à avaliação. Esses parâmetros incluem o tipo de avaliação (exemplo, simulação transiente) e o tempo para o qual deseja-se obter as métricas.

O objetivo da construção da API de comunicação é possibilitar que, uma vez que o algoritmo proposto encontre soluções de alocação de redes virtuais, estas soluções sejam avaliadas automaticamente pela *Mercury*. Tal abordagem permite a verificação automática das métricas de dependabilidade nas redes virtuais, assim como o descarte das soluções que não atendam às necessidades dos clientes.

5.4 ALGORITMO DE ALOCAÇÃO

Este trabalho adota a metaheurística GRASP (Seção 2.6) para a alocação de requisições de redes virtualizadas, pois é uma abordagem largamente utilizada para resolução de problemas combinatoriais (RESENDE; RIBEIRO, 2001). Como explicado no Capítulo 2,

em termos gerais, GRASP contempla duas fases (Figura 37): construção e busca local. A fase de construção é responsável pela construção de uma solução aleatória, a qual deve satisfazer todas as restrições consideradas. Após a construção de uma solução viável, a fase de busca local investiga a solução ótima relacionada à solução inicial. A iteração principal do GRASP possui o tempo máximo de execução como critério de parada adotado neste trabalho.

```

procedure GRASP( $G^S = (N^S, E^S), G^V = (N^V, E^V)$ )
1  Melhor_Solucao  $\leftarrow \emptyset$ ;
2  while critérios de parada não foram satisfeitos do
3      Solucao  $\leftarrow$  GreedyRandomized( $G^S, G^V$ );
4      if Solucao  $\neq \emptyset$  then
5          Solucao  $\leftarrow$  LocalSearch(Solucao,  $G^S, G^V$ );
6          if  $f(\text{Solucao}) < f(\text{Melhor\_Solucao})$  then
7              Melhor_Solucao  $\leftarrow$  Solucao;
8          end if
9      end if
10 end while
11 if Melhor_Solucao =  $\emptyset$  then
12     Requisição de RV não pode ser satisfeita;
13 end if
14 return Melhor_Solucao;
end procedure

```

Figura 37 – Pseudo-código da metaheurística GRASP

A Figura 38 apresenta o pseudo-código da fase de construção. Cada nó virtual $n^V \in N^V$ da requisição é aleatoriamente alocado em um nó físico $M_N(n^V)$ com capacidade disponível suficiente para a alocação. Caso a redundância seja considerada, um segundo nó físico $M_{SN}(n^V)$ com capacidade suficiente é aleatoriamente selecionado, o qual será o componente redundante para n^V . Após a seleção dos nós, a fase de alocação dos enlaces é inicializada.

Para cada enlace virtual $e^V(i, j) \in E^V$, o menor caminho físico $M_{ME}(e^V)$ entre $M_N(i)$ e $M_N(j)$ é selecionado através do algoritmo de menor caminho de Dijkstra (onde o caminho físico deve possuir largura de banda disponível suficiente). Caso a alocação adote redundância, então os outros enlaces são alocados através do mesmo procedimento. Mais especificamente, três caminhos adicionais são alocados: $M_{SP}(e^V)$ (redundante-principal), $M_{PS}(e^V)$ (principal-redundante) e $M_{SS}(e^V)$ (redundante-redundante). Após a alocação dos recursos físicos, o custo total da solução é calculado, bem como suas métricas de dependabilidade.

A Figura 39 apresenta o pseudo-código da fase de busca local. Para cada nó virtual $n^V \in N^V$, é criada uma lista de todos os nós físicos vizinhos em relação a $M_N(n^V)$, contendo capacidade suficiente para alocar n^V . Para cada nó físico n^S com capacidade suficiente, uma nova solução é criada através da substituição de $M_N(n^V)$ por n^S (MACAMBIRA; MENESES, 1998). Caso uma nova solução viável seja encontrada através de tal substituição, e esta possua o custo menor que a solução original, o mesmo processo

```

procedure GreedyRandomized ( $G^S = (N^S, E^S), G^V = (N^V, E^V)$ )
1  for all  $n^V \in N^V$  do
2      Selecione um nó aleatório  $M_N(n^V)$  de  $N^S$ ;
3      if possui redundância
4          Selecione um nó aleatório  $M_{SN}(n^V)$  de  $N^S$ ;
5      end if
6  end for
7  for all  $e^V(i, j) \in E^V$  do
8      Selecione o menor caminho  $M_{ME}(e^V)$  entre  $M_N(i)$  e  $M_N(j)$ ;
9      if possui redundância
10         Selecione o menor caminho  $M_{SP}(e^V)$  entre  $M_{SN}(i)$  e  $M_N(j)$ ;
11         Selecione o menor caminho  $M_{PS}(e^V)$  entre  $M_N(i)$  e  $M_{SN}(j)$ ;
12         Selecione o menor caminho  $M_{SS}(e^V)$  entre  $M_{SN}(i)$  e  $M_{SN}(j)$ ;
13     end if
14 end for
15 Solucao  $\leftarrow \{ M_N, M_{SN}, M_{NE}, M_{SP}, M_{PS}, M_{SS} \}$ ;
16 Calcula o custo de Solucao;
17 translate_vn( $G^S, G^V$ );
18 Calcula as métricas de dependabilidade de Solucao;
19 if Solucao não satisfaz restrições then
20     return  $\emptyset$ ;
21 end if
22 return Solucao;
end procedure

```

Figura 38 – Pseudo-código da fase de construção

de busca local é repetido para a nova solução. Após esse passo, caso haja redundância, o processo (de investigação de nós vizinhos) é repetido para todos os nós físicos redundantes ($M_{SN}(n^V)$) da alocação.

```

procedure LocalSearch(Solucao,  $G^S = (N^S, E^S), G^V = (N^V, E^V)$ )
1  for all  $M_N(n^V) \in N^V$  do
2      Vizinhos  $\leftarrow$  Selecione todos os nós vizinhos à  $M_N(n^V)$ ;
3      for all  $n^S \in$  Vizinhos do
4          Solucao_Local  $\leftarrow$  Construa uma solução válida substituindo  $M_N(n^V)$  por  $n^S$ ;
5          if  $f(\text{Solucao\_Local}) < f(\text{Solucao})$  then
6              return LocalSearch(Solucao_Local,  $G^S, G^V$ );
7          end if
8      end for
9      if possui redundância
10         Vizinhos_Replica  $\leftarrow$  Selecione todos os nós vizinhos à  $M_{SN}(n^V)$ ;
11         for all  $n^S \in$  Vizinhos_Replica do
12             Solucao_Local  $\leftarrow$  Construa uma solução válida substituindo  $M_{SN}(n^V)$  por  $n^S$ ;
13             if  $f(\text{Solucao\_Local}) < f(\text{Solucao})$  then
14                 return LocalSearch(Solucao_Local,  $G^S, G^V$ );
15             end if
16         end for
17     end if
18 end for
19 return Solucao;
end procedure

```

Figura 39 – Pseudo-código da fase de busca local

5.5 CONVERSÃO DE REDES VIRTUAIS EM MODELOS DE DISPONIBILIDADE

Esta seção apresenta os algoritmos adotados para tradução de redes virtuais nos respectivos modelos de disponibilidade adotados. Inicialmente, é executado o algoritmo de triagem (Figura 40), responsável por escolher o tipo de modelo a ser criado a partir da rede virtual.

```

procedure translate_vn( $G^S = (N^S, E^S)$ ,  $G^V = (N^V, E^V)$ )
1  if  $G^V$  não utiliza redundância then
2    translate_vn_noRedundancy( $G^S$ ,  $G^V$ );
3  else if  $G^V$  utiliza redundância hot standby then
4    translate_vn_hotStandby( $G^S$ ,  $G^V$ );
5  else then
6    translate_vn_coldStandby( $G^S$ ,  $G^V$ );
7  end if
end procedure

```

Figura 40 – Tradução de rede virtual em modelo de dependabilidade

Conforme apresentado na Figura 40, há três fluxos possíveis para o processo de tradução de redes virtuais em modelos de disponibilidade. O primeiro é dado quando a rede virtual é criada sem redundância. Nesse caso, traduz-se a rede virtual em um modelo RBD (*translate_vn_noRedundancy*), através de composição em série. O segundo caso ocorre quando a rede virtual utiliza redundância *hot standby*, e é gerado um modelo RBD através de arranjos em série e paralelo (*translate_vn_hotStandby*). Finalmente, caso o tipo de redundância utilizada seja *cold standby*, o formalismo SPN é adotado para representação da rede virtual (*translate_vn_coldStandby*).

5.5.1 Sem Redundância

Nesta subseção será apresentado o algoritmo de tradução de uma rede virtual sem redundância no modelo de disponibilidade correspondente. Conforme citado na Seção 5.1.3.1, para redes virtuais sem redundância o formalismo RBD é adotado. A Figura 41 apresenta o algoritmo de tradução de uma rede virtual no modelo RBD equivalente. Existe uma iteração que percorre os nós $n^V \in N^V$ da rede virtual. Então, cada nó físico $M_N(n^V)$ em que os nós virtuais foram alocados é adicionado no modelo (variável *model*). Ou seja, todos os nós físicos utilizados na alocação da rede virtual são dispostos em um arranjo em série.

Similarmente, para cada enlace $e^V(i, j) \in E^V$ da requisição, percorrem-se todos os enlaces físicos p utilizados na alocação $M_{ME}(e^V)$, inserindo (em série) os que ainda não estiverem contidos no modelo RBD.

Ao fim do processo descrito, têm-se um modelo RBD composto por todos os componentes da infraestrutura física utilizados na alocação da requisição.

```

procedure translate_vn_noRedundancy( $G^S = (N^S, E^S), G^V = (N^V, E^V)$ )
1  var model;
2
3  for all  $n^V \in N^V$  do
4      if model.naoContem( $M_N(n^V)$ ) then
5          model.inserirEmSerie( $M_N(n^V)$ );
6      end if
7  end for
8  for all  $e^V(i, j) \in E^V$  do
9      for all  $p \in M_{ME}(e^V)$  do
10         if model.naoContem( $p$ ) then
11             model.inserirEmSerie( $p$ );
12         end if
13     end for
14 end for
15
16 return model;
end procedure

```

Figura 41 – Tradução de rede virtual sem redundância em modelo RBD

5.5.2 Hot Standby

O algoritmo para tradução de uma rede virtual com *hot standby* no modelo de disponibilidade RBD equivalente é apresentado na Figura 42. Similarmente, a variável *model* representa o modelo de RBD final.

Basicamente, existe uma iteração (linha 3) responsável por percorrer cada enlace virtual (variável *modelE*) e incluí-lo no modelo final. É importante lembrar que, no caso das redes virtuais com redundância, cada enlace virtual é alocado em quatro caminhos físicos: (i) M_{ME} ; (ii) M_{SP} ; (iii) M_{PS} ; e (iv) M_{SS} (ver Seção 4.2.4.2). Além disso, uma vez que para o enlace virtual estar operacional, pelo menos um dos quatro caminhos físicos deve estar disponível e esses caminhos são dispostos em um arranjo em paralelo. Ou seja, o submodelo de cada enlace virtual (variável *modelE*) é composto por um arranjo em série de *modelME*, *modelSP*, *modelPS* e *modelSS*.

Por fim, é importante mencionar que, cada um dos quatro submodelos que compõem *modelE*, é formado pela composição em série dos componentes físicos em questão. Por exemplo, o submodelo *modelME*, o qual representa $M_{ME}(e^V(i, j))$, é formado por um arranjo em série com os nós físicos $M_N(i)$ e $M_N(j)$ e enlaces físicos que compõem o caminho entre esses nós.

5.5.3 Cold Standby

A Figura 43 apresenta o algoritmo responsável pela conversão de uma rede virtualizada com redundância *cold standby* no modelo de disponibilidade proposto.

Primeiramente, o algoritmo percorre os nós virtuais $n^V \in N^V$ da requisição. Para cada nó virtual, existe uma chamada a *insertColdStandbyModel*, responsável por inserir um componente *cold standby* (Figura 18) no modelo SPN. Em seguida, cada enlace virtual

```

procedure translate_vn_hotStandby( $G^S = (N^S, E^S), G^V = (N^V, E^V)$ )
1  var model;
2
3  for all  $e^V(i, j) \in E^V$  do
4    var modelE;
5
6    for all  $m \in (M_{ME} \cup M_{SP} \cup M_{PS} \cup M_{SS})$  do
7      var model_m;
8      model_m.inserirEmSerie(  $i$  );
9      model_m.inserirEmSerie(  $j$  );
10     for all  $p \in m(e^V(i, j))$  do
11       model_m.inserirEmSerie(  $p$ );
12     end for
13     modelE.inserirEmParalelo(model_m);
14   end for
15   model.inserirEmSerie(modelE);
16 end for
17
18 return model;
end procedure

```

Figura 42 – Tradução de rede virtual com hot standby em modelo RBD

$e^V(i, j)$ é incluído no modelo como um componente simples (ver Figura 17).

```

procedure translate_vn_coldStandby( $G^S = (N^S, E^S), G^V = (N^V, E^V)$ )
1  var model;
2
3  for all  $n^V \in N^V$  do
4    model.inserirModeloColdStandby(  $M_N(n^V), M_{SN}(n^V)$ );
5  end if
6 end for
7 for all  $e^V(i, j) \in E^V$  do
8   for all  $p \in M_{ME}(e^V)$  do
9     if model.naoContem(  $p$  ) then
10      model.inserirComponenteSimples(  $p$ );
11    end if
12  end for
13  for all  $p \in M_{SP}(e^V)$  do
14    if model.naoContem(  $p$  ) then
15      model.inserirComponenteSimples(  $p$ );
16    end if
17  end for
18  for all  $p \in M_{PS}(e^V)$  do
19    if model.naoContem(  $p$  ) then
20      model.inserirComponenteSimples(  $p$ );
21    end if
22  end for
23  for all  $p \in M_{SS}(e^V)$  do
24    if model.naoContem(  $p$  ) then
25      model.inserirComponenteSimples(  $p$ );
26    end if
27  end for
28 end for
29
30 return model;
end procedure

```

Figura 43 – Tradução de rede virtual com cold standby em modelo SPN

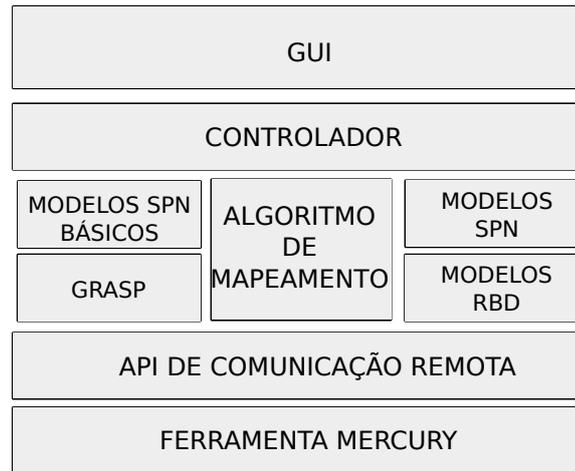


Figura 44 – Arquitetura geral da ferramenta

5.6 VISÃO GERAL DA FERRAMENTA

Esta seção apresenta uma visão geral da MODERADOR - uma ferramenta para alocação de redes virtuais confiáveis de computadores considerando disponibilidade e consumo de energia. Inicialmente, a arquitetura geral da ferramenta é apresentada. Posteriormente, é demonstrada a usabilidade da ferramenta para a alocação das redes virtuais.

5.6.1 Arquitetura da Ferramenta

MODERADOR é um ambiente integrado que permite a alocação de redes virtuais confiáveis de computadores, considerando métricas de desempenho, dependabilidade e consumo de energia. A ferramenta adota os modelos RBD e SPN apresentados nesta tese para avaliar de forma automatizada as redes virtuais. Além disso, o algoritmo GRASP apresentado é utilizado para realizar as alocações das requisições. A Figura 44 apresenta a estrutura geral da ferramenta, a qual foi escrita na linguagem de programação Java. Consequentemente, a ferramenta é portátil e pode ser executada em qualquer ambiente que possua a Máquina Virtual Java instalada.

A interface gráfica da ferramenta permite que o usuário envie os arquivos contendo as especificações de topologia da rede física, bem como das requisições de redes virtuais a serem alocadas. O controlador da ferramenta é responsável por coordenar o processo de alocação, utilizando o algoritmo que implementa a metaheurística GRASP. Para cada solução de alocação encontrada, são utilizados os modelos básicos SPN, bem como os modelos de alto nível RBD e SPN para representação da rede virtual. Em seguida, a API de comunicação remota é adotada para que a ferramenta Mercury avalie o modelo e estime as métricas de dependabilidade da rede virtual. Finalmente, a solução de alocação encontrada é apresentada na interface gráfica para o usuário.

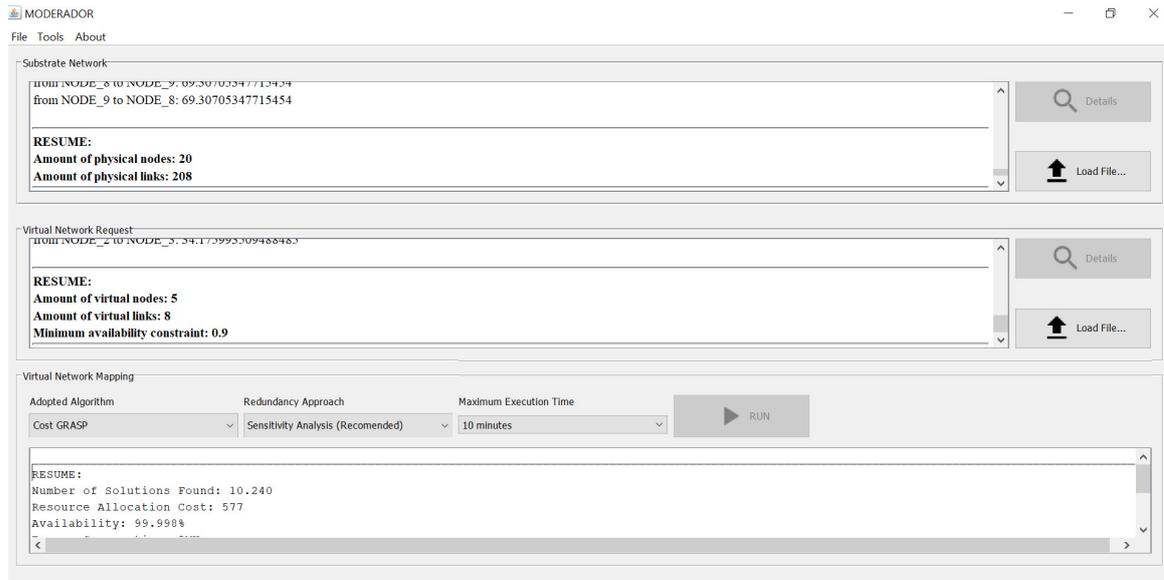


Figura 45 – Captura de tela ferramenta MODERADOR

5.6.2 Ambiente de Alocação de Redes Virtuais

Uma captura de tela da ferramenta é apresentada na Figura 45, a qual é composta por três painéis principais. O painel superior é utilizado para que usuário envie o arquivo contendo a especificação da topologia de rede física a ser adotada. Essa especificação deve conter os nós físicos que compõem a rede, suas respectivas capacidades, bem como os enlaces físicos e as larguras de banda associadas. Analogamente, o painel intermediário da ferramenta representa as redes virtuais a serem alocadas. Nele, é possível que o usuário selecione o arquivo que contém a especificação das requisições de redes virtuais, com as informações dos nós e enlaces virtuais a serem adotados. Finalmente, o painel inferior da ferramenta permite que o usuário selecione as configurações de alocação desejadas e execute o algoritmo de alocação.

A ferramenta permite que o usuário selecione o algoritmo desejado, a abordagem de redundância, bem como o tempo máximo de execução a ser utilizado. Em relação ao algoritmo, duas opções estão disponíveis: (i) Cost GRASP e (ii) GRASP Energy-Aware. A diferença entre os dois está na métrica que o algoritmo visa otimizar, que pode ser o custo ou o consumo de energia, respectivamente. As abordagens de redundância disponíveis são: sem redundância, *hot standby*, *cold standby* e utilizar análise de sensibilidade. O tempo máximo de execução do algoritmo varia entre 1 minuto e 10 horas. Após o fim da execução do algoritmo, a solução de alocação encontrada é apresentada ao usuário, bem como os respectivos valores das métricas de desempenho, dependabilidade e consumo de energia.

Através do uso da MODERADOR, gestores, administradores de rede e usuários não especializados podem se utilizar de todas as técnicas desenvolvidas neste trabalho para projetar redes virtuais de computadores considerando diferentes métricas de desempenho, dependabilidade e consumo de energia.

5.7 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os modelos SPN e RBD propostos para avaliação de dependabilidade em redes virtualizadas, bem como o algoritmo de alocação adotado. Primeiramente, os componentes básicos dos modelos finais foram apresentados individualmente. Posteriormente, os modelos para redes virtuais sem redundância, e com redundância *hot* e *cold standby* foram descritos. Em seguida, o algoritmo de alocação proposto foi apresentado, o qual se baseia na metaheurística GRASP para realizar as alocações. Em seguida, foram apresentados os algoritmos de tradução de redes virtuais para os modelos de dependabilidade. Por fim, este capítulo apresentou a MODERADOR - uma ferramenta para a alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia. Foi apresentada a arquitetura geral da ferramenta e explicada a interface gráfica da ferramenta e seus principais componentes.

6 RESULTADOS EXPERIMENTAIS

Este capítulo apresenta a validação do modelo energético proposto, bem como dois estudos de caso. No primeiro estudo de caso, foram gerados dados de entrada para o algoritmo de alocação proposto, com o objetivo de verificar o comportamento do sistema. Foram coletadas as métricas de consumo de energia, custo de energia acumulado, disponibilidade, custo, receita e taxa de aceitação, considerando restrições de disponibilidade em diferentes faixas de acordos de níveis de serviço (SLA - *Service Level Agreement*). Por fim, o segundo estudo de caso analisa o impacto da distância do caminho físico dos enlaces virtuais em diferentes métricas de desempenho e dependabilidade, como a vazão orientada à disponibilidade e a taxa de perda de pacotes.

6.1 VALIDAÇÃO DO MODELO DE CONSUMO ENERGÉTICO

Esta seção apresenta os experimentos conduzidos para validar o modelo de energia proposto, bem como estimar os valores de α e u_{PP} . A Figura 46 apresenta a arquitetura de medição de consumo de energia adotada no experimento, a qual é explicada detalhadamente nas próximas seções.

6.1.1 Framework de Medição

Neste trabalho, foi concebido um *framework* de medição chamado E-meter, com o objetivo de coletar os dados de consumo energético em um ambiente de virtualização de redes. O *framework* E-meter contempla componentes de hardware e ferramentas de software utilizadas para armazenar e visualizar as métricas coletadas. Além disso, o *framework* adota uma máquina monitora, na qual o software E-meter executa e se comunica com um hardware EVM430-F6736 (Texas-Instruments 2015). O hardware adotado é um dispositivo especializado, o qual coleta a corrente instantânea e a voltagem utilizados para estimar o consumo de energia elétrica instantâneo. O consumo energético é calculado através de uma integração numérica, adotando a potência instantânea e o intervalo de tempo da execução da rede virtual. O *firmware* existente no EVM430-F6736 foi modificado, de forma a possibilitar a comunicação e coleta dos dados requeridos através de uma interface serial. Para um melhor controle e gerenciamento, o E-meter possui uma interface gráfica (GUI), na qual os dados de consumo podem ser visualizados em tempo real. Finalmente, os dados coletados são armazenados em um banco de dados ou em um arquivo csv. Adicionalmente, o E-meter suporta controle remoto via comunicação TCP/IP, o que permite que todos os experimentos possam ser controlados externamente.

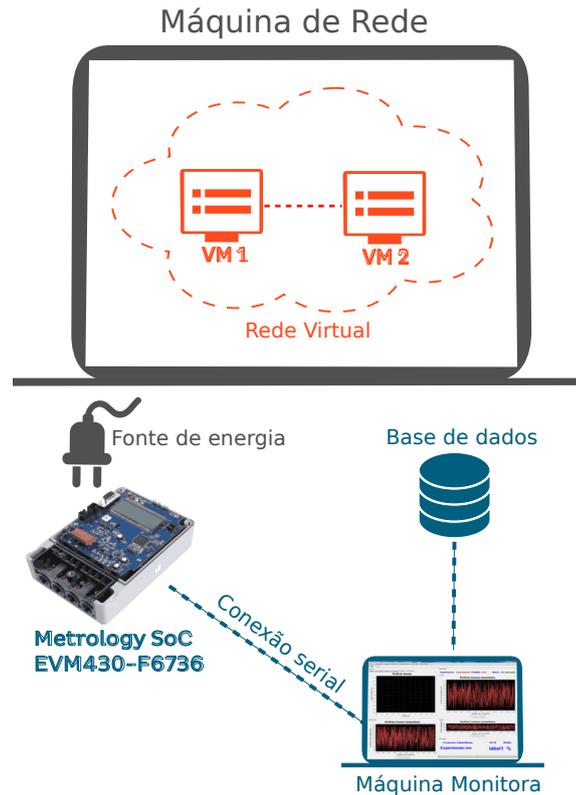


Figura 46 – Arquitetura de validação do modelo de consumo de energia

Tabela 10 – Caracterização dos nós físicos

<i>Propriedade</i>	<i>Valor</i>
CPU	i5 3427U @ 1.80GHz
Disco Rígido (HD)	465GB
Memória	8,00GB
Sistema Operacional (SO)	Ubuntu 15.10 64 bits
Software de Virtualização (VMM)	VMware Workstation Player 12

6.1.2 Resultados da Medição

Duas alocações diferentes de redes virtuais foram considerados para estimar α e u_{PP} , considerando as especificações físicas e virtuais apresentadas nas Tabelas 10 e 11, respectivamente. É importante observar que os Cenários 1 e 2 são similares, sendo compostos por um único nó virtual alocado em um único nó físico. A diferença entre os cenários é a quantidade de recursos adotados para o nó virtual. Para este trabalho, nós assumimos que os nós virtuais variam apenas na quantidade de memória adotada, e as outras características de processamento (ex: CPU) são mantidas equivalentes. O Cenário 1 assume um nó virtual com 1 GB de memória, enquanto o Cenário 2 considera um nó virtual com 2 GB. Além disso, o nó virtual possui um *workload* o qual aloca memória até que toda a memória disponível esteja sendo utilizada. Os resultados obtidos são apresentados na Tabela 12.

Tabela 11 – Caracterização dos nós virtuais

<i>Propriedade</i>	<i>Valor</i>
CPUs	1
Disco Rígido (HD)	20GB
Sistema Operacional	Ubuntu 14.04.3 LTS 32 bits

Tabela 12 – Resultados da medição - Cenários 1 e 2

<i>Cenário</i>	<i>Amostras</i>	<i>Média</i>	<i>Desvio Padrão</i>	<i>I. C. - 95%</i>
Alocação 1	300	15,044 Watts	0,428	(14,996W; 15,093W)
Alocação 2	300	16,175 Watts	2,790	(15,937W; 16,421W)

Tabela 13 – RESULTADO DO TESTES ANOVA - Cenários 1 e 2

<i>F</i>	<i>Valor de P</i>	<i>F (Tabulado)</i>
82,963	1,24475E-18	3,857

u_{PP} é calculado pela diferença das médias nas alocações 1 e 2, o que é igual a $16,175W - 15,044W = 1,131W$. Mais especificamente, u_{PP} é o consumo de energia variável do nó. Tendo em vista que as alocações 1 e 2 se diferenciam apenas na quantidade de memória alocada (diferença igual a 1 GB), nós assumimos que cada 1 GB requerido por um nó virtual corresponde a $1,131W$ de consumo energético adicional. Além disso, nós atribuímos a α , variável que representa o consumo de energia base de um nó ativo, o resultado da diferença de médias na alocação 1 e u_{PP} , ou seja: $15,044W - 1,131W = 13,913W$. É importante mencionar que, de acordo com resultados obtidos utilizando o teste ANOVA (apresentados na Tabela 13) com nível de significância de 0,05, há uma diferença significativa entre os resultados obtidos nas duas alocações. Haja vista que o único parâmetro de diferenciação utilizado nos dois cenários é a quantidade de memória alocada e os resultados obtidos são estatisticamente diferentes, nós assumimos que o acréscimo do consumo de energia acontece devido à quantidade de capacidade adicional alocada no segundo cenário.

Por fim, uma terceira alocação de requisição de redes virtuais (G_V) é utilizado para a validação do modelo de energia proposto. Considerando os valores obtidos nas alocações 1 e 2, os seguintes passos foram executados:

1. O consumo de energia instantâneo de G_V ($Pw_u(G_V)$) é estimado através da Equação 4.16;
2. O consumo de energia de G_V ($\Delta ECN(G_V)$) é estimado utilizando a Equação 4.20;
e
3. Em seguida, o consumo de energia real é medido na infraestrutura física (ou seja, máquina de rede) e os valores resultantes são comparados com os obtidos no passo 2.

Tabela 14 – Resultados da medição - Cenário 3

<i>Amostras</i>	<i>Média</i>	<i>Desvio padrão</i>	<i>Intervalo de confiança - 95%</i>
50	1534,5J	219,2	(1472,2J; 1596,8J)

G_V é composta por dois nós virtuais conectados, com 1 GB e 2 GB de memória. Além disso, G_V possui um tempo de vida de 90 segundos. O consumo de energia instantâneo é $Pw_u(G_V) = 13,913W + 1,131W \times 3 = 17,306W$, logo, o consumo de energia é obtido através de $\Delta ECN(G_V) = 17,306W \times 90s = 1.557,54J$. Finalmente, uma amostra de 50 dados é coletada na máquina de rede, e os resultados são apresentados na Tabela 14. O intervalo de confiança dos valores medidos contempla o valor estimado de consumo de energia, o que sugere que o modelo de energia proposto é viável. Logo, tal modelo é adotado na próxima seção para estimar o consumo de energia para diferentes técnicas de alocação de redes virtuais.

6.2 COMPARAÇÃO DAS POLÍTICAS DE ALOCAÇÃO

Esta seção apresenta a aplicabilidade da abordagem proposta, a qual adota análise de sensibilidade (SA). Mais especificamente, a técnica proposta provê resultados melhores que as políticas de alocação sem redundância, *hot* e *cold standby*. Além disso, é importante mencionar que nos experimentos realizados, os nós alocados com redundância pela política SA adotam *hot standby*.

As métricas de interesse são o custo de alocação de recursos, o consumo de energia, o custo da energia elétrica, a receita, a disponibilidade e a taxa de aceitação. Tais métricas são especialmente importantes uma vez que demonstram o comportamento de cada política em termos de desempenho, dependabilidade e consumo de energia.

6.2.1 Configurações do Experimento

Este estudo de caso adotou topologias de rede sintéticas geradas pela ferramenta GT-ITM (ZEGURA; CALVERT; BHATTACHARJEE, 1996), considerando a mesma metodologia. Isso deve-se ao fato de que tal abordagem tem sido adotada por diversos trabalhos como forma de avaliação de algoritmos de alocação de redes virtuais (RAHMAN; AIB; BOUTABA, 2010), (CHOWDHRY; RAHMAN; BOUTABA, 2009), (SZETO; IRAQI; BOUTABA, 2003). Mais especificamente, a rede física gerada consiste de 50 nós randomicamente conectados com probabilidade igual a 0.5. As capacidades dos nós, bem como as larguras de banda dos enlaces são números reais uniformemente distribuídos entre 50 e 100.

Além disso, cerca de 300 requisições são consideradas durante um período de 2 anos, onde cada requisição possui o tempo de vida exponencialmente distribuído com a média igual a 1.000 horas. O número de nós virtuais em cada requisição segue uma distribuição

uniformemente variada entre 2 e 10, e suas respectivas capacidades são números reais uniformemente distribuídos entre 0 e 20, expressas em termos GB de memória (os nós virtuais adotam a mesma capacidade de CPU). As larguras de banda dos enlaces variam uniformemente entre 0 e 50Mbps. O tempo máximo de execução para o algoritmo de alocação (critério de parada para encontrar uma solução) é de 10 minutos. Tal critério de parada foi adotado baseado em observações experimentais, uma vez que tal valor é relativamente pequeno quando comparado ao tempo de vida das requisições de redes virtuais (1.000 horas), e grande o suficiente para permitir ao algoritmo encontrar um grande número de possíveis soluções durante a sua execução.

Para os experimentos, foi considerado que cada nó é composto por CPU, *Hard Disk (Disco Rígido)* (HD), memória, cartão de interface de rede, sistema operacional (OS), VMM (programa de virtualização) e VM (máquina virtual). Além disso, um enlace contempla um switch (assumindo que este pode criar enlaces virtuais), cabo de fibra óptica, VMM e VM. A Tabela 15 apresenta os valores de MTTFs e MTTRs adotados em horas (FERNANDES et al., 2012). Cada nó e enlace físico está associado a um número aleatório entre 35 e 100, o qual representa a porcentagem a ser adotada em relação aos MTTFs do componente. Por outro lado, os valores dos MTTRs são os mesmo para todos os nós e enlaces da rede.

Para associar os valores dos MTTFs e MTTRs aos nós e enlaces, é adotado o processo apresentado na Seção 5.1.3. Inicialmente, todos os componentes que compõem o nó ou enlace são dispostos em um arranjo RBD em série, com os respectivos MTTFs e MTTRs. Então, aplica-se uma redução do modelo em um único componente. Posteriormente, os valores resultantes representam o MTTF e MTTR do nó ou enlace em questão. Para propósito de comparação (não por limitação), adota-se a distribuição exponencial para os MTTFs e MTTRs.

Tabela 15 – MTTFs E MTTRs DOS NÓS E ENLACES FÍSICOS

Componente	MTTF(h)	MTTR(h)
CPU	2500000	1
Disco Rígido	200000	1
Memória	480000	1
Cartão de Interface de Rede	6200000	1
Sistema Operacional	1440	2
Máquina Virtual (VM)	2880	2
Monitor de VM	2880	2
Switch/Roteador	320000	1
Enlace Óptico	19996	12

Além disso, são utilizados os valores obtidos na seção anterior para as variáveis α_n e u_{PP} . O valor utilizado para β_e é 1,1W, baseado nos experimentos conduzidos em (CHABAREK et al., 2008) e (SIVARAMAN et al., 2011). Para estimar o custo da energia elétrica, foram assumidos os preços da eletricidade entre 1 de Junho de 2014 e 1 de Junho

de 2016, obtidos da capital de Nova Iorque (NYISO), Estados Unidos (disponíveis publicamente em <<http://www.ferc.gov>>). A janela de intervalo de tempo é de 1 hora, já que está é a unidade de variação do preço da energia elétrica nos dados reais adotados.

6.2.2 Resultados

Foram coletadas as seguintes métricas de consumo de energia, custo de energia acumulado, disponibilidade, custo, receita e taxa de aceitação.

As Figuras 47 até 57 apresentam os resultados obtidos. O teste ANOVA foi aplicado para as amostras e os resultados, apresentados na Tabela 16, indicam a existência de uma diferença significativa nos resultados provido por cada política e métrica. Uma análise detalhada é provida a seguir, utilizando como base resultados obtidos com o teste Tukey (WALLERSTEIN; LUCKER; FLIN, 1980).

Tabela 16 – RESULTADOS DO TESTE ANOVA

<i>Métrica</i>	<i>SLA</i>	<i>F</i>	<i>Valor de P</i>	<i>F (Tabulado)</i>
Disponibilidade	-	173,329	2,086E-71	2,258
Consumo de energia	-	22,069	1,183E-17	2,258
Custo	-	243,874	1,617E-54	2,671
Taxa de aceitação	[0,9 até 0,95]	107,033	1,246E-35	2,671
Taxa de aceitação	[0,95 até 0,99]	68,547	4,577E-27	2,671
Taxa de aceitação	[0,99 até 0,9999999]	75,807	6,980E-29	2,671

Disponibilidade. A Figura 47 apresenta os resultados de disponibilidade utilizando distribuição cumulativa empírica (ECDF). Esse tipo de gráfico foi adotado pelo fato de que diversos outros trabalhos da área de alocação de redes virtuais apresentam as métricas utilizando ECDF (por exemplo, CHOWDHURY; RAHMAN; BOUTABA, 2009). É importante mencionar que, nas figuras apresentadas utilizando ECDF, o eixo y ($P\{X \leq x\}$) representa a probabilidade de que uma rede virtual possua, para a métrica apresentada, um valor menor ou igual ao que é apresentado no eixo x . Ou seja, na Figura 47 pode-se observar que uma rede virtual alocada utilizando *cold standby* possui 100% de probabilidade de ter uma disponibilidade inferior a 98%. Por outro lado, uma rede alocada utilizando a política sem redundância possui cerca de 20% de probabilidade de ter uma disponibilidade inferior a 97%. Além disso, uma rede virtual alocada utilizando *hot standby* possui 0% de probabilidade de entregar uma disponibilidade inferior a 99%. O experimento adota uma restrição de disponibilidade entre 0,9 (90%) e 0,95 (95%), uma vez que nesse cenário todas as políticas estão aptas a produzir resultados viáveis. Para a política a qual utiliza análise de sensibilidade (SA), consideramos 2 acordos de nível de serviço (SLA) adicionais: [0,95 até 0,99]; e [0,99 até 0,9999999]. A faixa de SLA indica que as requisições de redes virtuais possuem uma restrição de disponibilidade dentro de tal faixa de valores. A abordagem SA é adaptativa, o que significa que a técnica de análise de importância de disponibilidade só

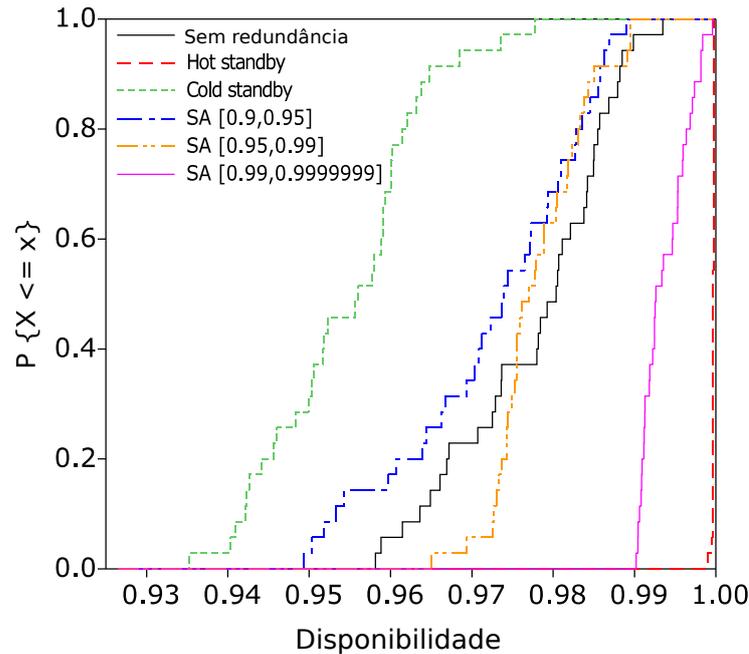


Figura 47 – Resultados de disponibilidade

é utilizada nos casos em que o algoritmo de alocação não está apto a encontrar soluções viáveis. Em outras palavras, a abordagem SA apenas considera redundância caso esta seja de fato necessária para satisfazer as restrições de disponibilidade, e na medida que a redundância é necessária (o que minimiza os custos relacionados à alocação de recursos físicos). Por outro lado, para as políticas sem redundância, *hot* e *cold standby*, as variações nos valores de restrição de disponibilidade não sugerem que, necessariamente, os algoritmos produzam melhores soluções. Por exemplo, a abordagem com *hot standby* considera sempre redundância para todos os componentes, o que afeta negativamente a receita e o custo das alocações, como mostrado a seguir.

Os resultados mostram que a técnica de *hot standby* conduz à alocação de redes virtuais com as maiores disponibilidades, seguida pela abordagem SA adotando o SLA [0,99 até 0,9999999]. A técnica de *cold standby*, por outro lado, possui os menores valores de disponibilidade, uma vez que em tal técnica os nós redundantes precisam ser alocados em nós físicos os quais não são adotados por qualquer outro nó virtual. Mais especificamente, as alocações de redes virtuais iniciais possuem uma disponibilidade alta, contudo as alocações subsequentes adotam componentes até então inutilizados e que são menos confiáveis. Essa particularidade também afeta negativamente a taxa de aceitação da política *cold standby*. Além disso, todas as outras políticas comparadas não proveem soluções que são estatisticamente diferentes em relação à disponibilidade.

Para melhor visualização do impacto da disponibilidade na qualidade do serviço (QoS), é apresentado o respectivo *downtime* (tempo *offline*) em um ano, o qual é calculado através de $downtime = (1 - availability) \times 24 \times 30 \times 12$. A Figura 48 apresenta os resultados obtidos, e pode-se observar que a abordagem *hot standby* provê redes virtuais

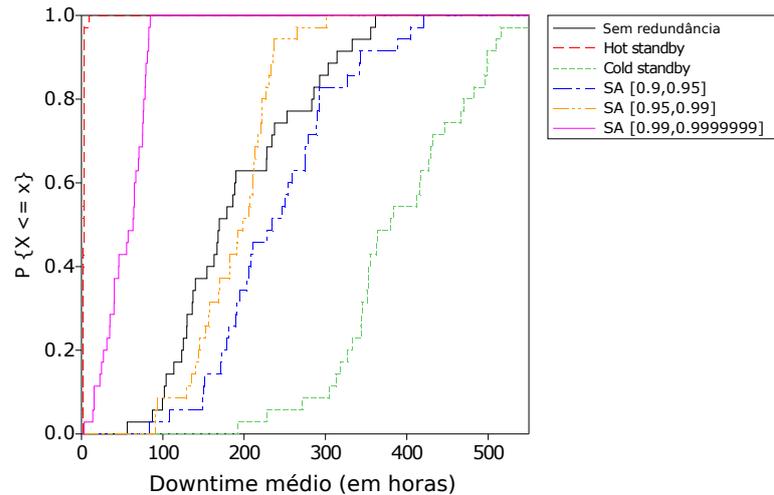


Figura 48 – Resultados de downtime

com a média de *downtime* de 10 horas em um ano. Do outro lado, a abordagem sem redundância, por exemplo, conduz a centenas de horas de *downtime*, impactando o QoS negativamente. Serviços críticos, os quais podem requerer alta disponibilidade para o seu correto funcionamento, podem ser consideravelmente impactados.

Eletricidade, Custo de Alocação de Recursos e Receita.

A Figura 49 apresenta os resultados do preço acumulado da energia elétrica. Para os valores observados, as abordagens *hot* e *cold standby* proveem os maiores valores acumulados de custos com energia devido aos componentes redundantes. Uma análise separada (utilizando o teste ANOVA) para as abordagens sem redundância e SA indicam que não há diferença significativa entre os resultados obtidos com essas duas políticas. O consumo médio de energia, apresentado na Figura 50, demonstra que a política *cold standby* possui os maiores valores. Isso deve-se ao fato dos nós redundantes serem alocados em nós exclusivos. Para o SLA [0,99 até 0,9999999], a política SA possui o mesmo consumo de energia quando comparada com a política de *hot standby*. As políticas sem redundância e SA possuem valores de consumo energético equivalentes para as outras faixas de SLA, uma vez que a abordagem SA evita a utilização de recursos desnecessários para satisfazer as restrições de disponibilidade.

As Figuras 51 a 54 apresentam os valores médios para o custo de alocação de recursos e receita acumulada. Para este experimento, ambas as métricas são expressas em termos de \$1, representando o preço de 1Mb de largura de banda de um enlace ou 1 GB de memória de um nó. O custo da técnica SA é menor que o das políticas *hot* e *cold standby*. A política sem redundância possui os menores valores de custo. Contudo, apesar da política sem redundância prover custos reduzidos, tal abordagem não é viável para alocar redes virtuais as quais necessitem de SLA na faixa [0,99 até 0,9999999], devido à ausência de componentes redundantes necessários à elevação da disponibilidade até tais valores. Nesse caso, a política SA é a mais indicada, uma vez que possibilita a alocação de redes virtuais

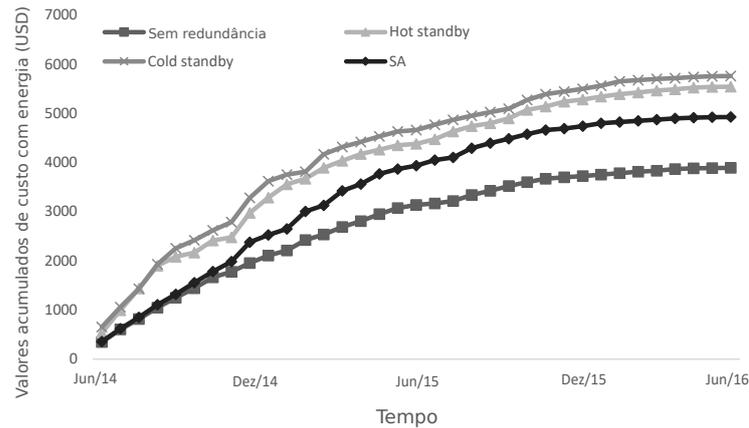


Figura 49 – Resultados do preço da eletricidade

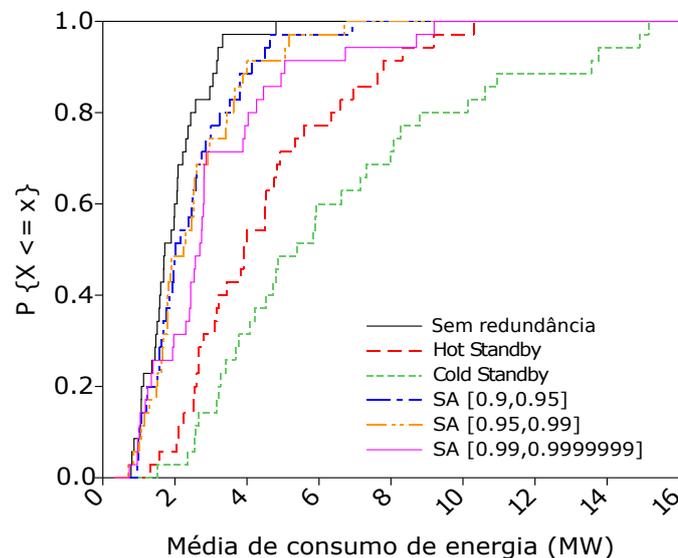


Figura 50 – Resultados de consumo de energia

com altas disponibilidades (como na política *hot standby*) sem ter como contrapartida o grande impacto no custo da alocação de recursos.

Em relação à receita, as políticas SA e sem redundância obtiveram os melhores valores quando a faixa de SLA [0,9 até 0,95] é adotada. Contudo, considerando as faixas de SLA com maiores restrições de disponibilidade(ex: [0,99 até 0,9999999]), a política SA possui os melhores valores de receita entre todas as abordagens.

Taxa de Aceitação. As Figuras 55 a 57 apresenta a ECDF da taxa de aceitação. Os resultados indicam que as políticas sem redundância e SA proveem os maiores valores para as faixas de SLA [0,9 até 0,95] e [0,95 até 0,99]. *Hot* e *cold standby* utilizam mais recursos físicos devido aos componentes redundantes, o que diminui a taxa de aceitação de novas requisições. Por exemplo, no caso de *hot standby*, redes virtuais com alta disponibilidade (ex: 0,999) são geradas mesmo quando as requisições precisam de disponibilidades bem mais baixas. Já abordagem SA evita a utilização desnecessária de recursos. Por fim, uma

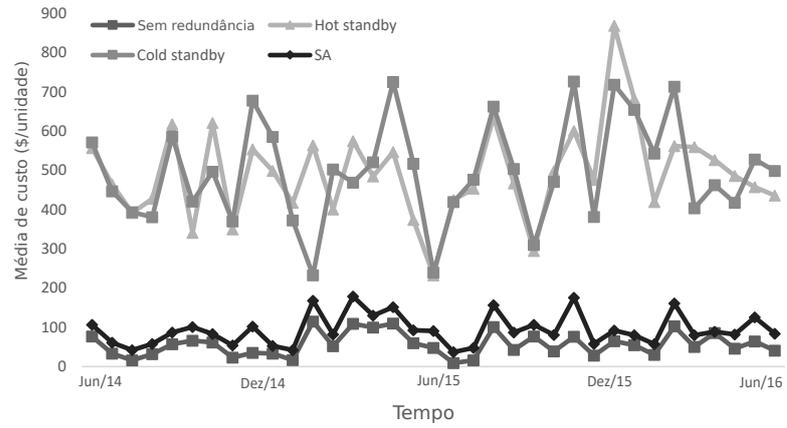


Figura 51 – Resultados do custo de alocação de recursos

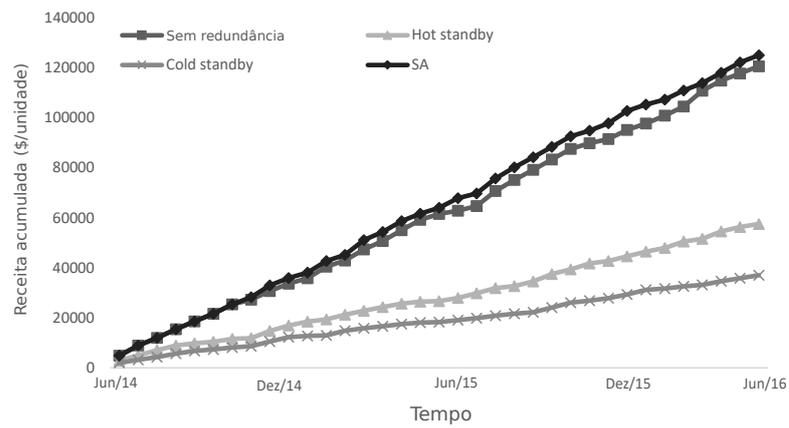


Figura 52 – Resultados de receita - SLA 90-95%

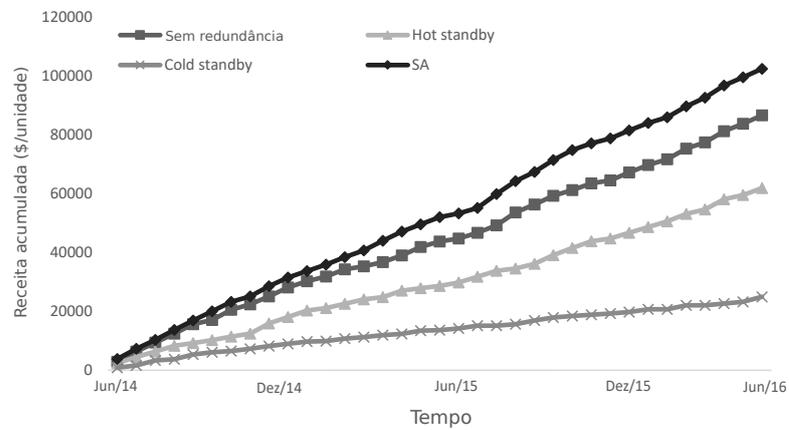


Figura 53 – Resultados de receita - SLA 95-99%

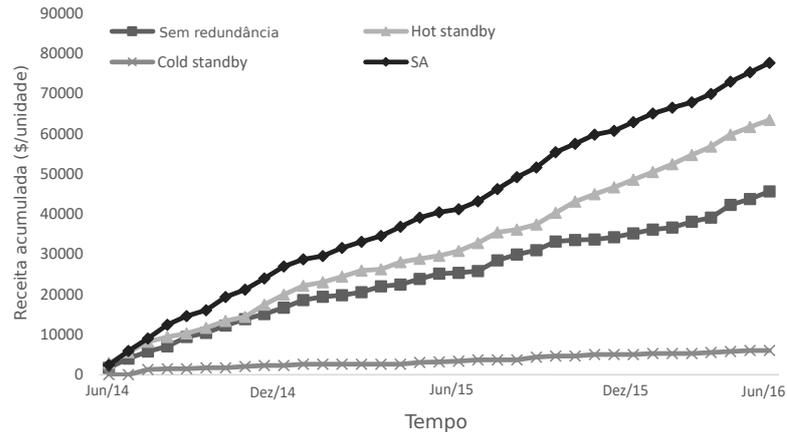


Figura 54 – Resultados de receita - SLA 99-99,99999%

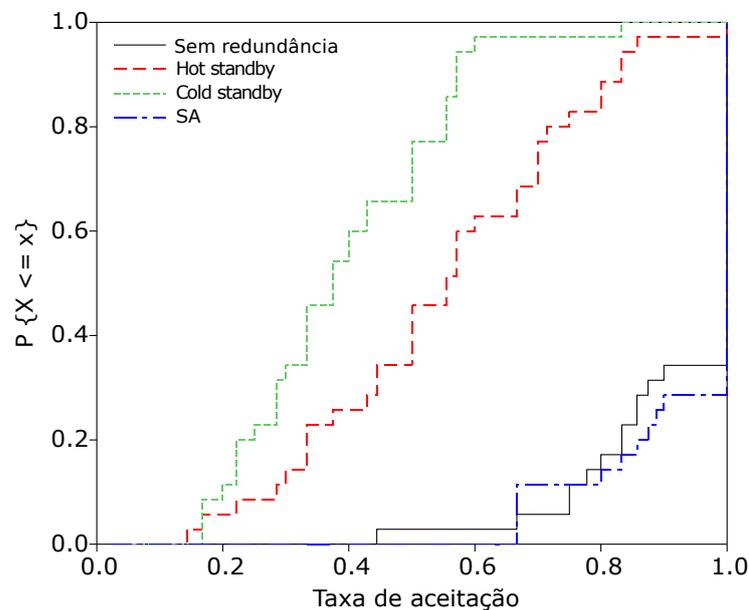


Figura 55 – Resultados da taxa de aceitação - SLA 90-95%

vez que os clientes possuem requisitos de redes virtuais com alta disponibilidade ([0,99 até 0,9999999]), as abordagens SA e *hot standby* possuem os melhores valores de taxa de aceitação.

Esses resultados demonstram a que a abordagem SA proposta é realmente viável. De fato, para a faixa de SLA [0,9 até 0,95] a política SA possui eficiência equivalente à política sem redundância. Contudo, para a faixa de SLA [0,95 até 0,99], os resultados demonstram que a abordagem SA possui os melhores valores de receita e taxa de aceitação. Finalmente, para [0,99 até 0,9999999], *hot standby* e SA possuem uma taxa de aceitação equivalente. Contudo, SA atinge níveis significativamente maiores de receita, bem como custos reduzidos.

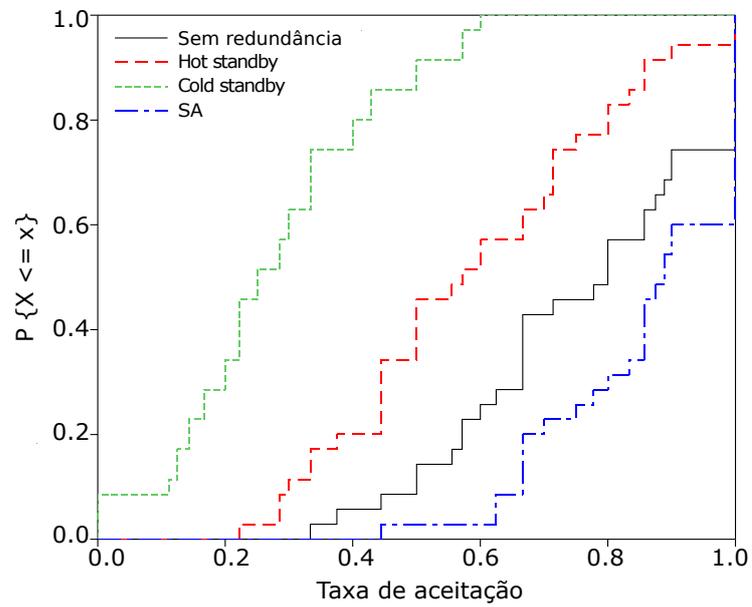


Figura 56 – Resultados da taxa de aceitação - SLA 95-99%

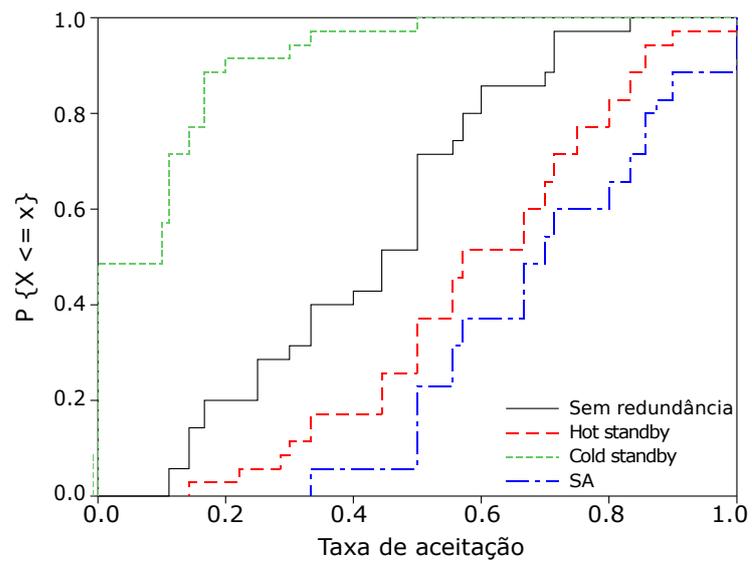


Figura 57 – Resultados da taxa de aceitação - SLA 99-99,99999%

6.3 AVALIAÇÃO DA VAZÃO ORIENTADA À DISPONIBILIDADE EM ENLACES VIRTUAIS

Esta seção apresenta os experimentos conduzidos para verificar o impacto da distância do caminho físico dos enlaces virtuais nas seguintes métricas: vazão orientada à disponibilidade, disponibilidade, confiabilidade, custo e perda de pacotes dos enlaces virtuais. Neste experimento, uma taxa de entrada de dados $\lambda = 50Mbps$ foi adotada para a comunicação dos enlaces.

6.3.1 Resultados

Considerando as políticas sem redundância e *hot standby*, foram coletadas as seguintes métricas: vazão, taxa de perda de pacotes, disponibilidade, confiabilidade e custo. Além disso, é importante mencionar que as métricas são apresentadas de acordo com a distância dos caminhos físicos de cada enlace virtual (Seção 4.2.5). O objetivo é demonstrar em termos de desempenho e dependabilidade, o impacto da distância entre os nós físicos utilizados na alocação dos enlaces virtuais.

Os experimentos foram conduzidos considerando as alocações de redes virtuais com distâncias entre nós físicos de 1 até 10. Os valores médios foram calculados e são apresentados nas Figuras 58 a 62.

As Figuras 58 e 59 apresentam, respectivamente, os resultados da vazão orientada à disponibilidade e perda de pacotes. A política de alocação *hot standby* apresenta a melhor vazão, uma vez que os pacotes podem ser entregues através de diferentes caminhos físicos redundantes. Consequentemente, a perda de pacotes também é significativamente menor em comparação com a política sem redundância. Por exemplo, considerando a distância entre os nós físicos igual a 10, a política sem redundância possui uma perda de pacote de 15%, enquanto o uso da técnica de *hot standby* atinge apenas 0,07% de perdas.

A Figura 60 apresenta os resultados médios do custo, os quais estão relacionados à quantidade de recursos físicos alocados para os enlaces virtuais (Seção 4.2.6). Os resul-

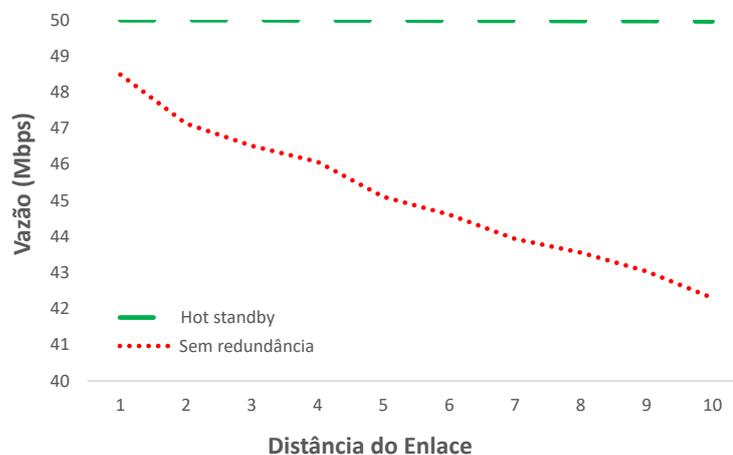


Figura 58 – Resultados da vazão

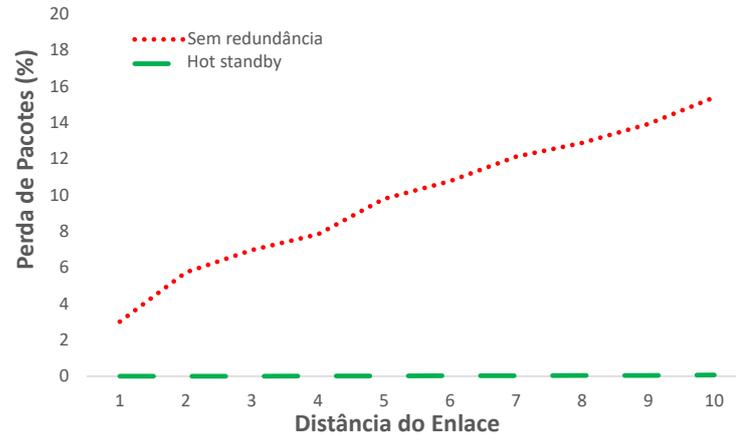


Figura 59 – Resultados de perda de pacotes

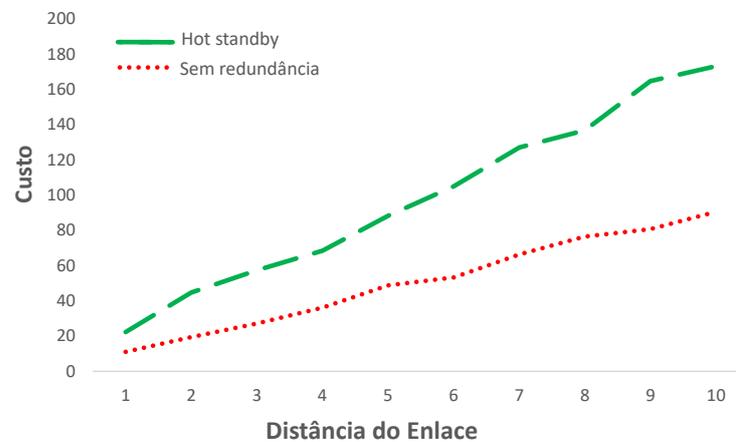


Figura 60 – Resultados dos custos de alocação dos enlaces virtuais

tados indicam que a política de alocação sem redundância entrega os menores valores de custo devido à ausência de nós e enlaces redundantes na alocação. Entretanto, apesar dos custos resultantes da técnica de *hot standby* serem maiores, as métricas de disponibilidade e confiabilidade são positivamente impactadas (Figuras 61 e 62). Os resultados da técnica de alocação sem redundância são influenciados significativamente pela distância do caminho físico utilizado na alocação dos enlaces virtuais. Como mostram os resultados, nos caminhos físicos com maior distância entre os nós, os resultados são de apenas 82% para disponibilidade e de menos de 10% para confiabilidade. Nesse sentido, serviços críticos, os quais necessitam de alta disponibilidade, podem não operar de maneira viável sem o uso de redundância. Por outro lado, o uso da técnica de *hot standby* garante a entrega de alta disponibilidade e confiabilidade em todos os cenários de alocação de redes virtuais considerados.

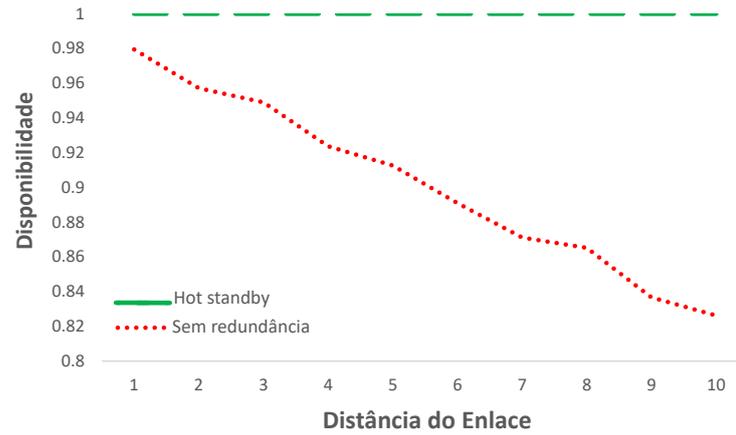


Figura 61 – Resultados de disponibilidade

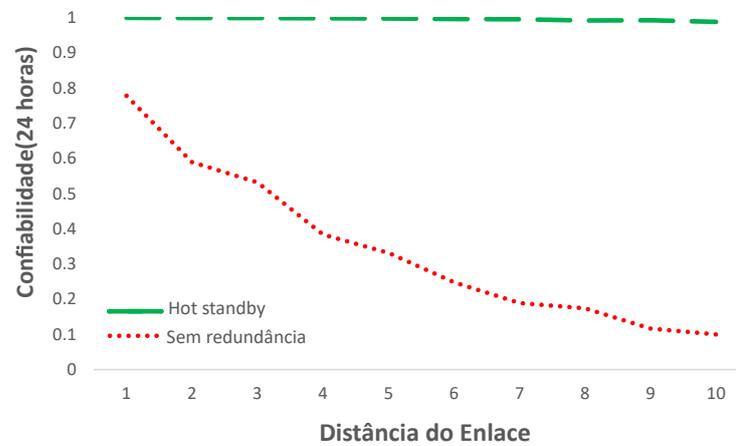


Figura 62 – Resultados de confiabilidade

6.4 CONSIDERAÇÕES FINAIS

Com esses resultados, concluímos que os provedores de serviços podem se beneficiar através do uso da técnica de alocação com análise de sensibilidade baseada em índices de importância de disponibilidade. Tal abordagem permite lidar com diferentes requisitos de redes virtuais, minimizando custos e adotando redundância apenas nos casos em que alta disponibilidade é necessária. Mais especificamente, o uso da técnica de *cold standby* apresenta os piores resultados de custo, disponibilidade e consumo de energia, portanto não se apresenta viável para alocação de redes virtuais. Por outro lado, a técnica de *hot standby* apresenta os melhores resultados para disponibilidade, permitindo a alocação de redes virtuais com o menor *downtime* entre todas as técnicas. Contudo, o uso desse tipo de redundância se mostrou excessivamente custoso em termos de consumo de energia e do uso de recursos físicos na alocação. Por exemplo, se o cliente possui a necessidade de uma rede virtual com disponibilidade de 99,5%, ao invés de atingir apenas tal faixa de valores, a técnica de *hot standby* resulta na entrega de uma rede virtual com disponibilidade muito acima do necessário (ex: 99,999%), resultando em um desnecessário custo e consumo de

energia. O contrário acontece com o uso da técnica sem redundância, a qual otimiza excessivamente o custo e consumo de energia das redes virtuais e não consegue atender a requisições de redes virtuais com restrições de altos níveis de disponibilidade. Finalmente, a abordagem de alocação utilizando análise de sensibilidade permite aos projetistas e administradores de rede realizarem a alocação de redes virtuais de forma adaptativa para cada requisição. Ou seja, caso o cliente necessite de uma rede virtual com determinada disponibilidade (ex: 99,5%), o algoritmo adiciona progressivamente redundância na alocação até que os níveis de disponibilidade sejam elevados para os padrões desejados. Logo, essa técnica se apresenta como a mais indicada, já que é flexível para atender às requisições com diferentes níveis de restrições de disponibilidade, porém sem adicionar redundância desnecessariamente. Adicionalmente, os provedores podem planejar e estimar o consumo energético de suas infraestruturas em um determinado período de interesse, adotando o *framework* E-meter desenvolvido, bem como as equações propostas. Finalmente, a ferramenta MODERADOR permite que gestores e administradores de rede utilizem as técnicas de alocação desenvolvidas para realizar a alocação de redes virtuais de computadores, sem a necessidade de possuir conhecimento especializado das técnicas utilizadas (ex: SPN e RBD).

7 CONCLUSÃO E TRABALHOS FUTUROS

A *Internet* é uma infraestrutura vital para o funcionamento de serviços em diversas áreas, como saúde, entretenimento e educação. Contudo, a *Internet* sofre efeitos da inércia, uma vez que há particular resistência na criação e introdução de inovações tecnológicas importantes e necessárias em sua arquitetura. Além disso, algumas deficiências significativas da *Internet* (como em segurança) têm sido expostas devido a sua rápida expansão, bem como uso em larga escala.

O conceito de virtualização de redes permite a coexistência de múltiplas instâncias de redes virtuais em uma única infraestrutura física compartilhada, e tem sido apontado recentemente pela comunidade científica como uma abordagem promissora para resolver o problema atual de ossificação da *Internet* (CHOWDHURY et al., 2017, NGUYEN et al., 2017, ARAÚJO et al., 2018, HOU et al., 2018). Adicionalmente, os benefícios inerentes ao uso de virtualização de redes incluem flexibilidade na topologia, gerenciabilidade, escalabilidade e isolamento de tráfego, além de permitir o uso de tecnologias heterogêneas.

Recentemente, diversos trabalhos têm sido propostos para permitir a alocação de redes virtuais de uma maneira eficiente. Contudo, as abordagens existentes lidam com questões de desempenho, mas os atributos de dependabilidade e eficiência energética são negligenciados na alocação. Logo, isso pode impactar de forma negativa nos serviços críticos que dependem de altos níveis de disponibilidade para seu correto funcionamento, bem como resultar em altos custos relacionados ao consumo de energia.

Esta tese apresentou uma estratégia para a alocação de redes virtualizadas considerando disponibilidade e consumo de energia. Foi implementado um algoritmo baseado na metaheurística GRASP e análise de sensibilidade baseada em índices de importância de disponibilidade para a realização das alocações. Além disso, foi proposta uma abordagem híbrida de modelagem, com redes de Petri estocásticas (SPN) e diagramas de bloco de confiabilidade (RBD) para avaliar a disponibilidade e a vazão orientada à disponibilidade em redes virtuais. Foram definidas diferentes políticas na alocação das redes virtuais, utilizando ou não redundância de infraestrutura física para a alocação das requisições dos clientes. Não obstante, foi proposto um modelo energético para estimar o consumo de energia e o preço da eletricidade em redes virtuais. Finalmente, as técnicas de modelagem e alocação desenvolvidas foram reunidas em uma ferramenta de *software* chamada MODERADOR, a qual permite que gestores e administradores de rede possam realizar a alocação de redes virtuais confiáveis de computadores, considerando questões de consumo energético na alocação. Tal ferramenta permite que as alocações sejam obtidas através do uso de uma interface gráfica amigável ao usuário, sem a necessidade de que o mesmo possua conhecimento específico de programação ou mesmo do funcionamento interno dos modelos formais adotados.

7.1 LIMITAÇÕES

Ao longo do desenvolvimento e escrita desta tese, podem ser observadas diversas limitações, as quais estão descritas a seguir.

- **Uso de apenas uma heurística na alocação das redes virtuais:** Neste trabalho, a técnica de alocação adotada é baseada na metaheurística GRASP. Contudo, é interessante a implementação e comparação dos resultados obtidos com outras heurísticas existentes.
- **Uso de infraestruturas físicas de um único provedor:** O algoritmo de alocação implementado considera a topologia de um único provedor de infraestrutura física.
- **O modelo de precificação de energia:** O modelo de precificação de energia adotado considera que todos os componentes de hardware adotados estão alocados em uma região com o mesmo preço de energia elétrica. Tal assertiva pode não ser realista em alguns cenários, considerando que a infraestrutura física dos provedores pode ser geograficamente distribuída em diferentes regiões, cada uma com um preço de energia elétrica diferente.

7.2 CONTRIBUIÇÕES

Como resultado das atividades desenvolvidas nesta tese, podemos identificar as seguintes contribuições:

- Proposição de um modelo energético validado em uma infraestrutura física real para estimar o consumo de energia de redes virtuais.
- Proposição de uma técnica de modelagem híbrida, a qual adota modelos combinatoriais e baseados em estados para representação de redes virtualizadas. Além disso, pode-se destacar a metodologia hierárquica adotada, que, em conjunto com a técnica de modelagem proposta, permite mitigar a complexidade de representação de redes virtuais.
- Elaboração de uma técnica de modelagem, baseada no uso de redes de Petri estocásticas (SPN), para modelar e estimar a vazão orientada a disponibilidade em enlaces virtuais.
- Um algoritmo, baseado na metaheurística GRASP, para alocação de redes virtuais considerando disponibilidade e consumo de energia na alocação. Através da técnica de alocação proposta, a qual utiliza análise de sensibilidade baseada em índices de importância de disponibilidade é possível realizar a alocação das redes virtuais utilizando diferentes políticas de redundância, a depender das necessidades específicas de cada requisição.

- A ferramenta de *software* MODERADOR, a qual integra as técnicas e modelos desenvolvidos, bem como os algoritmos propostos para a realização de alocações de redes virtuais considerando as métricas de desempenho e dependabilidade exploradas neste trabalho.

A ferramenta MODERADOR permite que administradores e gestores de redes possam prover redes virtuais capazes de satisfazer clientes com diferentes necessidades de restrições de disponibilidade, de forma a minimizar o consumo de energia, bem como o custo advindo do uso dos recursos da infraestrutura física.

7.3 TRABALHOS FUTUROS

Embora esta tese tenha alcançado diversos resultados e coberto alguns pontos relacionados à alocação de redes virtuais de computadores considerando disponibilidade e consumo de energia, há muitas possibilidades de estender o trabalho atual. Algumas dessas possibilidades podem ser implementadas em trabalhos futuros, os quais estão listados a seguir:

- **Uso de outras heurísticas para a alocação:** Pode-se avaliar o impacto do uso de outras heurísticas na alocação, como por exemplo, algoritmos genéticos.
- **Extensão do modelo energético atual para considerar infraestruturas geograficamente distribuídas:** É possível a proposição de um modelo energético que considere preços diferentes de eletricidade para nós e enlaces, de acordo com a localidade onde os equipamentos físicos estão localizados.
- **Uso de infraestruturas físicas compartilhadas:** O algoritmo de alocação implementado pode ser estendido para permitir a alocação das redes virtuais utilizando a combinação de diferentes topologias, de diferentes provedores de infraestrutura.
- **Estudo do impacto das técnicas de alocação na latência das redes virtuais:** Devido ao crescente número aplicações multimídia e de outros serviços sensíveis a atraso na entrega de pacotes, um importante requisito a ser considerado é o impacto do uso de redes virtuais e das técnicas de alocação na latência, visando prover técnicas que permitam atender a requisitos de latência entre determinados pares de nós.
- **Utilização da vazão orientada à disponibilidade como restrição na alocação das requisições:** A técnica de alocação adotada pode ser estendida, para que o usuário informe o nível máximo aceitável de degradação da vazão dos enlaces virtuais durante a alocação.

REFERÊNCIAS

- SCHAFFRATH, G. *et al.* **Network virtualization architecture: proposal and initial prototype.** Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures (VISA-09), 2009.
- CLARK, D. *et al.* **Tussle in Cyberspace: Defining Tomorrow's Internet.** In Proceedings of SIGCOMM -02, pages 347-356. ACM, 2002.
- FERNANDES, S. *et al.* **Dependability Assessment of Virtualized Networks.** In: Proceedings of the IEEE ICC - Next-Generation Networking Symposium in conjunction with IEEE International Conference on Communications, Ottawa, Canada, 2012.
- MACIEL, P. *et al.* **Dependability Modeling.** In: Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions. Ed. Hershey: IGI Global, Pennsylvania, USA, 2010.
- SZETO, W.; IRAQI, Y.; BOUTABA, R. **A multi-commodity flow based approach to virtual network resource allocation.** In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM-03), , pp. 3004-3008, 2003.
- KOSLOVSKI, G. *et al.* **Reliability support in virtual infrastructures.** In Proceedings of the IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci., pp.49 -58, 2010.
- SUN, D. *et al.* **A Dependability Model to Enhance Security of Cloud Environment Using System-Level Virtualization Techniques.** 1st Pervasive Intl.Conf. on Computing Signal Processing and Applications (PCSPA), 2010,
- SARIPALLI, P.; WALTERS, B. **QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security.** Cloud Computing (CLOUD) IEEE, 2010,
- HU, X.; LIU, S.; MA, L. **Research on dependability of virtual computing system based on Stochastic Petri nets.** Computer Application and System Modeling (ICCASM), 2010.
- CHOWDHRY, N.; RAHMAN, M.; BOUTABA, R. **Virtual network embedding with coordinated node and link mapping.** In Proc. IEEE INFOCOM, 2009.
- SILVA, B. *et al.* **ASTRO: An Integrated Environment for Dependability and Sustainability Evaluation.** Sustainable Computing: Informatics and Systems, v. 2, p. 1-31, 2012.
- RESENDE, M.; RIBEIRO, C. **Greedy randomized adaptive search procedures.** In F. Glover and G. Kochenberger, editors, State-of-the-Art Handbook in Metaheuristics. Kluwer Academic Publisher, 2001.
- ZEGURA, E.; CALVERT, K.; BHATTACHARJEE, S. **How to model an Internet network.** In Proceedings of IEEE INFOCOM, pp. 594-602, 1996.
- BALBO, G. **Introduction to stochastic petri nets.** Lectures on Formal Methods and Performance Analysis, 2001.

- TRIVEDI, K. **Probability and Statistics with Reliability, Queueing, and Computer Science Applications**. Wiley Interscience Publication, 2 edition, 2002.
- RAHMAN, M.; AIB, I.; BOUTABA, R. **Survivable virtual network embedding**. In Proc. 9th IFIP NETWORKING Conference, Chennai, India, 2010.
- KUO, W.; ZUO, M. **Optimal reliability modeling: principles and applications**. Wiley, 2003.
- ZHU, Y.; AMMAR, M. **Algorithms for assigning substrate network resources to virtual network components**. In Proceedings of IEEE INFOCOM, 2006.
- News Briefs, IEEE Computer Magazine (2011). Páginas 18-20, <http://dx.doi.org/10.1109/MC.2011.185>, 2011.
- TURNER, J.; TAYLOR, D. **Diversifying the internet**. IEEE Globecom, 2005.
- ARAUJO, C. *et al.* **Performance modeling for evaluation and planning of electronic funds transfer systems with bursty arrival traffic**. First International Conference on Intensive Applications and Services, 2009.
- SCHRIJVER, A. **Theory of linear and integer programming**. New York, NY, USA: John Wiley & Sons, Inc, 1986.
- NATKIN, G. **Matrix Product Form Solution For Closed Synchronized Queuing Networks**. In Petri Nets and Performance Models, page 29. IEEE Computer Society, 1990.
- RAUSAND, M.; HOYLAND, A. **System Reliability Theory - Models, Statistical Methods, and Applications**. Wiley Series in Probability and Statistics. Wiley, second edition, 2004.
- CALLOU, G. *et al.* **A formal approach to the quantification of sustainability and dependability metrics on data center infrastructures**. Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, p.274-281, Boston, Massachusetts, 2011.
- DANTAS, J. *et al.* **Models for Dependability Analysis of Cloud Computing Architectures for Eucalyptus Platform**. International Transactions on Systems Science and Applications (Print), v. 8, p. 13-25, 2012.
- WAN, Y. *et al.* **The adaptive heartbeat design of high availability raid dual-controller**. In Multimedia and Ubiquitous Engineering, 2008. MUE 2008. International Conference on, pages 45-50. IEEE, 2008.
- FEAMSTER, N.; GAO, L.; REXFORD, J. **How to lease the Internet in your spare time**. SIGCOMM Computer Communication Review, vol. 37, no. 1, pp. 61-64, 2007.
- FELDMANN, A. **Internet clean-slate design: What and why?** SIGCOMM CCR, 37(3):59-64, 2007.
- HONG, Z.; WANG, Y.; SHI, M. **Ctmc-based availability analysis of cluster system with multiple nodes**. In Advances in Future Computer and Control Systems, volume 160, pages 121-125, 2012.
- WEI, B.; LIN, C.; KONG, X. **Dependability modeling and analysis for the**

- virtual data center of cloud computing.** In High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on, pages 784-789, 2011.
- TRIVEDI, K. *et al.* **Modeling high availability systems.** In Proc. Pacific Rim Dependability Conference, pages 11-20, 2006.
- KIM, D.; MACHIDA, F.; TRIVEDI, K. **Availability modeling and analysis of a virtualized system.** In Dependable Computing. PRDC -09. 15th IEEE Pacific Rim Int. Symp. on, pages 365-371, 2009.
- MACIEL, P.; LINS, R.; CUNHA, P. **Introdução às redes de Petri e aplicações.** Campinas: Instituto de Computação, UNICAMP, 1996.
- PETRI, C. **Kommunikation mit Automaten.** Schriften des RheinischWestfälischen Institutes für Instrumentelle Mathematik an der Universität Bonn Nr. 2, 1962.
- CHANIN, R. *et al.* **Avaliação Quantitativa de Sistemas.** Faculdade de Informática, PUCRS, Porto Alegre, 2005.
- MARSAN, M. *et al.* **Modelling with Generalized Stochastic Petri Nets.** ACM SIGMETRICS Performance Evaluation Review, 26(2), 1998.
- NORRIS, J. **Markov Chains.** Cambridge University Press, 1998.
- LILJA, D. **Measuring Computer Performance: A Practitioner's Guide.** Cambridge University Press, 2000.
- MARSAN, M. **Stochastic Petri Nets: an elementary introduction.** Advances in Petri Nets, 1990.
- ROZENBERG, G. Lecture Notes in Computer Science 424, pp. 1-29, 1989.
- LAPRIE, J. **Dependable Computing: Concepts, Limits, Challenges.** in Proc. 25th Int. Symp. on Fault-Tolerant Computing (FTCS-25). Special Issue, (Pasadena, CA, USA), pp. 42-54, IEEE Computer Society Press, 1995.
- LAPRIE, J.; AVIZIENIS, A.; KOPETZ, H. **Dependability: Basic concepts and terminology.** Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1992.
- FIGUERÊDO, J. **Análise de Dependabilidade de Sistemas Data Center Baseada em Índices de Importância.** Dissertação - Centro de Informática, Universidade Federal de Pernambuco, Recife, 2011.
- OSMAN, I. **Metaheuristics: A bibliography.** Institute of Mathematics and Statistics, University of Kent, Canterbury, 1996.
- RANGEL, M.; ABREU, N.; BOAVENTURA NETTO, P. **GRASP para o PQA: Um Limite de Aceitação para Soluções Iniciais.** Pesquisa Operacional SciELO Brasil, Vol. 20, No 1, 2000.
- RESENDE, M.; RIBEIRO, C. **An annotated bibliography of GRASP.** AT&T Labs Research, Florham Park, NJ, 2004.
- MACAMBIRA, E.; MENESES, C. **A GRASP algorithm for the maximum weighted edge subgraph problem.** Departamento de Estatística e Computação,

Universidade do Ceará, Fortaleza, Ceará, 1998.

KLINCEWICZ, J. **Avoiding local optima in p-hub location problem using tabu search and GRASP.** Annals of Operations Research, 40, 238-302, 1992.

LAGUNA, M.; FEO, T.A.; ELROD, H. **A greedy randomized adaptive search procedures for the 2-partition problem.** tech. report., Graduate School of Business and Administration, The University of Colorado at Boulder, Boulder, CO 80309-0419, 1993.

LAGUNA, M.; GONZÁLEZ-VELARDE, J. **A search heuristic for just-in-time scheduling in parallel machines.** Journal of Intelligent Manufacturing, 253-260, 1991.

FEO, T. A.; RESENDE, M.G.C. **Greedy randomized adaptive search procedures.** Journal of Global Optimization, 1995.

MURATA, T. **Petri nets: Properties, analysis and applications.** Proc. IEEE, 77(4), 541-580, 1989.

VAUGHAN-NICHOLS, S. **We Love IPv6, We Love IPv6 Not.** in Enterprise IT Planet, 2004.

CALLOU, G. *et al.* **An Integrated Modeling Approach to Evaluate and Optimize Data Center Sustainability, Dependability and Cost.** Energies. 7(1):238-277, 2014.

FERREIRA J.; CALLOU G.; MACIEL P. **A Power Load Distribution Algorithm to Optimize Data Center Electrical Flow.** Energies. ; 6(7):3422-3443, 2013.

MARWAH, M. *et al.* **Quantifying the sustainability impact of data center availability.** ACM SIGMETRICS Performance Evaluation Review, 37(4):64-68, 2010.

NIST, Standards for Security Categorization of Federal Information and Information Systems. FIPS-199, <csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf>, Acessado em março de 2014.

EUCALYPTUS. Eucalyptus - the open source cloud platform. Eucalyptus Systems. Available in: <http://open.eucalyptus.com/>, Acessado em 2012.

LU, J.; TURNER, J. **Efficient mapping of virtual networks onto a shared substrate.** Washington University, Tech. Rep. WUCSE-2006-35, 2006.

YU, M. *et al.* **Rethinking virtual network embedding: Substrate support for path splitting and migration.** ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 17-29, 2008.

MELO, M. *et al.* **Optimal Virtual Network Embedding: Node-Link Formulation.** Network and Service Management, IEEE Transactions on , vol.10, no.4, pp.356,368, 2013.

WOLSEY, L. **Integer programming.** Wiley-Interscience Series In Discrete Mathematics and Optimization, 1998.

LISCHKA, J.; KARL, H. **A virtual network mapping algorithm based on subgraph isomorphism detection.** In: Proceedings of ACM SIGCOMM VISA, 2009.

-
- GROETSCHHEL, M.; LOVASZ, L.; SCHRIJVER, A. **Geometric Algorithms and Combinatorial Optimization**. Springer, 1988.
- ECKERSON, W. **Three tier client/server architecture: Achieving scalability, performance, and efficiency in client server applications**. Open Information Systems Journal, 1, 1995.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **Unified Modeling Language User Guide**. Reading, MA: Addison-Wesley, 1998.
- KELTON, W.; SADOWSKI, R.; STURROCK, D. **Simulation with Arena**. McGrawHill, 2004.
- CHOWDHURY, N.; BOUTABA, R. **Network virtualization: state of the art and research challenges**. Communications Magazine, IEEE, v. 47, n. 7, pp. 20-26, 2009.
- COSTA, L.; FDIDA, S.; DUARTE, O. **An Introduction to Virtual Private Networks: Towards D-VPNs**. Networking and Information Systems Journal, v. 2, n. 6, pp. 575-594, 2000.
- LUA, E. *et al.* **A survey and comparison of peer-to-peer overlay network schemes**. Communications Surveys Tutorials, IEEE, v. 7, n. 2, pp. 72-93, 2005.
- KHAN, M. *et al.* **Multi-Path Link Embedding for Survivability in Virtual Networks**. in IEEE Transactions on Network and Service Management, vol. 13, no. 2, pp. 253-266, 2016.
- BOTERO, J. *et al.* **Energy Efficient Virtual Network Embedding**. IEEE Comm. Letters, vol. 16, no. 5, pp. 756-759, 2012.
- Texas-Instruments (2015). Evm430-f6736 - msp430f6736 evm for metering. Disponível em <http://www.ti.com/tool/EVM430-F6736>. Acessado em 22 de março de 2016.
- SU, S. *et al.* **Energy-Aware Virtual Network Embedding**. in IEEE/ACM Transactions on Networking, vol. 22, no. 5, pp. 1607-1620, 2014.
- WANG, B. *et al.* **Reducing power consumption in embedding virtual infrastructures**. in Proc. IEEE Globecom Workshops, Dec. 3-7, 2012, pp. 714-718, 2012.
- GONG, S. *et al.* **Energy-efficient virtual network embedding for heterogeneous networks**. 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), Wuhan, 2016, pp. 85-90, 2016.
- WALLERSTEIN, S.; LUCKER, C.; FLIN, J. **Some statistical methods useful in circulation research**. Circ Res, 47 (1980), pp. 19, 1980.
- WANG, W.; EUGENE NG T. **The Impact of Virtualization on Network Performance of Amazon EC2 Data Center**. 2010 Proceedings IEEE INFOCOM, San Diego, CA, 2010, pp. 1- 9. doi: 10.1109/INFOCOM.2010.5461931, 2010.
- ARAÚJO, S.; SOUZA, F.; MATEUS, G. **Virtual network embedding in multi-domain environments with energy efficiency concepts**. 2018 International Conference on Information Networking (ICOIN), Chiang Mai, 2018, pp. 205-210. doi: 10.1109/ICOIN.2018.8343111, 2018.
- NGUYEN, L. *et al.* **RT-VNE: A real-time strategy for Virtual Network Em-**

bedding towards resource efficiency. 2017 International Conference on Information Networking (ICOIN), Da Nang, 2017, pp. 185-190. doi: 10.1109/ICOIN.2017.7899501, 2017.

HOU, W. *et al.* **A heuristic survivable virtual network mapping algorithm.** in Soft Computing, 2018.

PAPASTERGIOU, G. *et al.* **De-Ossifying the Internet Transport Layer: A Survey and Future Perspectives.** in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 619-639, Firstquarter 2017. doi: 10.1109/COMST.2016.2626780, 2017.

SHAHRIAR, N. *et al.* **Generalized Recovery From Node Failure in Virtual Network Embedding.** in IEEE Transactions on Network and Service Management, vol. 14, no. 2, pp. 261-274, June 2017. doi: 10.1109/TNSM.2017.2693404, 2017.

CHOWDHURY, S. *et al.* **ReViNE: Reallocation of Virtual Network Embedding to eliminate substrate bottlenecks.** 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, 2017, pp. 116-124. doi: 10.23919/INM.2017.7987271, 2017.

CAO, H. *et al.* **A Novel Optimal Mapping Algorithm With Less Computational Complexity for Virtual Network Embedding.** in IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 356-371, March 2018. doi: 10.1109/TNSM.2017.2778106, 2018.

YU, C. *et al.* **PAME: Evolutionary membrane computing for virtual network embedding.** in Journal of Parallel and Distributed Computing, vol. 111, pp. 136-151, January 2018. doi: 10.1016/j.jpdc.2017.08.005, 2018.

JAVA website. Disponível em: <<https://www.java.com/en/>>. Acessado em Março de 2016.

Especificação da Máquina Virtual JAVA. Disponível em: <<http://docs.oracle.com/javase/specs/jvms/se7/html/>>. Acessado em Março de 2016.

CHIARAVIGLIO, L.; MELLIA, M.; NERI, F. **Minimizing ISP Network Energy Cost: Formulation and Solutions.** Networking, IEEE/ACM Transactions on , vol.20, no.2, pp.463,476, 2012.

PICKAVET, M. *et al.* **Worldwide energy needs for ICT: The rise of power-aware networking.** in Proc. IEEE ANTS, Bombay, India, 2008.

SIVARAMAN, V. *et al.* **Profiling perpacket and per-byte energy consumption in the netfpga gigabit router.** In Proc. IEEE INFOCOM WKSHPS, 2011.

CHABAREK, J. *et al.* **Power awareness in network design and routing.** In Proc. IEEE INFOCOM, 2008.