ESTRATÉGIAS E ANÁLISE DE RESILIÊNCIA EM REDES DE CENTROS DE DADOS

Rodrigo de Souza Couto

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Luís Henrique Maciel Kosmalski Costa
Miguel Elias Mitre Campista

Rio de Janeiro
Janeiro de 2015

ESTRATÉGIAS E ANÁLISE DE RESILIÊNCIA
EM REDES DE CENTROS DE DADOS

Rodrigo de Souza Couto

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

Prof. Miguel Elias Mitre Campista, D.Sc.

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Prof. Célio Vinicius Neves de Albuquerque, Ph.D.

Profa. Michele Nogueira Lima, Dr.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2015

Couto, Rodrigo de Souza

Estratégias e Análise de Resiliência em Redes de Centros de Dados/Rodrigo de Souza Couto. – Rio de Janeiro: UFRJ/COPPE, 2015.

XVII, 109 p.: il.; 29, 7cm.

Orientadores: Luís Henrique Maciel Kosmalski Costa
Miguel Elias Mitre Campista

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 102 – 109.

1. Redes de Centro de Dados. 2. Resiliência. 3. Confiabilidade. 4. Sobrevivência. 5. Computação em Nuvem. I. Costa, Luís Henrique Maciel Kosmalski *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Às mulheres da minha vida,
minha noiva Marcela, minha
mãe Laura e minha irmã
Fernanda. Ao meu avô
Vicente (in memoriam).

Agradecimentos

A conquista do título de Doutor se tornou concreta com o auxílio de muitas pessoas em diferentes etapas da minha vida. Primeiramente, agradeço aos meus pais por todo investimento feito em minha educação, mesmo quando a situação financeira não era favorável. Também agradeço por todo o carinho e afeto recebido, que me possibilitaram finalizar esses anos de estudo e, mais importante, a construir o meu caráter. Obrigado também pelo exemplo de vida e pelos conselhos dados. À minha mãe Laura agradeço também pelos puxões de orelha, que me possibilitaram aprender como ser uma boa pessoa, a todos os dias que deixou de preocupar consigo mesma para cuidar de mim e da minha irmã, batalhando arduamente pelo nosso bem-estar. Ao meu pai Alvaro agradeço também pelas longas conversas sobre carreira, que me ajudaram a escolher qual caminho seguir, a ouvir com interesse sobre assuntos acadêmicos e as memoráveis brincadeiras na infância.

À minha noiva Marcela por me acompanhar desde o início da minha carreira científica, dando força e incentivo. Obrigado por fazer a minha vida mais completa, estando sempre ao meu lado e me dando muito orgulho de tê-la como futura esposa. Fico muito feliz de ter ao meu lado alguém que possui exatamente os mesmos objetivos de vida e profissionais que os meus. Obrigado por todos os bons momentos que já passamos juntos.

Aos meus avós maternos Maria e Vicente pelo carinho e por me acolherem em sua casa durante vários anos. Esse gesto de carinho levarei pelo resto da minha vida. Agradeço também à minha avó paterna Maria também pelo carinho e pelas reuniões de família feitas em sua casa e sua dedicação em torná-las muito agradáveis.

À minha irmã Fernanda por ser uma ótima companhia em toda minha vida e pela dedicação à família. Por ter feito a minha infância e adolescência mais divertidas, apesar de eu sempre reclamar quando ela precisava de mim para comprar pipoca. Ao meu cunhado Pedro por cuidar muito bem da minha irmã.

Agradeço à minha prima Cláudia pelas aulas de Matemática e sua constante torcida e à tia Clair pelos ótimos momentos passados em sua casa. Gostaria muito de agradecer também ao tio Nanando por tudo que me ensinou e, junto com o tio Gerardo, pelo grande apoio dado nos momentos mais difíceis. Aos meus sogros Márcia e Nestor e à Dona Cléia, pelo carinho e pela amizade.

Aos meus orientadores Luís e Miguel pela imensa atenção recebida durante esses quase sete anos de GTA. Muito obrigado pelos conselhos dados e pelas sugestões profissionais, que me permitiram crescer academicamente e me abriram muitas portas para no mundo profissional. Certo de que me espelharei em vocês na continuação da minha vida acadêmica, esperando fornecer aos meus futuros orientados uma qualidade de orientação ao menos próxima à que vocês forneceram para mim.

Aos professores do Pedro II pela base que me foi dada. Especialmente, agradeço à professora Solveig pelas correções na minha forma de escrever, o que possibilitou escrever esta tese e as diversas publicações relacionadas. Agradeço os professores da UFRJ por me fornecerem uma base sólida, tanto na graduação quanto na pós-graduação. Gostaria de agradecer também aos funcionários da COPPE/UFRJ, em especial à Daniele, Rosa e Mauricio pela dedicação ao trabalho e presteza no atendimento.

Gostaria de agradecer a todos amigos que fiz através do GTA, que me ajudaram bastante a crescer profissionalmente e tornar o trabalho mais divertido, Júnior, Vitor Borges, Igor Moraes, Fábio Vieira, Marcelo Rubinstein, Otto, Pedro Velloso, Marcelo Amorim, Célio Albuquerque, Hugo Sadok, Tatiana, Bernardo Camilo, Dianne, Lyno, Alyson e Marcus. Agradeço também aos outros amigos, que estão há muitos anos em minha vida ou em poucos anos se tornam pessoas importantes, Gabriel de Almeida de Barros, Clécio De Bom, João Guilherme Cardoso, Leonardo Martins, Bruno Linhares, Ana Coimbra, Telma Almeida, Laura Moraes, Felipe Bogossian, Luís Felipe Moraes, Eric Vinícius Leite, Leandro Borges, Olavo Machado, Juliana Joannou e Facundo Cosmai. Aos amigos da *Maison du Brésil*, que tornaram o ano em Paris ainda mais inesquecível, Karen Fukushima, Jane Barbosa, Monize Moura, Raissa Mussara, João Paulo Jeannine, Junior Xavier, Felipe Cunha e Hudson Polononi.

Gostaria de agradecer aos membros do LIP6/UPMC, que me receberam muito bem durante meu período na França, em especial ao professor Stefano Secci, cuja contribuição para este trabalho foi de elevada importância.

Gostaria ainda de agradecer aos professores Aloysio Pedroza e Célio Albuquerque e à professora Michele Nogueira pela presença na banca examinadora e pelos comentários pertinentes que impulsionarão futuros trabalhos.

Por fim, agradeço aos órgãos FAPERJ, CNPq e CAPES pelo financiamento da pesquisa.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

ESTRATÉGIAS E ANÁLISE DE RESILIÊNCIA EM REDES DE CENTROS DE DADOS

Rodrigo de Souza Couto

Janeiro/2015

Orientadores: Luís Henrique Maciel Kosmalski Costa
Miguel Elias Mitre Campista

Programa: Engenharia Elétrica

A sociedade requer cada vez mais o uso dos centros de dados, visto que esses são responsáveis por hospedar serviços importantes como aplicações de computação em nuvem e serviços web. Devido a essa importância, é necessário estudar e planejar a resiliência dos centros de dados aos mais diversos tipos de falha, desde o rompimento de cabos de rede até grandes desastres. Um centro de dados pode ser composto por sítios espalhados geograficamente, cada um empregando uma determinada arquitetura de rede para conectar seus servidores. Assim, é necessário estudar a resiliência tanto da rede interna a um sítio, como também da rede que interliga os sítios. Primeiramente, esta tese analisa a resiliência de três novas topologias internas ao sítio: Fat-tree, BCube e DCell. Os resultados indicam características das topologias considerando falhas de enlace, servidor ou comutador. Em particular, conclui-se que a BCube e a DCell possuem, respectivamente, melhor resiliência para falhas de enlace e de comutador. Em seguida, esta tese aborda a distribuição geográfica do centro de dados, que reduz o impacto das falhas, mas aumenta a latência entre os servidores, causada pela maior distância entre eles. Esse compromisso é analisado através da formulação de um problema de otimização. Os resultados mostram que o aumento na latência é significativo apenas no caso de exigências muito fortes de resiliência, porém é insignificante para exigências moderadas. Ainda considerando a geodistribuição, nesta tese formula-se um problema de posicionamento de backups em uma rede de longa distância, considerando a replicação contínua e confirmada de máquinas virtuais. Um dos objetivos do problema é reduzir o número de servidores de backup necessários. Os resultados mostram que a estratégia proposta reduz o número de servidores de backup em pelo menos 40%.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

RESILIENCE STRATEGIES AND ANALYSIS
IN DATA CENTER NETWORKS

Rodrigo de Souza Couto

January/2015

Advisors: Luís Henrique Maciel Kosmalski Costa
Miguel Elias Mitre Campista

Department: Electrical Engineering

Our society increasingly requires the use of data centers, since they are responsible for hosting important services, such as cloud computing applications and web services. Due to this importance, we need to study and plan the resilience of data centers to different types of failure, from network cable cuts to big disasters. A data center can be composed of geographically spread sites, each one employing a given network architecture to interconnect its servers. Hence, we need to study the resilience of the network inside the sites, as well as the resilience of the network between sites. First, we analyze in this thesis the resilience to failures of three novel intra-site topologies: Fat-tree, BCube, and DCell. Results indicate characteristics of the topologies considering link, server or switch failures. In particular, we conclude that BCube is more resilient to link failures than the other topologies, whereas DCell has the most resilient topology considering switch failures. Next, this thesis addresses the geographical data center distribution, which reduces failure impact, but increases the latency between servers, as a consequence of their geographic distances. This trade-off is analyzed through an optimization problem formulation. Results show that the latency increase is significant only in the case of very strong resilience requirements, whereas it is negligible for moderate resilience requirements. Still considering geo-distribution, in this thesis we formulate a problem to place backups in a wide area network, considering the continuous and acknowledged replication of virtual machines. One of the problem goals is to reduce the amount of backup servers required. Results show that the proposed strategy reduces the number of backup servers by at least 40%.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

A2TR	*Average Two Terminal Reliability* - Confiabilidade Média de Dois Terminais, p. 32
ABT	*Aggregate Bottleneck Throughput* - Vazão agregada de gargalo, p. 46
ASPL	*Average Shortest Path Length* - Comprimento Médio dos Caminhos Mais Curtos, p. 33
AS	Aproveitamento de Servidores, p. 87
AWS	*Amazon Web Services*, p. 55
BCP	*Business Continuity Planning* - Plano de continuidade de negócios, p. 49
CAPEX	*CAPital EXpenditures* - Despesas de capital, p. 1
CDF	*Cumulative Distribution Function* - Função de distribuição acumulada, p. 19
CDN	*Content Delivery Networks* - Redes de distribuição de conteúdo, p. 91
DCN	*Data Center Network* - Rede de centro de dados, p. 1
DC	*Data Center* - Centro de dados, p. 1
DNS	*Domain Name System*, p. 55
FER	*Failed Elements Ratio* - Fração de elementos defeituosos, p. 16
GPD	*Gateway Port Density* - Densidade de Portas de Gateway, p. 20
ILP	*Integer Linear Programming* - Programação linear inteira, p. 82

ITIL	Information Technology Infrastructure Library, p. 49
ITSCM	*Information Technology Service Continuity Management* - Gerenciamento da continuidade de serviços de tecnologia da informação, p. 49
IaaS	*Infrastructure as a Service* - Infraestrutura como serviço, p. 2
MDC	*Modular Data Center* - Centro de dados modular, p. 2
MILP	*Mixed Integer Linear Programming* - Programação linear inteira mista, p. 66
MTTF	*Mean Time To Failure* - Tempo médio até uma falha, p. 5
NMTTF	*Normalized MTTF* - MTTF Normalizado, p. 22
OPEX	*OPerational EXpenditure* - Despesas Operacionais, p. 1
PoP	*Point of Presence* - Ponto de presença, p. 56
QoE	*Quality of Experience* - Qualidade de experiência, p. 52
QoR	*Quality of Resilience* - Qualidade de Resiliência, p. 48
QoS	*Quality of Service* - Qualidade de serviço, p. 48
REN	*Research and Education Network* - Rede de educação e pesquisa, p. 56
RE	*Relative Error* - Erro Relativo, p. 23
RPO	*Recovery Point Objective* - Objetivo do ponto de recuperação, p. 51
RSR	*Reachable Server Ratio* - Fração de servidores alcançáveis, p. 31
RTO	*Recovery Time Objective* - Objetivo do tempo de recuperação, p. 51
RTT	*Round-Trip Time* - Tempo de ida e volta, p. 64
SC	*Server Connectivity* - Conectividade entre servidores, p. 32
SDN	*Software Defined Network*, p. 92
SLA	*Service Level Agreement* - Acordo de nível de serviço, p. 48

SRG	*Shared Risk Group* - Grupo de risco compartilhado, p. 53
TCP	*Transmission Control Protocol*, p. 53
TI	Tecnologia da Informação, p. 2
TTF	*Time To Failure* - Tempo até uma falha, p. 17
ToR	*Top-of-Rack* - Topo de bastidor, p. 14
VM	*Virtual Machine* - Máquina virtual, p. 2
WAN	*Wide Area Network* - Rede de longa distância, p. 3

Capítulo 1

Introdução

A sociedade encontra-se cada vez mais dependente dos centros de dados (DCs - *Data Centers*), devido ao papel essencial que desempenham em serviços web, aplicações de computação em nuvem e de manipulação de grandes massas de dados. Os centros de dados são infraestruturas computacionais, compostas por diversos servidores interconectados por uma rede. Os DCs, como qualquer outro tipo de infraestrutura, estão sujeitos a falhas em seus componentes, como rompimento de cabos, quedas de energia e desastres. Essas falhas podem levar à indisponibilidade completa ou parcial dos seus servidores. Como a sociedade depende bastante do funcionamento do DCs, a resiliência a falhas e desastres de centros de dados deve ser cuidadosamente planejada. Utiliza-se nesta tese o termo resiliência para representar os diversos aspectos do comportamento de um DC quando sujeito a falhas, como a confiabilidade e sobrevivência, definidos ao longo do trabalho.

Como o tamanho dos centros de dados vem crescendo intensamente ao longo dos anos, além da resiliência, as despesas operacionais (OPEX - *OPerational EXpenditure*) e de capital (CAPEX - *CAPital EXpenditures*) se tornam cada vez mais importantes na escolha de como organizar a arquitetura de um DC. As topologias utilizadas nos DCs tradicionais são estruturadas e construídas geralmente utilizando uma árvore hierárquica de três níveis: o núcleo, a agregação e a borda [1]. As arquiteturas que adotam essas topologias empregam equipamentos de ponta, sofrendo assim custos proibitivos na construção de grandes DCs, como mostrado em [2]. Como consequência, diversas arquiteturas de rede de centros de dados (DCN - *Data Center Network*) têm sido propostas para lidar de forma eficiente com o custo do DC, sua escalabilidade e seus requisitos de comunicação. Entre as novas arquiteturas de DCN mais relevantes, podem ser mencionadas a Fat-tree [2], a BCube [3] e a DCell [4]. Essas arquiteturas possuem diferentes topologias mas compartilham o mesmo objetivo de fornecer uma infraestrutura modular utilizando equipamentos de baixo custo. As topologias de DCN convencionais e a Fat-tree são topologias centradas em comutadores, nas quais apenas comutadores encaminham pacotes; enquanto

a BCube e a DCell são topologias centradas em servidores, nas quais os servidores também participam do encaminhamento. Apesar de a utilização de equipamentos de baixo custo reduzir o CAPEX de um DC, isso pode tornar a rede mais suscetível a falhas [2, 3, 5]. Assim, a médio e longo prazo, alternativas de baixo custo podem acarretar um aumento do OPEX, causado pela necessidade de realizar mais rotinas de manutenção na rede.

O compromisso entre CAPEX e OPEX pode ser mais significativo considerando a tendência atual de instalar DCs em ambientes que possuem difícil manutenção como os que são colocados dentro de contêineres fechados. Por exemplo, centros de dados modulares (MDCs -*Modular Data Centers*) [3] são instalados dentro de contêineres para permitir maior facilidade de instalação e migração física do DC. Nesse caso, reparar ou substituir elementos defeituosos pode ser uma tarefa árdua, devido às restrições de espaço e dificuldade de acesso. Visto isso, a DCN deve permanecer o maior tempo possível operacional sem necessitar de procedimentos de manutenção. Dessa forma, a resiliência a falhas da rede é uma preocupação importante no projeto da arquitetura de redes de centros de dados.

Além da utilização de equipamentos de baixo custo, outro fator que motiva o estudo da resiliência em redes de centros de dados é a intensa adoção de serviços de computação em nuvem. A computação em nuvem está revolucionando a forma na qual os serviços de TI (Tecnologia da Informação) são implementados e utilizados. No Modelo de Infraestrutura como Serviço (IaaS - *Infrastructure as a Service*), os clientes da nuvem terceirizam suas infraestruturas de TI, executando seus serviços dentro de máquinas virtuais (VMs - *Virtual Machines*) hospedadas na infraestrutura física do provedor. Com o modelo IaaS, as empresas clientes reduzem seus custos de instalação e manutenção de serviços de TI, podendo centralizar seus esforços no seu negócio principal. Por exemplo, uma empresa de comércio eletrônico pode hospedar todos seus serviços em VMs de um provedor IaaS, não necessitando construir uma infraestrutura própria de TI. Assim, o provedor IaaS se encarregará da manutenção dos servidores, enquanto a empresa de comércio eletrônico desenvolverá suas aplicações para executar nas VMs hospedadas no provedor.

Ao utilizar serviços IaaS, as empresas abandonam o controle de suas infraestruturas de TI, e então só poderão confiar nesses serviços se os provedores puderem garantir certos níveis de resiliência e desempenho. Assim, para atrair o uso de IaaS, provedores de nuvem geralmente buscam utilizar infraestruturas resilientes de servidores e de rede [6]. Para tal, os provedores empregam bastante redundância na infraestrutura, de forma a superar diversos tipos de falhas, como de *hardware* (p.ex., falhas em discos rígidos, em cabos de rede e em sistemas de refrigeração), de *software* (p.ex., erros de programação) e humanas (p.ex., execução de procedimentos de manutenção incorretos). Essa estratégia, entretanto, não garante a disponibilidade

do serviço em eventos de força maior e de desastres, que estão fora do controle do provedor. Esses eventos são situações causadas por catástrofes naturais ou falhas causadas por agentes humanos não relacionados ao provedor, que podem danificar diversos enlaces da rede além de construções inteiras de um centro de dados [7]. Um exemplo de desastre de larga escala que afetou a comunidade de redes recentemente foi o furacão Sandy em Novembro de 2012: os servidores da *IEEE Communications Society* (web, email, FTP, DNS, etc.) permaneceram completamente desconectados durante 5 dias.

Um provedor IaaS geralmente não se responsabiliza por falhas causadas por eventos que estão fora de seu controle [8]. Mesmo considerando que os provedores IaaS não precisam se responsabilizar por eventos catastróficos, esses podem oferecer serviços de recuperação de desastres, como replicação de VMs e uso de componentes redundantes, visando reduzir o risco de perda de dados e indisponibilidade das VMs nessas situações. Mesmo antes da utilização de serviços de nuvem, muitas empresas utilizavam esquemas de recuperação de desastres para manterem operacionais seus serviços de rede e de computação após situações imprevistas, como desastres e ataques. Como as corporações estão cada vez mais migrando suas infraestruturas de TI para a nuvem utilizando o paradigma IaaS, os provedores de nuvem precisam estar preparados para oferecer serviços resilientes a desastres. Para tal, a nuvem IaaS deve empregar uma infraestrutura de centro de dados geodistribuída, de forma a eliminar pontos únicos de falha e fornecer mecanismos para realizar cópias de segurança (backups) dos serviços em execução. Quanto maior a região utilizada para distribuir o DC, menor o risco de a infraestrutura inteira ser afetada por um pequeno conjunto de falhas inter-relacionadas como, por exemplo, o rompimento de fibras ópticas, quedas de energia ou outros desastres de larga escala [9]. A utilização de uma arquitetura geograficamente distribuída em um DC é de fato uma outra tendência atual, realizada por fatores não necessariamente relacionados à resiliência. Essa tendência consiste em distribuir o DC em alguns poucos sítios de uma rede de longa distância (WAN - *Wide Area Network*), tornando-o mais próximo dos clientes, de forma a reduzir a latência percebida pelos usuários finais [10]. As VMs alocadas para um determinado usuário podem, dessa forma, estar distribuídas em diversas localidades definidas pelo mecanismo de orquestração de recursos da nuvem [11]. Além da maior proximidade dos usuários finais e melhora da resiliência, a geodistribuição do DC possibilita uma ampliação do DC em localidades com limitação da capacidade (p.ex., fornecimento de energia, espaço físico, etc.).

Apesar de a distribuição geográfica possuir efeitos positivos, o projeto desse tipo de DC deve considerar o custo necessário para interconectar sítios em WANs, além do aumento da latência da comunicação entre seus servidores, causado pela maior distância entre eles. O primeiro aspecto geralmente depende de vários fatores ex-

ternos, como a matriz de tráfego do DC e o montante de investimento necessário para sua construção física. O segundo aspecto, entretanto, é de natureza operacional e possui crescente importância em redes para nuvem, visto que um acréscimo na latência de apenas alguns milissegundos pode causar considerável impacto nos serviços fornecidos [12, 13].

1.1 Objetivos

A resiliência de uma DCN depende de sua topologia física e da habilidade de seus protocolos reagirem a falhas. Os protocolos de tolerância a falhas são indispensáveis para garantir boa resiliência da rede, reagindo às diferentes situações de falhas. Entretanto, existem situações de falhas nas quais os protocolos não são efetivos e dependem da organização topológica do DC. Por exemplo, se um servidor do DC se conectar à rede através de apenas um enlace, a falha desse enlace torna impossível encontrar alguma outra rota na DCN para esse servidor. Entretanto, em uma topologia na qual cada servidor se conecta à rede utilizando mais de um enlace, os protocolos de tolerância a falhas podem encontrar novos caminhos para um servidor que possui um determinado enlace com falha. Assim, este trabalho foca na topologia da DCN, pois essa serve como base para a resiliência do DC.

Neste trabalho consideram-se dois tipos de cenário. O primeiro cenário é a rede intra-sítio do DC, ou seja, a rede utilizada para interconectar os servidores do DC dentro de uma mesma localização geográfica (p.ex., uma sala na qual os servidores estão instalados e conectados por uma rede local). Esse cenário é utilizado para o estudo da resiliência a pequenas falhas que podem ocorrer dentro de um sítio, como falhas de comutadores e rompimento de cabos de rede local. O segundo cenário é a rede inter-sítio, necessária para conectar diversos sítios quando o DC é geodistribuído. Esse cenário é utilizado para o estudo da resiliência a falhas de larga escala ou desastres, como a destruição de um sítio completo do DC.

No cenário intra-sítio, analisa-se as topologias de DCN recentemente propostas quando submetidas a falhas. Além disso comparam-se essas novas topologias com uma topologia convencional de DC. Na literatura atual, as topologias de DCN são analisadas em termos de custo [14], escalabilidade [15] e capacidade da rede [3]. Esse último trabalho também analisa a resiliência da DCN, comparando as arquiteturas Fat-tree, BCube e DCell quando a rede está sujeita a falhas de comutadores e servidores. Todavia, como a comparação não é o foco principal do trabalho citado, as topologias são analisadas do ponto de vista de apenas um critério. Além disso, as conclusões apresentadas são limitadas a protocolos de roteamento específicos e a um padrão de tráfego. Assim, como primeiro objetivo, este trabalho de tese fornece uma análise da resiliência da DCN. Essa análise é genérica, sendo independente

de protocolos e fabricantes de equipamento, focando assim nas características topológicas da rede. A motivação para considerar o caso genérico é o crescente uso de equipamentos de prateleira nas DCNs, permitindo que projetistas de DCs possuam uma ampla e heterogênea gama de escolhas de fabricantes. Assim, a análise não é limitada a fabricantes específicos. Em outras palavras, não se utiliza valores de taxas de falhas de equipamentos, que podem variar de acordo com modelo e fabricante. Outra motivação do estudo é o fato de que uma topologia de DC poder ser empregada por diferentes aplicações [16]. Assim, a análise deve ser independente da matriz de tráfego do DC. Os aspectos de resiliência são analisados para as novas topologias Fat-tree, BCube e DCell. Como detalhado posteriormente, foca-se nessas topologias pois elas têm recebido bastante atenção na literatura e também pelo fato de utilizarem equipamentos de baixo custo. Além disso, comparam-se as novas topologias de DCNs com uma topologia convencional em três níveis.

As contribuições desta tese, no que diz respeito às topologias internas ao sítio, são resumidas a seguir:

- Identificam-se as características que tornam as topologias mais vulneráveis ou robustas para um determinado tipo de falhas. Mostra-se que a BCube e a DCell possuem melhor desempenho em comparação à Fat-tree considerando tanto falhas de enlace quanto falhas de comutador, visto que a Fat-tree apresenta algumas vulnerabilidades severas. A topologia BCube é a mais robusta a falhas de enlaces, enquanto a DCell apresenta o melhor desempenho para falhas de comutador. Observa-se também que a robustez a falhas cresce proporcionalmente ao número de interfaces de rede nos servidores para a BCube e a DCell. Finalmente, mostra-se que todas as novas topologias de DCN possuem melhor desempenho que uma topologia convencional de três níveis, tanto para falhas de enlace como para falhas de comutador;

- Caracteriza-se e analisa-se o tempo de vida completo da DCN, tanto de forma teórica quanto através de simulações, verificando o impacto de cada tipo de falha separadamente. A metodologia proposta baseia-se na métrica MTTF (*Mean Time To Failure* - tempo médio até uma falha) e outras métricas de qualidade de caminho e alcançabilidade da DCN. Em particular, são fornecidas fórmulas fechadas para modelar o MTTF das topologias consideradas e prever desconexões de servidores, auxiliando o planejamento de rotinas de manutenção.

No cenário inter-sítio, ou seja, considerando que o DC é geograficamente distribuído, esta tese foca no projeto do DC resiliente a desastres em uma rede WAN. Primeiramente, são descritas as diversas etapas desse tipo de projeto e, em seguida,

são propostas estratégias de posicionamento resiliente de servidores do DC em WANs reais existentes, através da formulação de dois problemas de otimização.

Em um panorama geral, o projeto de DCs geograficamente distribuídos levando em consideração a resiliência começou a ser abordado recentemente na literatura [17–19]. O estado da arte foca no provisionamento de capacidade de fibra óptica entre os sítios. Dessa forma, a proposta de diretrizes para o projeto de DC desta tese busca consolidar a pesquisa nesse domínio e listar diversas direções de pesquisa. Uma das direções listadas, relacionada ao posicionamento de servidores do DC na WAN, culmina nas duas contribuições da tese em relação à formulação de problemas de otimização. O primeiro problema de otimização proposto aborda a otimização conjunta da latência e a resiliência a falhas. Assim, preenche uma lacuna no projeto de DCs geograficamente distribuídos, pois foca nesses dois importantes objetivos, ignorando outros fatores como a capacidade da rede. Os resultados ótimos gerados pelo problema formulado mostram que um nível moderado de resiliência consegue ser alcançado em redes em malha WAN, sem comprometer significativamente a latência. Considerando todas as redes analisadas, o problema proposto encontra configurações de DC que, após a falha em um elemento, mantêm 80% dos servidores disponíveis, enquanto o aumento de latência em relação ao uso de apenas um sítio (resiliência zero) é de 3,6 ms. Por outro lado, o aumento da resiliência, quando esta já possui um nível muito elevado, acarreta em um alto aumento da latência. Por exemplo, aumentar a porcentagem de servidores disponíveis após o pior caso de falhas de 94% para 95% pode levar a um aumento de 46% na latência.

O segundo problema de otimização proposto foca no projeto de DCs resilientes a desastres com perda zero de dados ou de estado das VMs. Basicamente, esse tipo de serviço consiste em VMs que enviam continuamente cópias de backup a um servidor. Nesse caso, uma determinada operação requisitada por um usuário final só será concluída após a VM receber uma confirmação do sítio de backup de que seu estado foi corretamente replicado [13]. Como exige replicação contínua de dados, esse serviço acarreta alta utilização da banda passante da rede. Além disso, como exige confirmação dos backups, demanda baixa latência entre o sítio primário e o de backup. Assim, é proposto um esquema de posicionamento de servidores físicos em centros de dados, que aloca cada servidor primário e seu respectivo backup na rede. O posicionamento leva em consideração um modelo de falhas, de forma que um desastre não destrua um servidor primário e seu backup ao mesmo tempo. Além disso, o posicionamento proposto tira proveito da virtualização para reduzir o número de servidores de backup necessários. A ideia básica é que um servidor de backup necessita instanciar uma VM apenas após um desastre. Assim, ao invés de adotar um esquema no qual cada servidor primário possui um servidor de backup dedicado; no posicionamento proposto, um único servidor de backup poderá ser

compartilhado, recebendo réplicas de diferentes servidores primários. Para tal, esses servidores primários não podem falhar ao mesmo tempo. A partir da análise das topologias WAN, mostra-se que o compartilhamento de backups permite reduzir em pelo menos 40% o número de servidores necessários. Além disso, quantifica-se a capacidade de cada uma das WANs em número de servidores primários que podem ser instalados, o que impacta diretamente o número de VMs suportadas. Requisitos mais estritos de resiliência podem reduzir em pelo menos 50% o número de servidores primários suportados.

Dado o exposto, esta tese possui as seguintes contribuições no cenário inter-sítio:

- Propõem-se diretrizes para projetar uma infraestrutura de DCN que suporte uma nuvem IaaS resiliente a desastres, permitindo a identificação de importantes direções de pesquisa em computação em nuvem;
- Formula-se um problema de otimização que permite quantificar o aumento da latência do DC quando sua resiliência é melhorada a partir da distribuição geográfica;
- Formula-se um problema de otimização para planejar a implantação de um serviço de backup de VMs em redes WAN existentes, analisando os compromissos da implantação desse tipo de serviço.

1.2 Organização do Texto

Esta tese está organizada da seguinte forma. O Capítulo 2 apresenta detalhes dos cenários considerados, explicando o modelo de rede utilizado e introduzindo as topologias intra-sítio utilizadas. O Capítulo 3 apresenta a metodologia e os resultados da comparação realizada entre as topologias intra-sítio. O Capítulo 4 propõe as diretrizes para o projeto de DC resiliente a desastres, introduzindo a parte da tese dedicada ao projeto da rede inter-sítio do DC. Baseando-se nessas diretrizes, os Capítulos 5 e 6 descrevem os problemas de otimização propostos para o projeto de DCs geodistribuídos, apresentando os resultados e análises correspondentes. Por fim, o Capítulo 7 apresenta as conclusões da tese e direções futuras.

Capítulo 2

Modelagem das Redes de Centros de Dados

Este capítulo aborda o modelo de rede utilizado neste trabalho. Assim, a Seção 2.1 detalha as topologias intra-sítio e a Seção 2.2 fornece o modelo geral do DC completo, que consiste em um DC formado por um ou mais sítios.

2.1 Topologias de DCN Intra-sítio

As topologias de DCN intra-sítio podem ser classificadas como estruturadas ou não-estruturadas. As topologias estruturadas possuem uma regra de formação determinística e são construídas a partir de módulos básicos. As redes baseadas nessas topologias podem ser construídas utilizando apenas enlaces de cobre (p.ex., Ethernet Gigabit), como a topologia convencional em três níveis, a Fat-tree, a BCube e a DCell; ou podem ser híbridas, utilizando também enlaces ópticos para melhorar a eficiência energética e a capacidade da rede, como as topologias C-Through e Helios [20]. Por sua vez, as topologias não-estruturadas não possuem uma lei de formação determinística. Essas topologias podem ser formadas a partir de um algoritmo estocástico (p.ex., Jellyfish [21]) ou a partir de um problema de otimização (p.ex., REWIRE [22]). As topologias não-estruturadas permitem uma melhor capacidade de expansão, pois podem ser estendidas de forma incremental por não possuírem uma estrutura rígida. Neste trabalho a análise é focada nas topologias estruturadas, visto que elas vêm recebendo uma maior atenção na literatura atual [23, 24], detalhadas a seguir.

2.1.1 Three-layer

A maioria das DCNs atuais emprega uma topologia hierárquica convencional, composta de três níveis: a borda, a agregação e o núcleo [1]. Não existe uma de-

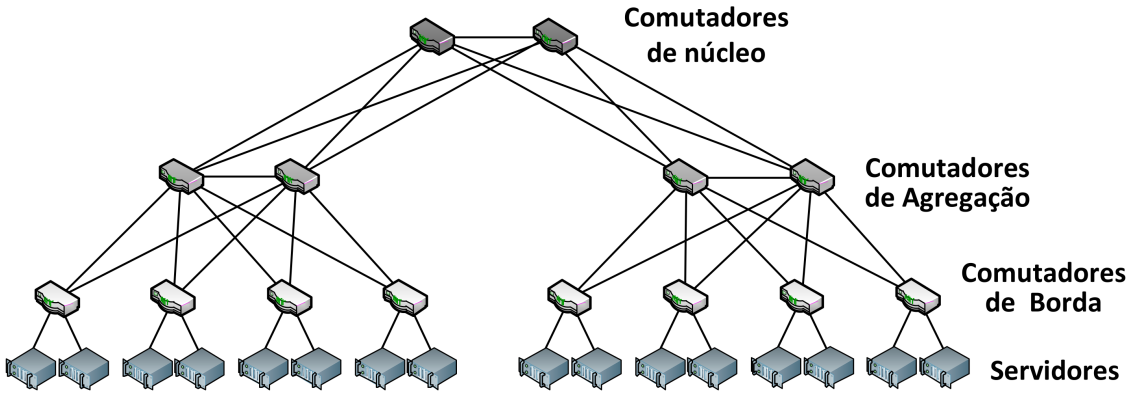


Figura 2.1: Topologia Three-layer como 2 portas de borda ($n_e = 2$) e 4 portas de agregação ($n_a = 4$).

finição única na literatura para uma topologia de DC convencional em três níveis, já que a organização topológica depende fortemente das decisões de projeto e das especificações de equipamento. Assim, define-se neste trabalho a topologia Three-layer (três níveis) baseando-se na arquitetura de DCN recomendada em [1]. Na topologia Three-layer o nível de núcleo é composto de dois comutadores diretamente conectados entre si, que atuam como *gateways* do DC. Cada comutador de núcleo está conectado a todos os comutadores de agregação. Os comutadores de agregação estão organizados em pares. Em cada par os comutadores de agregação estão diretamente conectados entre si, além de estarem conectados ao mesmo grupo de n_a comutadores de borda, como visto na Figura 2.1. Cada comutador de borda possui n_e portas conectadas diretamente aos servidores. Assim, cada par de comutadores de agregação fornece conectividade a $n_a * n_e$ servidores, sendo necessários $\frac{|\mathcal{S}|}{n_a * n_e}$ pares para formar um DC com $|\mathcal{S}|$ servidores. A Figura 2.1 mostra um exemplo de uma topologia Three-layer com 2 pares de comutadores de agregação, com $n_a = 4$ e $n_e = 2$, possuindo $2x4x2 = 16$ servidores.

Os comutadores de borda em DCNs comerciais geralmente estão conectados aos servidores por portas 1 Gbps Ethernet. Por outro lado, as portas que conectam os comutadores de agregação ao núcleo e aos comutadores de borda são geralmente 10 Gbps Ethernet. Assim, as topologias do tipo Three-layer empregam equipamentos de alta capacidade nos níveis de núcleo e de agregação. As novas topologias de DC propõem melhorias topológicas para permitir a utilização de comutadores de prateleira por toda a rede, como descrito a seguir.

2.1.2 Fat-tree

Neste trabalho, denomina-se como Fat-tree a topologia proposta por Al-Fares *et al.* em [2]. Al-Fares *et al.* utilizam o conceito de fat-tree, que é um caso especial de uma rede de Clos [25], para definir uma topologia de DC organizada na forma

de uma árvore n-ária. O conceito de rede de Clos foi originalmente idealizado para o projeto de matrizes de comutação telefônica. Em uma rede de Clos é sempre possível realizar a conexão entre dois terminais inativos, independente de quantas conexões estão ativas na matriz de comutação. Além disso, no projeto original [25], a rede de Clos mantém essa característica sem a necessidade de rearranjar conexões já estabelecidas. Uma rede com essas propriedades é chamada de estritamente não-blocante. A Fat-tree, como mostrado na Figura 2.2, possui dois tipos de conjuntos: o núcleo e os *pods*. O núcleo é formado por comutadores que possuem cada uma de suas portas conectada a um *pod* diferente. O *pod* é formado por comutadores de agregação e de borda, e também pelos servidores do DC. Os comutadores de agregação realizam a conexão entre o *pod* e o núcleo e possuem conexão com os comutadores de borda e os de núcleo. Já os comutadores de borda possuem ligações com um conjunto diferente de servidores. Todos os comutadores da rede são idênticos e possuem n portas. Assim, a rede possui n *pods*, sendo que cada *pod* possui $\frac{n}{2}$ comutadores de agregação e outros $\frac{n}{2}$ de borda. Em um *pod*, cada comutador de agregação está ligado a todos os comutadores de borda, que estão individualmente ligados a $\frac{n}{2}$ servidores diferentes. Assim, a topologia Fat-tree possui capacidade para $\frac{n}{2} * \frac{n}{2} * n = \frac{n^3}{4}$ servidores. A Figura 2.2 mostra uma Fat-tree para $n = 4$. Note que o Servidor de índice 0 no *Pod 0* ($S_{0.0}$) está se comunicando com o Servidor 1 ($S_{1.0}$) no seu mesmo *pod*, sendo que ambos estão conectados através do mesmo comutador de borda. Já o Servidor 3 do *Pod 0* ($S_{3.0}$) se comunica com um servidor em outro *pod*, $S_{3.1}$, e por isso mesmo precisa utilizar os comutadores de núcleo. A Fat-tree permite que todos os servidores se comuniquem ao mesmo tempo utilizando a capacidade total de suas interfaces de rede. Entretanto, diferente do projeto original da rede de Clos, a Fat-tree é rearranjável não-blocante, o que significa que as conexões devem ser rearranjadas para permitir essa característica. Em termos práticos, isso significa que, considerando padrões arbitrários de tráfego, alguns enlaces da rede poderão ser saturados. Outra característica da Fat-tree é que todos os elementos de rede são idênticos, não necessitando de comutadores de alto custo com grande número de portas em níveis mais altos da hierarquia da árvore. Note também que a Fat-tree utiliza um núcleo mais redundante do que a topologia Three-layer.

A arquitetura VL2 [26] também usa uma rede de Clos e não foi utilizada neste trabalho por se assemelhar com a Fat-tree [14]. Porém, diferente da Fat-tree, a VL2 não é completamente formada por elementos idênticos, possuindo maior capacidade nos enlaces do núcleo do que na borda da rede.

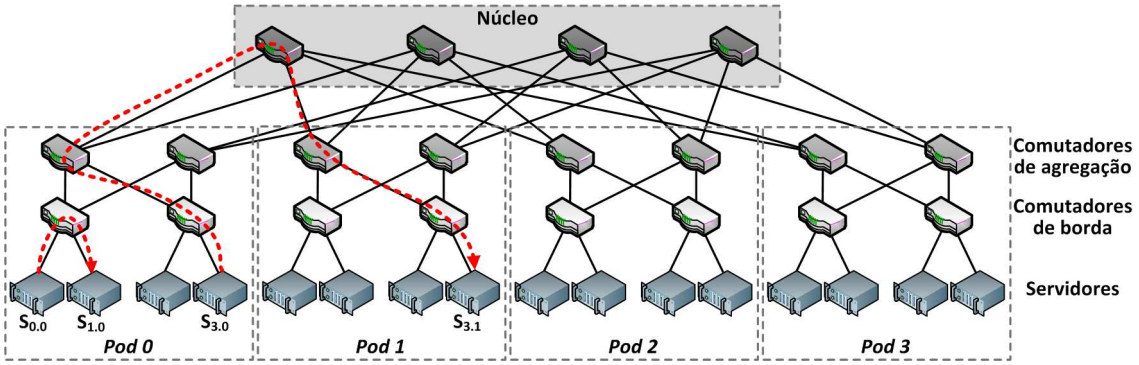


Figura 2.2: Fat-tree com comutadores de 4 portas ($n = 4$).

2.1.3 BCube

A topologia BCube [3] foi proposta para utilização em DCs modulares (MDCs), que são geralmente montados em contêineres. Os MDCs possuem a facilidade de migração entre localidades geográficas, assim como a montagem sob demanda. Essa migração é útil em termos de economia de energia, possibilitando o transporte do DC para regiões com menor custo de refrigeração em uma determinada época ou para atender melhor a demanda por um serviço em uma localidade específica. Por exemplo, um MDC pode ser utilizado para suprir uma demanda temporária de poder computacional, como em eventos. Por serem montadas em contêineres fechados e com alta densidade de equipamentos, essas redes possuem manutenção difícil, necessitando de uma alta tolerância a falhas. Assim, é desejável que o desempenho diminua lentamente com o aumento do número de falhas de seus equipamentos. Além disso, como no caso da Fat-tree, a rede precisa ter alta capacidade de transferência de bits e baixo custo devido ao grande tamanho do DC. Para tal, a topologia BCube emprega mini-comutadores e servidores que participam do encaminhamento. Esses servidores devem, então, possuir mais de uma interface de rede, que tipicamente não é maior que cinco [3].

A topologia BCube é definida de forma recursiva, sendo uma rede $BCube_n$ constituída por n redes do tipo $BCube_{n-1}$. O componente fundamental da topologia é uma rede $BCube_0$, composta por um único comutador de n portas ligado a n servidores. Para a construção de uma $BCube_1$ utilizam-se n redes $BCube_0$ e n comutadores. Cada comutador é conectado a todas as redes $BCube_0$ através da conexão com um dos servidores de cada $BCube_0$. A Figura 2.3 ilustra uma rede $BCube_1$. De forma mais geral, uma rede $BCube_l$ ($l \leq 1$) é formada por n unidades $BCube_{l-1}$ e n^l comutadores de n portas. Para construir uma $BCube_l$ numera-se as n unidades $BCube_{l-1}$ de 0 a $n-1$ e os servidores de cada uma delas são numerados de 0 até $n^l - 1$. Em seguida, conecta-se a porta de nível l do i-ésimo servidor ($i \in [0, n^l - 1]$), situado no j-ésimo $BCube_l$ ($j \in [0, n-1]$), à j-ésima porta do i-ésimo comutador de nível

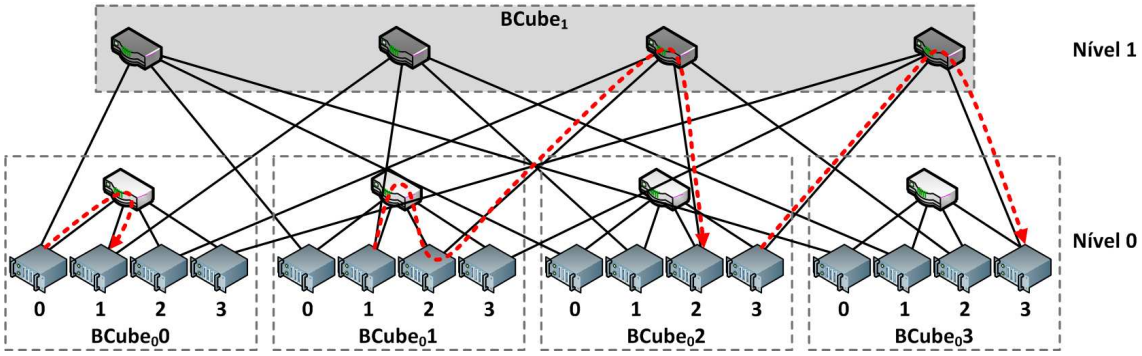


Figura 2.3: BCube com comutadores de 4 portas ($n = 4$) e servidores com 2 interfaces de rede ($l = 1$).

l. Uma rede $BCube_l$ possui capacidade de n^{l+1} servidores. Note na Figura 2.3 que no $BCube_0 0$, o Servidor 0 se comunica através de um comutador com o Servidor 1 na mesma rede. Já o Servidor 1 do $BCube_0 1$ utiliza o comutador da rede para encaminhar os seus dados para o Servidor 2 que é quem possui o enlace até a rede do destino, no caso, o $BCube_0 2$. Entretanto, $BCube$s diferentes que estão em um mesmo nível podem se comunicar envolvendo também apenas um comutador de nível superior, como é o caso do Servidor 3 do $BCube_0 2$ com o Servidor 3 do $BCube_0 3$. Logo, diferente da Fat-tree, os servidores podem participar do encaminhamento de dados, dependendo das suas posições na topologia. Note que na BCube os servidores participam do encaminhamento mas não estão diretamente conectados.

2.1.4 DCell

Assim como a Fat-tree e a BCube, a DCell [4] foi proposta para prover alta capacidade de transferência e tolerância a falhas. Da mesma forma que a BCube, a DCell é definida recursivamente e utiliza encaminhamento pelos servidores e mini-comutadores. O componente fundamental da topologia é a rede $DCell_0$ constituída, assim como a $BCube_0$, por um comutador ligado a n servidores. Constrói-se uma $DCell_1$ utilizando $n + 1$ redes $DCell_0$, na qual duas $DCell_0$ estão conectadas entre si através de um enlace formado por um de seus servidores. Um exemplo de $DCell_1$ está ilustrado na Figura 2.4. Note que as comunicações internas à célula são realizadas localmente através do comutador, como visto na comunicação entre o Servidor 2 e 3 da $DCell_0 0$. Já a comunicação entre servidores em células diferentes ou são realizadas de maneira direta, como a comunicação entre o Servidor 1 na $DCell_0 2$ e o Servidor 2 na $DCell_0 3$, ou através de uma combinação de servidores e comutadores, como visto na comunicação entre o Servidor 1 na $DCell_0 1$ e o Servidor 1 na $DCell_0 4$. Note que, nesse último exemplo, o caminho total percorrido é o maior já visto entre as topologias apresentadas, englobando dois comutadores e dois servidores. Esse comportamento é confirmado nos resultados deste trabalho.

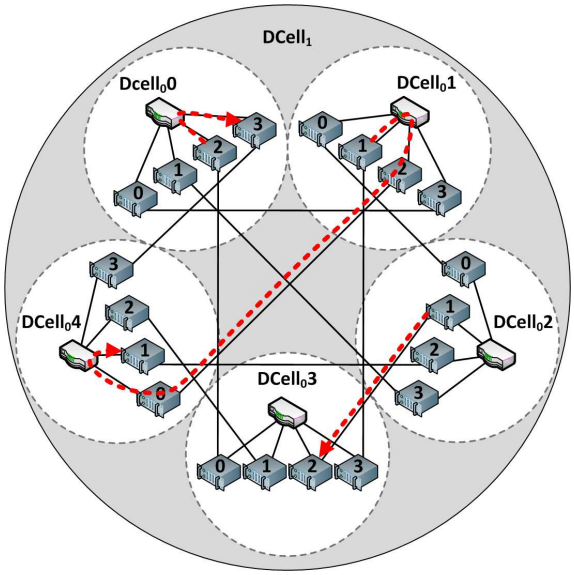


Figura 2.4: DCell com comutadores de 4 portas ($n = 4$) e servidores com 2 interfaces de rede ($l = 1$).

Observa-se na arquitetura DCell que, diferentemente da BCube, os comutadores apenas estão conectados aos servidores de seu DCell e a ligação direta entre diferentes redes DCell é sempre realizada através dos servidores. Para a construção de uma $DCell_l$ são necessárias $n + 1$ redes $DCell_{l-1}$. Cada servidor em uma rede $DCell_l$ possui $l + 1$ enlaces, sendo que em cada servidor, o primeiro enlace (enlace de nível 0) é conectado ao comutador da $DCell_0$ que ele faz parte. Já o segundo enlace conecta o servidor a um nó de uma mesma $DCell_1$, mas em uma $DCell_0$ vizinha. Genericamente, o enlace de nível i de um servidor o conecta a uma $DCell_{i-1}$ vizinha dentro de uma mesma $DCell_i$. O procedimento de construção é mais complexo que o da rede BCube, sendo executado a partir de um algoritmo de formação proposto por Guo *et al.* [4].

A capacidade da rede em número de servidores pode ser calculada de forma recursiva utilizando as seguintes relações: $g_l = t_{l-1} + 1$ e $t_l = g_l \times t_{l-1}$, onde g_l é o número de redes $DCell_{l-1}$ no $DCell_l$ e t_l é o número de servidores no $DCell_l$. A $DCell_0$ é um caso especial na qual $g_0 = 1$ e $t_0 = n$ [4].

2.2 Modelo do DC Completo

Considerando o DC completo, adota-se o seguinte modelo de rede:

- o DC é formado por um ou mais sítios espalhados em uma região geográfica;
- em um sítio, os servidores são interligados por uma topologia intra-sítio como, por exemplo, as citadas na Seção 2.1;

- a rede inter-sítio é uma WAN;
- os usuários do DC acessam os serviços através de *gateways* espalhados pela rede.

A Figura 2.5 exemplifica o modelo acima. Note que, apesar de todos os sítios utilizarem uma simples árvore como topologia intra-sítio, outras topologias poderiam ser utilizadas como a Fat-tree, BCube e DCell. Nesse exemplo, os servidores de um sítio estão organizados em bastidores. Os servidores de cada bastidor são interconectados através de um comutador de topo de bastidor (ToR - *Top-of-Rack*). Em cada sítio o comutador de núcleo é responsável por fornecer o acesso externo, conectando o sítio aos demais sítios e fornecendo acesso aos *gateways*. Note que, apesar de no exemplo cada sítio possuir apenas um comutador de núcleo, as suas funções podem ser realizadas por diversos comutadores e roteadores.

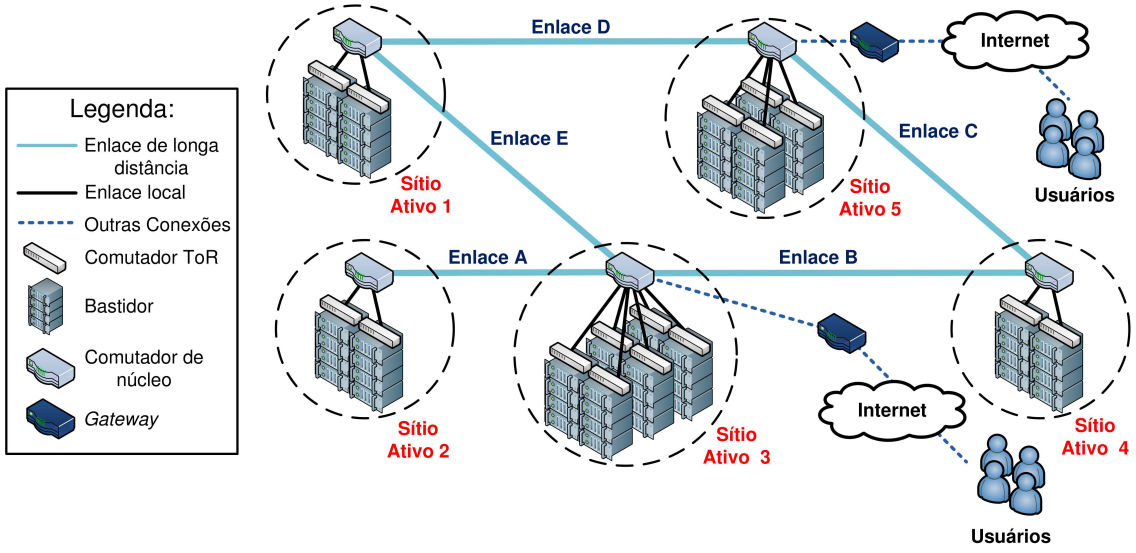


Figura 2.5: Exemplo de um DC geodistribuído, composto por diferentes sítios.

Na abordagem intra-sítio do Capítulo 3, considera-se que o DC é formado por apenas um sítio e adota-se uma das topologias descritas na Seção 2.1. Por outro lado, as abordagens inter-sítio dos Capítulos 4, 5 e 6 ignoram a topologia intra-sítio, considerando apenas aspectos da WAN. A partir do Capítulo 4 serão detalhadas mais características do modelo específicas da abordagem inter-sítio. O próximo capítulo apresenta a abordagem intra-sítio, que consiste na análise de resiliência das topologias intra-sítio Three-layer, Fat-tree, BCube e DCell. Essa análise apresenta o comportamento de cada topologia quando submetida a falhas de enlace, comutador ou servidor.

Capítulo 3

Análise da Resiliência em Topologias Intra-sítio

Neste capítulo, considera-se a análise de topologias internas ao sítio, descritas na Seção 2.1. Dessa forma, o DC é considerado como formado por apenas um sítio. A análise deste capítulo considera a resiliência utilizando os aspectos confiabilidade e sobrevivência. O primeiro quantifica o tempo que o DC permanece com todos seus servidores alcançáveis [27], enquanto o segundo quantifica métricas de desempenho quando o DC é submetido a uma determinada situação de falhas [28]. Este Capítulo está organizado da seguinte forma. A Seção 3.1 descreve a metodologia de análise proposta. As métricas de confiabilidade e suas respectivas análises estão presentes na Seção 3.2, enquanto a Seção 3.3 apresenta as métricas e análises do ponto de vista da sobrevivência. A Seção 3.4 resume os resultados obtidos com uma análise qualitativa das topologias intra-sítio. A Seção 3.5 aborda a sensibilidade das métricas utilizadas de acordo com a escolha dos *gateways*. Por fim, a Seção 3.6 descreve os trabalhos relacionados.

3.1 Metodologia

À medida que o tempo de operação de um DC avança, mais elementos de rede tendem a falhar, sendo assim esperada uma redução no número de servidores alcançáveis [29, 30]. *Um servidor é considerado como desconectado, ou indisponível, quando este não possui caminhos para os gateways da DCN, isto é, aos comutadores que fornecem acesso às rede externas, como a Internet.* Neste capítulo, analisa-se o tempo de vida completo de um DC, considerando falhas de um dado tipo de elemento: enlace, comutador ou servidor. Cada tipo de falha é avaliado separadamente de forma a analisar sua influência individual na rede. Independentemente do tipo de elemento, o tempo de vida do DC é definido como *a quantidade de tempo até*

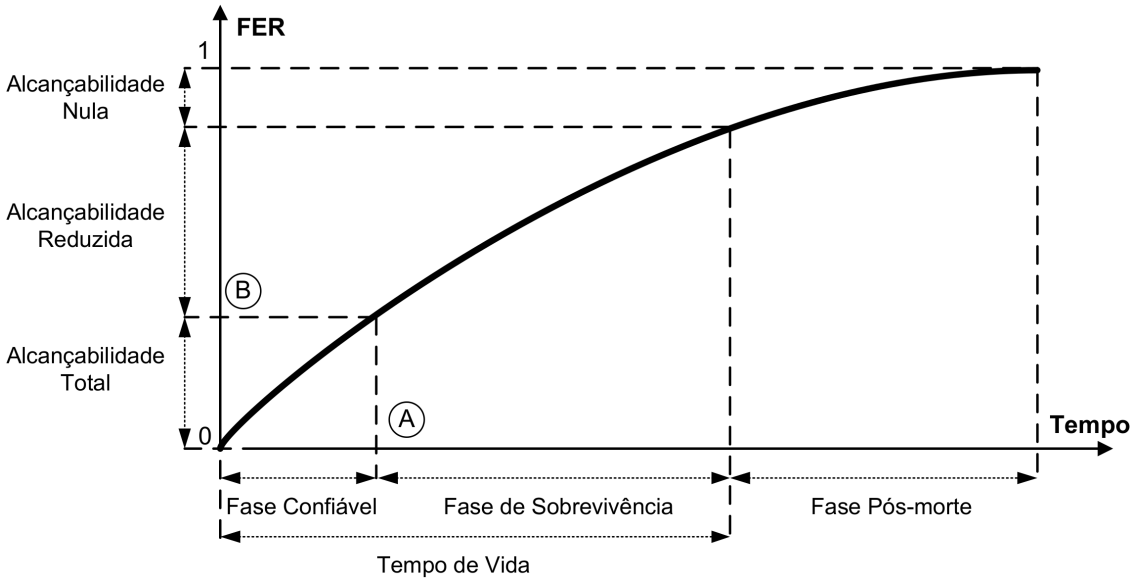
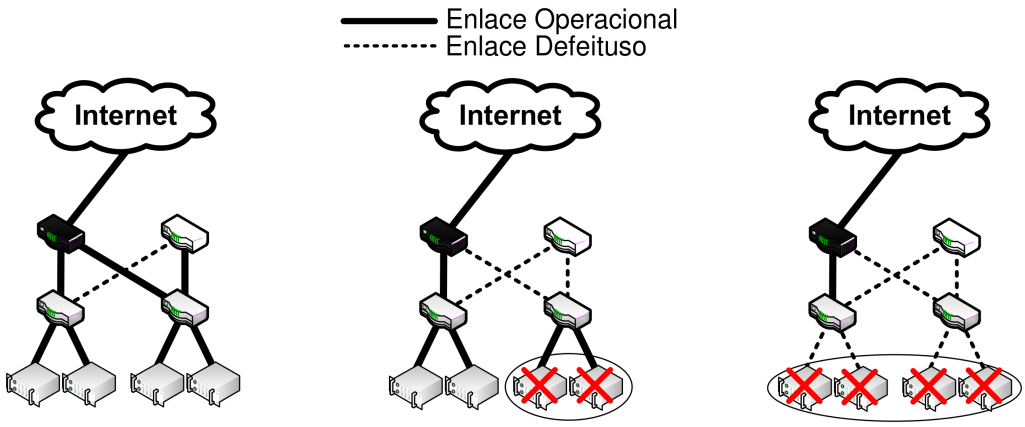


Figura 3.1: Evolução da alcançabilidade dos servidores. À medida que os elementos de rede falham, mais servidores são desconectados e assim a alcançabilidade do serviço diminui.

a desconexão de todos os seus servidores. Para quantificar a extensão das falhas, define-se neste capítulo a *Fração de Elementos Defeituosos* (FER - *Failed Elements Ratio*), que consiste na quantidade de elementos defeituosos de um determinado tipo (enlace, comutador ou servidor), normalizada pelo número total de elementos desse tipo. Se nenhuma manutenção é executada no DC, que é o caso considerado neste trabalho, a FER para um determinado tipo de elemento aumentará ao longo do tempo, significando que mais elementos de rede apresentarão falhas. A Figura 3.1 ilustra uma situação hipotética da evolução da FER ao longo do tempo e demais fases associadas, detalhadas a diante.

O tempo de vida tem início a partir do momento da realização de manutenção completa da DCN, isto é, da restauração de todos elementos de rede. A partir disso, considera-se que o DC se encontra na *Fase Confiável*, na qual todos os servidores estão disponíveis, mesmo após algumas falhas de elementos de rede. A partir do momento que uma falha causa desconexão de pelo menos um servidor, o DC entra na *Fase de Sobrevivência*. O fim da *Fase de Sobrevivência* coincide com o fim do tempo de vida do DC, entrando assim na *Fase Pós-morte*, na qual todos os servidores estão desconectados mesmo se alguns elementos de rede ainda se encontram ativos. A Figura 3.2 mostra cada fase de uma rede hipotética, considerando apenas falhas de enlace. Nessa figura, cada enlace defeituoso é representado por uma linha tracejada, enquanto um servidor inalcançável está marcado com uma cruz e o comutador que atua como *gateway* está pintado de preto. A fração desconectada do DC está circulada na figura. Nota-se que na *Fase Confiável* a DCN pode possuir enlaces com falha e na *Fase Pós-morte* pode possuir enlaces que ainda estão operacionais.



(a) Fase Confiável. (b) Fase de Sobrevivência. (c) Fase Pós-morte.

Figura 3.2: Exemplo das diferentes fases que um DC percorre, quando suscetível a falhas de enlace.

Em relação à *Fase Confiável*, as letras circuladas na Figura 3.1 apontam duas medidas de interesse desta primeira análise, que são:

- **A**: Indica o tempo decorrido até a primeira desconexão de servidor, chamado de TTF (*Time To Failure* - tempo até uma falha). Neste capítulo calcula-se o valor médio dessa métrica, chamado de MTTF (*Mean Time To Failure* - Tempo médio até uma falha), que é o valor esperado do TTF em uma rede (isto é, tempo médio decorrido até a primeira desconexão de servidor).

- **B**: Indica o menor valor de FER que produz uma desconexão de servidores. Neste capítulo, calcula-se essa métrica a partir de seu valor médio, chamado de *FER Crítica*. Por exemplo, em uma rede com 100 comutadores na qual os servidores são desconectados, em média, após a remoção aleatória de 2 comutadores, possui uma FER crítica de $\frac{2}{100} = 0,02$. O tempo médio para a rede apresentar a FER Crítica é então o MTTF.

A Fase de Sobrevivência merece atenção especial por quantificar a degradação da rede. Para essa fase, a Seção 3.3 define e analisa duas métricas representativas: Alcançabilidade do Serviço e Qualidade dos Caminhos.

3.1.1 Falhas de Nó e Enlace

Modelo de Falhas

Para topologias intra-sítio, o modelo de falhas utilizado neste trabalho é baseado nas seguintes considerações:

- **Isolamento das falhas:** Cada tipo de falha (enlace, comutador ou servidor) é analisado separadamente. Isso é importante para quantificar o impacto de

um determinado elemento nas topologias consideradas.

- **Probabilidade de falha:** Para simplificar, todos os elementos possuem a mesma probabilidade de falhas, e as falhas são independentes entre si.

- **Reparos:** Os elementos não sofrem reparos durante o funcionamento do DC. Isso é importante para possibilitar o estudo de quanto tempo a rede pode operar sem sofrer manutenção (p.ex., Centros de Dados Modulares).

Métricas de Falha

Neste capítulo, as falhas são analisadas de forma espacial e temporal a partir das seguintes métricas:

Fração de Elementos Defeituosos (FER). Definida anteriormente, essa métrica quantifica apenas a extensão das falhas e não depende da distribuição de probabilidade do tempo de vida dos elementos. Em seguida, também utiliza-se o termo mais específico "Fração de Enlaces/Comutadores/Servidores Defeituosos" para enfatizar o tipo de falha.

Tempo Decorrido. À medida que o tempo passa, mais elementos podem falhar. Nesse caso, o tempo decorrido desde a última manutenção completa pode caracterizar indiretamente o estado de falhas da rede. Para uma dada FER, existe um tempo esperado para o qual essa fração de falhas irá aparecer. O tempo pode ser definido de duas formas: Absoluto e Normalizado. Na primeira, mede-se o tempo em horas, dias ou meses. Na segunda, normaliza-se o tempo pelo tempo médio de vida de um nó ou enlace individual, como detalhado adiante. Essa medida é importante para tornar a análise independente do tempo médio de vida dos elementos, sendo assim independente das características dos equipamentos de rede utilizados.

3.1.2 Simulação de Falhas

A topologia do DC é modelada como um grafo $G = (\mathcal{V}, \mathcal{E})$, onde $\mathcal{V}$ é o conjunto de servidores e comutadores, e $\mathcal{E}$ é o conjunto de enlaces. O conjunto $\mathcal{V}$ é dado por $\mathcal{V} = \mathcal{S} \cup \mathcal{C}$, onde $\mathcal{S}$ é o conjunto de servidores e $\mathcal{C}$ o conjunto de comutadores. Os pesos dos enlaces são unitários pois todas as topologias consideradas utilizam apenas um tipo de enlace. Para simular o cenário de falhas da Seção 3.1.1, escolhe-se aleatoriamente um conjunto de elementos $\mathcal{S}'$, $\mathcal{C}'$, ou $\mathcal{E}'$ para remover do grafo G, onde $\mathcal{S}' \subset \mathcal{S}$, $\mathcal{C}' \subset \mathcal{C}$ e $\mathcal{E}' \subset \mathcal{E}$, gerando o subgrafo G'. Note que cada conjunto de elementos (comutadores, servidores e enlaces) é analisado separadamente. As métricas de interesse são assim calculadas utilizando o grafo G'. Salvo indicação contrária, todas as métricas de interesse são valores médios e possuem um intervalo confiança com nível de 95%. A geração de topologia, simulação de falhas e cálculo de

métricas são obtidos utilizando o NetworkX [31]. Essa é uma ferramenta de análise de grafos que permite, entre outras funções, o cálculo dos caminhos mais curtos entre os nós do grafo e a análise de conectividade entre os nós quando arestas e outros nós são removidos. Assim, a simulação e cálculos de métricas são realizados através de *scripts* em Python que fazem chamadas aos métodos do NetworkX.

A análise tem início removendo aleatoriamente f elementos de G, onde $0 \leq f \leq F$ e F é quantidade total de elementos do tipo em análise (enlace, comutador ou servidor) presente no grafo original G. Após isso, calculam-se as métricas de interesse para esse determinado f. As métricas de falhas FER e Tempo Decorrido (Seção 3.1.1) são calculadas, respectivamente, por $\frac{f}{F}$ e pelo tempo médio que f elementos falham, dado que existem F possibilidades de falhas (isto é, o número total de elementos de um determinado tipo). Como visto na Seção 3.1.1, o Tempo Decorrido pode ser definido tanto pelo Tempo Absoluto como também pelo Tempo Normalizado. Para calcular ambas quantidades de tempo, é necessário primeiro definir a distribuição de probabilidade de falha dos elementos. Para simplificar e seguindo uma abordagem amplamente utilizada, considera-se que as falhas são independentes e que o instante τ no qual um elemento falha é aleatório e segue uma distribuição exponencial dada pela CDF (*Cumulative Distribution Function* - Função de Distribuição Acumulada) $Z(t; E[\tau]) = 1 - e^{\frac{t}{E[\tau]}}, t \geq 0$, onde $E[\tau]$ é o valor do tempo médio no qual cada falha individual ocorre [32, 33]. A partir disso, o tempo médio para existirem f elementos defeituosos (Tempo Decorrido) é dado pelas Estatísticas de Ordem (*Order Statistics*). Para tal, considerando todos os F elementos, ordena-se de forma crescente seus instantes de falha da seguinte forma $Y_{1:F}, Y_{2:F}, \cdots, Y_{F:F}$, onde $Y_{f:F}$ é uma variável aleatória indicando o instante τ da f-ésima falha, com $f \leq F$. Essa ordenação consiste nas Estatísticas de Ordem da distribuição $Z(t; E[\tau])$, e o valor esperado da variável aleatória $Y_{f:F}$, dado por $E[Y_{f:F}]$, é o tempo médio para ocorrerem f falhas. Assim, utilizando a expressão do valor esperado de $E[Y_{f:F}]$ para uma distribuição exponencial com média $E[\tau]$, demonstrada em [34], define-se o Tempo Absoluto da seguinte forma:

$$AT = E[Y_{f:F}] = E[\tau] \sum_{i=0}^{f-1} \frac{1}{F-i}, \text{ para } f \leq F. \tag{3.1}$$

Note que a Equação 3.1 pode se tornar independente de $E[\tau]$, dividindo o termo à direita por $E[\tau]$. O resultado é o Tempo Normalizado dado por:

$$NT = \sum_{i=0}^{f-1} \frac{1}{F-i}, \text{ para } f \leq F. \tag{3.2}$$

3.1.3 Sub-redes Operacionais Após Falhas

Nesta análise, é necessário identificar se uma rede está operacional para calcular as métricas de interesse. Como as falhas podem dividir a rede, define-se como operacionais todas as sub-redes que possuem acesso a pelo menos um *gateway*. Neste trabalho, chama-se de *gateway* o comutador que é responsável pelo acesso à rede exterior ao DC. Na prática, a função de *gateway* é executada por um roteador conectado a esse comutador. Esse nó possui um papel fundamental, visto que é responsável por conectar o DC com redes externas, como a Internet. Assim, uma sub-rede que não possui acesso externo não é considerada como operacional, pois não pode receber chamadas remotas para atribuir tarefas a seus servidores. Um servidor em uma rede operacional é considerado como *alcançável*.

As definições típicas de topologias de DCN geralmente não especificam os *gateways* do DC. Assim, assume-se nesta análise que todos os comutadores do mais alto nível hierárquico de cada topologia são responsáveis por essa tarefa. Dessa forma, para as topologias consideradas têm-se os possíveis *gateways*:

- **Three-layer:** Os dois comutadores de núcleo.
- **Fat-tree:** Todos os comutadores de núcleo.
- **BCube:** Para uma BCube de nível l, todos os comutadores de nível l.
- **DCell:** Como não há hierarquia de comutadores nessa topologia, considera-se que todos os comutadores estão no nível mais alto e então podem ser *gateways*.

Uma possível desvantagem da escolha apresentada acima pode ser o fato da comparação entre as topologias ser injusta dependendo de quantos *gateways* são escolhidos para cada uma. Assim define-se uma métrica de referência denominada Densidade de Portas de Gateway (GPD - *Gateway Port Density*):

$$GPD = \frac{n * g}{|\mathcal{S}|}, \tag{3.3}$$

onde n é o número de portas por comutador, g é o número de *gateways* e $|\mathcal{S}|$ é o número total de servidores na rede. A GPD fornece uma ideia do número de portas de comutador por servidor disponíveis nos *gateways*. Como um *gateway* possui n portas, o DC possui $n * g$ portas atuando como o último acesso do tráfego antes de sair do DC. Note que não contabiliza-se em n o número de portas que conectam o *gateway* às redes externas, uma vez que n é o número de portas de comutador presente na definição de cada topologia (Seção 2.1). Assim, assume-se que cada *gateway* possui uma ou mais portas que fornecem acesso externo. Além disso, assume-se que essas portas não são suscetíveis a falhas. A GPD máxima (isto

é, se todos os comutadores possíveis são utilizados) das topologias Fat-tree, BCube e DCell é igual a 1. Como a topologia Three-layer utiliza apenas dois comutadores de núcleo, seu GPD máximo é muito pequeno (p.ex., 0,007 para uma rede com 3.456 servidores). Assim, salvo indicação contrária, o estudo apresentado neste capítulo considera que todos os possíveis *gateways* são utilizados nas topologias analisadas. Neste trabalho não equaliza-se o GPD de todas as topologias com a Three-layer de forma a possibilitar uma melhor comparação entre as novas topologias. Além disso, mostra-se mais adiante ainda neste capítulo que essa escolha não muda as conclusões em relação à comparação entre a topologia Three-layer e as novas topologias.

3.2 Fase Confiável

A Fase Confiável corresponde ao período até a desconexão do primeiro servidor. Assim, ela quantifica o tempo que o administrador do DC pode esperar até a próxima manutenção, para tornar o DC completamente disponível. Neste capítulo, analisa-se a Fase Confiável de forma teórica e por simulação, como detalhado adiante.

3.2.1 Análise Teórica

O MTTF pode ser calculado como função da confiabilidade $R(t)$. $R(t)$ é definida como a probabilidade de a rede estar na Fase Confiável (isto é, todos seus servidores estão alcançáveis) no instante t. Em outras palavras, considerando que a quantidade de tempo que o DC permanece na Fase Confiável é uma variável aleatória T, a confiabilidade é definida como $R(t) = P(T > t) = 1 - P(T \leq t)$. Note que $P(T \leq t)$ é a CDF da variável aleatória T. Como o MTTF é o valor esperado de T, dado por $E[T]$, é possível utilizar a definição de valor esperado para variáveis aleatórias contínuas não negativas como a seguir:

$$MTTF = \int_0^\infty 1 - P(T \leq t)\, dt = \int_0^\infty R(t)\, dt. \tag{3.4}$$

Neste trabalho, $R(t)$ é calculado utilizando a aproximação Burtin-Pittel [27] para confiabilidade de redes, dada por:

$$R(t) = 1 - \frac{t^r c}{E[\tau]^r} + O\left(\frac{1}{E[\tau]}^{r+1}\right) \approx e^{-\frac{t^r c}{E[\tau]^r}}, \tag{3.5}$$

onde c e r são, respectivamente, o número de conjuntos de corte mínimo e o tamanho desses conjuntos. Um conjunto de corte mínimo é um conjunto com o número mínimo possível de elementos que, se removidos, causam desconexão de servidor. Por exemplo, considerando apenas falhas de enlace na rede da Figura 2.1, um conjunto

de corte mínimo consiste no enlace entre um servidor e um comutador de borda. Considerando apenas falhas de comutador na Figura 2.1, um conjunto de corte mínimo é um comutador de borda. O tamanho do corte mínimo é o número de elementos (enlace, comutadores ou servidores) em um único conjunto (p.ex., igual a 1 nos exemplos mencionados anteriormente). Na Equação 3.5, $\frac{t^r c}{E[\tau]^r}$ é a contribuição dos conjuntos de cortes mínimos para $R(t)$ e $O\left(\frac{1}{E[\tau]}^{r+1}\right)$ é um limitante superior da contribuição dos outros cortes. A ideia por trás dessa aproximação é se $E[\tau]$ é alto (isto é, a probabilidade de falhas é baixa), $R(t)$ é principalmente afetado pelos cortes mínimos. Isso é válido para uma DCN, visto que é esperado um alto tempo de vida mesmo para equipamentos de baixo custo [35]. A aproximação é realizada utilizado o fato de o termo $1 - \frac{t^r c}{E[\tau]^r}$ da Equação 3.5 coincidir com os dois primeiros termos da expansão em série de Taylor para a função $e^{-\frac{t^r c}{E[\tau]^r}}$. Assim, considerando que a contribuição dos outros cortes é tão pequena quanto os termos restantes da série de Taylor, escreve-se $R(t) \approx e^{-\frac{t^r c}{E[\tau]^r}}$.

Combinando as Equações 3.4 e 3.5, como detalhado no Apêndice A, o MTTF pode ser escrito como:

$$MTTF \approx \frac{E[\tau]}{r} \sqrt[r]{\frac{1}{c}} \Gamma\left(\frac{1}{r}\right), \tag{3.6}$$

onde $\Gamma(x)$ é a função gama de x, definida no Apêndice A. Com essa equação, o MTTF é escrito em função de c e r que, como mostrado mais adiante, dependem da topologia empregada e de seus parâmetros.

3.2.2 Análise Baseada em Simulação

A simulação é realizada para medir a acurácia da aproximação do MTTF descrita anteriormente. Para cada amostra da simulação, encontra-se o menor número f de elementos de um determinado tipo que desconecta um servidor da rede. Esse valor é chamado de ponto crítico. Assim, em uma amostra o MTTF Normalizado (NMTTF - *Normalized MTTF*) pode ser calculado colocando o valor f na Equação 3.2. O valor simulado do NMTTF ($NMTF_{sim}$) é então a média dos valores do NMTTF considerando todas as amostras. O Algoritmo 1 resume o procedimento de simulação. A função `removerElementoAleatoriamente` remove aleatoriamente um elemento de um determinado tipo (enlace, comutador ou servidor) do grafo G', seguindo o procedimento descrito na Seção 3.1.2. A função `checarAlcancabilidade` verifica se todos os servidores da rede G' (isto é, a rede com f elementos de um tipo removidos) estão alcançáveis, como definido na Seção 3.1.3. Quando a função `removerElementosAleatoriamente` leva a função `checarAlcancabilidade` a detectar ao menos um servidor inalcançável, a simulação para e a linha 10 calcula o MTTF Normalizado (NMTTF) utilizando a Equação 3.2. Esse valor de NMTTF é adicio-

nado à variável `accNMTTF`. Essa variável é a soma de todos os valores de NMTTF em todas as amostras. No final, a variável `accNMTTF` é dividida pelo número total de amostras $nrAmostras$ para alcançar o valor médio de NMTTF ($NMTF_{sim}$). Note que o MTTF simulado pode ser calculado pela multiplicação de $NMTF_{sim}$ por $E[\tau]$, como indicado pela Equação 3.1. O parâmetro `nrAmostras` é escolhido na ordem de milhares de amostras para alcançar um intervalo de confiança pequeno.

Algoritmo 1: Simulação do NMTTF

Entrada: tipo dos elementos *tipo*, número de amostras da simulação $nrAmostras$, número total de elementos F, rede original G.
Saída: NMTTF Simulado $NMTTF_{sim}$.

```
1  amostra = 1;
2  accNMTTF = 0;
3  enquanto amostra ≤ nrAmostras faça
4  |  G' = G;
5  |  f = 0;
6  |  enquanto (f < F) e checarAlcancabilidade(G') faça
7  |  |  f += 1;
8  |  |  G' = removerElementoAleatoriamente (tipo,G');
9  |  fim
10 |  accNMTTF += Σ_{i=0}^{f-1} 1/(F-i);
11 |  amostra += 1;
12 fim
13 NMTTF_sim = accNMTTF/nrAmostras;
```

A comparação entre o MTTF simulado e teórico é calculada pelo Erro Relativo (RE - *Relative Error*), definido como:

$$RE = \frac{|NMTTF_{sim} - NMTTF_{theo}|}{NMTTF_{sim}}, \tag{3.7}$$

onde $NMTTF_{theo}$ é o MTTF Normalizado teórico, obtido dividindo o MTTF teórico por $E[\tau]$, e $NMTTF_{sim}$ é o valor obtido na simulação. É importante notar que, como mostrado na Equação 3.6, o MTTF pode ser expresso como um termo em primeira ordem de $E[\tau]$. Consequentemente, a utilização do valor $E[\tau]$ não é necessária na prática para normalizar o MTTF teórico. Assim, a normalização é realizada simplesmente removendo $E[\tau]$ da equação. Utilizando os resultados de RE, mostra-se na Seção 3.2.3 para quais casos a Equação 3.6 é uma aproximação acurada do MTTF. Nesses casos, os resultados apresentados mais adiante mostram que o MTTF de cada topologia pode ser aproximado como uma função do número de interfaces de rede e do número de servidores.

3.2.3 Resultados

Nesta seção, utilizam-se as métricas detalhadas anteriormente para analisar as topologias da Tabela 3.1 na Fase Confiável. A comparação entre as topologias é rea-

Tabela 3.1: Configurações de topologias de rede intra-sítio utilizadas neste trabalho.

Tamanho	Nome	Portas de comutador	Portas de servidor	Enlaces	Comutadores	Servidores
500	Three-layer	2(núcleo)	1	605	16	576
	Fat-tree	12	1	1296	180	432
	BCube2	22	2	968	44	484
	BCube3	8	3	1536	192	512
	DCell2	22	2	759	23	506
	DCell3	4	3	840	105	420
3k	Three-layer	12(núcleo)	1	3630	86	3456
	Fat-tree	24	1	10368	720	3456
	BCube2	58	2	6728	116	3364
	BCube3	15	3	10125	670	3375
	BCube5	5	5	15625	3125	3125
	DCell2	58	2	5133	59	3422
	DCell3	7	3	6384	456	3192
8k	Three-layer	28(núcleo)	1	8470	198	8064
	Fat-tree	32	1	24576	1280	8192
	BCube2	90	2	16200	180	8100
	BCube3	20	3	24000	1190	8000
	BCube5	6	5	38880	6480	7776
	DCell2	90	2	12285	91	8190
	DCell3	9	3	16380	910	8190

lizada entre configurações que possuem o mesmo número aproximado de servidores, designado como "Tamanho" na tabela. É importante observar que, apesar de ser possível construir de forma incremental algumas dessas topologias, considera-se apenas configurações de topologias completas, nas quais todas as portas de servidores e comutadores são utilizadas. Além disso, para as novas topologias, o número de portas de comutador não é limitado pelo número de portas comumente encontrado em equipamentos comerciais (p.ex., 8, 24 e 48) para tornar possível obter um número similar de servidores entre configurações a serem comparadas. Como um dos principais objetivos de um DC é fornecer capacidade de processamento e redundância no armazenamento de dados, que aumenta com o número de servidores, balancear o número de servidores por topologia é uma tentativa de fornecer uma análise justa. Para a topologia Three-layer, utiliza-se sempre, $n_e = 48$ e $n_a = 12$, baseando-se na descrição dos equipamentos comerciais encontra em [1]. Assim, para todas as configurações, cada par de comutadores de agregação fornece conectividade para 576 servidores. Como a análise emprega, para a topologia Three-layer, o mesmo número de porta nas camadas de agregação e de borda, a Tabela 3.1 especifica apenas o número de portas de um comutador de núcleo conectadas aos comutadores de agregação. Apresenta-se abaixo os resultados da análise para cada tipo de falha. A confiabilidade para falhas de servidores não é considerada visto que uma falha do DC ocorre sempre que um servidor é desconectado. Assim, uma única falha de servidor já é necessária para mudar da Fase Confiável para a Fase de Sobrevivência.

Tabela 3.2: Tamanho e número de cortes mínimos, considerando falhas de enlace.

Topologia	**Tamanho do corte mínimo (r)**	**Número de cortes mínimos (c)**
Three-layer	1	$\|\mathcal{S}\|$
Fat-tree	1	$\|\mathcal{S}\|$
BCube	$l+1$	$\|\mathcal{S}\|$
DCell	$l+1$	$1,5\|\mathcal{S}\|$ se $l=1$, S caso contrário

Falhas de Enlace

Para encontrar o MTTF teórico para falhas de enlace, utiliza-se a Equação 3.6 com os valores r e c correspondentes a cada topologia. A Tabela 3.2 mostra esses valores para as topologias consideradas. Para todas as topologias, o tamanho do corte mínimo é o número de interfaces por servidor, que é sempre 1 para a Three-layer e a Fat-tree, e $l+1$ para BCube e DCell. Além disso, à exceção da DCell com $l=1$, o número de cortes mínimos é igual ao número de servidores. Para a DCell com $l=1$ tem-se outra possibilidade de corte mínimo, além da desconexão de $l+1=2$ enlaces de um único servidor, chamada adiante de "ilha de servidores". Essa situação ocorre quando dois servidores que estão diretamente conectados perdem, cada um, o enlace com o seu comutador correspondente. Como exemplo, considere que na Figura 2.4 o Servidor 0 na $DCell_0 0$ e o Servidor 3 na $DCell_0 1$ perderam o enlace com seus comutadores correspondentes. Esses dois servidores permanecem conectados entre si, mas estão desconectados da rede (isto é, não possuem acesso a um *gateway*), formando uma ilha de servidores. Na DCell com $l=1$, cada servidor está diretamente conectado com apenas um outro servidor, uma vez que cada um possui duas interfaces. Assim, o número possível de ilhas de servidores é $0,5|\mathcal{S}$ e o número de cortes mínimos é dado por $|\mathcal{S}+0,5|\mathcal{S}=1,5|\mathcal{S}$. Para uma DCell com $l>1$, o número de falhas de enlaces que produzem uma ilha de servidores é maior que $l+1$, e então essa situação não é um corte mínimo.

Utilizando os valores da Tabela 3.2 na Equação 3.6, obtém-se as seguintes aproximações para o MTTF considerando falhas de enlace:

$$MTTF_{threeLayer} = MTTF_{fatTree} \approx \frac{E[\tau]}{|\mathcal{S}|}. \tag{3.8}$$

$$MTTF_{dcell} \approx \begin{cases} \frac{E[\tau]}{2}\sqrt{\frac{1}{1,5|\mathcal{S}|}}\Gamma\left(\frac{1}{2}\right) & \text{se } l=1, \\ \frac{E[\tau]}{l+1}\sqrt[l+1]{\frac{1}{|\mathcal{S}|}}\Gamma\left(\frac{1}{l+1}\right) & \text{caso contrário.} \end{cases} \tag{3.9}$$

$$MTTF_{bcube} \approx \frac{E[\tau]}{l+1}\sqrt[l+1]{\frac{1}{|\mathcal{S}|}}\Gamma\left(\frac{1}{l+1}\right). \tag{3.10}$$

Os resultados da Figura 3.3(a) mostram o RE (Equação 3.7) para diferen-

tes tamanhos de rede. Note que a medida de RE não possui intervalo de confiança, visto que o RE é um valor único baseado na média do MTTF obtido na simulação ($NMTTF_{sim}$). A figura mostra que a estimativa do MTTF utilizando cortes mínimos possui um erro menor que 10%.

Dadas as equações acima e a respectiva comparação entre elas no Apêndice B, é possível afirmar que:[1]

- *Sobre o desempenho das topologias centradas em comutador.* As duas topologias apresentam o mesmo valor de MTTF, possuindo o MTTF mais curto considerando falhas de enlace. De acordo com as equações, o MTTF da Three-layer e da Fat-tree é $\sqrt{\frac{|\mathcal{S}|\pi}{6}}$ vezes mais baixo que o pior caso de uma topologia centrada em servidores (DCell2). Assim, para uma DCN com 3400 servidores, o MTTF da Fat-tree é pelo menos 42 vezes menor que o da BCube ou o da DCell.

- *Sobre o desempenho das topologias centradas em servidor.* A BCube possui o mesmo MTTF que a DCell, à exceção do caso de duas interfaces por servidor no qual a BCube apresenta melhor desempenho. Entretanto, como mostrado pelas equações, a BCube2 é apenas $1,23$ vezes melhor que a DCell2 para qualquer $|\mathcal{S}|$. Observa-se também que, na BCube e na DCell, o aumento do número de interfaces de servidor aumenta o MTTF.

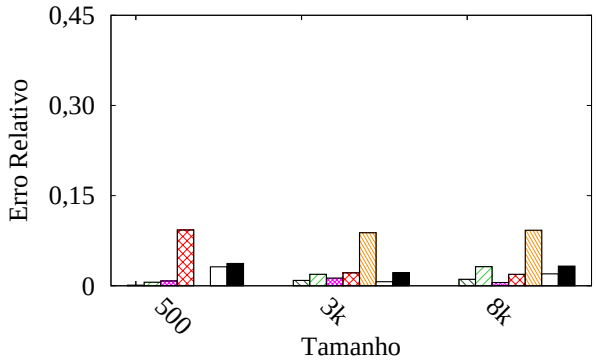
- *Observações Gerais.* O MTTF reduz à medida que o número de servidores $|\mathcal{S}|$ aumenta. Esse resultado ressalta a importância de se preocupar com a confiabilidade em grandes DCs, nos quais $|\mathcal{S}|$ pode ser da ordem de milhares de servidores.

A Figura 3.3(b) mostra um exemplo da simulação do MTTF Normalizado e da FER Crítica para topologias com 3k (três mil) servidores. Note que a confiabilidade da Three-layer e da Fat-tree é consideravelmente menor que das outras topologias. Consequentemente, as barras correspondentes a essas topologias na Figura 3.3(b) são quase imperceptíveis.

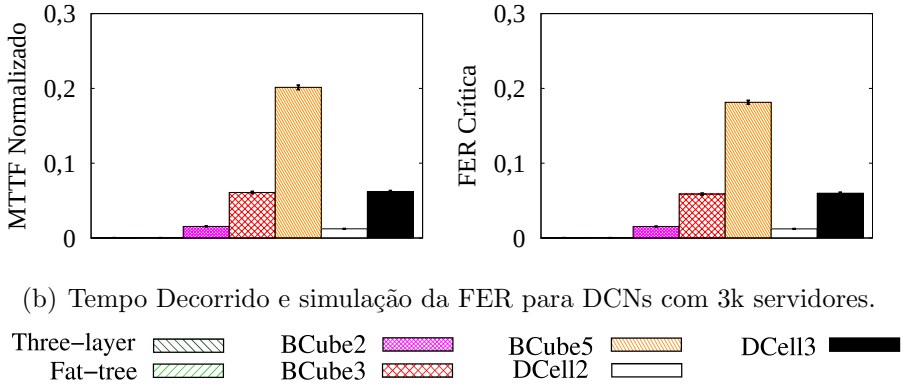
Falhas de Comutador

Para analisar o uso de cortes mínimos para aproximar o MTTF em falhas de comutador, emprega-se a mesma metodologia do caso de falhas de enlace apresentada anteriormente. A Tabela 3.3 mostra os valores de c e r para falhas de comutador. Na

[1]No restante deste capítulo divide-se em três itens as observações sobre cada resultado. O primeiro comenta o desempenho das topologias centradas em comutador (isto é, Three-layer e Fat-tree), enquanto o segundo destaca os resultados para topologias centradas em servidor (isto é, BCube e DCell). O último item, quando disponível, indica observações gerais considerando as três topologias.



(a) Erro Relativo da aproximação do MTTF.



(b) Tempo Decorrido e simulação da FER para DCNs com 3k servidores.

Figura 3.3: Análise da Fase Confiável para falhas de enlace.

Three-layer e na Fat-tree, uma única falha em um comutador de borda é suficiente para desconectar um servidor. Assim, o tamanho e número de cortes mínimos são, respectivamente, 1 e o número de comutadores de borda. Na Three-layer o número de comutadores de borda é simplesmente $\frac{|\mathcal{S}|}{n_e}$, onde n_e é o número de portas de comutador de borda. Como em uma Fat-tree de n portas cada comutador de borda está conectado a $\frac{n}{2}$ servidores, o número de comutadores de borda é $\frac{|\mathcal{S}|}{\frac{n}{2}}$. Como $|\mathcal{S}| = \frac{n^3}{4}$, escreve-se $n = \sqrt[3]{4|\mathcal{S}|}$, e então o número de cortes mínimos é $\sqrt[3]{2|\mathcal{S}|^2}$. Para a BCube, um corte mínimo de comutador ocorre quando há falha dos $l + 1$ comutadores conectados a um único servidor. O número de cortes mínimos é então igual ao número de servidores $|\mathcal{S}|$, já que cada servidor possui um conjunto diferente de comutadores conectados. No caso da DCell o raciocínio é mais complexo, pois o corte mínimo é o conjunto de comutadores necessários para formar uma ilha de servidores. Apesar de o caso de falhas de enlace gerar ilhas de servidores apenas na DCell2, para falhas de comutadores todos os cortes mínimos das DCell2 e DCell3 apresentam essa situação. Para a DCell2, é fácil observar que uma ilha de servidores é formada quando dois servidores que estão diretamente conectados perdem seus comutadores correspondentes, assim $r = 2$. Como observado para falhas de enlace na Seção 3.2.3, o número de ilhas de servidores possíveis é o número de pares de servidores, dado

por $0,5|\mathcal{S}|$. Para a DCell3, os valores de r e c são obtidos analisando grafos DCell para diferentes valores de n com $l = 2$. Nessa análise o valor de r sempre apresentou valor igual a 8, independente de n. Além disso, comprovou-se a formação de ilhas de servidores para os cortes mínimos encontrados. Outro fator observado na análise foi um padrão para as ilhas de servidores: cada ilha possui servidores de 4 diferentes módulos de DCell de nível 1. Ademais, cada DCell de nível $l = 1$ possui 2 servidores da ilha. Obviamente, esses 2 servidores estão diretamente conectados e pertencem a diferentes módulos de DCell com $l = 0$. Baseado na análise de diferentes grafos, observou-se que a DCell3 possui $c = \binom{n+2}{4}$. Assim, os cortes mínimos da DCell2 e da DCell3 podem ser formulados como $r = 2l^2$ e $c = \binom{n+l}{2l}$. Note que, para a DCell2 $c = \binom{n+l}{2l} = \binom{n+1}{2} = 0,5|\mathcal{S}|$, correspondendo ao valor encontrado anteriormente. Para a DCell3 tem-se $c = \binom{n+2}{4} = 0,125(2|\mathcal{S}| - 3\sqrt{4|\mathcal{S}| + 1} + 3)$, ao substituir n pela solução da equação $|\mathcal{S}| = [n(n+1)][(n+1)n+1]$. O cálculo de r e c para redes DCell com $l > 2$ é deixado como trabalho futuro.

Tabela 3.3: Tamanho e número de cortes mínimos, considerando falhas de comutador.

Topologia	**Tamanho do corte mínimo (r)**	**Número de cortes mínimos (c)**
Three-layer	1	$\frac{\lvert\mathcal{S}\rvert}{n_e}$
Fat-tree	1	$\sqrt[3]{2S^2}$
BCube	$l+1$	S
DCell ($l \leq 2$)	$2l^2$	$\binom{n+l}{2l}$

Utilizando os valores da Tabela 3.3 na Equação 3.6, avalia-se o MTTF teórico para falhas de comutador. A comparação entre valores simulados e teóricos é realizada utilizando a mesma metodologia anterior, resultando nos valores RE apresentados na Figura 3.4(a). Como mostrado, a aproximação por cortes mínimos para falhas de comutadores não é satisfatória para algumas topologias. As topologias que apresentam boas aproximações são Three-layer, Fat-tree, BCube5 e BCube3. O erro da BCube2 é próximo de 40%. Os resultados para DCell mostram uma baixa acurácia na aproximação, uma vez que o RE mínimo obtido é de 27%. Entretanto, para a DCell2 é possível formular o MTTF exato, visto que uma falha em qualquer grupo de dois comutadores é suficiente para formar uma ilha de servidores, como visto na Figura 2.4. Assim, o MTTF da DCell2 é o tempo necessário para haver 2 falhas de comutadores, encontrado fazendo $f = 2$ e $F = n + 1$ (isto é, número total de comutadores) na Equação 3.1, e escrevendo o número de portas de comutadores

em função do número de servidores[2] como $n = 0,5(-1 + \sqrt{4|\mathcal{S}| + 1})$:

$$MTTF_{dcell} = \frac{E[\tau]\sqrt{4|\mathcal{S}| + 1}}{|\mathcal{S}|}, \text{ para } l = 1. \tag{3.11}$$

Baseado na análise anterior do RE, encontra-se um baixo erro da aproximação Burtin-Pittel para estimar o MTTF da Three-layer, Fat-Tree e BCube para $l > 1$. Assim, o MTTF dessas topologias pode ser escrito utilizando as seguintes equações:

$$MTTF_{threeLayer} \approx \frac{E[\tau]n_e}{|\mathcal{S}|}; \tag{3.12}$$

$$MTTF_{fatTree} \approx \frac{E[\tau]}{\sqrt[3]{2|\mathcal{S}|^2}}; \tag{3.13}$$

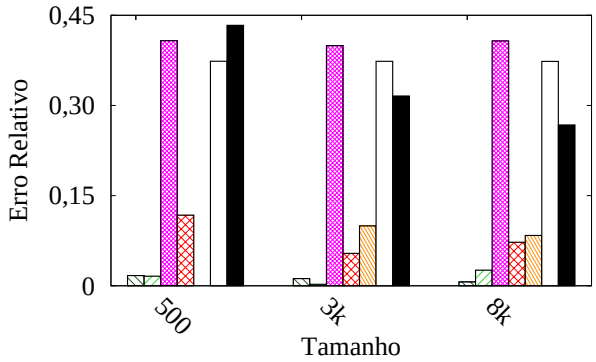
$$MTTF_{bcube} \approx \frac{E[\tau]}{l+1} \sqrt[l+1]{\frac{1}{|\mathcal{S}|}}\Gamma\left(\frac{1}{l+1}\right), \text{ para } l > 1. \tag{3.14}$$

A Figura 3.4(b) mostra resultados de simulação da Fase Confiável considerando uma rede com 3k servidores. Como não há equações de MTTF para todas as configurações utilizadas, a comparação das topologias utiliza esses resultados de simulação. É importante notar que essa mesma comparação é válida para redes de tamanho 500 e 8k. Baseando-se nos resultados, conclui-se:

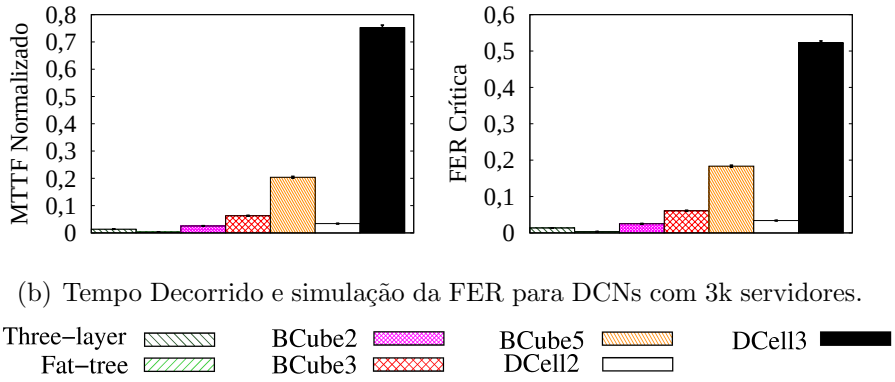
- *Sobre o desempenho das topologias centradas em comutador.* A Three-layer e a Fat-tree apresentam valores de MTTF muito curtos se comparadas às outras topologias, pois uma única falha em um comutador de borda já desconecta a rede. O MTTF da Fat-tree para topologias com 3k servidores é aproximadamente $7,3$ vezes mais baixo do que para a BCube2, que é a topologia centrada em servidores com o MTTF mais baixo.

- *Sobre o desempenho das topologias centradas em servidor.* O número de interfaces por servidor aumenta o MTTF, como no caso de falhas de enlace. Além disso, a DCell possui um MTTF mais longo que o da BCube. Isso ocorre pois a DCell possui uma menor dependência de comutadores, pois na DCell cada servidor está conectado a 1 comutador e a l servidores, enquanto na BCube apenas comutadores estão ligados aos servidores. Apesar de o desempenho da DCell2 ser próximo do da BCube2, o MTTF e a FER Crítica são muito maiores na DCell3 do que na BCube3. Os resultados mostram que, para redes com 3k servidores, todos os servidores da DCell3 permanecem disponíveis até a situação na qual 50% dos comutadores estão defeituosos. Dessa forma, seu MTTF é 12 vezes maior que o da BCube3.

[2] O número de portas de comutador n em função de $|\mathcal{S}|$ é calculado resolvendo a equação $|\mathcal{S}| = n(n + 1)$.



(a) Erro Relativo da aproximação do MTTF.



(b) Tempo Decorrido e simulação da FER para DCNs com 3k servidores.

Figura 3.4: Análise da Fase Confiável para falhas de comutador.

3.3 Fase de Sobrevivência

Após a primeira desconexão de servidor, se nenhum reparo é realizado, o DC entra na Fase de Sobrevivência. Assim, o DC continua em operação porém com alguns servidores inalcançáveis. Nessa fase é interessante analisar outras métricas de desempenho, como o tamanho dos caminhos afetados pelas falhas. As métricas analisadas nesta seção podem ser vistas como medidas de sobrevivência, considerando a definição de Liew e Lu [28]. A sobrevivência é então definida nesta tese como uma métrica de desempenho do DC quando sujeito a uma determinada situação de falhas. Por exemplo, a sobrevivência de um DC pode ser medida pelo número de servidores alcançáveis após a falha de uma determinada fração dos enlaces. Assim, diferentes métricas de desempenho são propostas ao longo dessa tese para analisar a sobrevivência, tanto no caso intra-sítio quanto no caso inter-sítio. Vale notar que existem diversas outras definições de sobrevivência na literatura [36, 37]. Por exemplo, algumas definições consideram a sobrevivência como a habilidade de um sistema fornecer serviços críticos após sofrer algum tipo de falha [37]. As métricas desta tese, entretanto, consideram que todos os servidores do DC possuem igual importância, não distinguindo um conjunto de serviços críticos.

A sobrevivência é avaliada nesta seção com métricas de desempenho para uma

determinada FER e Tempo Decorrido, que correspondem à Fase de Sobrevivência. A FER é importante na Fase de Sobrevivência para quantificar o quanto o DC é resistente a uma determinada quantidade de falhas. Por exemplo, mede-se o número esperado de servidores alcançáveis quando 10% dos enlaces não estão funcionando. Pela FER é possível medir o número esperado de servidores alcançáveis para uma certa porcentagem de enlaces defeituosos. Isso pode mostrar, por exemplo, qual topologia é mais adequada para um ambiente no qual são esperados defeitos em uma grande quantidade de elementos de rede. O Tempo Decorrido, por sua vez, é importante para medir a degradação do DC provocada pelo seu uso contínuo sem manutenção. Por exemplo, é possível medir o número esperado de servidores alcançáveis após seis meses de operação do DC. Isso quantifica a degradação da capacidade de processamento do DC com o tempo. A sobrevivência é então analisada a partir de simulações baseadas na metodologia da Seção 3.1.1, através de métricas de Alcançabilidade do Serviço e Qualidade dos Caminhos detalhadas a seguir.

3.3.1 Alcançabilidade do Serviço

A Alcançabilidade do Serviço indica a quantidade de servidores que estão disponíveis para executar as tarefas desejadas. Para tal, calcula-se o número de servidores alcançáveis e a conectividade entre eles. Essa medida é importante para quantificar o poder de processamento do DC, uma vez que esse poder depende do número de servidores disponíveis. Além disso, pode representar a capacidade do DC em alocar VMs em um cenário de computação em nuvem. A Alcançabilidade do Serviço pode assim ser medida pelas duas métricas seguintes:

Fração de Servidores Alcançáveis (RSR - *Reachable Server Ratio*). Essa métrica é a razão entre o número de servidores alcançáveis e o número total de servidores na rede original, dado o estado atual da rede (isto é, uma dada FER). Assim, a RSR é definida como:

$$RSR = \frac{\sum_{k \in \mathcal{A}} s_k}{|\mathcal{S}|} \tag{3.15}$$

onde s_k e $|\mathcal{S}|$ são, respectivamente, o número de servidores na k-ésima sub-rede operacional ($k \in \mathcal{A}$) e na rede original (isto é, quando não há falhas). O conjunto de redes alcançáveis é dado por $\mathcal{A}$. A métrica RSR é baseada na métrica proposta em [38] para avaliar a resiliência em redes complexas. Nesse trabalho, Albert *et al.* medem a resiliência da rede como a fração dos nós que, após uma falha aleatória, permanecem na sub-rede com o maior número de nós. Entretanto, essa métrica não é adequada para redes de centros de dados, já que é necessário considerar a existência de *gateways* e a existência de múltiplas sub-redes operacionais, como destacado na

Seção 3.1.3.

Conectividade entre Servidores (SC - *Server Connectivity*). A RSR é importante para quantificar os servidores que ainda estão alcançáveis na rede após uma situação de falhas. Entretanto, quando utilizada individualmente, essa métrica pode não representar a verdadeira capacidade do DC em realizar processamento paralelo ou fornecer redundância. Isso ocorre pois servidores alcançáveis não estão necessariamente interconectados dentro de um DC. Por exemplo, uma rede com 100 servidores alcançáveis em 2 sub-redes isoladas, com 50 servidores cada uma, possui melhor desempenho ao executar uma tarefa paralela do que uma rede com 100 servidores alcançáveis em 100 sub-redes isoladas. Como consequência, complementa-se a métrica RSR com a noção de conectividade entre servidores. Essa conectividade é medida através do cálculo da densidade de um grafo auxiliar simples e não-direcionado, no qual os nós são os servidores alcançáveis (isto é, servidores que possuem um caminho ao *gateway*) e uma aresta entre dois nós indica se esses nós podem se comunicar utilizado a rede interna ao DC. Note que as arestas do grafo auxiliar, que representam a interconexão entre os servidores, não estão relacionadas aos enlaces físicos. Em outras palavras, a métrica proposta é a densidade do grafo de enlaces lógicos entre os servidores alcançáveis. A densidade de um grafo simples e não-direcionado com $|\mathcal{E}|$ arestas e S_a nós é definida da seguinte forma [39]:

$$D = \frac{2|\mathcal{E}|}{S_a(S_a - 1)}. \tag{3.16}$$

Neste trabalho $|\mathcal{E}|$ é o número de enlaces lógicos, e $S_a = \sum_{k \in \mathcal{A}} s_k$ é o número de servidores alcançáveis. Note que uma DCN sem falhas possui densidade igual a 1 pois todos os servidores podem se comunicar. Uma rede com falhas que apresenta apenas uma sub-rede operacional também possui densidade igual a 1. O cálculo da métrica acima pode ser simplificado utilizando o fato de que, após uma falha, em cada sub-rede isolada o grafo dos enlaces lógicos é um grafo completo. Além disso, como as sub-redes estão isoladas entre si, o valor $|\mathcal{E}|$ é a soma do número de enlaces de cada sub-rede. Como a sub-rede é um grafo completo, ela possui $\frac{s_k(s_k-1)}{2}$ arestas (isto é, pares de servidores alcançáveis). Assim, substitui-se o valor $|\mathcal{E}|$ na Equação 3.16 de acordo com o raciocínio anterior, definindo SC como:

$$SC = \begin{cases} \frac{\sum_{k \in \mathcal{A}} s_k(s_k-1)}{S_a(S_a-1)}, & \text{if } S_a > 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.17}$$

A SC definida neste trabalho é similar à métrica A2TR (*Average Two Terminal Reliability* - Confiabilidade Média de Dois Terminais) [40]. A A2TR é definida como a probabilidade de um par de nós aleatoriamente escolhido estar conectado na rede, e

também é calculada como a densidade do grafo de enlaces lógicos. Entretanto, a SC difere da A2TR uma vez que considera apenas os servidores alcançáveis, enquanto a A2TR considera qualquer nó. Assim, se aplicada no cenário deste trabalho, a A2TR consideraria comutadores, servidores alcançáveis e servidores inalcançáveis.

3.3.2 Qualidade dos Caminhos

Na análise de sobrevivência, mede-se a Qualidade dos Caminhos calculando os caminhos mais curtos de cada topologia após as falhas. O cálculo dos caminhos mais curtos é adequado para medir a qualidades dos caminhos da rede, já que é a base de novos mecanismos de roteamento utilizados em DCNs, como o TRILL [41], IEEE 802.1aq [42], e SPAIN [43]. Dessa forma, define-se a seguinte métrica:

Comprimento Médio dos Caminhos Mais Curtos (ASPL - *Average Shortest Path Length*). Essa métrica é a média dos comprimentos dos caminhos mais curtos entre os servidores da rede. Para essa métrica não se considera os caminhos entre servidores de diferentes sub-redes, visto que não possuem caminhos entre eles. Assim, essa métrica captura o aumento da latência causado pelas falhas.

3.3.3 Resultados

Como indicado na Seção 3.1.1, a caracterização das falhas utiliza a FER e o Tempo Decorrido. A FER é independente da distribuição de probabilidade do tempo de vida do elemento de rede, enquanto o Tempo Decorrido assume uma distribuição exponencial. Para obter maior concisão, apresenta-se a maioria dos resultados desta seção como função da FER, visto que não dependem de uma distribuição de probabilidade específica. As conclusões oriundas da comparação entre topologias em relação à FER são também válidas considerando o Tempo Decorrido. Isso ocorre pois, utilizando a Equação 3.2, o Tempo Normalizado para uma dada FER é praticamente independente do número total de elementos F, sendo independente da topologia ou tipo de falha. Por exemplo, utiliza-se a Equação 3.2 para plotar na Figura 3.5 o Tempo Normalizado em função da FER (isto é, $\frac{f}{F}$) para diferentes valores de número total de elementos F (p.ex., número total de enlaces). Essa figura mostra que, para uma grande faixa de valores de FER, a relação entre o Tempo Normalizado e a FER é independente de F.

Como feito na Seção 3.2.3, as topologias comparadas entre si possuem aproximadamente o mesmo tamanho, medido em número de servidores. Para maior concisão, a análise apresenta resultados para topologias com 3k servidores, detalhadas na Tabela 3.1. Por outro lado, observa-se que esse tamanho da rede é suficiente para desvendar as diferenças entre as topologias investigadas. Além disso, como as topologias possuem uma estrutura regular, as conclusões também podem ser ex-

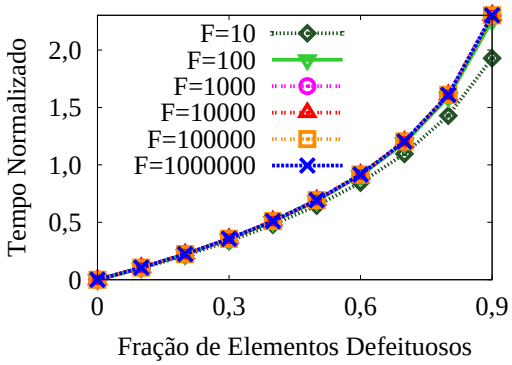


Figura 3.5: Tempo Normalizado (NT) em função da FER.

trapoladas para um maior número de servidores suportados. Finalmente, a análise desta seção apresenta resultados para uma larga faixa de valores de FER (de 0 até 0,4). Apesar dessa faixa cobrir altos valores de FER, que podem ser irreais para DCs tradicionais, escolhe-se apresentar os resultados também para esses valores de forma a realizar uma análise genérica, apropriada para diferentes novos cenários que vierem a surgir. Por exemplo, um centro de dados modular (MDC) apresenta alguns desafios em relação à sua manutenção, que pode levar o operador do DC a esperar por um alto número de falhas de equipamentos antes de reparar a rede [3].

Falhas de Enlace

As Figuras 3.6(a) e 3.6(b) apresentam, respectivamente, as métricas RSR e SC em função da FER. A partir dos resultados observa-se os seguintes aspectos:

- *Sobre o desempenho das topologias centradas em comutador.* A Three-layer e a Fat-tree possuem o pior desempenho pois, nessas topologias, cada servidor está conectado ao seu comutador de borda utilizando apenas um enlace. Assim, a falha nesse enlace desconecta completamente o servidor. As topologias centradas em servidor, por outro lado, possuem uma deterioração mais lenta ao aumentar a RSR, visto que os servidores possuem enlaces redundantes. Os resultados para a Fat-tree mostram que uma dada Fração de Elementos Defeituosos reduz o número de servidores alcançáveis na mesma proporção (p.ex., uma FER de $0,3$ produz uma RSR de $1 - 0,3 = 0,7$), mostrando uma rápida deterioração na Alcançabilidade do Serviço. Como a Three-layer em comparação com a Fat-tree possui menos enlaces redundantes no núcleo e na agregação, seu RSR tende a decair mais rápido do que na Fat-tree. Como exemplo, a Tabela 3.1 mostra que, para uma rede com 3k servidores, a Fat-tree possui aproximadamente um número de enlaces três vezes maior do que o da Three-layer.

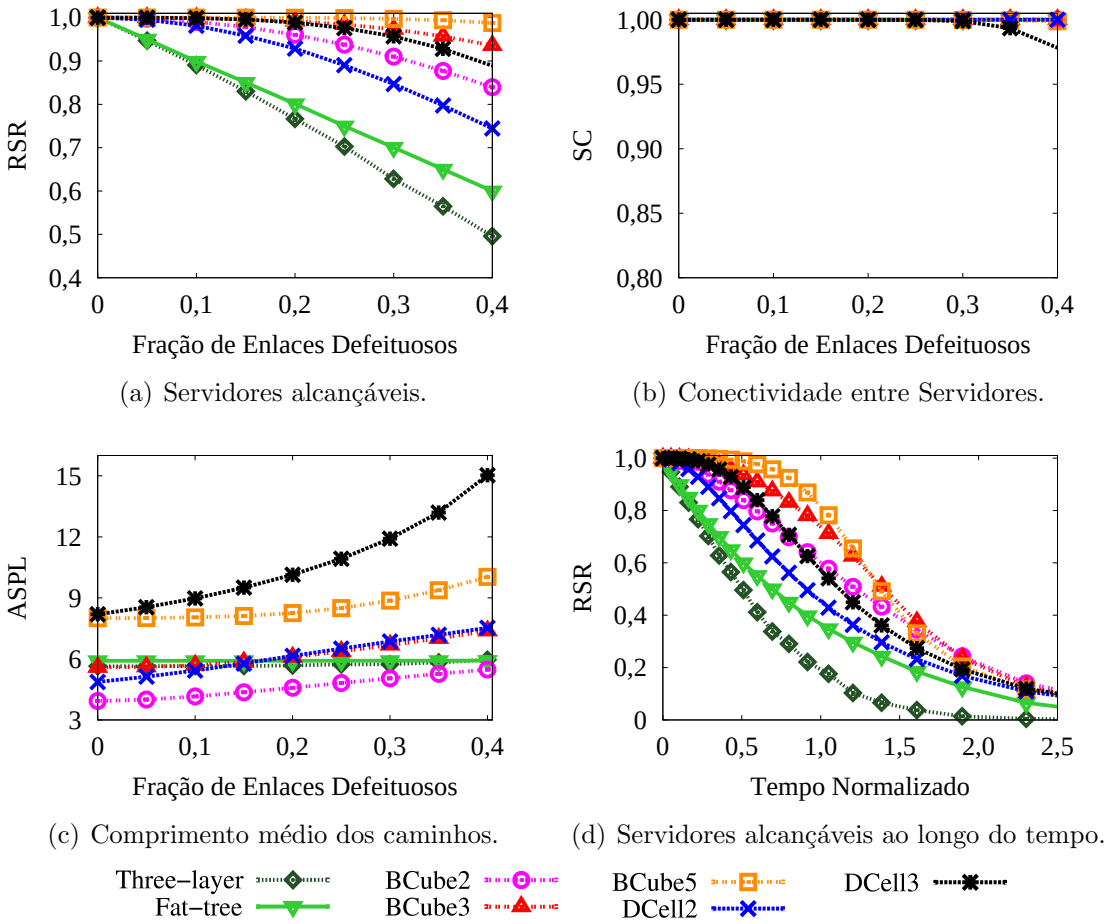
- *Sobre o desempenho das topologias centradas em servidor.* Para um mesmo tipo de topologia centrada em servidor, a sobrevivência pode ser melhorada aumentando o número de interfaces de rede por servidor. Quando um servidor possui mais interfaces, a sua desconexão por falhas de enlaces será mais difícil e então uma dada FER desconecta menos servidores. Por exemplo, considerando uma FER de $0,4$, a RSR é melhorada em 11% na BCube e 19% na DCell se o número de interfaces por servidor for aumentado de dois para três. A comparação das duas topologias permite verificar que, para o mesmo número de interfaces de servidor, a sobrevivência da BCube é superior à da DCell. Por exemplo, a BCube mantém no mínimo uma RSR de $0,84$ quando 40% dos seus enlaces estão defeituosos, enquanto na DCell esse limitante inferior é $0,74$. Esse comportamento ocorre pois na DCell cada servidor está conectado a 1 comutador e l servidores, enquanto os servidores da BCube estão conectados a $l+1$ comutadores. Como um comutador possui mais interfaces de rede que um servidor, as falhas de enlaces tendem a desconectar menos comutadores que servidores. Consequentemente, os servidores na BCube se desconectam com maior dificuldade da rede do que na DCell. Obviamente, o preço a pagar por essa melhor sobrevivência é um maior número de enlaces e comutadores necessários na BCube se comparada à DCell, como visto na Tabela 3.1.

- *Observação Geral.* Para todas as topologias a conectividade SC é bem próxima de 1, mostrando que as falhas de enlace produzem aproximadamente apenas 1 sub-rede operacional.

A Figura 3.6(c) apresenta a Qualidade dos Caminhos em função da FER, possibilitando as seguintes observações:

- *Sobre o desempenho das topologias centradas em comutador.* A Three-layer e a Fat-tree mantêm o comprimento médio original independente da FER, mostrando uma melhor Qualidade dos Caminhos em relação às outras topologias quando a FER aumenta.

- *Sobre o desempenho das topologias centradas em servidor.* O comprimento dos caminhos das topologias centradas em servidor aumenta com a FER. A BCube mantém um comprimento médio mais baixo que a DCell, comparando topologias com o mesmo número de interfaces de servidor. Por exemplo, para uma alta FER $(0,4)$ a DCell apresenta um aumento de 7 saltos no ASPL, enquanto na BCube o aumento máximo dessa métrica é de 2 saltos. Outro fator observado é que, para uma mesma topologia, o ASPL é maior quando há mais interfaces de rede por servidor, mesmo quando a rede não sofre falhas.



(a) Servidores alcançáveis.

(b) Conectividade entre Servidores.

(c) Comprimento médio dos caminhos.

(d) Servidores alcançáveis ao longo do tempo.

Figura 3.6: Análise da Fase de Sobrevivência para falhas de enlace.

Como mais interfaces de rede implicam em mais níveis na BCube e na DCell, os caminhos contêm nós pertencentes a mais níveis e assim são mais longos.

A partir dos resultados acima, observa-se um compromisso entre Alcançabilidade do Serviço e Qualidade dos Caminhos. Por um lado, quanto mais interfaces por servidor, melhor a sobrevivência em relação ao número de servidores alcançáveis. Por outro lado, o aumento de interfaces por servidor causa um aumento no comprimento médio dos caminhos. Assim, *aumentar a Alcançabilidade do Serviço pela adição de interfaces de servidor implica exigências mais relaxadas em relação à Qualidade dos Caminhos.*

A Figura 3.6(d) ilustra como a sobrevivência evolui ao longo tempo, mostrando a RSR em função do Tempo Normalizado. Esse é o mesmo experimento do mostrado na Figura 3.6(a), porém utilizando como eixo X o resultado da Equação 3.2, ao invés de $\frac{f}{F}$. Note que, apesar de a Figura 3.6(a) mostrar a RSR para uma FER de $0,4$, o último ponto de resultado no eixo X da Figura 3.6(d) é $2,3$, que corresponde a uma FER de aproximadamente $0,9$. O Tempo Normalizado fornece uma ideia de como a sobrevivência está relacionada ao tempo de vida individual de um único

elemento, que é um enlace nesse caso. Assim, um Tempo Normalizado de valor 1 representa o tempo médio de vida de um enlace, dado por $E[\tau]$. Como mostrado na Figura 3.6(d), a maioria das topologias apresenta uma degradação substancial da RSR quando o Tempo Decorrido é igual ao tempo médio de vida do enlace (Tempo Normalizado igual a 1). Além disso, os resultados mostram que todas as topologias possuem uma sobrevivência muito baixa quando o Tempo Normalizado é o dobro do tempo de vida do enlace (Tempo Normalizado igual a 2).

Falhas de Comutador

As Figuras 3.7(a) e 3.7(b) plotam, respectivamente, a RSR e a SC de acordo com a Fração de Comutadores Defeituosos. É possível observar que:

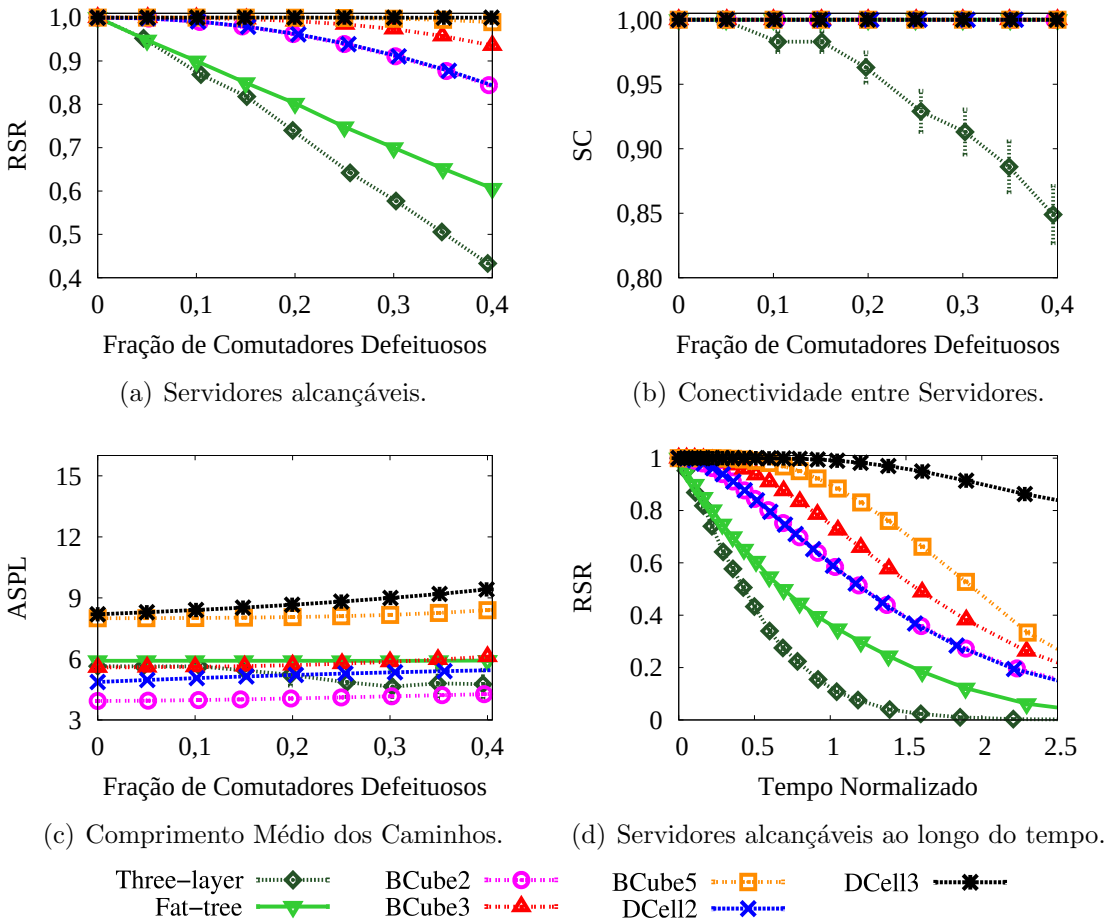
- *Sobre o desempenho das topologias centradas em comutador.* A Three-layer e Fat-tree apresenta o pior desempenho devido à fragilidade que apresentam no nível de borda. Na Three-layer, uma única falha em um comutador de borda é suficiente para desconectar 48 servidores da rede, que é o número de portas nesse tipo de comutador. Para a Fat-tree, uma única falha em um comutador de borda é suficiente para desconectar $\frac{n}{2}$ servidores, onde n é o número de portas de comutador. Assim, para uma configuração de 3k servidores, a Fat-tree perde $\frac{24}{2} = 12$ servidores em uma falha de comutador de borda. Note que esse número é quatro vezes menor do que no caso da Three-layer. Além disso, a topologia Three-layer utiliza no núcleo comutadores de alta capacidade que atuam como *gateways* e mantêm a conectividade de todo o DC, enquanto a Fat-tree utiliza 24 elementos menores do núcleo atuando como *gateways*. Apesar de essa comparação não ser necessariamente justa, visto que essas topologias possuem diferentes valores de GPD (Seção 3.1.3), os resultados mostram como utilizar um pequeno número de elementos no núcleo pode diminuir a sobrevivência da topologia. Como visto para o caso de enlaces, uma dada Fração de Comutadores Defeituosos reduz em média a RSR na mesma proporção para a Fat-tree, enquanto na Three-layer o impacto no desempenho é mais severo.

- *Sobre o desempenho das topologias centradas em servidor.* Como no caso de falhas de enlace, aumentar o número de interfaces por servidor aumenta a sobrevivência a falhas de comutador. Considerando uma FER de $0,4$ para BCube e DCell, a RSR aumenta respectivamente em 11% e 19% se o número de interfaces aumenta de dois para três. No caso da BCube, um maior número de interfaces de servidor representa um maior número de comutadores conectados em cada servidor. Consequentemente, mais falhas de comutador são necessárias para desconectar um servidor. Para a DCell, um maior número

de interfaces de servidor representa uma menor dependência de comutador, já que cada servidor está conectado a 1 comutador e l servidores. Pode também ser observado que a DCell3 possui desempenho superior à BCube3, mostrando uma RSR 6% maior para uma FER de $0,4$, enquanto a BCube2 e a DCell2 apresentam o mesmo desempenho. A diferença de desempenho entre a DCell3 e a BCube3 pode ser explicada pela menor dependência de comutadores na DCell, como observado na Seção 3.2.3. No caso particular de duas interfaces de servidor, essa explicação não é válida. Considerando que a sobrevivência é fortemente afetada pelos cortes mínimos, cada corte mínimo de comutador na DCell2 desconecta dois servidores; na BCube2 cada corte mínimo desconecta apenas um servidor. Por outro lado, cada valor de FER na BCube2 apresenta aproximadamente o dobro do número absoluto de comutadores defeituosos do que na DCell2. Isso pode ser observado na Tabela 3.1, na qual o número total de comutadores na BCube2 é aproximadamente o dobro de comutadores que na DCell2. Por esse motivo, e como os cortes mínimos possuem o mesmo tamanho em ambas as topologias (Tabela 3.3), uma dada Fração de Comutadores Defeituosos na BCube2 escolhe em média o dobro de conjuntos de cortes mínimos que na DCell2. Assim, a BCube2 possui o dobro de cortes mínimos afetados, enquanto a DCell2 possui o dobro de desconexões de servidores por corte mínimo. Consequentemente, o número de servidores desconectados é aproximadamente o mesmo em ambas as topologias para uma dada FER.

- *Observação Geral.* Os resultados da métrica SC são muito próximos de 1 para todas as topologias, à exceção da Three-layer. Uma única amostra de simulação da Three-layer pode apresentar apenas dois possíveis resultados. No primeiro, pelo menos um *gateway* (comutador de núcleo) está em funcionamento e existe uma sub-rede operacional, sendo então $SC = 1$. No segundo, os dois *gateways* foram removidos da rede, e então $SC = 0$. Como a Figura 3.7(b) apresenta a média de todas as amostras da simulação, a medida de SC é simplesmente a fração das amostras que obtiveram $SC = 1$. Como pode ser observado, o resultado $SC = 1$ é mais frequente, uma vez que $SC > 0.8$ para a faixa de FER considerada. Assim, mesmo no caso da topologia Three-layer, que possui apenas dois *gateways*, existe uma probabilidade pequena que todos os servidores do DC estejam inalcançáveis após a remoção aleatória de comutadores.

Os resultados para comprimento médio dos caminhos na Figura 3.7(c) mostram que, *para todas as topologias, falhas de comutador não reduzem significativamente a Qualidade dos Caminhos.*



(a) Servidores alcançáveis.

(b) Conectividade entre Servidores.

(c) Comprimento Médio dos Caminhos.

(d) Servidores alcançáveis ao longo do tempo.

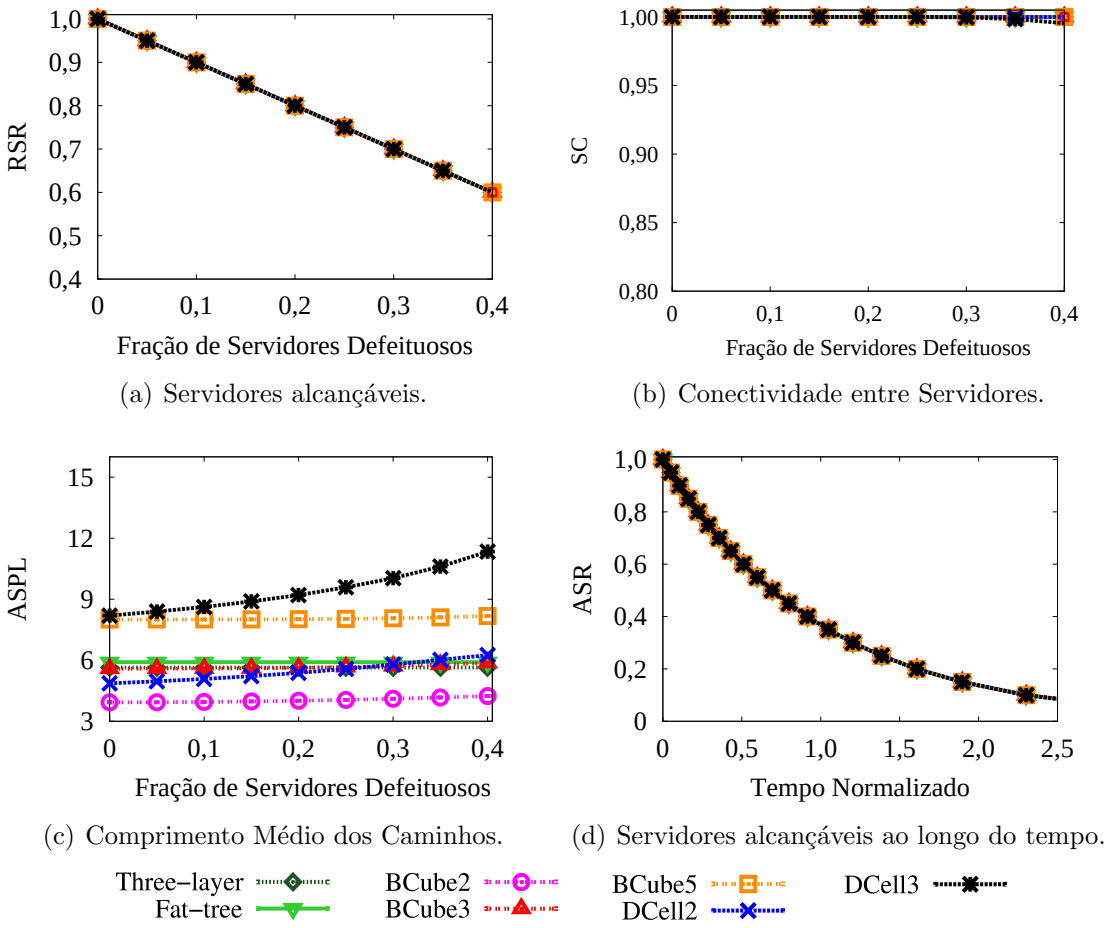
Figura 3.7: Análise da Fase de Sobrevivência para falhas de comutador.

A Figura 3.7(d) mostra a evolução da RSR ao longo do tempo, considerando falhas de comutador. Como no caso de falhas de enlaces, o último resultado do gráfico é obtido para um Tempo Normalizado de $2,3$, que corresponde a uma Fração de Comutadores Defeituosos de $0,9$. Comparando com os resultados de falhas de enlace (Figura 3.6(d)), mostra-se que as topologias possuem degradação mais lenta considerando falhas de comutador do que para falhas de enlace. Além disso, é possível notar a alta sobrevivência da DCell3, que mantém uma alta RSR durante um longo período de falhas de comutadores. Como observado anteriormente, esse comportamento mostra que essa configuração possui uma baixa dependência de comutadores.

Falhas de Servidor

A Figura 3.8(a) mostra que, para todas as topologias, a RSR reduz linearmente de acordo com a Fração de Servidores Defeituosos. Apesar de a BCube e a DCell dependerem do encaminhamento por servidores, a Alcançabilidade do Serviço é igual à da Fat-tree e da Three-layer quando servidores são removidos. Isso indica que na

BCube e na DCell em média *uma falha de servidor não causa desconexão de outros servidores da rede.* Para todas as topologias, os resultados de SC são sempre 1 para a faixa de FER considerada.



(a) Servidores alcançáveis.

(b) Conectividade entre Servidores.

(c) Comprimento Médio dos Caminhos.

(d) Servidores alcançáveis ao longo do tempo.

Figura 3.8: Análise da Fase de Sobrevivência para falhas de servidor.

Apesar dos resultados favoráveis de Alcançabilidade do Serviço para falhas de servidor, a Figura 3.8(c) mostra que o comprimento dos caminhos na DCell apresenta certo acréscimo com o aumento das falhas (p.ex., até 3 saltos com uma FER de $0,4$). Isso ocorre pois a DCell é mais dependente do encaminhamento por servidores do que outras topologias.

A evolução da RSR ao longo do tempo é apresentada na Figura 3.8(d). Esse resultado indica que a Alcançabilidade do Serviço dos servidores restantes (isto é, que não estão defeituosos) não é afetada por falhas de servidores por um longo período.

3.4 Análise de Desempenho Qualitativa

Considerando os resultados para a Fase Confiável e para a Fase de Sobrevivência, a Tabela 3.4 fornece uma comparação qualitativa das topologias de DCN em termos

Tabela 3.4: Análise de desempenho qualitativa das topologias de DC, considerando igualmente a Fase Confiável e a Fase de Sobrevivência.

Tipo de Falha	Quesito	Three-layer	Fat-tree	BCube	DCell
Enlace	**Alcançabilidade**	ruim	pobre	boa	razoável
	Qualidade dos Caminhos	excelente	excelente	boa	razoável
Comutador	**Alcançabilidade**	ruim	pobre	boa	excelente
	Qualidade dos Caminhos	excelente	excelente	excelente	boa
Servidor	**Alcançabilidade**	excelente	excelente	excelente	excelente
	Qualidade dos Caminhos	excelente	excelente	excelente	boa

da Alcançabilidade e da Qualidade dos Caminhos. O quesito de Alcançabilidade agrupa as métricas MTTF e Alcançabilidade do Serviço (RSR), visto que os resultados quantitativos mostram que essas métricas estão intimamente relacionadas (isto é, uma boa Alcançabilidade do Serviço implica em um bom MTTF). As topologias são avaliadas considerando cinco níveis qualitativos: "ruim", "pobre", "razoável", "boa" e "excelente". Os critérios para a escolha de cada um estão detalhados adiante na Seção 3.4.1. Note que falhas de comutador não produzem queda severa de desempenho em topologias centradas em servidores. Assim, mesmo a DCell possuindo melhor desempenho que a BCube para falhas de comutador, essa última ainda apresenta um melhor desempenho global, visto que não é classificada como "ruim", "pobre" ou "razoável" em nenhum quesito. Além disso, a tabela mostra que a Qualidade dos Caminhos não é afetada substancialmente por nenhum tipo de falha.

3.4.1 Critérios Utilizados

Detalha-se a seguir os critérios considerados na análise qualitativa de Disponibilidade e Qualidade dos Caminhos.

Disponibilidade

Utilizaram-se cinco níveis qualitativos de desempenho: ruim, pobre, razoável, boa e excelente. Os critérios adotados são detalhados a seguir:

- todas as topologias são consideradas como "excelente" na análise de servidor pois as falhas de servidores não causam impacto nos servidores restantes;
- a topologia Three-layer é sempre considerada como "ruim" para as análises de falhas de enlace e de comutador, pois apresenta o pior desempenho nas simulações;
- para as falhas de enlace e comutador, considerou-se como valor de referência "excelente" o desempenho da topologia DCell3 para falhas de comutador e

FER de $0,4$. Essa topologia apresenta uma RSR muito próxima de 1 para um alto valor de FER $(0,4)$ e também um alto MTTF;

- para as falhas de enlace e comutador, considerou-se como valor de referência "pobre" o desempenho da topologia Fat-tree para falhas de comutador e FER de $0,4$. Nessa topologia a RSR decresce linearmente de acordo com a FER, e seu MTTF é significativamente menor que todas as novas topologias para ambos tipos de falha.

- o desempenho da BCube5 não foi considerado, visto que não foi utilizada na análise uma DCell com o mesmo número de interfaces de rede;

- para um determinado tipo de falhas (enlace ou comutador), a topologia é classificada como "excelente" se, para um valor de FER de $0,4$, **pelo menos** uma de suas configurações (número de interfaces de rede) possuir desempenho próximo (diferença de $0,01$ na RSR) ao valor de referência "excelente", e **todas** as configurações possuírem uma RSR superior a $0,8$;

- para um determinado tipo de falhas (enlace ou comutador), a topologia é classificada como "pobre" se, para um valor de FER de $0,4$, **pelo menos** uma de suas configurações (número de interfaces de rede) possuir desempenho próximo (diferença de $0,01$ na RSR) ao valor de referência "pobre", e **todas** as configurações possuírem uma RSR inferior a $0,8$;

- caso a topologia não atenda aos critérios para ser "pobre" ou "excelente", classifica-se essa topologia como "boa" se, para **todas** as configurações, possuir uma RSR superior a $0,8$ para um valor de FER de $0,4$. Senão, é classificada como "razoável".

Qualidade dos Caminhos

Utilizou-se cinco níveis qualitativos de desempenho: ruim, pobre, razoável, boa e excelente. Os critérios adotados são detalhados a seguir:

- como a Qualidade dos Caminhos não altera significativamente para todos os tipos de falha, nenhuma topologia foi considerada como "ruim" ou "pobre" nesse quesito;

- o desempenho da BCube5 não foi considerado, visto que não foi utilizada na análise uma DCell com mesmo número de interfaces de rede;

- para um determinado tipo de falhas, a topologia é considerada como "excelente" se, para um valor de FER de $0,4$, possuir **todas** suas configurações com

um ASPL menor ou igual a 6. Esse valor de referência "excelente" é a métrica calculada pela Fat-tree, que não se altera com o aumento de falhas;

- para um determinado tipo de falhas, a topologia é considerada como "razoável" se, para um valor de FER de $0,4$, possuir **pelo menos** uma de suas configurações com um ASPL maior que 12. Esse valor de referência 'razoável" corresponde ao dobro do valor de referência "excelente";
- para um determinado tipo de falhas, a topologia é classificada como "boa" caso não atenda aos critérios para ser "razoável" ou "excelente". Note que, para a Qualidade dos Caminhos, os critérios para a topologia ser considerada como "excelente" são menos severos do que no caso da Disponibilidade. Essa abordagem foi escolhida pois, como dito anteriormente, a curva da RSR sofre mais variações com o aumento do FER do que a curva do ASPL.

3.5 Análise de Sensibilidade à GPD

Nesta seção estuda-se como a escolha do número de *gateways*, ou a Densidade de Portas de Gateway (GPD), influencia a confiabilidade e a sobrevivência. No caso da sobrevivência, calcula-se apenas a Alcançabilidade do Serviço. A Qualidade dos Caminhos diz respeito aos caminhos entre servidores dentro do DC, não dependendo da escolha dos *gateways*.

Os resultados das Seções 3.2 e 3.3 foram obtidos com a máxima GPD (Seção 3.1.3), que é 1 para as topologias Fat-tree, BCube e DCell, e 0,007 para a Three-layer. Nesta seção, inicialmente calcula-se as métricas com a mínima GPD de cada topologia. Em outras palavras, escolhe-se em cada topologia apenas um comutador como *gateway*. Como a rede possui um *gateway*, calcula-se apenas a RSR, já que o SC é sempre 1 (isto é, existe apenas uma sub-rede operacional) quando ao menos um servidor está alcançável. As simulações revelaram que as métricas MTTF e Alcançabilidade do Serviço considerando falhas de enlace e de servidor não são substancialmente afetadas se a mínima GPD é escolhida, quando comparado ao caso de máxima GPD. Dessa forma, os resultados para esses tipos de falha são omitidos, apresentando-se apenas resultados para falhas de comutador.

A Figura 3.9 mostra os resultados para a Fase Confiável quando a GPD é a mínima possível. À exceção da DCell3, a redução do MTTF e da FER Crítica é pequena se comparada aos resultados da Figura 3.4(b). Os resultados pra a DCell3 mostram que:

- Apesar de a DCell3 ser altamente confiável para falhas de comutador, a escolha da mínima GPD produz um ponto único de falha, reduzindo em 29% seu MTTF;

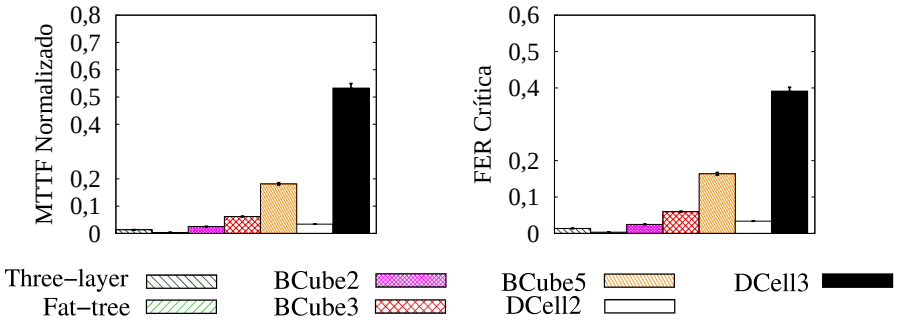


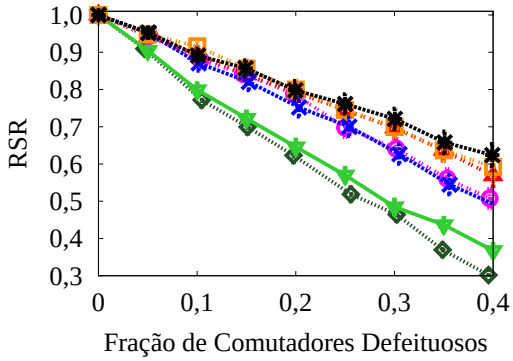
Figura 3.9: Análise da Fase Confiável para falhas de comutador, utilizando o GPD mínimo.

- Mesmo com a GPD mínima, a confiabilidade da DCell3 permanece maior que das outras topologias, como também observado nos resultados de GPD máxima (Figura 3.4(b).)

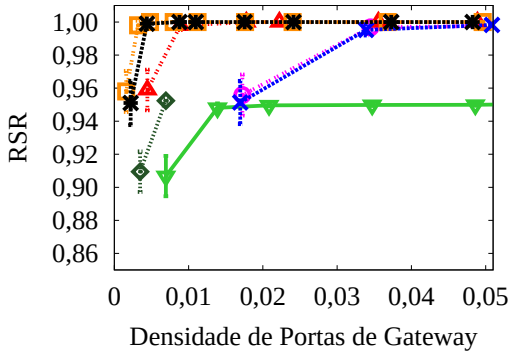
A Figura 3.10(a) mostra que a sobrevivência considerando falhas de servidores é altamente afetada pela mínima GPD, se comparada com os resultados de máxima GPD da Figura 3.7(a). A figura mostra também que topologias de um mesmo tipo (isto é, centradas em comutadores ou em servidores) possuem um comportamento bem próximo para a RSR. Com a mínima GPD, uma alta redução na sobrevivência é esperada pois as topologias possuem um único elemento responsável pela conectividade de toda a rede. Assim, a rede torna-se totalmente desconectada se houver uma falha no *gateway*. Além disso, ao aumentar a FER, a probabilidade de falha nesse comutador aumenta, reduzindo em média a RSR.

Para completar a análise anterior, a Figura 3.10(b) mostra a RSR em função da escolha da GPD. Esse resultado é obtido para uma Fração de Comutadores Defeituosos de $0,05$, pois nesse valor a maioria das topologias possuem desempenho próximo mas já apresentam alguma degradação do RSR. Assim, para esse valor de Fração de Comutadores Defeituosos, varia-se a GPD de cada topologia desde seu valor mínimo até um valor de aproximadamente $0,05$. Esse último valor de GPD foi escolhido, pois a partir dele os valores de RSR não se alteram para todas as topologias. Note que, apesar de as curvas serem mostradas como linhas contínuas para facilitar a visualização, para cada topologia as duas GPDs mais baixas na figura correspondem à utilização de 1 e 2 *gateways*. Os resultados indicam que a RSR reduz significativamente apenas quando a GPD é mínima, mostrando a alta robustez das topologias em relação a esse valor. Assim, conclui-se que:

- A robustez à GPD está mais relacionada à probabilidade de um *gateway* ser escolhido para ser retirado na simulação, do que à perda de acesso a esse elemento (isto é, quebras dos caminhos internos que levam aos *gateways*). Isso pode ser concluído pois as falhas de enlace não reduzem significativamente



(a) Fração de Servidores Alcançáveis.



(b) RSR em função da GPD para uma Fração de Comutadores Defeituosos de 0,05.

Figura 3.10: Análise de Sensibilidade à Densidade de Portas de Gateway, para falhas de comutador.

a confiabilidade e a sobrevivência, enquanto que para falhas de comutador o impacto é significativo.

- A escolha do número de *gateways* possui pequena influência na Alcançabilidade do Serviço. Uma degradação severa causada pela escolha da GPD é observada apenas quando um único *gateway* é escolhido e a rede é suscetível a falhas de comutador.

É importante observar que os resultados desta seção não abordam a confiabilidade e sobrevivência de acordo com falhas do próprio *gateway* (p.ex. falha do enlace com a Internet.). Os resultados desta seção apenas provam que o acesso aos *gateways* não é afetado substancialmente pela falha aleatória dos elementos de rede, à exceção do próprio *gateway*. Obviamente, se as falhas no *gateway* fossem consideradas, a alcançabilidade do DC inteiro aumentaria ao aumentar a redundância dos *gateways*. Entretanto, nesta seção esse tipo de falha não é analisado, visto que o interesse é analisar as características internas ao sítio. Além disso, *gateways* são geralmente mais fáceis de monitorar e reparar visto que são menos numerosos que

outros elementos de rede.

3.6 Trabalhos Relacionados

Este capítulo fornece uma análise da resiliência considerando a utilização de topologias de DCN propostas recentemente. De certa forma, este trabalho complementa um estudo existente presente em [3] no qual compara-se o desempenho das topologias Fat-Tree, BCube e DCell considerando falhas de comutador e servidores. Nesse trabalho, Guo *et al.* avaliam a sobrevivência das topologias definindo a métrica ABT (*Aggregate Bottleneck Throughput* - Vazão agregada de gargalo) Para calcular essa métrica, o trabalho considera que cada servidor envia um fluxo de pacotes a todos os outros servidores. A vazão de gargalo (*Bottleneck Throughput*) é definida como a menor vazão alcançada considerando todos os fluxos. Assim, a ABT é definida como o número de fluxos existentes multiplicado pela vazão de gargalo. Na avaliação de Guo *et al.* utiliza-se apenas uma configuração para cada topologia (BCube e DCell com respectivamente 4 e 3 interfaces por servidor, e uma Fat-tree com 5 níveis de comutadores) com um número total de 2.048 servidores. Além disso, consideram a utilização dos esquemas de roteamento originalmente propostos para cada uma das três topologias. O trabalho conclui que a BCube possui bom desempenho para falhas de comutador e servidores, enquanto a Fat-tree sofre de uma alta queda da ABT quando comutadores falham. Por outro lado, os resultados mostram que a DCell possui uma baixa ABT mesmo no caso sem falhas, mas esse valor não muda significativamente quando a rede é submetida a falhas. O trabalho desta tese difere de [3] visto que a análise proposta não é restrita a padrões de tráfego específicos e esquemas de roteamento, sendo assim genérica e com foco em aspectos topológicos. Além disso, este trabalho fornece métricas que permitem a avaliação do tempo de vida completo da topologia, e analisa a relação entre o número de interfaces de servidor e a resiliência de cada topologia. Finalmente, este trabalho avalia a Alcançabilidade do Serviço, que não é abordada por Guo *et al.*.

Bilal *et al.* [44] analisam a resiliência a falhas da Fat-tree, DCell e de um topologia convencional em três níveis. Eles demonstram que as métricas clássicas de resiliência, derivadas da Teoria dos Grafos (p.ex., o grau médio dos nós), não fornecem sozinhas uma medida acurada de resiliência em DCNs. Da mesma forma que Guo *et al.* e a análise desta tese, Bilal *et al.* medem diferentes métricas a partir da variação do número de elementos defeituosos. Além disso, propõem uma métrica que contabiliza todas as métricas que eles analisam. A partir dessa métrica, concluem que a DCell possui melhor resiliência a falhas do que uma topologia convencional em três níveis e do que a Fat-tree, enquanto a convencional possui o pior desempenho. Diferente desse trabalho, esta tese analisa em mais detalhes a alcançabilidade

dos servidores de acordo com falhas na rede, permitindo enfatizar algumas características topológicas que permitem uma maior resiliência das topologias, para um determinado tipo de falhas. Ademais, esta tese fornece uma análise da BCube e DCell com diferentes números de portas de servidores, enquanto o trabalho de Bilal *et al.* foca em uma especifica configuração da DCell com dois tamanhos diferentes e não analisa a BCube. Finalmente, esta tese fornece uma avaliação da degradação do DC ao longo do tempo, o que permite modelar e analisar o MTTF das topologias consideradas.

Ainda considerando topologias intra-sítio, Ni *et al.* [45] fornecem uma análise teórica dos requisitos de banda passante nas redes centradas em comutadores Fat-tree e VL2, considerando a falha de k enlaces da rede. Esse trabalho conclui que a Fat-tree necessita de menos capacidade nos enlaces que a VL2 para sobreviver (isto é, possibilitar que os servidores utilizem toda a capacidade de suas interfaces de rede) a k falhas quando k é pequeno. Para valores grandes de k, o trabalho mostra que a Fat-tree necessita de mais capacidade que a VL2.

Existem ainda estudos que fornecem medidas de resiliência em DCs reais. Vishwanath e Nagappan [46] fornecem uma caracterização de falhas de servidores em DCs, analisando um ambiente com mais de 100.000 servidores espalhados em diferentes países e continentes. Entre outras observações, esse trabalho conclui que a causa da maioria das falhas de servidores é a falha de seus discos rígidos. Gill *et al.* [35] medem o impacto dos elementos de rede na resiliência do DC. Para isso, utilizam *logs* de eventos de falha em alguns DCs reais. Apesar de Gill *et al.* não considerarem medidas em DCs utilizando as novas topologias de DCN, eles concluem que comutadores de prateleira possuem um alto nível de resiliência. Consequentemente, uma baixa fração de elementos defeituosos (FER) pode ser alcançada na utilização de topologias de baixo custo como Fat-tree, BCube e DCell. Além disso, eles destacam que as DCNs convencionais são altamente resilientes, apresentando mais de $0,9999$ de disponibilidade em aproximadamente 80% dos enlaces e 60% dos elementos de rede. Entretanto, como esse estudo foca apenas em DCNs convencionais, essa conclusão pode não ser válida para cenários DCN emergentes, como MDCs e arquiteturas de baixo custo.

Os próximos capítulos focam na DCN inter-sítio, apresentando diretrizes e estratégias para projetar um DC resilientes a desastres, que são falhas que abrangem uma grande região geográfica e podem tornar indisponíveis sítios inteiros.

Capítulo 4

Diretrizes para o Projeto de Centros de Dados Resilientes a Desastres

A resiliência de serviços de rede pode ser expressa através de métricas de qualidade de serviço (QoS - *Quality of Service*) ou, mais especificamente, como métricas de qualidade de resiliência (QoR - *Quality of Resilience*) [47]. Como exemplo de métricas de QoR, podem ser citadas a disponibilidade do serviço (ou seja, a fração do tempo na qual o serviço permanece operacional) e o tempo de recuperação do serviço após uma falha. Por outro lado, a avaliação de QoS utiliza outras métricas de desempenho, como a latência na rede e a taxa de perda de pacotes. Os provedores de IaaS geralmente expressam seus níveis de QoR em termos da disponibilidade das VMs durante um certo intervalo, definindo-os em um acordo de nível de serviço (SLA - *Service Level Agreement*). Por exemplo, os serviços IaaS *Amazon Elastic Computer Cloud* (*Amazon EC2*) [8] e *Rackspace Cloud Servers* [48] garantem uma disponibilidade de 99,95% e 100%, respectivamente. Nesses casos, o serviço é considerado indisponível se todas as VMs de um cliente não possuírem conectividade externa (ou seja, não estão acessíveis pela Internet). Dessa forma, o compromisso do provedor de IaaS é reembolsar seus clientes de forma proporcional ao tempo de indisponibilidade, caso este atinja um nível superior ao especificado no SLA. Alguns provedores de IaaS definem seus níveis de resiliência em termos da redundância de suas infraestruturas. A Rackspace, por exemplo, especifica que seus servidores físicos estão equipados com a tecnologia RAID 10 e possuem fontes de alimentação redundantes. Além disso, a Rackspace menciona que sua rede é dimensionada de tal forma que uma falha em um comutador pode reduzir à metade a capacidade de rede, ao invés de causar a indisponibilidade do serviço.

Uma característica comum dos SLAs mencionados anteriormente é o fato de

eles não cobrirem falhas fora do controle do provedor de IaaS (p.ex., um ataque de negação de serviço) e outros eventos de força maior, como enchentes. Em outras palavras, um SLA típico para um serviço IaaS não considera a resiliência a desastres [49]. Entretanto, um provedor de IaaS poderia optar por ser resiliente a desastres, garantindo um certo QoR após a ocorrência de um desastre [50].

Para oferecer resiliência a desastres, um provedor de IaaS deve realizar backups das VMs operacionais e os deixar em modo de espera, sendo ativados apenas após um desastre. Além disso, uma VM operacional deve estar geograficamente isolada de seu backup de forma que um determinado desastre não afete os dois ao mesmo tempo. Para tal, o centro de dados que hospeda as VMs deve ser geograficamente distribuído e requer uma rede que, por si só, seja resiliente a desastres e possua uma boa relação custo-benefício. O levantamento de requisitos de projeto de centro de dados IaaS resilientes a desastres é ainda um problema em aberto apesar da importância para permitir um plano de continuidade de negócios (BCP - *Business Continuity Planning*) em provedores de IaaS. O BCP consiste em diferentes requisitos, técnicos ou não, para garantir que alguns serviços estejam disponíveis mesmo quando desastres ocorrerem. Para fazer a infraestrutura de TI estar de acordo com o BCP de uma determinada empresa, a equipe de TI precisa adotar um processo chamado Gerenciamento da Continuidade de Serviços de Tecnologia da Informação (ITSCM - *Information Technology Service Continuity Management*), que pode ser realizado através de diferentes arcabouços, como os procedimento definidos no estágio *Service Design* (Projeto de Serviço) da *Information Technology Infrastructure Library* (ITIL) [51] e o padrão ISO/IEC 24762:2008 [52]. A implementação e o teste de sistemas de recuperação são exemplos de tais procedimentos.

Este capítulo fornece diretrizes para projetar uma infraestrutura de DC que suporte uma nuvem resiliente a desastres. Essas diretrizes são organizadas em fases inter-relacionadas. A primeira fase é iniciada com considerações preliminares de projeto, como a avaliação de riscos de desastre e definição dos requisitos dos clientes. Nas fases seguintes, escolhem-se os mecanismos de recuperação de desastres, além da infraestrutura de rede e o mecanismo de posicionamento de máquinas virtuais. É importante observar que a proposta das diretrizes neste capítulo não possui como intenção substituir os arcabouços de ITSCM existentes, que possuem um escopo mais amplo, mas atuar em conjunto com eles para projetar uma nuvem IaaS resiliente a desastres. Além disso, neste capítulo chama-se a atenção para tópicos de pesquisa incipientes, como o projeto físico de centros de dados geodistribuídos e o posicionamento dos backups das VMs. Finalmente, motivados pelas diretrizes propostas, os dois próximos capítulos apresentam uma análise do compromisso entre a resiliência e latência no projeto de DCs geodistribuídos, além de uma estratégia de posicionamento de servidores primários e de backup nesse tipo de DC.

4.1 Projeto de Centros de Dados Geodistribuídos

Baseando-se na literatura sobre resiliência a desastres em WANs ópticas [9, 53, 54] e nuvens [6, 50], determinam-se neste capítulo as principais fases de projeto de uma infraestrutura de centro de dados que suporte uma nuvem IaaS resiliente, como mostrado na Tabela 4.1. Na primeira fase, denominada "Planejamento", define-se todos os requisitos de projeto, como características dos desastres a serem suportados e níveis de resiliência oferecidos. A fase de "Modelagem" é empregada para descrever a relação entre esses requisitos e os componentes do centro de dados projetados nas próximas três fases, através de modelos matemáticos e de simulação. Dessa forma, a fase de "Modelagem" deve ser executada após o fim de cada uma das três próximas fases, objetivando considerar nos modelos os novos mecanismos adicionados por cada fase. Com base nos requisitos de projeto já definidos, na fase de "Escolha dos Mecanismos de Recuperação de Desastres", o projetista do centro de dados escolhe, para cada tipo de cliente, a frequência na qual os backups são sincronizados com as VMs operacionais, além de realizar outras decisões em relação à recuperação de falhas. A fase de "Posicionamento de Sítios e Projeto da Topologia" é empregada para projetar a infraestrutura WAN, baseando-se na frequência que os backups são realizados entre sítios (ou seja, no consumo de banda gerado pelos backups) e em outros requisitos, como o máximo custo financeiro admitido. O projeto realizado nessa fase consiste no dimensionamento da capacidade da rede do centro de dados e na escolha do número de servidores instalados em cada sítio. Finalmente, na fase de "Escolha dos Mecanismos de Posicionamento de VMs", o projetista seleciona e configura mecanismos para alocar as VMs e seus backups a cada nova requisição dos usuários, baseando-se no conhecimento da topologia da WAN e nos requisitos e restrições do projeto. Normalmente, as três últimas fases devem ser executadas na ordem dada pela Tabela 4.1. Entretanto, o projetista pode retornar a uma fase anterior se as decisões impedirem a realização da fase atual. Por exemplo, quando a frequência escolhida para realizar sincronização do backup necessita de um aprovisionamento inviável de capacidade na rede WAN. As subseções a seguir detalham cada uma das fases propostas. Algumas dessas fases estão organizadas em tarefas, que consistem em procedimentos genéricos importantes para concluir cada fase. Todavia, a lista de tarefas não é exaustiva e, assim, tarefas mais específicas podem ser adicionadas a cada fase dependendo do cenário considerado e tecnologias empregadas.

4.1.1 Planejamento

Nesta fase, realiza-se o planejamento inicial do projeto do DC através das seguintes tarefas.

Tabela 4.1: Principais fases para o projeto de um centro de dados geodistribuído.

Nome	**Objetivos**
Planejamento	Verificar os possíveis riscos de desastres, definir requisitos de QoR e QoS e traçar restrições de custo.
Modelagem	Definir modelos de rede e de falha a serem utilizados em todas as fases do projeto.
Escolha dos Mecanismos de Recuperação de Desastres	Escolher os mecanismos de detecção de desastres, recuperação de VMs e de recuperação da rede após desastres.
Posicionamento de Sítios e Projeto da Topologia	Definir quais localizações da área geográfica serão utilizadas pelos sítios e projetar a WAN de interconexão de sítios.
Escolha dos Mecanismos de Posicionamento de VMs	Selecionar os mecanismos para alocar as VMs no centro de dados, especificando suas políticas em relação ao isolamento entre VMs de backup e operacionais, além do cumprimento dos requisitos de QoS e QoR.

Definição dos Riscos de Desastre

Os riscos de desastre são situações a serem consideradas no projeto da nuvem resiliente, como quedas de energia em larga escala e furacões. Para tal, todos os possíveis desastres devem ser listados, assim como é necessário analisar o efeito de cada um na infraestrutura do DC. A partir dessa lista, um subconjunto dos tipos de desastres é selecionado para as próximas fases a partir da análise da importância de cada um. Por exemplo, o provedor pode escolher ignorar um desastre muito raro, como a ocorrência de terremotos graves no Brasil.

Como desastres são difíceis de prever, situações mais genéricas de desastres podem ser consideradas. Por exemplo, uma estratégia pode ser projetar um DC resiliente a qualquer falha de um sítio inteiro ou de um link, ou a falhas em todos os elementos dentro de uma região [53].

Definição dos Requisitos de Resiliência a Desastres

Nesta tarefa, o provedor de nuvem define os níveis de QoR e os SLAs correspondentes. Os valores médios de QoR, como a disponibilidade utilizada pela Amazon EC2, geralmente não são adequados para quantificar a resiliência a desastres de uma infraestrutura, visto que desastres podem ser muito raros [6]. Ao invés disso, as métricas mais comuns de QoR para desastres são o Objetivo do Tempo de Recuperação (RTO - *Recovery Time Objective*) e Objetivo do Ponto de Recuperação (RPO - *Recovery Point Objective*). O RTO caracteriza a quantidade de tempo ne-

cessária para restaurar um serviço após esse ser afetado por um desastre. Para um determinado cliente IaaS, o RTO depende do tempo para detectar a falha, restaurar as VMs afetadas a partir do sítio de backup, reiniciar todos os serviços que estavam executando nessas VMs e redirecionar o tráfego de rede do sítio original até o sítio de backup. A outra métrica de interesse, o RPO, é a diferença de tempo entre o momento da última sincronização do backup (p.ex., cópia dos discos virtuais) e o momento do desastre. O RPO fornece uma ideia de perda de dados após o desastre. Alguns serviços necessitam de valores baixos ou nulos de RPO (p.ex., transações bancárias), exigindo replicação contínua de dados. Um baixo RPO acarreta em alto uso da capacidade da rede entre os sítios do DC, pois realiza-se o envio de uma grande quantidade de dados. Tanto o RTO como o RPO podem variar de alguns segundos até várias horas [6], apesar de ser possível um valor zero de RPO, como mostrado adiante no Capítulo 6.

Definição das Exigências do Projeto

As exigências do projeto são aspectos a serem considerados independentemente da resiliência a desastres. Dentre as exigências mais importantes estão os requisitos de QoS. Esses requisitos influenciam a qualidade de experiência (QoE - *Quality of Experience*), definida na recomendação ITU-T Rec. P.10 como "a aceitação global de uma aplicação ou serviço, percebida subjetivamente pelo usuário final" [55]. Isso significa que a QoE depende de todos os elementos da infraestrutura, como a rede e os servidores físicos, e de fatores externos como o valor cobrado pelo serviço. De fato, é necessário garantir tanto os requisitos de QoR como de QoS para fornecer um QoE satisfatório. Entretanto, os níveis de QoR devem ser garantidos sem comprometer a QoS, pois eventos de desastres são raros, enquanto as métricas de QoS estão em constante percepção pelos usuários finais, tanto diretamente quando indiretamente. Por exemplo, a geodistribuição do DC, com o objetivo de melhorar a resiliência, aumenta a distância entre os sítios. Consequentemente, a latência percebida pelos usuários, que é uma métrica de QoS, pode aumentar quando múltiplas VMs estão espalhadas em diferentes sítios. Esse aspecto, em particular, é analisado adiante no Capítulo 5. Dado o exposto, nesta tarefa o provedor de nuvem deve listar todas os requisitos de QoS para assegurar que os próximos passos do projeto considerem esses requisitos. Além disso, os requisitos de QoR que não estão necessariamente relacionados a desastres, como a disponibilidade, devem ser considerados se constarem nos SLAs.

Dentre as exigências, também se encontram outros fatores como o orçamento disponível para construir o DC, as possíveis localidades para instalar sítios, além de outras exigências relacionadas à capacidade do sítio. Como exemplo dessa última, o

provedor de IaaS pode escolher instalar um determinado número de servidores em um sítio de acordo com a demanda esperada em uma região.

4.1.2 Modelagem

Esta fase define os modelos que capturam as características do cenário definido na fase "Planejamento". Além disso, o modelo descreve como os componentes do DC, escolhidos nas próximas três fases do projeto, afetam as características da infraestrutura (p.ex., custo, métricas de QoS, RPO e RTO). Note que esta fase define as relações entre todas as outras fases, que podem variar de acordo com cenário considerado. Os modelos definidos nesta fase basicamente dependem das características dos desastres considerados e os parâmetros da rede.

As características dos desastres, obtidas na fase "Planejamento", são utilizadas para construir modelos de desastre. Os modelos de desastres para redes de comunicação podem ser determinísticos, probabilísticos, ou podem considerar as diferentes camadas de rede [53] Um modelo determinístico clássico é dado pela utilização de grupos de risco compartilhado (SRGs - *Shared Risk Groups*). Um SRG é definido como um grupo de elementos suscetíveis a uma falha comum. Por exemplo, considerando quedas de energia, um SRG pode ser composto de sítios ligados a uma mesma subestação elétrica. As propostas apresentadas adiante nos Capítulos 5 e 6 utilizam esse tipo de modelo para o projeto de DCs. Os modelos do próximo tipo, denominados probabilísticos, consideram que cada componente ou grupo de componentes falha com uma determinada probabilidade. Como desastres e seus impactos na rede são difíceis de prever e são muito raros, os modelos determinísticos são preferíveis. Por fim, as abordagens que consideram as diferentes camadas fazem parte de um tema de pesquisa ainda incipiente, que modela o efeito da falha em uma camada nas camadas superiores a ela. Por exemplo, uma única falha na camada física, como rompimento de cabos, pode afetar múltiplas rotas IP, podendo acarretar a queda de diversas conexões TCP. Por outro lado, a recuperação de um rompimento de cabo pode ser rapidamente realizada por camadas inferiores, sendo assim imperceptível para as camadas superiores [9]. Os modelos que consideram a organização da rede em camadas são mais complexos do que os determinísticos e probabilísticos, necessitando de mais informações dos protocolos utilizados e das características das falhas.

Os parâmetros da rede, como a distribuição do tráfego, comprimento dos enlaces e banda disponível são modelados por técnicas convencionais de modelagem de rede. Por exemplo, a rede pode ser modelada como um grafo, no qual os nós são os sítios do DC e as arestas são os enlaces entre eles. Os pesos das arestas podem ser atribuídos de acordo com parâmetros de QoS, como a latência entre os sítios. A

latência pode ser obtida através da execução de algoritmos de menores caminhos no grafo. Os modelos em grafo podem ser combinados com informações de SRG para capturar os riscos de desastre. Nesse caso, os SRGs definidos no modelo de desastre são compostos por nós e arestas. Através da utilização de algoritmos oriundos da Teoria dos Grafos, é possível determinar quais sítios são afetados por cada SRG. A proposta de modelos que capturam todas as métricas de resiliência com base nos parâmetros da rede é ainda um problema de pesquisa em aberto. Esse modelo deve capturar, por exemplo, como o aumento da banda entre sítios do DC afeta os níveis de RPO oferecidos pelo provedor de IaaS. Além disso, o modelo pode descrever como a reconfiguração da rede e a ativação dos backups das VMs, descritos na próxima seção, afetam os níveis de RTO. No Capítulo 6 estuda-se, entre outros fatores, a capacidade de serviços IaaS com RPO zero quando implantados em redes WAN reais, com topologia e banda disponível nos enlaces conhecidas.

4.1.3 Escolha dos Mecanismos de Recuperação de Desastres

Nesta fase, o provedor de nuvem escolhe os mecanismos de recuperação, que impactam diretamente o RTO e o RPO, executando as seguintes tarefas.

Escolha dos Mecanismos de Detecção de Desastres

Apesar de todos mecanismos de detecção de falhas empregados nas várias camadas da rede (p.ex., reação dos protocolos de roteamento), o DC deve utilizar mecanismos que definam quando as VMs no sítio de backup se tornarão operacionais. Como o RTO depende do tempo de reação a um desastre, a detecção de falhas possui um papel importante na QoR. A detecção pode ser realizada através de sondas periódicas enviadas ao sítio a partir de um ou diversos pontos na rede, utilização de alarmes de rede, etc. Obviamente, quanto mais frequentes forem as sondas e os alarmes de rede, mais curto será o RTO, acarretando um maior tráfego de gerenciamento.

Escolha dos Mecanismos de Recuperação de VMs

O uso de *snapshots* é uma estratégia apropriada para a recuperação de VMs após desastres [6]. Um *snapshot* é uma cópia do estado da VM em um dado momento, que pode incluir seu disco, seu estado da memória e suas configurações. A maioria das plataformas de virtualização suportam *snapshots*, o que permite um DC manter o *snapshot* de suas VMs em sítios de backup e ativá-los após um desastre. Note que esse esquema força o retorno dos serviços em execução nas VMs a retornarem a um

estado anterior, afetando o RPO. De fato, *snapshots* mais frequentes se traduzem em menores valores de RPO, apesar de utilizarem mais recursos da rede para realizar as transferências. Recentemente, esquemas foram propostos para fornecer zero RPO para VMs utilizando redes de longa distância [13]. Entretanto, como mostrado mais adiante no Capítulo 6, esses esquemas exigem uma alta capacidade da rede. Dado o exposto, a escolha da frequência para realizar *snapshots* depende das classes de QoR escolhidas, bem como das exigências do projeto, como a quantidade de banda alocada para o serviço.

Escolha dos Mecanismos de Reconfiguração da Rede

Quando a VM passa a executar em outro sítio após um desastre, a infraestrutura de rede precisa re-rotear o tráfego destinado a essa VM. Assim, esta tarefa de projeto escolhe os mecanismos apropriados para realizar a reconfiguração da rede após um desastre. Os provedores de nuvem geralmente empregam serviços de *Domain Name System* (DNS) para redirecionar o tráfego da VM após a mudança de sua localização física. O servidor DNS da nuvem é então responsável por responder às consultas DNS com o endereço IP atual. Por exemplo, a Amazon Web Services (AWS) oferece um serviço de DNS denominado Amazon Route 53. Os serviços de DNS na nuvem geralmente se baseiam em roteamento *anycast*, no qual qualquer nó que execute um determinado serviço pode responder às requisições, permitindo a reconfiguração da rede após desastres [56]. Alternativamente, os provedores podem utilizar protocolos de redes de sobrecamada na nuvem, que suportam diversas formas de encapsulamento IP ou Ethernet, permitindo a localização flexível das VMs além de isolamento das fatias da rede entre os clientes [57]. É importante observar que os mecanismos de configuração de rede possuem um alto impacto no RTO, visto que afetam o intervalo no qual os pontos de rede da VM permanecem inacessíveis após um desastre.

4.1.4 Posicionamento de Sítios e Projeto da Topologia

Nesta fase escolhe-se a topologia de rede WAN do centro de dados, como também a localização geográfica de cada sítio. Apesar de a topologia da rede interna ao sítio ser também uma decisão de projeto, geralmente a resiliência a desastres não é considerada, pois um desastre geralmente leva à falha de um sítio inteiro. Assim, a redundância da estrutura interna ao sítio, como considerada pela análise do Capítulo 3, geralmente é empregada para fins de alta disponibilidade, mas não resiliência a desastres.

Posicionamento de Sítios do DC

Esta tarefa define em quais localidades instalar os sítios do DC em uma dada região geográfica, de forma a minimizar o impacto dos desastres. Além disso, o posicionamento de sítios define quantos servidores serão instalados em cada sítio, e quantos servidores serão utilizados para hospedar os backups das VMs.

Os DCs geodistribuídos tendem a ser espalhados por diferentes SRGs, sendo mais resilientes a desastres do que DCs centralizados. A Figura 4.1 ilustra diferentes níveis de distribuição de um DC, utilizando uma topologia WAN da rede de educação e pesquisa (REN - *Research and Education Network*) francesa, denominada RENATER. Cada círculo da figura representa um ponto de presença (PoP - *Point of Presence*). Um sítio é representado por uma figura de um servidor, e consiste em um PoP com, no mínimo, um servidor instalado. Nos três níveis de distribuição (isto é, número de sítios utilizados) mostrados na figura, distribui-se 1024 servidores pela rede, escolhendo um dado número de PoPs para hospedar os servidores. Considerando um modelo de desastres de falha única, no qual cada sítio e enlace do DC pertence a um SRG diferente, a figura indica a fração de servidores disponíveis após a falha de pior caso (isto é, a falha que desconecta o maior número de servidores). Um servidor é considerado disponível se possui pelo menos um caminho para um *gateway*, representado por um triângulo na figura. Como a rede desse exemplo é altamente redundante em termos de caminhos para os *gateways*, o SRG de pior caso é sempre a falha de um sítio inteiro. Assim, um alto nível de distribuição torna o DC mais resiliente a desastres, pois cada sítio tende a abrigar menos servidores. Note, entretanto, que a diferença entre a "Média Distribuição" e "Alta Distribuição" é pequena em termos de resiliência, pois é mais difícil melhorar a resiliência a desastres após um determinado nível de distribuição. Mais detalhes sobre esse aspecto são abordados no Capítulo 5, no qual um problema de otimização é proposto para projetar um DC distribuído resiliente.

Outra vantagem da geodistribuição é a redução do número de servidores de backup necessários quando o DC se torna mais resiliente a desastres, como exemplificado na Figura 4.2. Assume-se, nesse caso, que um desastre pode tornar inacessível um sítio inteiro e os sítios do DC estão suficientemente distantes entre si, de forma a evitar falha em dois ou mais sítios ao mesmo tempo. A figura ilustra diferentes esquemas de posicionamento de DCs para fornecer possibilidade de backup para 12 servidores. O backup é realizado pelo envio periódico de *snapshots* das VMs dos sítios operacionais aos sítios de backup. Considere o esquema de 1 sítio, no qual o DC possui apenas um sítio primário e todas as VMs são copiadas para um sítio de backup. Se um sítio primário sofrer um desastre, as VMs do DC serão executadas no sítio de backup após os procedimentos de recuperação. Como o DC possui ape-

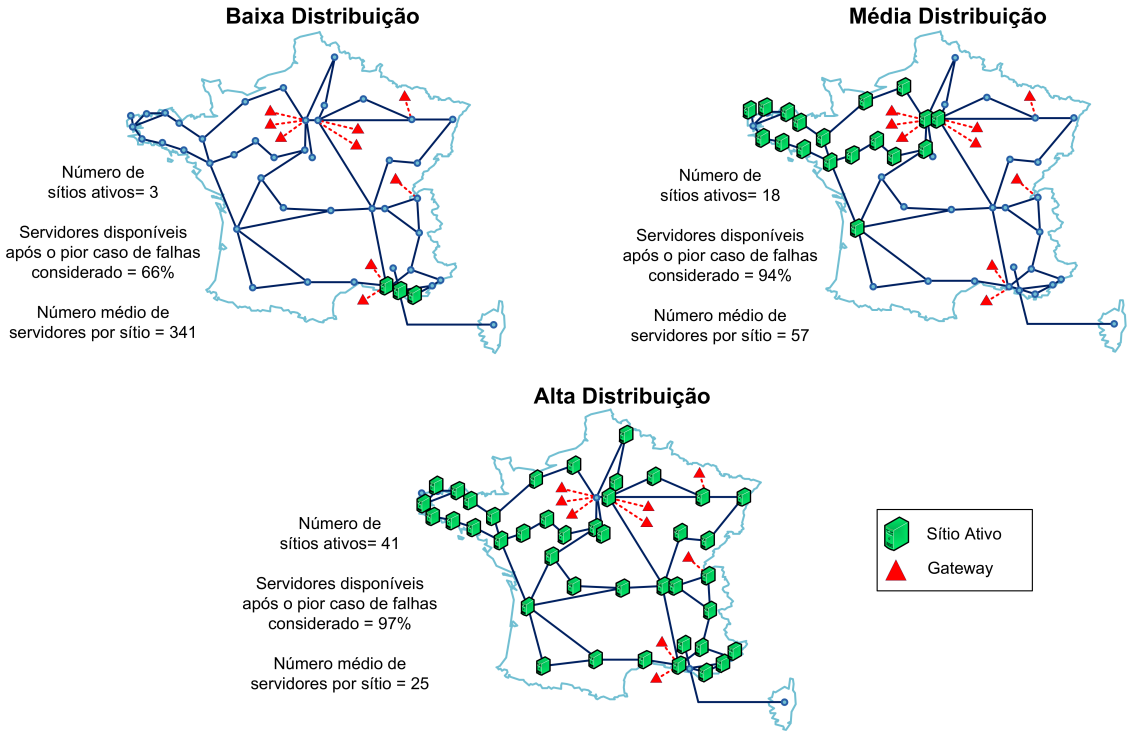


Figura 4.1: Impacto da distribuição do DC na resiliência.

nas um sítio primário e é necessário suportar a falha de um sítio inteiro, o sítio de backup possui a mesma capacidade que o primário em termos de VMs suportadas e armazenamento de discos virtuais. Consequentemente, o esquema com um único DC representa uma opção custosa em termos de capacidade de VM utilizada para backup, que se traduz em custos na aquisição de servidores de VMs. Essa situação não é diferente do caso no qual a virtualização não é utilizada e um DC inteiro é designado como backup. Entretanto, como mostrado no esquema com 2 sítios, é possível distribuir o sítio primário em dois sítios diferentes, cada um com metade do número de servidores do que no esquema com apenas 1 sítio. Como W1 e W2 não falham ao mesmo tempo, o sítio de backup não precisará executar as VMs de ambos W1 e W2 após uma situação de desastre. Assim, o sítio de backup precisa suportar apenas metade das VMs se comparado com o caso anterior. Vale notar, entretanto, que a capacidade de armazenamento necessária continua a mesma, visto que B1 precisa abrigar os *snapshots* de todas as VMs do DC. Utilizando o mesmo raciocínio, o esquema com 4 sítios reduz em quatro vezes a capacidade de servidores necessária em comparação com o esquema com 1 sítio. Mais detalhes sobre a economia de servidores permitida pelo geodistribuição são analisados no Capítulo 6, no qual um problema de otimização é proposto para alocar sítios operacionais e de backup em uma rede WAN.

Apesar das vantagens, a distribuição do DC pode ser custosa. Utilizando ainda o exemplo da Figura 4.2, cada sítio adicional no DC exige um enlace adicional

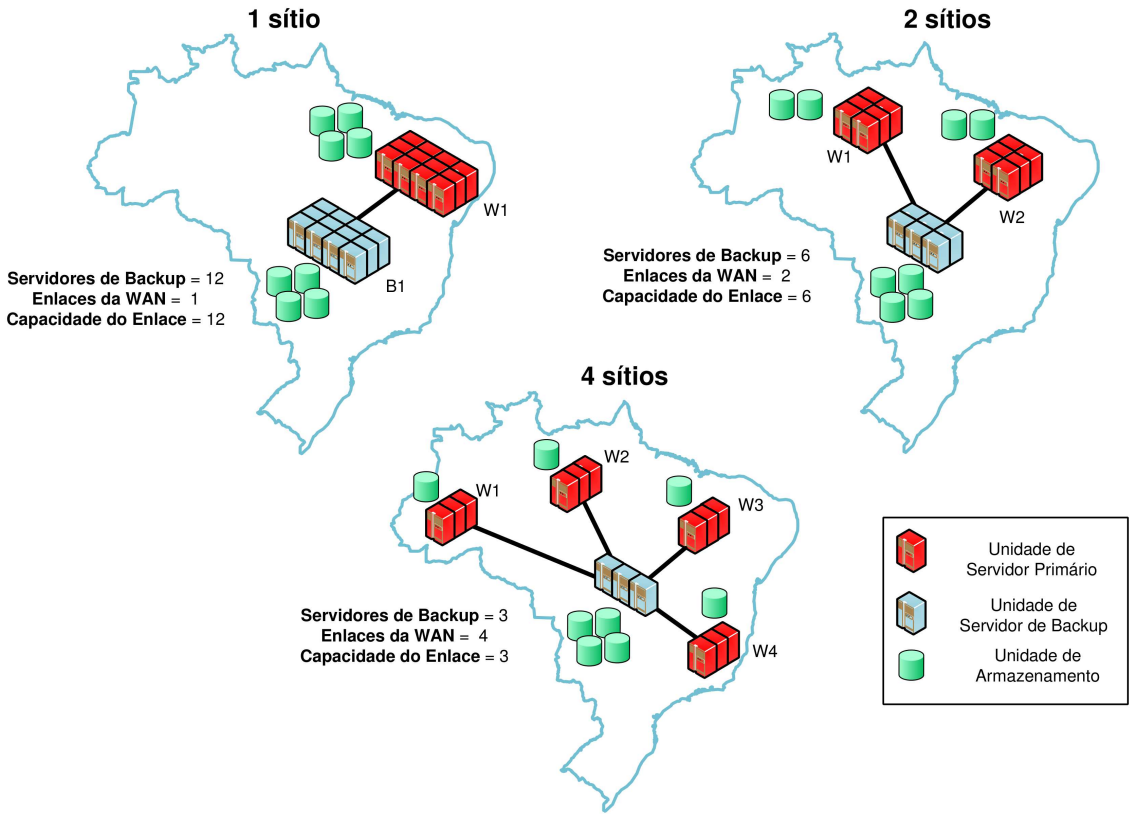


Figura 4.2: Número de servidores de backup e a distribuição do DC.

na WAN. Apesar de a capacidade necessária em cada enlace se reduzir quando aumenta-se a distribuição do DC (isto é, menos *snapshots* de VM são transferidos por cada enlace), o custo de interconexão de sítio aumenta com a distância entre eles. O custo para instalar novos sítios no DC também deve ser considerado e depende de diversos fatores, como a capacidade de rede necessária e preocupações com a segurança e a disponibilidade. Além disso, o custo para instalar um único sítio dependerá de sua localização geográfica, sendo afetado por fatores como o custo do metro quadrado na localização desejada, requisitos de refrigeração dadas as condições meteorológicas, impostos locais, etc. Maiores informações sobre custos de DC podem ser encontradas na página web [58], que consiste em uma iniciativa para estimar o custo dos componentes do DC de acordo com diferentes parâmetros.

Dada a discussão anterior, a distribuição do DC deve considerar as exigências definidas na fase "Planejamento", como o orçamento e métricas de QoS. Por exemplo, além de realizar a distribuição considerando a resiliência desastres, o provedor pode preferir instalar os sítios mais perto dos clientes que possuam requisitos de latência mais estritos.

Projeto da WAN Inter-sítio

Esta tarefa é principalmente caracterizada pelo projeto conhecido de redes WAN para operadores de telecomunicações. A literatura nesse tipo de problema é vasta e geralmente aborda o projeto de redes ópticas [53]. Um requisito de projeto de rede é melhorar a resiliência da topologia da rede, através do emprego de técnicas como restauração de caminhos e proteção (p.ex., provisionamento de caminhos de backup), roteamento de múltiplos caminhos, e p-cycles [9]. Diferentemente de redes tradicionais de telecomunicações, nas quais o principal objetivo é conectar diferentes PoPs, em redes de DC o objetivo é fornecer hospedagem de VM aos clientes em uma região. Assim, o projeto da rede inter-sítio precisa ser correlacionado com a tarefa "Posicionamento de Sítios do DC", visto que a capacidade da rede e o nível de proteção atribuídos a um sítio dependem do número de servidores instalados nele, como também de sua localização. Além disso, esta tarefa define a localização dos *gateways*, a instalação de enlaces e a capacidade alocada para um, a localização dos roteadores e comutadores utilizados para interconectar os sítios, etc [59].

4.1.5 Escolha dos Mecanismos de Posicionamento de VMs

Apesar de o posicionamento de sítios e projeto da topologia desempenharem papéis importantes na resiliência a desastres, apenas essas decisões de projeto não são suficientes para a garantia dos requisitos de QoR dos clientes. A QoR também é afetada pelo Mecanismo de Posicionamento de VMs, que aloca as VMs aos servidores físicos a partir das requisições recebidas dos clientes. Geralmente, as VMs operacionais são alocadas de forma a maximizar o lucro do provedor e atender aos requisitos dos usuários. Entretanto, ao oferecer recuperação de desastres, os provedores de IaaS devem também alocar uma VM de backup para cada VM operacional. Como a resiliência a desastres não deve afetar o QoE em situações normais de operação, o posicionamento das VMs de backup deve considerar os requisitos dos usuários como a QoS. Uma alternativa simples para assegurar esses requisitos é executar o posicionamento de VMs em duas fases. Na primeira fase são escolhidos os sítios para alocar as VMs operacionais, baseando-se nos requisitos de QoS, enquanto na segunda fase escolhe-se os sítios que hospedarão os backups das VMs dos clientes. Dessa forma, a localização dos backups deve ser escolhida de forma que a VM operacional e o backup não falhem ao mesmo tempo.

A Figura 4.3 mostra um exemplo do posicionamento de VMs em duas fases. O centro de dados da figura é distribuído em diferentes sítios de uma região e os SRGs, circulados por linhas tracejadas, indicam quais sítios podem falhar ao mesmo tempo. Para cada enlace na rede WAN, a figura mostra a capacidade de rede disponível entre os dois sítios. Nesse exemplo, após a primeira fase de posicionamento,

as VMs de um dado cliente são alocadas em dois sítios. Esse posicionamento é realizado de acordo com os requisitos de QoS do cliente e outras métricas de QoR não relacionadas a desastres, como a disponibilidade, não especificadas na figura. Na segunda fase, o mecanismo de posicionamento decide onde colocar o backup de cada VM. Esse segundo posicionamento deverá ser realizado de forma a reduzir a probabilidade de que a VM primária e seu backup correspondente sejam afetados pelo mesmo evento de desastre. Para isolar os backups, é necessário, na medida do possível, alocá-los em sítios que pertençam a SRGs que não contenham os sítios que hospedam as VMs operacionais. Essa abordagem deve considerar as métricas de QoR definidas pelo cliente, garantindo que os recursos disponíveis são suficientes para garantir os requisitos de RTO e RPO. A Figura 4.3 mostra três casos possíveis de posicionamento. O primeiro tira proveito da alta quantidade de banda disponível entre os sítios B e C, alocando os *snapshots* do sítio B no sítio C e vice versa. Apesar da alta quantidade de banda permitir uma alta frequência do envio de *snapshots* das VMs operacionais, permitindo um baixo RPO, esse posicionamento torna as VMs de backup e as operacionais suscetíveis a uma falha comum (SRG 4). A segunda possibilidade é aumentar a resiliência alocando todos os *snapshots* de VM no Sítio A, que não pertence aos mesmos SRGS dos Sítios B e C. Entretanto, em comparação com o primeiro caso, esse posicionamento acarreta uma menor banda disponível para transferir os *snapshots*. Uma desvantagem desse segundo caso é o fato de todos os *snapshots* estarem armazenados em um único sítio. Se o Site A falhar, todos os *snapshots* estarão indisponíveis e o provedor precisará de um maior esforço para restaurar o serviço de replicação de VMs. O terceiro caso de posicionamento melhora a resiliência de todos os casos anteriores, alocando as VMs de backup do Sítio B no Sítio A e o backup do Sítio C no Sítio D. Essa possibilidade possui a desvantagem de utilizar um enlace de baixa capacidade (o enlace de 1 Gbps entre os Sítios C e D) para transferir os *snapshots*. Note que em nenhuma das possibilidades o serviço é recuperado com as características exatas do serviço original, visto que as VMs operacionais estão espalhadas em dois sítios com um enlace de 100 Gbps entre eles. Esse fator também pode ser considerado na segunda fase de posicionamento: recuperar o serviço atendendo os mesmos requisitos de qualidade que esse possuía antes do desastre. Todavia, esse tipo de garantia requer uma infraestrutura de rede de alto custo. Dado o exposto, nesta fase de projeto o provedor de nuvem deve escolher ou desenvolver o mecanismo de posicionamento de VMs, que estará ciente de todos os requisitos de QoS e QoR.

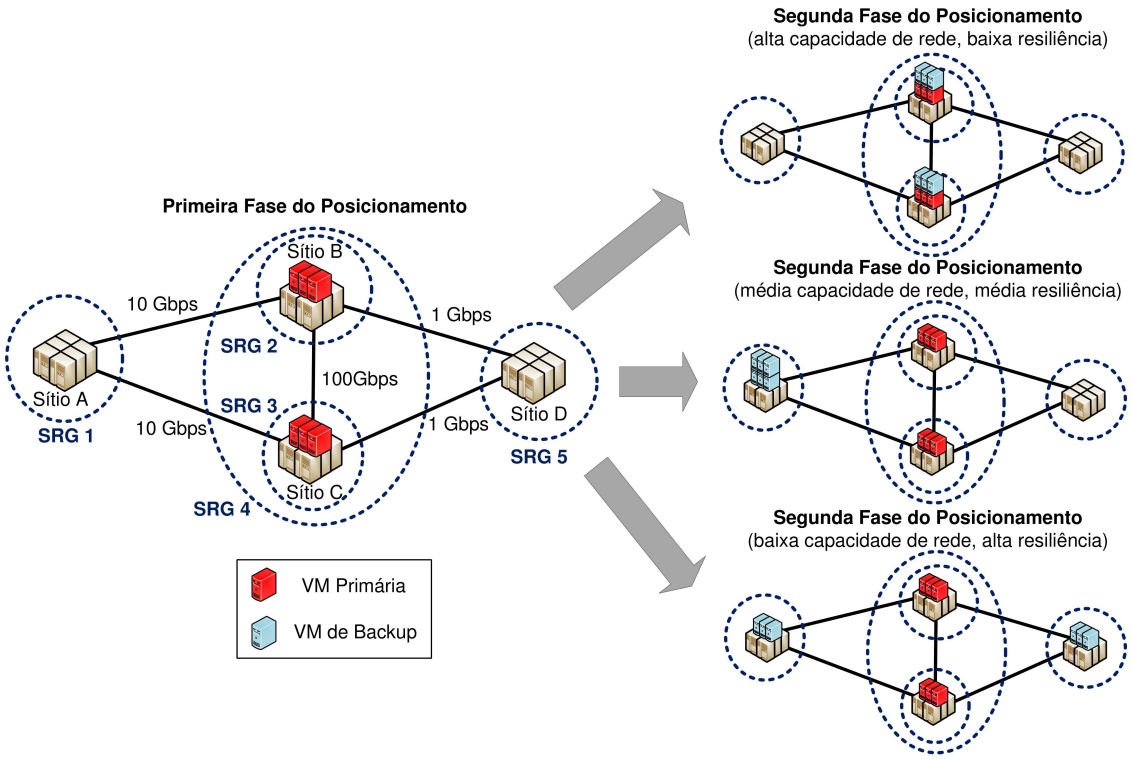


Figura 4.3: Posicionamento das VMs operacionais e a localização dos backups.

4.2 Desafios e Direções de Pesquisa

As fases de projeto descritas neste capítulo permitem a identificação de direções de pesquisa em resiliência a desastres em nuvens IaaS. Apesar de a fase "Escolha dos Mecanismos de Recuperação de Desastres" ser importante para resiliência a desastres, essa possui bastante interseções com outras áreas de pesquisa, como alta disponibilidade (isto é, resiliência a falhas sob controle do provedor) e técnicas como mobilidade de máquinas virtuais e virtualização de redes. Por outro lado, as fases "Posicionamento de Sítios e Projeto da Topologia" e "Escolha dos Mecanismos de Posicionamento de VMs" são as mais desafiadoras pois motivam uma nova área de pesquisa, que é o projeto de redes em nuvem resilientes a desastres. Por fim, a fase "Modelagem" traz importantes desafios de relacionar todos os componentes do centro de dados (p. ex., rede WAN e mecanismos de posicionamento e recuperação), modelando seus comportamentos em relação às métricas de QoR e QoS.

A fase "Posicionamento de Sítios e Projeto da Topologia" possui como maior desafio a otimização conjunta da rede WAN entre os sítios e o posicionamento geográfico de sítios. Por exemplo, um posicionamento no qual o centro de dados é altamente distribuído, de forma a eliminar pontos únicos de falha, pode necessitar de uma rede WAN complexa e custosa. Os trabalhos existentes nessa área geralmente propõem algoritmos de otimização para decidir onde instalar cada sítio do centro de dados e em quais sítios cada serviço será implantado, assim como seus

backups [19, 54]. Além disso, os trabalhos da literatura projetam a rede WAN entre os sítios escolhendo caminhos e seus esquemas de proteção (isto é, configuração de caminhos de backup e seus mecanismos de recuperação de falhas).

No que se refere à resiliência a desastres, esta tese foca na fase "Posicionamento de Sítios e Projeto da Topologia" da Tabela 4.1, em particular na tarefa de Posicionamento de Sítios do DC. Dessa forma, nos dois próximos capítulos são formulados problemas de otimização de forma a melhorar a resiliência do DC em redes WANs já existentes. No Capítulo 5 utiliza-se um problema de otimização para realizar uma análise do compromisso entre resiliência e latência em DCNs. Já no Capítulo 6, propõe-se uma estratégia de posicionamento de servidores primários e de backup em um DC geodistribuído, visando reduzir o número de servidores necessários e aumentar a capacidade de um serviço IaaS.

Capítulo 5

Latência versus Resiliência no Projeto de DCs Geodistribuídos

Este capítulo apresenta a formulação do problema de otimização para o projeto de DCs geodistribuídos, de forma a analisar o compromisso entre latência e resiliência. Esse compromisso é analisado visto que o aumento da resiliência pela geodistribuição torna os sítios do DC mais distantes entre si, o que aumenta o atraso de propagação do sinal entre eles, que é um dos componentes da latência. O estudo do aumento desse atraso é importante, pois alguns milissegundos de acréscimo na latência pode causar um impacto considerável nas aplicações do DC [12, 13]. Este capítulo está organizado da seguinte forma. A Seção 5.1 apresenta o modelo de DC utilizado e os critérios escolhidos. A Seção 5.2 descreve o problema de otimização proposto, enquanto a Seção 5.3 apresenta os resultados. Por fim, a Seção 5.4 descreve os trabalhos relacionados.

5.1 Modelo da Rede Inter-sítio

Utiliza-se nesse capítulo o modelo descrito na Seção 2.2, ao qual acrescenta-se as seguintes características:

- a menor unidade do DC é um bastidor (*rack*), que é um conjunto de servidores interligados. A resiliência da topologia intra-sítio (isto é, topologia que interconecta os servidores) não é considerada, sendo abordada especificamente no Capítulo 3;
- é possível instalar um certo número de bastidores em cada sítio disposto em uma determinada região geográfica. Um sítio ativo é aquele que possui pelo menos um bastidor instalado e operacional;
- o DC é suscetível a falhas de enlace e nó. O nó é um roteador da rede WAN

ou um sítio, enquanto o enlace é o meio físico que interconecta os nós, ou que provê acesso aos *gateways*. Utiliza-se o modelo de falhas determinístico de SRGs, definido na Seção 4.1.2. Cada enlace ou nó pertence a um ou mais SRGs.

Como mencionado anteriormente, a Figura 2.5 ilustra um exemplo do cenário descrito acima. Os sítios, que são interconectados pela WAN, podem hospedar diferentes números de bastidores. Dependendo da topologia da WAN e do número de *gateways*, a rede pode apresentar diferentes níveis de resiliência e latência de interconexão, como descrito nos próximos parágrafos.

O modelo descrito captura as características de um DC que fornece, principalmente, um serviço IaaS. Em serviços IaaS, cada cliente possui diversas VMs alocadas no DC. Além disso, o DC gerencia os recursos destinados às VMs, provendo alocação de recursos computacionais, enlaces virtuais, migração de VMs, etc. Um bastidor pode hospedar VMs de diversos clientes, e as VMs de cada cliente podem ser espalhadas por diferentes bastidores e sítios, de forma a aumentar a resiliência e a elasticidade.

Como as VMs de um cliente podem estar geograficamente distribuídas, a resiliência de um serviço IaaS pode ser aumentada espalhando seus recursos entre diferentes sítios [60, 61]. Entretanto, como as VMs podem possuir padrões de comunicação entre elas (p.ex., o tráfego interno a um DC pode ser maior que o tráfego oriundo de usuários [62]), a geodistribuição aumenta a distância entre bastidores e assim pode afetar o desempenho de suas aplicações. Evidentemente, uma infraestrutura na qual esse compromisso é bem ajustado pode prover uma melhor alocação de VMs. Esse problema é uma preocupação constante em redes para nuvem, especialmente para redes de armazenamento de dados, nas quais um acréscimo de apenas 1 milissegundo no tempo de ida e volta (RTT - *Round-Trip Time*) pode representar uma degradação de desempenho considerável [63]. A seguir são detalhados os dois objetivos analisados.

5.1.1 Resiliência

Para quantificar a resiliência de um DC geodistribuído, utiliza-se neste capítulo o conceito de "sobrevivência de pior caso" definido em [64]. Esse conceito é utilizado definindo a "sobrevivência" como *a menor fração dos bastidores que permanecem alcançáveis após a falha de um único SRG, considerando todos os possíveis SRGs.* Assim, o pior caso de falhas é definido pelo SRG que desconecta o maior número de servidores da rede. Essa definição é apropriada para um DC geodistribuído que fornece serviços IaaS, uma vez que possuir menos de 100% dos bastidores alcançáveis não significa que o DC perdeu sua capacidade de hospedar VMs. Como realizado nos

capítulos anteriores, a palavra "resiliência" é utilizada como um termo geral para caracterizar o comportamento do DC quando suscetível a falhas. A sobrevivência, seguindo a mesma definição da Seção 3.3, é utilizada neste capítulo como um aspecto da resiliência que quantifica métricas de desempenho do DC para diferentes casos de falhas (isto é, os SGRs no caso específico deste capítulo). Formalmente, a métrica de sobrevivência utilizada neste capítulo pode ser calculada por:

$$s = \min_{f \in \mathcal{F}} \left(\frac{\sum_{k \in \mathcal{A}_f} r_k}{R} \right), \tag{5.1}$$

onde $\mathcal{F}$ é o conjunto formado por todos os SRGs, R é o número total de bastidores, $\mathcal{A}_f$ é o conjunto das sub-redes alcançáveis após a falha do SRG f, e r_k é o número de bastidores na sub-rede alcançável $k \in \mathcal{A}_f$. Uma sub-rede alcançável é definida como uma parte da rede que está isolada das outras sub-redes, mas possui acesso a pelo menos um *gateway*. Note que, após uma falha, a rede pode ser particionada em diferentes sub-redes. Se uma sub-rede não possui acesso a um *gateway*, seus bastidores não estão alcançáveis ao mundo externo e então não podem fornecer os serviços IaaS. Por exemplo, se na Figura 2.5 os enlaces B e D falharem, a rede é dividida em duas sub-redes alcançáveis: uma composta pelos sítios 1, 2 e 3, e outra composta pelos sítios 4 e 5. Considerando outro cenário, no qual apenas o enlace A falha, a rede se particiona em duas sub-redes. Uma delas é alcançável e composta pelos sítios 1, 3, 4 e 5. A outra, composta pelo sítio 2, não é alcançável pois não possui caminhos para um *gateway*. Assim, $\sum_{k \in \mathcal{A}_f} r_k$ na Equação 5.1 é o número total de bastidores alcançáveis após a falha do SRG f. Dessa forma, o cálculo da Equação 5.1 é similar ao cálculo da métrica RSR (Equação 3.15) considerando o pior caso de falhas e contabilizando bastidores ao invés de servidores.

De acordo com a definição fornecida acima, a métrica de sobrevivência assume valores no intervalo $[0, 1]$. Seu valor mínimo (zero) ocorre quando todos os bastidores são afetados por um mesmo SRG. O valor máximo, por sua vez, ocorre quando a rede tem um certo nível de redundância e o DC é distribuído de forma que nenhuma falha única de SRG pode desconectar um bastidor.

5.1.2 Latência de interconexão

Neste trabalho assume-se que a latência de interconexão entre os diferentes sítios do DC é afetada principalmente pelo atraso de propagação de seus caminhos. Assim, considera-se que a capacidade de rede é bem provisionada, tornando desprezível o atraso devido ao congestionamento e às retransmissões nos nós intermediários. Partindo desse pressuposto, é apropriado quantificar a latência de interconexão como *o maior atraso entre pares de sítios ativos, considerando todas as possíveis com-*

binações. A escolha do valor máximo como métrica de referência é importante para considerar o fato de que VMs alocadas para um determinado cliente podem ser espalhadas por diversos sítios. Assim, a latência máxima corresponde ao pior caso, no qual a consolidação de VMs é realizada independentemente da localização dos sítios ou quando não há capacidade suficiente para realizar uma melhor alocação. Formalmente, a latência de interconexão é definida como:

$$l = \max_{i,j \in \mathcal{D}} (\Delta_{ij} u_i u_j), \tag{5.2}$$

onde $\mathcal{D}$ é o conjunto de todos os sítios, ativos ou não, Δ_{ij} é o atraso de propagação entre os sítios i e j como definido acima, e u_i é uma variável binária indicando se o sítio i está ativo ou não (em outras palavras, se ele possui pelo menos um bastidor instalado). Considera-se que $\Delta_{ij} = \Delta_{ji}$, uma vez que os caminhos em WANs são geralmente configurados simetricamente. É importante salientar que, neste trabalho, a latência de interconexão é calculada sem considerar situações de falhas, de forma a melhor analisar o compromisso entre latência e resiliência. Entretanto, após uma falha, caminhos alternativos podem ser escolhidos. Se esses caminhos possuírem maiores comprimentos, a latência aumenta.

5.2 Formulação do Problema de Projeto de DCs Geodistribuídos

O problema de projeto de DCs geodistribuídos, formulado neste trabalho com programação linear inteira mista (MILP - *Mixed Integer Linear Programming*), possui o duplo objetivo de maximizar a sobrevivência enquanto minimiza a latência de interconexão. A otimização considera como parâmetros a latência entre os sítios, o número de bastidores suportados em cada sítio, informação sobre SRGs e o número total de bastidores a serem posicionados. A saída do problema fornece a quantidade de bastidores alocados para cada sítio. A Tabela 5.1 resume as notações utilizadas, indicando o tipo de cada termo. As notações do tipo conjunto e parâmetro se referem aos dados do problema, enquanto as variáveis são ajustadas pelo algoritmo de otimização. A formulação MILP é apresentada a seguir:

$$\text{maximizar} \quad (1-\beta)s - \beta \frac{l}{L_{max}} \tag{5.3}$$

$$\text{sujeito a} \quad \sum_{i \in \mathcal{D}} M_{fi} x_i - sR \geq 0 \quad \forall f \in \mathcal{F}. \tag{5.4}$$

$$l - \Delta_{ij} u_i - \Delta_{ij} u_j \geq -\Delta ij \quad \forall i, j \in \mathcal{D},\ i < j. \tag{5.5}$$

Tabela 5.1: Notações utilizadas no problema.

Notação	Descrição	Tipo
$\mathcal{D}$	Sítios candidatos	Conjunto
$\mathcal{F}$	SRGs	Conjunto
M_{fi}	Valor binário indicando se o sítio i permanece conectado após a falha do SRG f	Parâmetro
Δ_{ij}	Atraso de propagação (latência) entre os sítios i e j	Parâmetro
L_{max}	Valor máximo do atraso de propagação entre todos os sítios da rede, ativos ou não.	Parâmetro
R	Número total de bastidores a serem posicionados	Parâmetro
Z_i	Capacidade (máximo número de bastidores suportados) do sítio i	Parâmetro
β	Valor de ajuste da importância da latência em relação à sobrevivência	Parâmetro
s	Sobrevivência do DC	Variável
l	Latência de interconexão do DC	Variável
x_i	Número de bastidores na localização i	Variável
u_i	Valor binário indicando se o sítio i está ativo ($x_i > 0$)	Variável

$$Ru_i - x_i \geq 0 \quad \forall i \in \mathcal{D}. \tag{5.6}$$

$$u_i \leq x_i \quad \forall i \in \mathcal{D}. \tag{5.7}$$

$$\sum_{i \in \mathcal{D}} x_i = R. \tag{5.8}$$

$$x_i \leq Z_i \quad \forall i \in \mathcal{D}. \tag{5.9}$$

$$s \geq 0, \quad l \geq 0, \quad x_i \geq 0 \ \forall \ i \in \mathcal{D}. \tag{5.10}$$

$$s \in \mathbb{R}; \quad l \in \mathbb{R}; \quad u_i \in \{0, 1\}, \ \forall \ i \in \mathcal{D}; \quad x_i \in \mathbb{Z}, \ \forall \ i \in \mathcal{D}. \tag{5.11}$$

O objetivo dado pela Equação (5.3) maximiza a sobrevivência s, como definida na Equação 5.1, enquanto minimiza a latência l, definida na Equação 5.2. O compromisso entre latência e sobrevivência é ajustado na Equação (5.3) pelo fator de escala $0 \leq \beta \leq 1$. Além disso, $L_{max} = \max_{i,j \in \mathcal{D}}(\Delta_{ij})$ normaliza l no intervalo $[0, 1]$. Assim, l e s assumem valores dentro do mesmo intervalo.

Como as Equações 5.1 e 5.2 não são lineares, a linearização de cada uma é dada respectivamente pelas Equações (5.4) e (5.5). Para a sobrevivência, aplicar a Equação (5.4) é equivalente a forçar s a ser menor ou igual ao valor definido na Equação 5.1. Como a Equação (5.3) tenta aumentar a sobrevivência, s assumirá o valor mais alto, que é dado de fato pela Equação 5.1. Utilizando o mesmo princípio, a Equação (5.5) força l a assumir a máxima latência entre dois sítios ativos. Para considerar apenas sítios ativos no cálculo de l, utilizam-se as variáveis binárias $u_i, i \in \mathcal{D}$. Assim, se u_i ou u_j possuírem valor zero para um determinado par de sítios, a restrição dada pela Equação (5.5) não será efetiva para esse par. Por exemplo, se $u_i = 0$ e $u_j = 1$, a restrição será $l \geq 0$. Os valores binários u_i são definidos pelas Equações (5.6) e (5.7), fazendo $u_i = 0$ se $x_i = 0$ e $u_i = 1$ se $x_i > 0$. A Equação (5.8) restringe o número total de bastidores do DC (R), enquanto a Equação (5.9) limita o número de bastidores (x_i) permitido em cada sítio i, respeitando sua capacidade Z_i. Finalmente, as Equações (5.10) e (5.11) definem, respectivamente, os limitantes

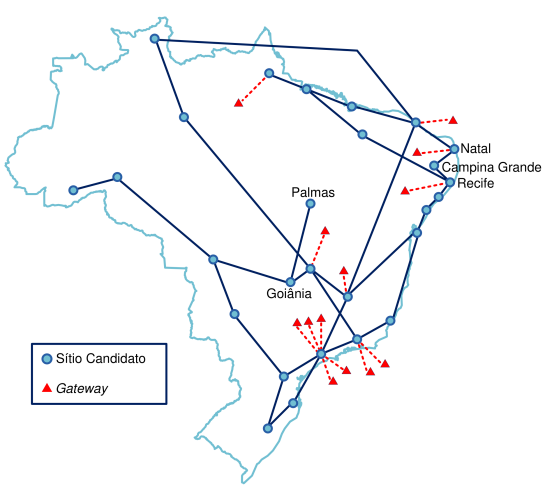
inferiores e o domínio de cada variável.

Para uma dada rede, os parâmetros de latência Δ_{ij} são calculados a partir dos caminhos mais curtos entre os sítios i e j. Os parâmetros binários M_{fi}, para um SRG f, são obtidos pela remoção de todos os elementos (nós e enlaces) que pertencem a esse SRG. Assim, após a remoção, verifica-se para cada SRG quais os sítios possuem acesso aos *gateways*. Obviamente, se um sítio pertence a um determinado SRG, ele já é considerado como desconectado na análise desse SRG.

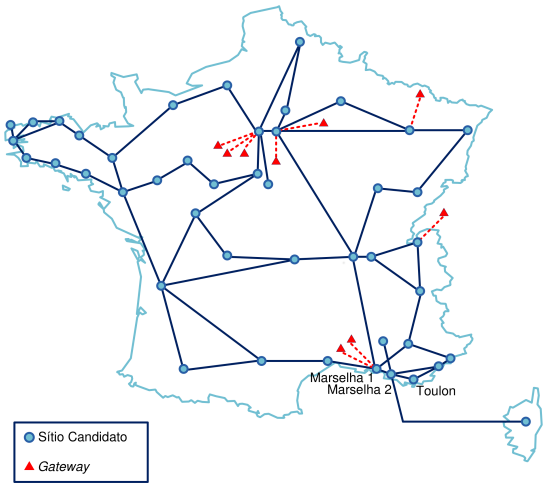
5.3 Avaliação do Compromisso entre Latência e Resiliência

DCs geodistribuídos são geralmente interconectados utilizando uma topologia em malha parcial, na qual alguns sítios possuem um roteador do *backbone* para encaminhar tráfego entre outros sítios da rede [65]. A rede desse tipo de DC é similar a uma WAN, que é tipicamente construída com topologias em malha irregulares. Dessa forma, neste trabalho realiza-se uma análise usando como base topologias de WANs reais de redes de educação e pesquisa (RENs), as quais são formadas por diversos pontos de presença (PoPs). Considera-se que cada PoP dessas redes é um sítio candidato a hospedar bastidores. Neste trabalho, são adotadas as topologias da RNP (Figura 5.1(a)) no Brasil, RENATER (Figura 5.1(b)) na França, e GEANT (Figura 5.1(c)) na Europa. Cada figura das redes utilizadas mostra seus sítios e *gateways*. Para maior clareza, são especificados nas figuras apenas os nomes de sítios mencionados ao longo do texto. Note que cada topologia cobre uma área de tamanho diferente. Em relação à RENATER, as redes RNP e GEANT cobrem uma área muito maior, com uma superfície mais de 10 vezes maior que a França metropolitana. Entretanto, a rede francesa possui mais nós que a RNP. Por fim, a GEANT possui um número de sítios próximo à RENATER.

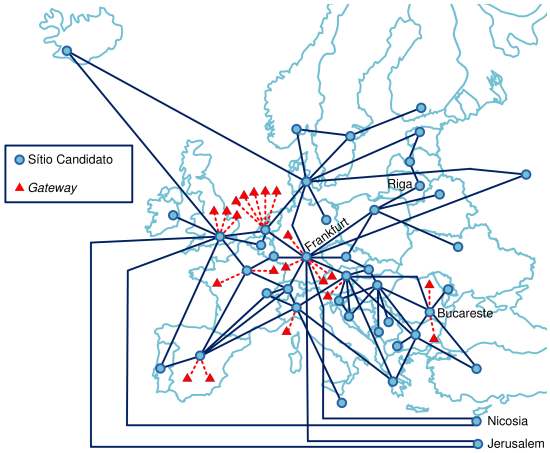
O problema MILP é resolvido neste trabalho utilizando a ferramenta IBM ILOG CPLEX 12.5.1. Além disso, cada nó ou enlace da rede é um SRG. Dessa forma, considera-se o modelo de falha única (*single failure model*) [53]. O modelo de falha única considera que cada elemento da rede (nó ou enlace) falha sozinho, ou seja, não se considera a falha de dois ou mais elementos simultaneamente. É importante notar que esse modelo determina as características das falhas, e não das desconexões, visto que uma única falha de enlace ou nó pode desconectar diversos nós e enlaces da rede. A distância de um enlace é estimada como o comprimento de uma linha reta entre os centros das duas cidades que hospedam os sítios. Assim, o atraso de propagação é diretamente proporcional à distância, e utiliza-se uma velocidade de propagação de 2×10^8 m/s, comumente adotada para análise de redes ópticas [66].



(a) RNP, 27 sítios conectados por 33 enlaces.



(b) RENATER , 45 sítios conectados por 54 enlaces.



(c) GEANT, 42 sítios conectados por 68 enlaces.

Figura 5.1: Topologias de redes de ensino e pesquisa consideradas na avaliação.

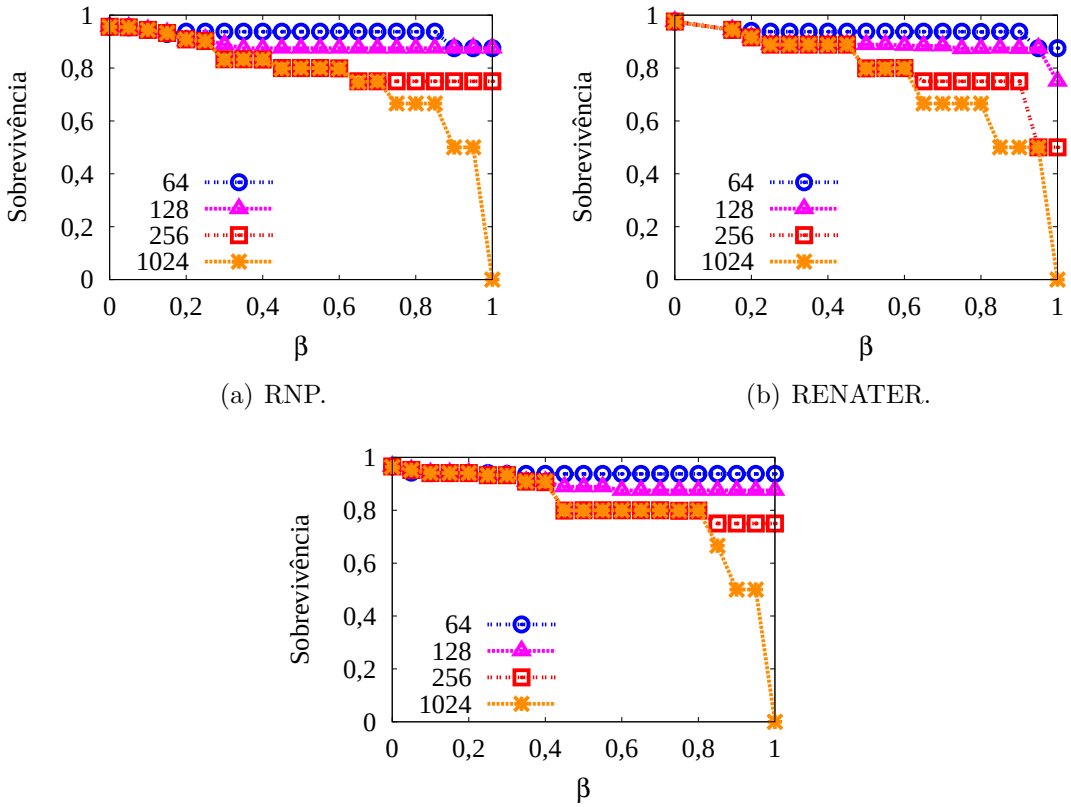
Para cálculo dos parâmetros da rede, utiliza-se a ferramenta de análise de grafos NetworkX [1] [31].

O número total de bastidores considerado nesta avaliação é $R = 1024$. Esse valor foi arbitrariamente escolhido, já que a alocação de bastidores depende da relação entre R e a capacidade Z_i de cada sítio i, e não de valores absolutos. Além disso, a análise é realizada para diferentes valores de Z_i considerando, para simplificar, que todos os sítios possuem a mesma capacidade. Por fim, o compromisso entre sobrevivência e latência é analisado variando o parâmetro β da Equação (5.3). Como visto anteriormente, um valor de β mais alto aumenta a importância da latência em relação à sobrevivência.

As Figuras 5.2 e 5.3 mostram, respectivamente, os valores de sobrevivência e latência obtidos para todas as redes analisadas. Cada curva representa uma capacidade Z_i diferente (64, 128, 256, ou 1024), atribuída a todos os sítios. Como o

[1] A ferramenta NetworkX está disponível em `http://networkx.github.io/`.

número total de bastidores é constante, a redução da capacidade dos sítios força o problema a encontrar soluções com mais sítios ativos. Por exemplo, fixando a capacidade em 64, impõe-se a utilização de pelo menos $\frac{1024}{64} = 16$ sítios ativos. Por outro lado, fixando a capacidade em 1024 é o mesmo que desconsiderar a capacidade no problema, visto que um sítio poderá hospedar todos os bastidores. Como esperado, os resultados mostram que o aumento de β compromete a sobrevivência enquanto melhora a latência (Figura 5.3). Além disso, ao diminuir a capacidade dos sítios o problema de otimização possui menos escolhas de alocação de bastidores, mantendo as métricas constantes para uma faixa de β mais larga.



(a) RNP.

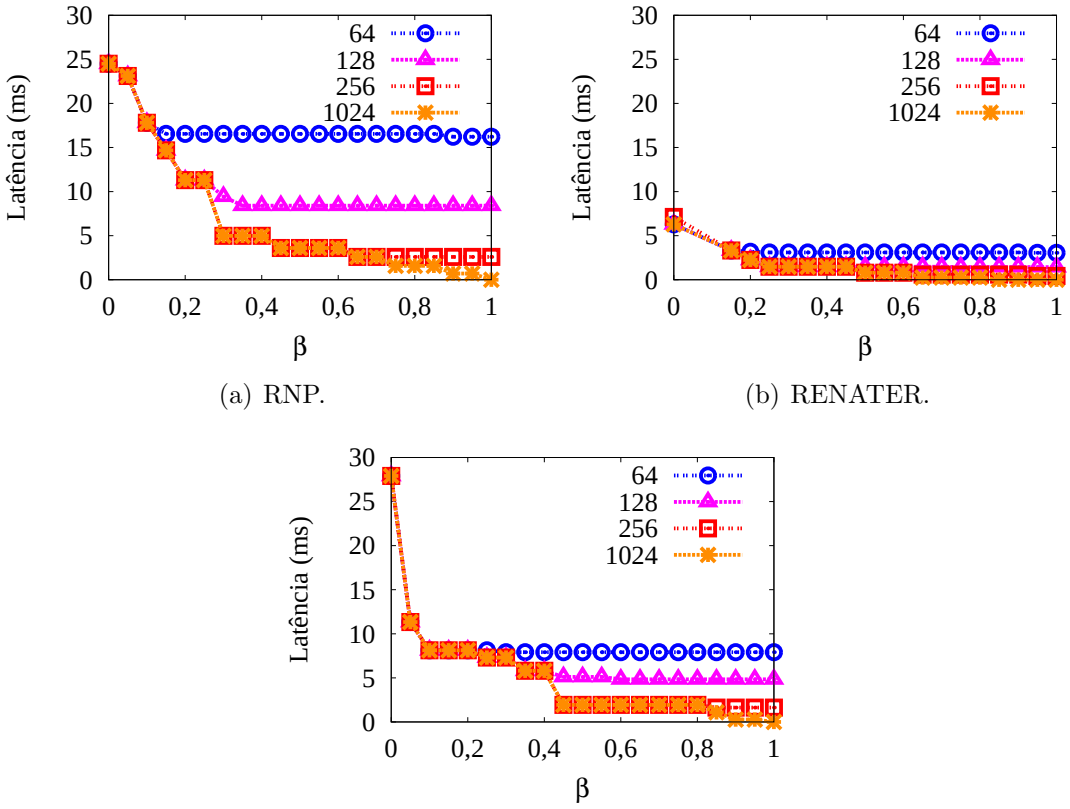
(b) RENATER.

(c) GEANT.

Figura 5.2: Sobrevivência em um DC com 1024 bastidores.

Como mostrado também nas Figuras 5.2 e 5.3, a capacidade dos sítios influencia os valores mínimos possíveis de sobrevivência e latência (valores alcançados para $\beta = 1$). Essa influência é consequência do número mínimo de sítios ativos imposto pela capacidade, como visto mais adiante. Finalmente, pode-se observar que os valores mais baixos de latência são alcançados pela rede RENATER, visto que ela cobre uma área menor que a RNP e a GEANT. Consequentemente, os caminhos na RENATER tendem a ser mais curtos se comparados com as outras duas redes.

Para melhor analisar o compromisso abordado neste trabalho, a Figura 5.4 mostra, para todas as redes, os valores de latência normalizada versus os valores de



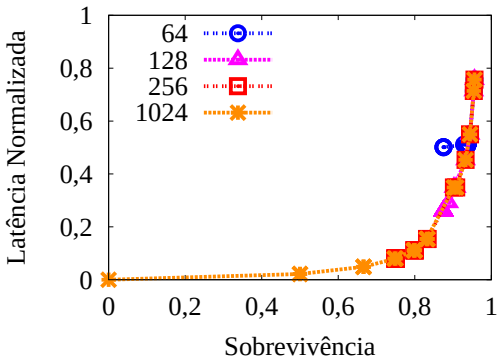
(a) RNP.

(b) RENATER.

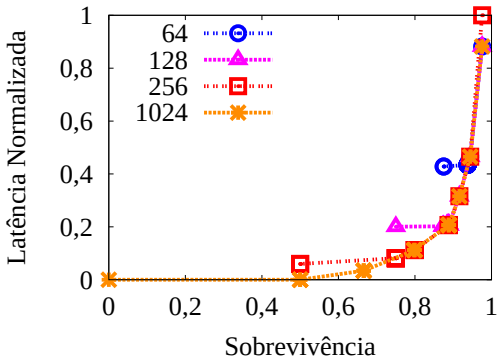
(c) GEANT.

Figura 5.3: Latência em um DC com 1024 bastidores.

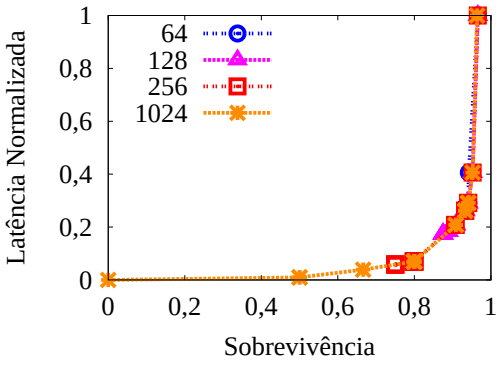
sobrevivência. A latência normalizada é simplesmente $\frac{l}{L_{max}}$. L_{max} é o maior valor possível de latência entre dois sítios, como definido anteriormente, e seu valor para cada rede está indicado nas legendas dos gráficos. Cada ponto da Figura 5.4 é obtido plotando o valor de latência e o valor de sobrevivência alcançados para um mesmo β. Os resultados mostram um comportamento similar para todas as topologias: para valores altos de sobrevivência, um pequeno ganho na sobrevivência representa um alto acréscimo na latência. Isso ocorre pois, em todas as redes consideradas, sempre existem alguns nós com caminhos significativamente longos entre si. Assim, quando os requisitos de sobrevivência aumentam (isto é, β diminui), esses nós são escolhidos devido à falta de opções. Consequentemente, uma pequena melhora na sobrevivência acarreta em um severo acréscimo na latência. Como exemplo, para uma capacidade de 1024 e $\beta = 0$ na GEANT, entre os sítios ativos encontram-se os de Nicósia (Chipre) e Jerusalém (Israel), cada um hospedando 34 bastidores. Apesar de esses sítios estarem próximos geograficamente, o caminho na rede entre eles é bastante longo. O tráfego entre eles deve passar por um nó em Frankfurt (Alemanha), percorrendo no total uma distância de 5.581 km, resultando em uma latência de 27,9 ms. Quando o valor de β é incrementado para 0,05, o que representa uma diminuição pequena da importância da sobrevivência, o pior caso de latência



(a) RNP, $L_{max} = 32,37$ ms.



(b) RENATER, $L_{max} = 7,12$ ms.



(c) GEANT, $L_{max} = 27,9$ ms.

Figura 5.4: Latência versus sobrevivência.

passa a ser o caminho entre Riga (Letônia) e Bucareste (Romênia). Esse caminho possui um comprimento de 2.267 km, possuindo uma latência de 11,33 ms. Note então que a latência sofre uma alta redução (46%) para uma pequena variação de β, como visto na Figura 5.3(c). Entretanto, a consequente redução de sobrevivência é de apenas 2%, como visto na Figura 5.2(c).

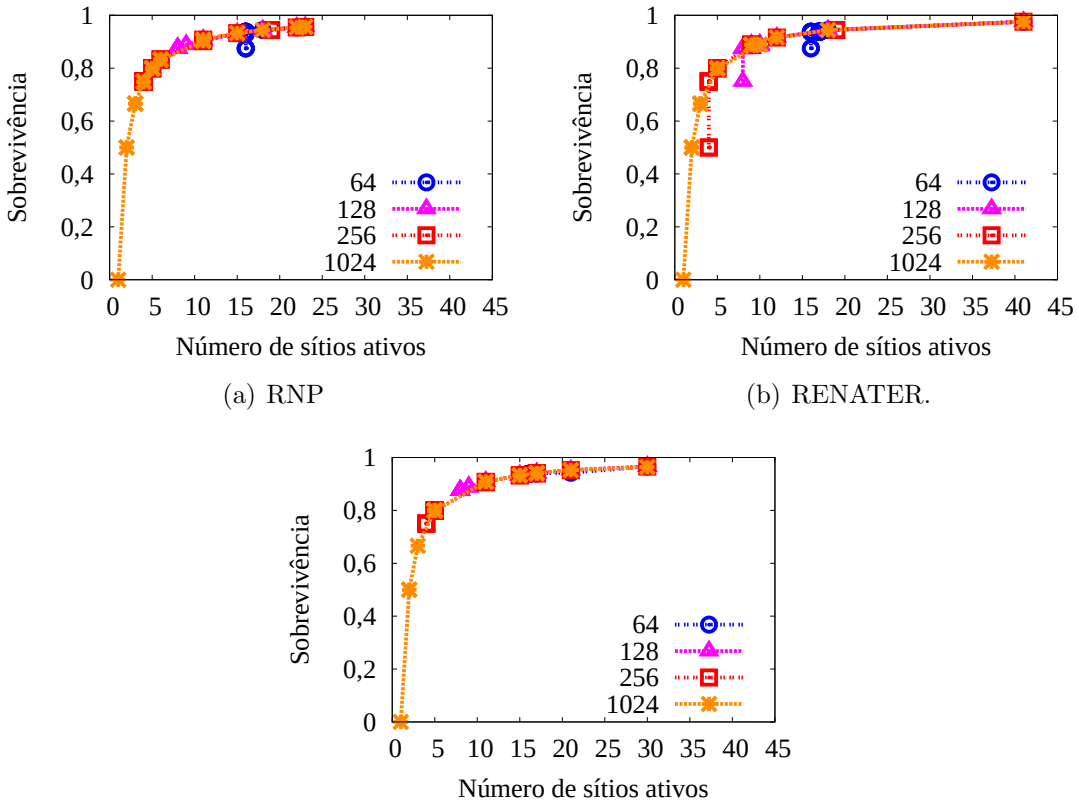
Inversamente ao exposto acima, outro comportamento é observado para todas as redes: para valores mais baixos de sobrevivência, uma redução significativa da sobrevivência resulta em uma melhora insignificante na latência. Esse comportamento ocorre para valores altos de β, em outras palavras, quando a sobrevivência possui baixa importância. Como exemplo, a Figura 5.3(b) mostra que variando o β de 0,90 para 0,95, com uma capacidade de 256 bastidores na RENATER, a latência é reduzida de 0,58 ms para 0,42 ms. Considerando a latência normalizada, isso representa uma redução de 0,081 para 0,059, como mostrado na Figura 5.4(b). Entretanto, essa variação de β reduz a sobrevivência de 0,75 para 0,5 (Figura 5.2(b)), que é uma redução significativa. Por outro lado, esse comportamento mostra que o DC pode atingir valores satisfatórios de sobrevivência com um acréscimo muito baixo na latência. Considerando todas as redes, o maior aumento na latência ao melhorar a sobrevivência de 0 para 0,5 é de 0,70 ms, que ocorre na rede RNP quando β é

reduzido de 1 para 0,95. Nessa mesma rede, uma sobrevivência de 0,8 é alcançada com um acréscimo de apenas 3,6 ms em relação ao caso de um único sítio ativo. Os baixos valores de latência alcançados, mesmo com valores de sobrevivência satisfatórios, são consequência de uma característica comum a todas as redes: todas as topologias consideradas possuem uma alta densidade de sítios em uma determinada região e eles não desconectam simultaneamente após a falha de qualquer SRG. Como exemplo, podem ser citados os sítios no Nordeste do Brasil (Natal, Campina Grande, Recife, etc.) e no Sul da França (Toulon, os dois sítios em Marselha, etc.). Assim, o DC pode ser espalhado nessas regiões sem um aumento significativo da latência.

Em suma, se a exigência de sobrevivência do DC não for muito alta, é fácil assegurar valores moderados de sobrevivência sem causar um aumento significativo da latência. Entretanto, aumentar a sobrevivência para valores próximos de 1 produz um aumento grande da latência entre os bastidores, podendo impactar o desempenho de aplicações com exigências mais estritas de latência. É importante salientar que essas conclusões são válidas para todas as topologias consideradas, e os indícios apontam que são independentes da área geográfica atendida pela WAN e do número de nós e enlaces.

A Figura 5.4 também mostra que, ao diminuir a capacidade dos sítios, a latência mínima possível mantém-se a mesma, ou aumenta. No último caso, isso ocorre apenas para o mínimo valor de sobrevivência ($\beta = 1$) e, após isso, os pontos do experimento encontram-se na mesma curva na qual a capacidade é desconsiderada (capacidade de 1024). Esse comportamento ocorre, pois a limitação da capacidade reduz as possibilidades de alocação de bastidores, podendo inviabilizar soluções melhores. Para $\beta = 1$, o problema não considera a sobrevivência e tenta melhorar a latência o quanto possível. Entretanto, como a gama de soluções possíveis é limitada pelo número mínimo de sítios (p.ex., 16 para uma capacidade de 64), o problema tende a escolher sítios que estejam mais próximos geograficamente de forma a melhorar a latência. Esses sítios mais próximos tendem a estar no mesmo SRG. Por exemplo, considerando a RNP com uma capacidade de 64, a redução ocorre pois a escolha ótima é alocar 2 sítios afetados pelo mesmo SRG: Goiânia e Palmas. Para a RNP isso ocorre apenas para a capacidade de 64, enquanto para a RENATER esse comportamento começa a partir de uma capacidade de 256. Para a GEANT, todos os pontos encontram-se na mesma curva, independente da capacidade.

A Figura 5.5 mostra a métrica de sobrevivência em função do número de sítios ativos, para o mesmo experimento realizado anteriormente. Apesar de a otimização não controlar diretamente essa métrica, o acréscimo da sobrevivência é influenciado pela distribuição do DC. Por sua vez, a distribuição do DC é influenciada pelo número de sítios ativos. Quanto mais sítios são usados, os bastidores do DC tendem



(a) RNP

(b) RENATER.

(c) GEANT.

Figura 5.5: Sobrevivência em função do número de sítios ativos.

a pertencer a mais SRGs e assim a sobrevivência tende a aumentar. Os resultados mostram que a sobrevivência cresce de forma logarítmica com o número de sítios ativos. Isso significa que o aumento do número de sítios ativos não melhora significativamente a sobrevivência quando o DC se torna amplamente distribuído em diferentes regiões. Além disso, os resultados mostram que a rede pode possuir diferentes níveis de sobrevivência para um mesmo número de sítios ativos. Esse comportamento é observado apenas para o valor mínimo de sítios ativos impostos pela capacidade (p.ex., $\frac{1024}{256} = 4$ para uma capacidade de 256). A razão é a mesma daquela observada na Figura 5.4, na qual ao fazer $\beta = 1$ obtém-se uma alta redução de sobrevivência e uma pequena melhora na latência.

5.4 Trabalhos Relacionados

O projeto de DCs geodistribuídos possui diversos fatores em comum com o projeto de redes para computação em grade. Essas redes também podem exigir uma infraestrutura geograficamente espalhada para executar cálculos distribuídos como, por exemplo, em aplicações de e-ciência. Em geral os trabalhos da literatura de projeto de DCs ou de redes em grade formulam problemas de otimização e propõem

heurísticas para resolver esses problemas.

Develder *et al.* [67] formulam um problema de otimização para o projeto de redes para computação em grade considerando a resiliência, que pode também ser utilizado para DCs, como mostrado pelos autores em [17]. O cenário de [67] consiste em diversas fontes que submetem tarefas para nós de destino. Os nós de destino possuem servidores, e todos os nós (fontes e destinos) são interconectados por uma rede óptica. Assim, o trabalho propõe um problema de otimização que determina quais nós da rede executarão cada tarefa, as rotas ópticas entre as fontes que solicitam a tarefa e os destinos que as executam, além da capacidade, em comprimentos de onda, alocada para cada rota. O objetivo do problema é minimizar a capacidade necessária nos servidores e na rede. Para prover resiliência, o problema escolhe destinos e rotas alternativos para cada serviço, considerando o modelo de falha única (isto é, falha de um sítio ou um enlace). Ao invés de simplesmente escolher caminhos alternativos que não possuam enlaces em comum com os respectivos caminhos primários, que é uma abordagem clássica [68], Develder *et. al* escolhe servidores de backup de um determinado destino. Esses servidores de backup podem ser usados para atender às demandas de uma fonte caso a rota até o destino primário apresente falha. Com esse esquema, Develder *et. al* mostram que é possível escolher caminhos de backup que utilizam menos comprimentos de onda do que o esquema clássico, economizando recursos de rede. Em [17], Develder *et al.* estende o trabalho mencionado anteriormente pela adição de proteção N:1 de servidores. Esse tipo de proteção significa que, para cada grupo de N servidores, um servidor adicional é alocado para backup.

Problemas similares são propostos em [19], [18] e [54] para DCs. Habib *et al.* [19], diferente dos outros trabalhos, considera um modelo de múltiplas falhas, no qual mais de um sítio ou enlace pode falhar ao mesmo tempo. Uma característica comum de todos os trabalhos mencionados é o fato de a função objetivo dos problemas de otimização minimizar o custo para prover capacidade de rede e de servidores, deixando a resiliência como uma restrição. Além disso, esses trabalhos assumem que todos os serviços do DC e suas respectivas demandas são conhecidos no momento de sua construção. O atraso de propagação devido à distribuição geográfica é considerado apenas em [18, 54], porém esses trabalhos não analisam o compromisso entre latência e resiliência.

O presente trabalho é diferente do atual estado da arte pois otimiza conjuntamente a latência e sobrevivência, de forma a explorar o compromisso entre eles. Assim, nesta tese isola-se essas duas métricas, ignorando outros fatores como o custo (p.ex., a capacidade utilizada pelas rotas e o custo para ativar um sítio). Além disso, as conclusões apresentadas aqui são independentes dos serviços suportados pelo DC e da matriz de tráfego gerada por esses serviços. O custo e a matriz de tráfego são,

sem dúvida, fatores importantes no projeto de um DC. Entretanto, esses fatores devem ser ignorados em uma primeira análise, para melhor observar o compromisso entre latência e sobrevivência.

Como visto, este capítulo analisa o compromisso entre a resiliência e latência entre os nós de um DC geodistribuído em um problema genérico, independente das aplicações executando no DC. O próximo capítulo foca o caso de IaaS com zero RPO, no qual a latência entre os sítios possui um papel importante, e considera os requisitos de banda passante e economia de servidores.

Capítulo 6

Posicionamento de Servidores Primários e de Backup para IaaS Resiliente

Este capítulo apresenta o problema de otimização proposto para posicionamento de servidores em DCs que fornecem IaaS com perda zero de dados ou de estado das VMs. Ou seja, as VMs possuem zero RPO. Este capítulo está organizado da seguinte forma. A Seção 6.1 apresenta o modelo do serviço considerado e as respectivas decisões de projeto. Com bases nessas decisões, a Seção 6.2 descreve o problema de otimização proposto. A Seção 6.3 apresenta os resultados da análise do problema propostos em redes WAN reais. Por fim, a Seção 6.4 descreve os trabalhos relacionados.

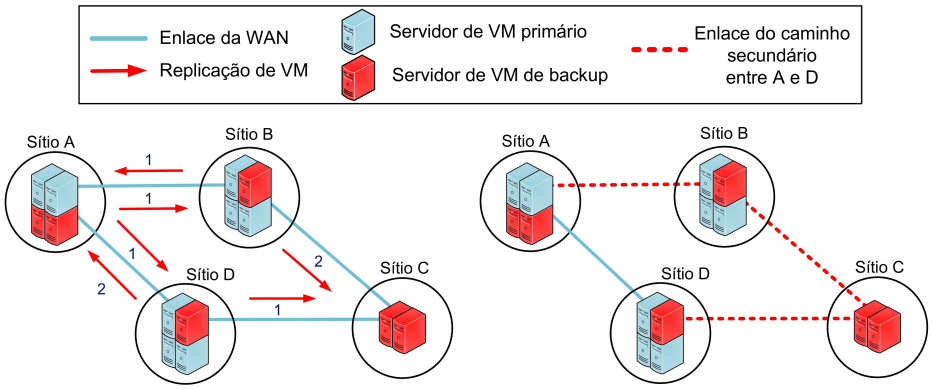
6.1 Modelagem e Decisões de Projeto

O problema proposto possui a função de distribuir servidores primários por uma WAN existente, que serão utilizados para hospedar VMs de clientes IaaS. Cada VM instalada será continuamente replicada para um servidor de backup, instalado em algum outro sítio da rede. Para tal, o problema também escolhe a distribuição de servidores de backup, que hospedarão as VMs dos servidores primários para o caso de falha causada por um desastre. Para cada servidor primário, escolhe-se um servidor de backup para receber as replicações das VMs. Esta seção detalha as decisões de projeto do DC, consideradas na formulação do problema de posicionamento, na Seção 6.2. Os detalhes do modelo completo do DC geodistribuído podem ser encontrados na Seção 2.2.

6.1.1 Replicação de VMs

O esquema de backup de VMs considerado neste trabalho é baseado na replicação contínua e confirmada, que permite garantir zero RPO (*Recovery Point Objective* - Objetivo do ponto de recuperação) em caso de desastres. Esse tipo de esquema já é comum para redes locais, sendo nativamente disponível em plataformas de virtualização como o Xen [69, 70]. Recentemente, esquemas de backup com zero RPO começaram a ser abordados na literatura para o caso de redes de longa distância [13, 71]. Um exemplo desse tipo de mecanismo é o SecondSite [13], utilizado como referência no restante deste capítulo. Para prover zero RPO, o SecondSite baseia-se em *checkpoints*. Um *checkpoint* é definido como o estado da VM (disco, memória, CPU) em determinado instante. Esse estado é enviado continuamente para o servidor de backup que, por sua vez, envia uma confirmação ao seu servidor primário a cada *checkpoint* recebido. Antes da confirmação de um *checkpoint*, os pacotes enviados da VM ao mundo externo são acumulados em fila, não sendo enviados aos usuários finais. O envio só é permitido após o recebimento de confirmação positiva do *checkpoint*. O usuário final só receberá a resposta de alguma operação que realizou após essa ter sido replicada no servidor de backup. Mais detalhes sobre o mecanismo de replicação do SecondSite podem ser encontrados na sua descrição, disponível em [13]. Note que o SecondSite possui exigências severas de banda passante e latência. O alto uso de banda passante se deve à contínua replicação de dados, maior quanto mais a VM muda de estado (p.ex., quanto mais o conteúdo da memória RAM é alterado). As exigências estritas de latência se devem à necessidade de confirmação dos *checkpoints* antes de avançar na execução das operações. Assim, quanto menor a latência, maior será a vazão com que uma determinada VM consegue realizar suas operações.

Ao detectar uma falha no servidor primário, o SecondSite ativa a VM no servidor de backup. A falha é detectada por cada nó a partir da ausência de dados de replicação entre os servidores. Isto é, o servidor de backup deduz que houve uma falha no servidor primário quando deixa de receber por um determinado tempo a replicação das VMs. Por sua vez, o servidor primário deduz que o backup falhou se não recebe confirmações de um *checkpoint* durante um determinado intervalo de tempo. Note que ambos os casos de falha podem ocorrer não só quando um dos servidores sofre uma falha, como também quando ocorre uma falha no enlace de replicação. Dessa forma, o servidor de backup pode deduzir de forma errônea que o servidor primário está operacional, ativando as VMs de backup. Isso pode causar o problema conhecido como *split brain*, no qual os dois servidores estão respondendo às requisições dos clientes, o que causa inconsistência entre a VM de backup e a primária. Para contornar esse problema, o SecondSite utiliza um terceiro tipo



(a) Posicionamento dos servidores. (b) Caminho secundário entre A e D.

Figura 6.1: Exemplo de DC geodistribuído com replicação contínua de VMs.

de servidor, denominado "servidor de quórum" . Ao suspeitar de uma falha, um servidor de backup ou primário contacta um servidor de quórum. Esse servidor de quórum, por ser consultado tanto pelo backup quando pelo primário, poderá informar o tipo da falha e os servidores envolvidos poderão realizar as operações apropriadas. Por exemplo, caso o enlace de replicação falhe, o servidor de backup é desligado. Mais detalhes sobre o esquema de detecção de falhas podem ser encontrados na descrição do SecondSite [13]. Obviamente, os servidores de quórum também devem ser posicionados na rede de forma a detectar as falhas de forma correta e rápida. Neste trabalho assume-se que esses servidores estão corretamente posicionados, conseguindo detectar todas as possíveis falhas. Esse tipo de posicionamento ainda não foi abordado na literatura, mas possui considerável semelhança com o posicionamento de controladores em SDNs [72].

6.1.2 Posicionamento dos Servidores

A Figura 6.1(a) exemplifica o cenário considerado. Cada círculo representa um sítio posicionado em uma localidade geográfica, e os sítios são interconectados por uma WAN. Note que as VMs hospedadas em cada servidor não estão mostradas. As setas indicam que um sítio envia continuamente réplicas das VMs para seus vizinhos, e os números ao lado de cada seta indicam quantos servidores primários do sítio de origem enviam backups para o sítio de destino. O Sítio D possui três servidores primários. O backup das VMs de dois de seus servidores é enviado para o Sítio A e o do terceiro servidor é enviado para o Sítio C. A figura mostra que um único sítio pode possuir servidores primários e de backup.

A Figura 6.1(a) mostra que um sítio usado para backup pode atender mais backups de servidores primários do que o número de servidores com essa função. Por exemplo, o Sítio C possui dois servidores de backup instalados. Por outro lado, dois servidores primários do Sítio B e um do Sítio D enviam seus backups para ele,

o que necessitaria de três servidores de backup no Sítio C. Entretanto, *considerando que o Sítio B e o Sítio D não estarão inalcançáveis ao mesmo tempo*, o Sítio B não precisará hospedar as VMs dos três servidores primários ao mesmo tempo. Como o serviço é baseado em VMs, durante a operação normal (isto é, sem falhas) os sítios que recebem os backups não precisam manter as VMs em operação, armazenando apenas os dados referentes ao disco, conteúdo da memória e outras informações das VMs enviadas pelo servidor primário, como visto na Seção 4.1.4. Assim, em operação normal, o servidor de backup necessita apenas de capacidade de armazenamento para as VMs, fornecida por servidores de armazenamento não mostrados na figura. A capacidade de memória e processamento, fornecida pelos servidores de VMs de backup, só será necessária após o desastre, quando os procedimentos de recuperação são executados e as VMs de backup passam a operar. Assim, é possível economizar o número de servidores de VMs de backup necessários, visto que um sítio precisa ter apenas o número de servidores de backup para suportar a falha de pior caso. No exemplo do Sítio C, o pior caso constitui na falha do Sítio B, exigindo que dois de seus servidores de backup se tornem operacionais. Essa característica é considerada no problema de otimização proposto, permitindo economia significativa do número de servidores de VM necessários. É importante notar que, independente do uso desse esquema de economia, o número de servidores de armazenamento necessários para backup é sempre o mesmo. Consequentemente, desconsidera-se o posicionamento desse tipo de servidor.

Para posicionar os servidores, um requisito básico é instalar um servidor primário e seu respectivo backup em sítios diferentes e que não falhem ao mesmo tempo. Para tal, utiliza-se a Matriz de Independência de Falhas (matriz I). Cada elemento corresponde a um valor binário I_{ij}, valendo 1 se o sítio i pode se tornar inalcançável ao mesmo tempo que o sítio j, e 0 senão. Um sítio é considerado inalcançável se, após uma falha, não possuir nenhum caminho para um *gateway* da rede ou se o próprio falhar, como definido no Capítulo 5. A matriz I é construída a partir do modelo de falhas. Neste capítulo considera-se o modelo de SRGs utilizado no Capítulo 5. Assim, utilizando a noção de SRGs, o valor de I_{ij} é 1 se os sítios i e j possuírem algum SRG em comum, e 0 caso contrário.

6.1.3 Enlace de Replicação e Caminho Secundário

Devido à severa restrição de latência imposta pela replicação de VMs, no problema formulado um determinado servidor realizará replicações de backup apenas aos vizinhos de um salto de seu sítio. Dessa forma, evita-se o aumento na latência causado por eventuais atrasos de processamento e de transmissão em roteadores intermediários. O enlace entre um sítio primário e seu respectivo backup, chamado

de enlace de replicação, é utilizado para transportar as réplicas das VMs. Entretanto, caso esse enlace falhe e os dois sítios não consigam se comunicar, o processo de replicação é interrompido e as VMs do servidor primário começam a executar no modo desprotegido [13]. Nesse modo as VMs continuam operacionais, porém não realizam replicações devido à ausência de comunicação com os servidores de backup. Como em serviços com zero RPO o tempo que a VM se encontra em modo desprotegido deve ser reduzido ao máximo, neste trabalho utiliza-se caminhos secundários entre o sítio primário e seu backup. Assim, quando for detectada uma queda no enlace de replicação, o servidor primário enviará suas replicações pelo caminho secundário. Obviamente, o caminho secundário escolhido não pode conter o enlace de replicação. A Figura 6.1(b) mostra o caminho escolhido para dar continuidade ao envio de replicações entre os Sítios A e D, quando o enlace entre eles falha. Mais adiante serão analisados os compromissos de se utilizar esse caminho adicional, bem como a qualidade desse tipo de caminho em termos de latência.

6.2 Formulação do Problema de Otimização do Posicionamento de Servidores

Neste trabalho maximiza-se o número de servidores primários cobertos pelo serviço de backup contínuo e, ao mesmo tempo, reduz-se a quantidade de servidores de backup adicionais necessários para abrigar as VMs após um desastre. O problema de otimização considera como parâmetros a latência e a capacidade (em Gbits/s) dos enlaces, a Matriz de Independência de Falhas, além da topologia da rede para cálculo dos caminhos secundários. A saída do problema fornece a quantidade de servidores primários e de backup a ser instalada em cada sítio, assim como o caminho secundário entre cada sítio primário e seu sítio de backup. O posicionamento proposto é realizado em duas etapas. Na primeira etapa calcula-se o caminho secundário entre cada par de sítios da rede. Na segunda etapa executa-se o problema de otimização do posicionamento de servidores físicos.

Na primeira etapa, modela-se no NetworkX a topologia WAN como um grafo, no qual os nós são os sítios e cada aresta um enlace entre os sítios, com o peso correspondente à distância geográfica entre eles. Para cada par de nós vizinhos, retira-se do grafo o enlace entre os dois sítios do par e calcula-se o menor caminho entre esses sítios utilizando o algoritmo de Dijkstra. Nessa estratégia serão escolhidos os caminhos secundários com a menor latência possível, independentemente do problema de otimização subsequente. Optou-se por adotá-la de forma a forçar o menor valor de latência possível, devido às restritas exigências em relação a essa métrica. Vale lembrar que, ao fim do problema de otimização, caminhos secundários entre sítios que

Tabela 6.1: Notações utilizadas no problema.

Notação	Descrição	Tipo
$\mathcal{D}$	Sítios candidatos	Conjunto
I_{ij}	Valor binário indicando se o sítio i pode se tornar inalcançável ao mesmo tempo que o sítio j	Parâmetro
Δ_{ij}	Atraso de propagação (latência) do enlace entre os sítios i e j	Parâmetro
W_{ij}	Capacidade do enlace entre os sítios i e j	Parâmetro
s_{ij}^{km}	Valor binário indicando se o enlace entre i e j pertence ao caminho secundário entre k e m	Parâmetro
α	Fração máxima da capacidade dos enlaces a ser utilizada pela replicação	Parâmetro
γ	Parâmetro binário indicando se existirá um caminho secundário entre sítios de backup e primários	Parâmetro
B	Consumo de banda, em Mbits/s, gerado pela transferência contínua do backup de um servidor primário	Parâmetro
L_{worst}	Máxima latência permitida entre um servidor primário e seu backup	Parâmetro
U_{max}	Máximo número de sítios utilizados	Parâmetro
x_i	Número de servidores primários na localização i	Variável
b_i	Número de servidores de backup na localização i	Variável
u_i	Valor binário indicando se o sítio i está ativo ($(x_i + b_i) > 0$)	Variável
c_{ij}	Número de backups do sítio i no sítio j	Variável
e_{ij}	Valor binário indicando se o sítio i possui backups no sítio j ($c_{ij} > 0$)	Variável
y_{kij}	Valor binário indicando se o sítio k abriga backups dos sítios i **e** j ($e_{ik} = 1$ e $e_{jk} = 1$)	Variável

não replicam backup entre si não são configurados. O resultado dessa etapa consiste nos parâmetros s_{ij}^{km}, que definem os enlaces contidos no caminho secundário entre cada par de sítios k e m. Ou seja, para um enlace entre i e j, tem-se $s_{ij}^{km} = 1$ se esse enlace está no caminho secundário entre k e m, ou $s_{ij}^{km} = 0$ senão.

Na segunda etapa, utiliza-se a ferramenta IBM ILOG CPLEX 12.5.1 para executar o problema ILP (ILP - *Integer Linear Programming*)formulado a seguir. Esse problema escolhe o posicionamento de cada servidor primário e seu respectivo backup, considerando os objetivos da otimização e restrições mencionados anteriormente. A Tabela 6.1 lista as notações utilizadas e seus tipos. Notações do tipo conjunto e parâmetro se referem aos dados do problema, enquanto as variáveis são ajustadas pelo algoritmo de otimização. A formulação ILP é a seguinte:

$$\text{maximizar} \quad \sum_{i \in \mathcal{D}} (x_i - b_i) \tag{6.1}$$

$$\text{sujeito a} \quad c_{ij} \cdot I_{ij} = 0 \quad \forall i, j \in \mathcal{D}. \tag{6.2}$$

$$\sum_{j \in \mathcal{D}} c_{ij} = x_i \quad \forall i \in \mathcal{D}. \tag{6.3}$$

$$M \cdot e_{ij} - c_{ij} \geq 0 \quad \forall i, j \in \mathcal{D}. \tag{6.4}$$

$$e_{ij} \leq c_{ij} \quad \forall i, j \in \mathcal{D}. \tag{6.5}$$

$$y_{kij} \geq e_{ik} + e_{jk} - 1 \quad \forall k, i, j \in \mathcal{D},\ i < j. \tag{6.6}$$

$$y_{kij} \leq e_{ik} \quad \forall k, i, j \in \mathcal{D},\ i < j. \tag{6.7}$$

$$y_{kij} \leq e_{jk} \quad \forall k, i, j \in \mathcal{D},\ i < j. \tag{6.8}$$

$$\sum_{i,j \in \mathcal{D},\ i<j} (I_{ij} \cdot y_{kij}) = 0 \quad \forall k \in \mathcal{D}. \tag{6.9}$$

$$b_j - c_{ij} \geq 0 \quad \forall i, j \in \mathcal{D}. \tag{6.10}$$

$$B \cdot c_{ij} \leq r_{ij} \quad \forall i, j \in \mathcal{D}. \tag{6.11}$$

$$r_{ij} \leq \alpha \cdot W_{ij} - \gamma \cdot B \cdot c_{km} \cdot s_{ij}^{km} \quad \forall i, j \in \mathcal{D} \quad \forall k, m \in \mathcal{D}. \tag{6.12}$$

$$e_{ij} \cdot \Delta_{ij} \leq L_{worst} \quad \forall i, j \in \mathcal{D}. \tag{6.13}$$

$$M \cdot u_i - (x_i + b_i) \geq 0 \quad \forall i \in \mathcal{D}. \tag{6.14}$$

$$u_i \leq (x_i + b_i) \quad \forall i \in \mathcal{D}. \tag{6.15}$$

$$\sum_{i \in \mathcal{D}} u_i \leq U_{max}. \tag{6.16}$$

$$x_i \geq 0, \quad b_i \geq 0,\ \forall\ i \in \mathcal{D}; c_{ij} \geq 0\ \forall\ i, j \in \mathcal{D}. \tag{6.17}$$

$$x_i \in \mathbb{Z}, \quad b_i \in \mathbb{Z}\ \forall\ i \in \mathcal{D}; \quad r_{ij} \in \mathbb{Z},\ c_{ij} \in \mathbb{Z}\ \forall\ i, j \in \mathcal{D}; u_i \in \{0, 1\},\ \forall\ i \in \mathcal{D}; \\ e_{ij} \in \{0, 1\}\ \forall\ i, j \in \mathcal{D}; y_{kij} \in \{0, 1\}\ \forall\ k, i, j \in \mathcal{D}. \tag{6.18}$$

O objetivo do problema, dado pela Equação (6.1), maximiza em cada sítio i o número de servidores primários (x_i) e minimiza o número de servidores utilizados para backup (b_i). Para cada servidor instalado em um sítio, o problema tentará reduzir em uma unidade o número de servidores de backup. Dessa forma, a função objetivo pode ser vista como a economia, em número de servidores, dos servidores de backup necessários para instalar o serviço. No pior caso, a função objetivo será zero, enquanto no melhor caso se aproximará do número de servidores primários instalados.

A Equação (6.2) força os servidores primários de um sítio i a só possuir backup em um sítio j (ou seja, $c_{ij} > 0$) se i e j não puderem se tornar inalcançáveis ao mesmo tempo ($I_{ij} = 0$). A Equação (6.3) define que o número de backups que um sítio i deverá espalhar pela rede seja igual ao número de servidores primários desse sítio.

As Equações (6.4) e (6.5) são utilizadas para calcular as variáveis binárias e_{ij}, que recebem valor 1 se $c_{ij} > 0$ e 0 caso contrário. O parâmetro M da Equação (6.4) é apenas um valor grande o suficiente para ser sempre maior ou igual a qualquer possível valor das variáveis c_{ij} dessa equação e das somas $x_i + u_i$ da Equação (6.14), mostrada adiante. Adota-se, de forma conservadora, $M = 1 \times 10^9$. As Equações (6.6), (6.7) e (6.8), utilizadas para calcular as variáveis binárias y_{kij}, são equivalentes à operação lógica E entre e_{ik} e e_{jk}.

A Equação (6.9) é responsável pela economia de servidores de backup. Essa equação garante que se dois sítios i e j podem se tornar inalcançáveis ao mesmo tempo (ou seja, $I_{ij} = 1$), logo eles não podem hospedar seus backups em um mesmo sítio k. Em outras palavras, se $I_{ij} = 1$, i e j não podem compartilhar servidores de backup. Note que a Equação (6.9) influencia as variáveis y_{kij} e, consequentemente, as variáveis e_{ij} e c_{ij}, que definem o posicionamento dos backups. Como o problema garante que sítios que podem se tornar inalcançáveis ao mesmo tempo não compartilham backups, o número de servidores de backup pode ser calculado utilizando a Equação (6.10), que é equivalente a fazer $b_i = \max_{j \in \mathcal{D}}(c_{ij})$. Isto é, o número de servidores de backup a ser instalado em um sítio i é dado pelo máximo número de backups que qualquer outro sítio da rede envia para ele. Voltando à Figura 6.1(a), o Sítio C recebe dois backups do Sítio B e um backup do Sítio D. Assim, o Sítio C possui dois servidores de backup para suportar a falha do Sítio B, que consiste no sítio que o envia o maior número de backups.

As Equações (6.11) e (6.12) tratam das restrições de banda passante. Na Equação (6.11), $B \cdot c_{ij}$ representa a quantidade de banda necessária para a replicação de c_{ij} servidores primários de i em j. Note que, para cada backup, uma quantidade de banda de B Mbits/s é utilizada continuamente no enlace entre os dois sítios. Essa equação especifica que o consumo de banda em um determinado enlace não pode ultrapassar o valor definido pela variável r_{ij}, dada pela Equação (6.12). Essa equação define a quantidade de banda disponível r_{ij} para realizar replicações entre os sítios i e j, considerando a banda reservada aos caminhos secundários que passam pelo enlace entre i e j. Nessa equação, $\alpha \cdot W_{ij}$ representa a quantidade de banda do enlace alocada para todo o serviço de IaaS com backup contínuo. O parâmetro W_{ij} representa a capacidade do enlace, zero caso os sítios i e j não possuam um enlace entre si. Dessa forma, apenas sítios vizinhos de um salto podem replicar backups entre si. O termo $\gamma \cdot B \cdot c_{km} \cdot s_{ij}^{km}$ é a quantidade de banda utilizada por um determinado caminho secundário entre dois sítios k e m, que passa pelo enlace entre i e j. Note que, nesse problema, a Equação (6.12) é equivalente a fazer $r_{ij} = \min_{k,m \in \mathcal{D}}(\alpha W_{ij} - B \cdot c_{km} \cdot s_{ij}^{km})$. Isto é, de todos os caminhos secundários que passam por um determinado enlace, contabiliza-se em r_{ij} apenas aquele que consome a maior quantidade de banda. Isso é possível assumindo que nenhum enlace da rede falhe ao mesmo tempo e, dessa forma, apenas um caminho secundário poderá estar ativo em todo o DC. Assim, em um determinado enlace, é necessário apenas reservar banda necessária para o pior caso de caminho secundário. Em outras palavras, os caminhos secundários são provisionados no esquema de proteção compartilhada [68]. O parâmetro γ da Equação (6.12) é utilizado para desabilitar a utilização de caminhos secundários, liberando recursos do enlace para replicações. Dessa forma, se $\gamma = 0$ tem-se simplesmente $r_{ij} = \alpha \cdot W_{ij}$.

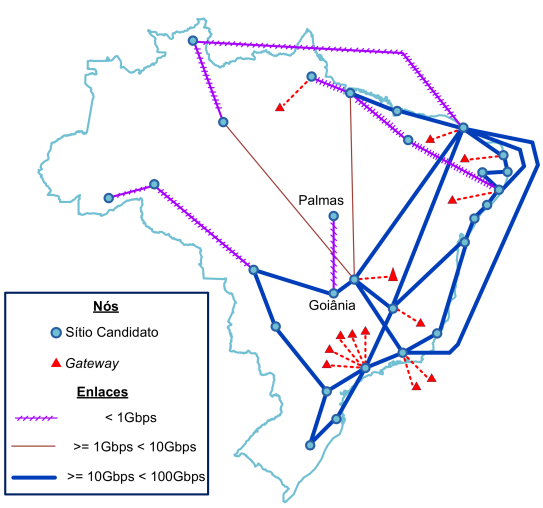
A Equação (6.13) define os requisitos de latência da replicação contínua. Um sítio i só poderá replicar backups para o sítio j (ou seja, $e_{ij} = 1$) se o enlace entre eles possuir latência Δ_{ij} inferior ou igual à máxima latência permitida (L_{worst}). As Equações (6.14), (6.15) e (6.16) limitam o número de sítios escolhidos para instalar servidores primários ou de backup, com base no número máximo de sítios, U_{max}. Finalmente, as Equações (6.17) e (6.18) definem, respectivamente, os limitantes inferiores e o domínio de cada variável.

Alguns requisitos comuns em serviços IaaS não são considerados no problema, como a proximidade dos servidores primários com os usuários finais. Desconsideram-se esses requisitos neste trabalho para permitir uma melhor análise de um serviço com zero RPO, cujos principais requisitos são a independência das falhas entre servidores operacionais e seus respectivos backups, além da baixa latência de replicação entre eles. Como visto anteriormente, a baixa latência é importante, pois influencia diretamente o tempo de resposta das aplicações executando nas VMs dos servidores primários.

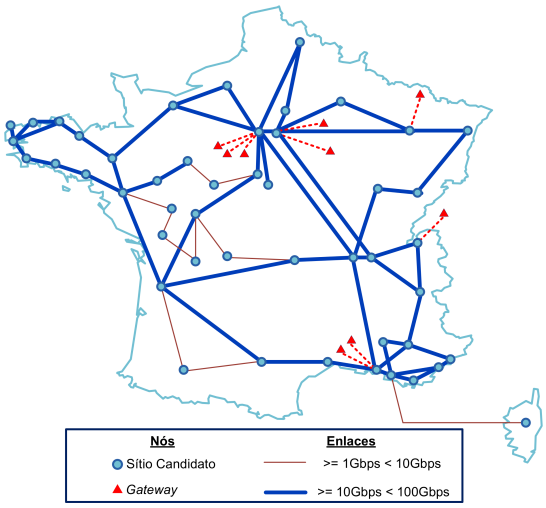
6.3 Avaliação

O problema de otimização formulado na Seção 6.2 é utilizado neste trabalho para posicionar servidores em WANs reais de RENs. Como visto no Capítulo 5, essas redes são formadas por PoPs, que são os possíveis sítios do DC no contexto deste trabalho. Assim, são adotadas as topologias da RNP (Figura 6.2(a)) no Brasil, RENATER (Figura 6.2(b)) na França, e GEANT (Figura 6.2(c)) na Europa. Cada figura das redes utilizadas mostra seus sítios e *gateways* de acesso à Internet.

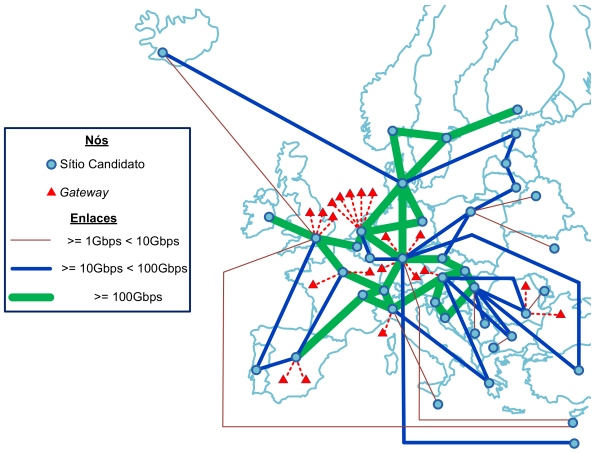
A partir das topologias mencionadas, calcula-se os parâmetros de entrada do problema de otimização. O valor de latência Δ_{ij} de cada enlace é dado pelo atraso de propagação entre os sítios i e j. Considera-se que a rede é bem provisionada e, assim, os atrasos de fila e de transmissão são desprezíveis. O atraso de propagação é calculado da mesma forma do Capítulo 5, como detalhado na Seção 5.3. A Matriz de Independência de Falhas (matriz I) é calculada a partir do modelo de SRGs de falha única, adotado também no Capítulo 5. Nesse modelo, apenas um enlace ou sítio da rede falha por vez. Note que, apesar do modelo de falha única, dois sítios podem se tornar inalcançáveis ao mesmo tempo. Por exemplo, na rede da Figura 6.2(a), se o sítio de Goiânia falhar, o sítio de Palmas também ficará inalcançável. A capacidade W_{ij} de cada enlace é mostrada na Figura 6.2. Esses valores foram obtidos no *website* oficial de cada uma das redes. A rede é modelada por um grafo direcionado, ou seja, a capacidade mostrada na figura corresponde à banda passante disponível em cada sentido do enlace entre dois sítios. Apesar de o grafo ser direcionado, considera-se no modelo de falhas que a ruptura de um determinado enlace representa a falha nos



(a) RNP, 28 sítios conectados por 38 enlaces.



(b) RENATER , 48 sítios conectados por 67 enlaces.



(c) GEANT, 41 sítios conectados por 58 enlaces.

Figura 6.2: Topologias de redes de ensino e pesquisa consideradas na avaliação.

dois sentidos desse enlace.

O consumo de banda gerado pela replicação contínua de cada servidor operacional, dado pelo parâmetro B, é fixado em 240 Mbits/s. Esse valor foi retirado do trabalho no qual o SecondSite foi proposto [13], e corresponde aproximadamente ao consumo de banda ao replicar um servidor primário que hospeda quatro VMs, sendo duas executando um *benchmark* de servidor web e duas executando um *benchmark* de banco de dados. O valor de B escolhido neste trabalho é usado apenas como referência, podendo variar consideravelmente dependendo da aplicação em execução nas VMs. Assim, no projeto de DCs, B deverá ser especificado em função de um SLA, no qual o provedor garantirá uma certa banda dedicada ao backup contínuo. Além disso, cada conjunto de servidores primários pode possuir um requisito diferente de banda de replicação, o que tornaria necessária a utilização de diferentes valores de B. O problema proposto pode ser facilmente modificado para considerar diferentes valores de B. Entretanto, optou-se por utilizar apenas um único valor de

B para simplificar a análise. Outro parâmetro que foi fixado para todas as avaliações desta seção é o máximo número de sítios utilizados, dado por U_{max}. Nesta análise escolheu-se um valor alto para esse parâmetro, para não limitar o número de sítios utilizados. Finalmente, salvo indicação contrária, o parâmetro γ é fixado em 1 na análise a seguir. Isso significa que o problema considera a utilização de caminhos secundários.

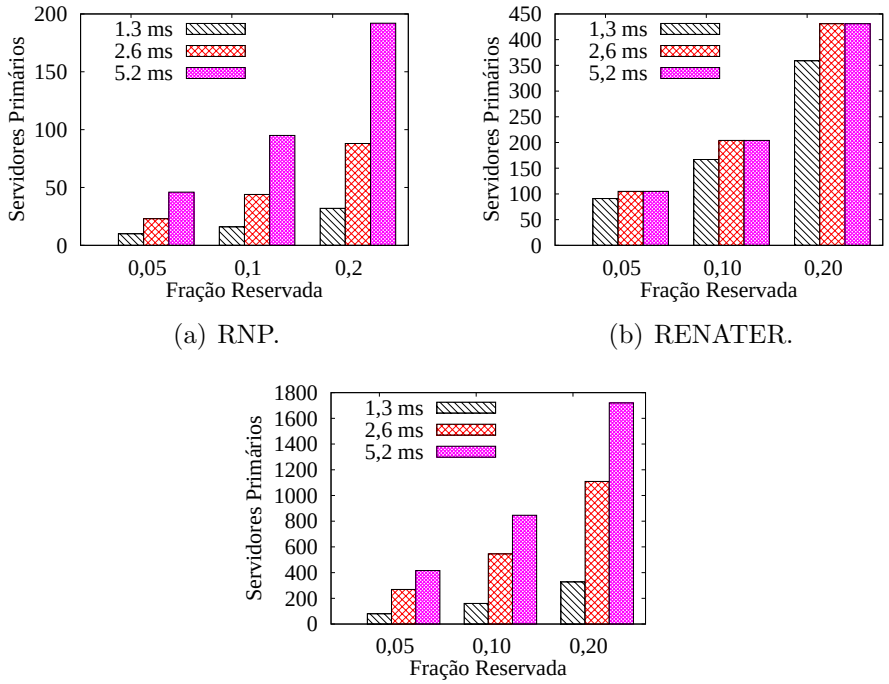
6.3.1 Capacidade do Serviço e Economia

A partir do problema proposto e dos parâmetros escolhidos anteriormente, analisa-se a capacidade do serviço de IaaS resiliente nas três redes consideradas. A capacidade do serviço, medida em número de servidores primários suportados, é analisada para diferentes valores de banda disponível (α) e diferentes valores de latência máxima tolerada (L_{worst}). A Figura 6.3 apresenta os resultados para cada rede, na qual cada gráfico agrupa no eixo X diferentes valores de α e cada barra representa um L_{worst} diferente. O L_{worst} mais baixo (isto é, 1,3 ms) foi escolhido com base nos experimentos realizados no artigo do SecondSite [13], nos quais replica-se um servidor em um enlace de 260 km entre as cidades canadenses de Vancouver e Kamloop. Essa distância acarreta uma latência de 1,3 ms, considerando a velocidade de propagação utilizada neste trabalho. Os demais valores utilizados são relaxamentos dos requisitos de latência, representando o dobro e o quádruplo desse valor de referência. Os resultados da Figura 6.3 mostram, como esperado, que quanto maior a banda reservada e quanto mais relaxado o requisito de latência, maior o número de servidores primários suportados. Note que na rede RENATER o número de servidores instalados não se altera ao relaxar o requisito de latência de 2,6 ms para 5,2 ms. Isso ocorre pois essa rede abrange a França metropolitana, uma área significativamente menor que a área coberta pelas outras redes. Assim, a maioria de seus enlaces já atende aos requisitos de latência quando $L_{worst} = 2,6\text{ms}$.

A Figura 6.4 mostra o resultado da economia de servidores de backup alcançada pela otimização do posicionamento. A economia é quantificada pela métrica Aproveitamento de Servidores (AS), definida como a relação entre o número de servidores de backup que não foram necessários instalar e o número total de servidores primários, dada por:

$$\text{AS} = \frac{\sum_{i \in \mathcal{D}}(x_i - b_i)}{\sum_{i \in \mathcal{D}}(x_i)} = 1 - \frac{\sum_{i \in \mathcal{D}}(b_i)}{\sum_{i \in \mathcal{D}}(x_i)} \tag{6.19}$$

Assim, AS varia no intervalo $[0, 1[$, e quanto maior seu valor, maior a economia. A Figura 6.4 mostra que, para todas as redes e todos os valores de α e L_{worst} avaliados, o aproveitamento é igual ou maior que 40%. Isso mostra que o posicionamento



(a) RNP.

(b) RENATER.

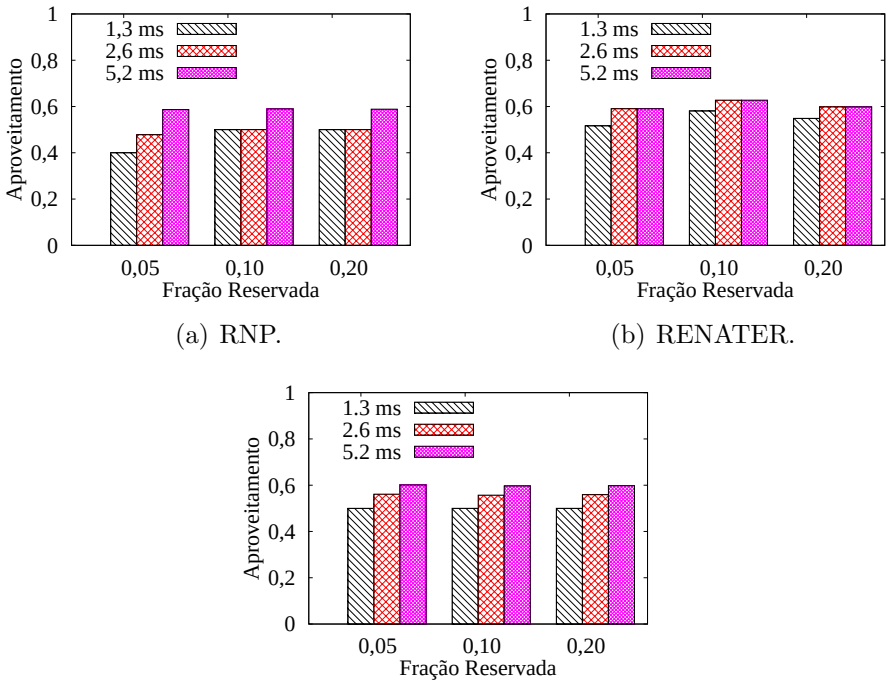
(c) GEANT.

Figura 6.3: Número de Servidores Primários.

proposto possibilita uma economia significativa de servidores de backup nas WANs consideradas. Além disso, os resultados mostram que relaxar o requisito de latência pode melhorar a economia de servidores, visto que mais opções de posicionamento estarão disponíveis.

6.3.2 Caminhos Secundários

Como já visto, o posicionamento deste trabalho restringe o valor máximo de latência no enlace entre dois sítios que replicam backups entre si. Entretanto, o problema não restringe esse valor para os caminhos secundários. Isto significa que, caso o enlace de replicação entre dois sítios falhe, esses realizarão suas replicações entre si utilizando um caminho que poderá não atender aos requisitos de latência definidos por L_{worst}. Dessa forma, o tempo de resposta das aplicações que as VMs executam poderá aumentar. A Figura 6.5 mostra características dos caminhos secundários, considerando apenas os caminhos entre sítios que replicam backups entre si. Os resultados são apresentados para $\alpha = 0,05$, mas os valores alcançados para os outros valores de α considerados são muito próximos aos da Figura 6.5. As Figuras 6.5(a), 6.5(b) e 6.5(c) apresentam a CDF da latência dos caminhos secundários para cada uma das redes utilizadas. Note que em todas as redes existem diversos caminhos secundários com valores de latência bem mais altos do que os valores L_{worst} especificados. Na RNP, alguns caminhos possuem valores de latência próximos a



(a) RNP.

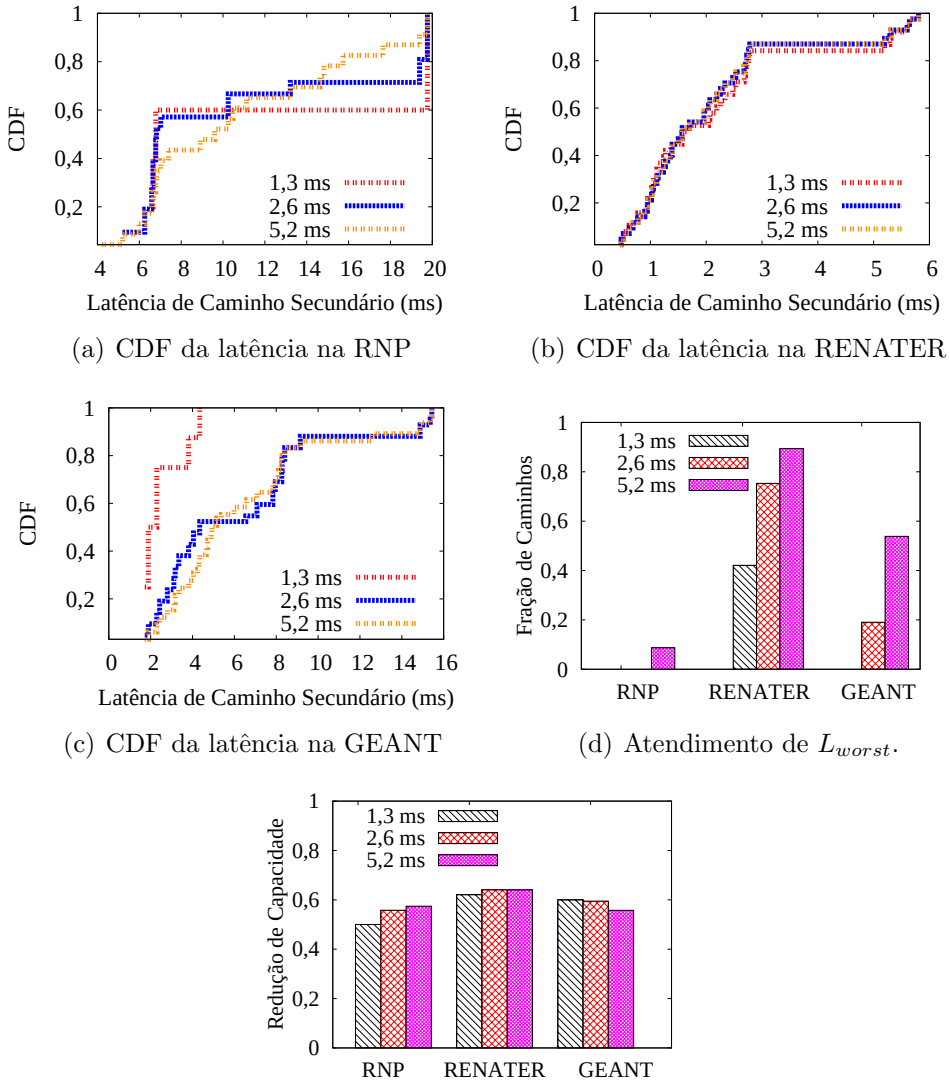
(b) RENATER.

(c) GEANT.

Figura 6.4: Economia de servidores físicos.

20 ms. Para uma melhor visualização dessa característica, a Figura 6.5(d) apresenta a fração de caminhos secundários que atendem ao requisito de latência L_{worst}. O eixo X agrupa os resultados obtidos com diferentes valores de L_{worst} para cada rede analisada. Note que o requisito mais estrito, de 1,3 ms, é atendido apenas por 40% dos caminhos na RENATER e por nenhum caminho da RNP e GEANT. Esse melhor desempenho da RENATER ocorre pois essa rede abrange uma área significativamente menor que as outras, como mencionado anteriormente. Para o requisito mais relaxado, de 5,6 ms, a RENATER é capaz de atender ao requisito L_{worst} em aproximadamente 90% dos caminhos. Apesar desse resultado favorável, as demais redes, e mesmo a própria RENATER para requisitos mais estritos, apresentam caminhos secundários com altos valores de latência. Isso mostra que apenas o posicionamento de servidores não é suficiente para garantir baixos valores de latência nos caminhos secundários, e que a topologia da WAN deve ser reprojetada para oferecer tal serviço. Esse projeto da WAN é importante, sobretudo em redes que abrangem grandes áreas geográficas. Este trabalho foca no posicionamento de servidores, considerando que a WAN já está instalada. O alinhamento do posicionamento de servidores ao projeto da WAN, recomendado no Capítulo 4, é um tema de trabalho futuro.

Como os possíveis caminhos secundários da rede podem apresentar alta latência, o projetista do DC pode escolher não configurar esses caminhos, fazendo $\gamma = 0$ no problema formulado na Seção 6.2. Dessa forma, mais banda passante estará



(a) CDF da latência na RNP

(b) CDF da latência na RENATER

(c) CDF da latência na GEANT

(d) Atendimento de L_{worst}.

(e) Sobrecarga.

Figura 6.5: Características dos caminhos secundários ($\alpha = 0,05$).

disponível para a instalação de servidores primários e seus respectivos backups. Por outro lado, se houver falha do enlace entre dois sítios que replicam entre si, os servidores primários correspondentes deverão ser pausados para evitar a execução de operações não replicadas. Essa estratégia, entretanto, diminui a disponibilidade dos servidores primários (isto é, a fração de tempo que permanecem operacionais). Outra estratégia é deixar o servidor primário em modo desprotegido (isto é, operacional, mas sem realizar backup), afetando os valores de RPO em prol da disponibilidade.

Para analisar o quanto a utilização de caminhos secundários reduz a capacidade de instalação de servidores primários, executou-se o problema com os mesmos parâmetros das seções anteriores, mas fazendo $\gamma = 0$. Denotando por S e S', respectivamente, o número de servidores primários suportados em uma rede quando utilizam-se caminhos secundários ($\gamma = 1$) e quando esses não são utilizados ($\gamma = 0$),

calcula-se a redução da capacidade pela expressão $1 - \frac{S}{S'}$. A Figura 6.5(e) mostra, para as diferentes redes e requisitos de latência, o valor da redução quando a fração máxima da banda utilizada nos enlaces (isto é, α) é de 0,05. Os resultados mostram que os caminhos secundários reduzem significativamente a capacidade do serviço, sendo a redução superior a 50% para todos os resultados da Figura 6.5(e). Os resultados para os outros valores de α, omitidos por questões de concisão, apresentam o mesmo comportamento, sendo sempre superiores a 50%. Assim, essa análise mostra que o projetista do DC pode aumentar significativamente a capacidade do serviço se escolher não utilizar caminhos secundários nas topologias consideradas.

6.4 Trabalhos Relacionados

Em relação ao posicionamento de servidores primários e de backup, os trabalhos mencionados anteriormente [17–19, 54, 67] consideram uma distribuição tradicional do DC, como a utilizada para redes de distribuição de conteúdo (CDN - *Content Delivery Networks*) [73], e assumem que cada serviço é conhecido no momento da construção do centro de dados. A replicação de serviços através de uma infraestrutura geodistribuída é alcançada nesses trabalhos a partir do princípio *anycast*, no qual qualquer nó que execute o serviço desejado pode responder às requisições, fazendo com que tanto os servidores primários quanto seus backups encontrem-se ativos ao mesmo tempo. Entretanto, esses trabalhos não consideram a sincronização entre as réplicas do serviço, desconsiderando requisitos de RPO. Além disso, ao considerar que os backups encontram-se ativos, não economizam o número de servidores de backup.

Yao *et al.* [74] propõem um problema de otimização para escolher os sítios de backup em um DC geodistribuído, além de agendar os períodos que esses sítios receberão as cópias. Diferente do esquema de backup contínuo considerado nesta tese, Yao *et al.* define janelas de backup. Essas janelas são intervalos pré-definidos nos quais as cópias são enviadas entre os sítios. Assim, o serviço considerado não aborda aplicações com zero RPO. O objetivo do problema proposto é minimizar o número de intervalos de tempo ocupados pelas janelas de backup ou, em outras palavras, a capacidade da rede consumida pelos backups. Como os backups não são realizados continuamente e não possuem com confirmação, ignora-se nesse trabalho o aumento na latência causado pelo atraso de propagação entre os sítios.

Bianco *et al.* [75] propõem um problema de posicionamento de backups similar ao desta tese, que consiste em alocar recursos primários e de backup para as VMs de um DC geodistribuído em uma rede já existente. Para tal, o problema de otimização escolhe o disco primário da VM e seu respectivo disco de backup, de forma que o disco de backup e o primário não estejam em um mesmo sítio. Esse trabalho propõe três

diferentes problemas de posicionamento. O primeiro minimiza o número de saltos entre um sítio que hospeda um disco primário e o sítio que hospeda seu respectivo backup, considerando todos os sítios da rede. Essa minimização é uma tentativa de minimizar a latência entre os sítios, já que a sincronização de discos é sensível à latência. Note que esse trabalho não considera o atraso de propagação entre os sítios, como realizado nesta tese, minimizando apenas o número de saltos. Essa abordagem pode não ser adequada para DCs abrangendo grandes regiões geográficas, nas quais o número de saltos não reflete necessariamente a latência entre sítios. O segundo problema proposto minimiza o número de backups hospedados em um sítio através de balanceamento de carga. Para tal, o problema tenta distribuir uniformemente os backups entre os sítios. De acordo com Bianco *et al.*, após uma falha no sítio primário, o processo de migração da VM para o sítio de backup é uma tarefa que demanda alto uso de CPU. Assim, minimizar o número de backups de um sítio reduz a capacidade de CPU necessária em cada sítio no processo de recuperação de desastres. Entretanto, a capacidade global da infraestrutura (isto é, considerando todos os sítios) de CPU necessária para o processo de recuperação será a mesma, independente dessa distribuição de carga. Note que a abordagem desse posicionamento é contrária à estratégia de economia de servidores de backup proposta nesta tese, visto que essa estratégia procura agrupar ao máximo os backups em um mesmo sítio. Por fim, o terceiro problema proposto é a abordagem híbrida, considerando tanto o número de saltos entre os sítios quanto o balanceamento de carga dos backups.

A contribuição desta tese em relação ao posicionamento de backups difere da literatura pois considera o backup contínuo, que acarreta requisitos estritos de latência e banda passante, além de prover economia de servidores de backup. Para tal, foca-se no modelo IaaS, diferente das CDNs tradicionais.

Uma área correlata ao posicionamento de servidores em um DC é o mapeamento resiliente de redes virtuais em redes físicas, introduzido pela primeira vez em [33]. O mapeamento de redes virtuais consiste em escolher os nós e enlaces físicos que serão utilizados por nós e enlaces virtuais requisitados. O mapeamento resiliente escolhe, além dos recursos físicos primários, os recursos físicos que serão utilizados pelas redes virtuais caso um nó ou enlace primário falhe. Em [33] considera-se apenas falhas de enlaces e alocação para recuperação rápida de falhas, na qual os recursos de backup são reservados *a priori*. Já em [76] propõe-se um problema de mapeamento resiliente considerando-se apenas falhas dos nós. Esse trabalho considera o compartilhamento de recursos de backup, reduzindo a quantidade de recursos necessária. Outra área relacionada a esta tese é o posicionamento de controladores em redes definidas por software (SDNs - *Software Defined Networks*). Em SDNs os elementos de comutação são gerenciados por controladores. Assim, diferentes trabalhos propõem esquemas

de posicionamento de um conjunto de controladores, de forma que os elementos de comutação consigam acesso a pelo menos um deles após falhas [72]. Por fim, outra área relacionada é o posicionamento resiliente de VMs. Bodík *et al.* [64] realizam um posicionamento resiliente de VMs considerando um único sítio no centro de dados e requisitos de alta disponibilidade. Entretanto, esse trabalho não considera centros de dados geodistribuídos nem a presença de backups e requisitos de QoR para desastres.

Capítulo 7

Conclusões e Trabalhos Futuros

Esta tese buscou salientar aspectos de rede de centros de dados (DCNs) que influenciam na resiliência a falhas. Esses aspectos consistem tanto na organização topológica da rede interna ao sítio, como também em características da rede entre os sítios de uma infraestrutura geodistribuída. Na rede intra-sítio o DC deve suportar pequenas falhas, como falhas em comutadores e rompimento de cabos de rede. Já no cenário inter-sítio, o DC deve ser projetado para suportar desastres ou outras falhas de larga escala, que podem destruir ou desconectar sítios inteiros. Dessa forma, primeiramente analisou-se a resiliência das principais novas topologias de rede intra-sítio. Em seguida, foram propostas linhas gerais para o projeto da rede inter-sítio de DCs resilientes a desastres. A partir dessas linhas gerais, identificaram-se direções de pesquisa e dois problemas de otimização foram formulados e resolvidos. O primeiro busca analisar o compromisso entre latência e resiliência de DCs geodistribuídos, enquanto o segundo posiciona servidores primários e de backup em uma WAN. De forma geral, os resultados dessa tese auxiliarão projetistas de DCs na construção de centros de dados resilientes. Assim, mostrou-se que é possível melhorar a resiliência do DC se a rede entre seus servidores for bem planejada, seja essa local ou de longa distância.

Na análise de resiliência das topologias intra-sítio, utilizaram-se novas arquiteturas de DCN propostas recentemente, considerando que seus diferentes elementos de rede são suscetíveis a falhas. Também utilizou-se uma topologia de DCN convencional, com três níveis de comutadores. Os resultados permitem concluir qual topologia possui melhor desempenho para um determinado cenário de falhas aleatórias. Observa-se que a topologia Three-layer, por possuir uma menor redundância de enlaces e comutadores em relação às novas topologias, apresenta o pior desempenho quando submetida a falhas nesses elementos. A Fat-tree, por sua vez, possui um núcleo altamente redundante mas uma borda vulnerável, o que reduz sua robustez a falhas. Quando há falha em uma dada porcentagem dos enlaces ou comutadores da rede, em média os servidores são desconectados da rede na mesma proporção. Por

exemplo, 40% de falhas aleatórias de enlace ou comutador representam, em média, uma perda de 40% dos servidores. Consequentemente, a Fat-tree mostra um desempenho substancialmente pior que a BCube e a DCell, que perdem no máximo 26% de seus servidores para uma alta porcentagem de falha de elementos (40%). Além disso, a Fat-tree apresenta um MTTF para falhas de enlace no mínimo 42 vezes menor que as outras topologias, e no mínimo $7,2$ vezes menor para falhas de comutadores. Por outro lado, a Fat-tree mantém seu comprimento médio de caminhos original, enquanto na BCube e DCell uma grande proporção de falhas pode aumentar o comprimento médio dos caminhos em 2 e 7 saltos, respectivamente. Apesar disso, o aumento do comprimento dos caminhos nas topologias centradas em servidor geralmente não é muito severo se comparado com a degradação do número de servidores conectados na Fat-tree. Sobre a BCube, é possível afirmar que seu desempenho em presença de falhas de enlace é superior a todas as topologias analisadas, mantendo no mínimo 84% de seus servidores quando 40% dos seus enlaces estão defeituosos, contra 74% de servidores na DCell. Isso é explicado pois, sendo uma rede centrada em servidores, a BCube utiliza interfaces redundantes nos seus servidores. Além disso, os servidores estão conectados diretamente apenas a comutadores. Como os comutadores possuem um grau (isto é, número de interfaces) mais alto que os servidores, a desconexão pela remoção de enlace é mais difícil na BCube que na DCell, visto que essa última possui servidores diretamente conectados entre si. Por fim, a DCell apresenta o melhor desempenho para falhas de comutadores, podendo apresentar um MTTF até 12 vezes maior que a BCube. Esse comportamento é explicado pela sua alta dependência em servidores para manter a rede conectada [29, 30].

Considerando ainda o cenário intra-sítio, mostrou-se que a melhora na resiliência pela adição de interfaces de servidores é limitada pelo máximo comprimento médio dos caminhos tolerado. Isso ocorre pois, mesmo sem falhas, aumentar o número de interfaces de servidor na BCube e na DCell aumenta o Comprimento Médio dos Caminhos Mais Curtos. Para as topologias BCube e DCell observou-se também que, apesar de utilizarem servidores para encaminhar pacotes, em média uma falha de servidor não leva à desconexão dos servidores restantes. A análise em topologias intra-sítio mostra que o corte mínimo é uma métrica apropriada para aproximar a confiabilidade para falhas de enlaces. Assim, foram propostas fórmulas fechadas para o MTTF das topologias consideradas. Para falhas de comutador, os resultados mostram que a utilização do corte mínimo não é adequada para algumas topologias.

Uma direção futura para a análise das topologias intra-sítio é considerar falhas correlacionadas (p.ex., falhas de um bastidor ou outras definições de SRGs), relaxando a consideração de independência entre as falhas. Além disso, uma direção interessante é construir um modelo de falhas no qual falhas de enlaces, comutadores e servidores coexistam. Entretanto, para construir tal modelo é necessário atri-

buir taxas de falha a cada tipo de equipamentos utilizado, perdendo a generalidade alcançada nesta tese.

As linhas gerais propostas para o cenário inter-sítio buscam projetar uma infraestrutura de DCN que suporte uma nuvem IaaS resiliente, a partir da redundância de seus componentes. Essas linhas gerais são divididas em fases de projeto, e permitem enumerar diversas direções de pesquisa. Em suma, essas direções consistem no posicionamento de nós em uma infraestrutura geodistribuída, tanto fisicamente (p.ex., sítios de um DC) como virtualmente (p.ex., *snapshots* de VMs), além do projeto da WAN para conectar esses nós [49]. A área de pesquisa motivada nesta tese permite que provedores IaaS (*Infrastructure as a Service* - Infraestrutura como serviço) ofereçam serviços mais sofisticados, garantindo a continuidade do serviço mesmo na ocorrência de eventos catastróficos. Além disso, um DC resiliente a desastres incentiva que mais empresas migrem suas infraestruturas de TI para uma nuvem IaaS. Uma direção futura para as linhas gerais de projeto propostas é adaptá-las aos arcabouços de ITSCM, como o estágio *Service Design* da ITIL e o padrão ISO/IEC 24762:2008.

Das direções de pesquisa levantadas para o cenário inter-sítio, focou-se nesta tese no posicionamento de servidores físicos em uma infraestrutura geodistribuída, considerando uma WAN já existente. Assim, primeiramente analisou-se o compromisso entre latência e resiliência em DC geodistribuídos. A resiliência nesse cenário é quantificada pela sobrevivência, definida como a menor fração dos bastidores que permanecem disponíveis após a falha de um único SRG (*Shared Risk Group* - Grupo de risco compartilhado), considerando todos os possíveis SRGs. Assim, a sobrevivência representa a redução da capacidade de oferta de serviços do DC quando ocorre falhas em seus componentes. As simulações realizadas em cenários diversos e realísticos mostram que quando a sobrevivência do DC já é muito alta, uma pequena melhora na resiliência acarreta um aumento significativo na latência. Esse comportamento é essencialmente devido à presença de caminhos muito longos para alguns poucos pares de sítios. Em níveis altos de resiliência, os resultados mostram que um aumento de 2% (de 0,94 para 0,96) na sobrevivência pode aumentar a latência em 46% (de 11,33 ms para 27,9 ms). Por outro lado, quando a exigência de resiliência permanece moderada, esta pode ser consideravelmente melhorada com um baixo acréscimo na latência. Considerando todas as redes analisadas, o máximo aumento na latência é de 0,7 ms quando a sobrevivência é melhorada de 0 para 0,5. Além disso, observa-se uma máxima latência de 3,6 ms quando a sobrevivência é 0,8. Os cenários atuais de IaaS apresentam geralmente centros de dados monolíticos com um único ou um número baixo de sítios, provendo baixos níveis de resiliência a desastres. Este estudo sugere que, considerando um cenário realístico de projeto de DCs em WANs, aumentar a resiliência a um nível moderado apresenta pouco ou nenhum

impacto no desempenho de um IaaS [60, 61]. Como direção futura a esse estudo, é possível estender a análise para considerar modelos mais sofisticados de falhas, modelando falhas de larga escala, típicas de casos de grandes catástrofes naturais. Esse tipo de falha pode afetar diversos sítios em uma grande área geográfica. Outra direção futura é estudar o compromisso da resiliência com outros fatores, como o custo para instalar o DC e a latência do ponto de vista dos usuários.

Além do compromisso entre latência e resiliência, no cenário inter-sítio foi proposto o posicionamento de servidores em DCs geodistribuídos, através da formulação de um problema de otimização com objetivo de suportar serviço IaaS com zero RPO (*Recovery Time Objective* - Objetivo do tempo de recuperação). Utilizando essa proposta, mostrou-se ser possível reduzir significativamente o número de servidores necessários, através do compartilhamento de servidores de backups. Os resultados para todas as redes consideradas mostram a economia de no mínimo 40% no número de servidores. Tal economia pode ser ainda maior se os requisitos de latência forem relaxados. Este trabalho também apresentou as características do DC geodistribuído se forem configurados caminhos secundários entre sítios que realizam réplicas entre si. Esses caminhos são utilizados caso o enlace entre dois sítios falhe. Os resultados mostram que os caminhos secundários podem apresentar alta latência, desrespeitando os requisitos do serviço de replicação. Por fim, mostra-se que a existência de caminhos secundários adiciona alta sobrecarga à rede, devido à quantidade necessária de banda reservada. Os resultados mostraram que, em todas as redes consideradas, seria possível adicionar pelo menos o dobro do número de servidores primários caso os caminhos secundários não fossem configurados. Como direção futura desta contribuição da tese é possível citar a proposta de algoritmos de projeto físicos de WANs (p.ex., a definição da localização de PoPs e da topologia entre eles) que atuem em conjunto com o posicionamento de servidores para, por exemplo, oferecer caminhos secundários com melhor qualidade. Outra direção promissora é considerar o posicionamento dos servidores que detectam as falhas, de forma a otimizar o tempo de recuperação das VMs e permitir uma detecção mais acurada.

Apêndice A

Cálculo da aproximação do MTTF

Neste apêndice obtém-se a Equação 3.6, resultante da combinação das Equações 3.4 e 3.5. Primeiramente, substitui-se o valor de confiabilidade $R(t)$ da Equação 3.4 pela estimativa de $R(t)$ da Equação 3.5, resultando em:

$$MTTF = \int_0^\infty R(t)\,dt \approx \int_0^\infty e^{-\frac{t^r_c}{E[\tau]^r}}\,dt. \tag{A.1}$$

Dessa forma, para encontrar o MTTF basta resolver a integral do último termo da Equação A.1, como realizado a seguir. Para a solução dessa integral realiza-se a seguinte substituição de variável:

$$t = x^{\frac{1}{r}} \Leftrightarrow dt = \frac{1}{r}x^{\left(\frac{1}{r}-1\right)}\,dx. \tag{A.2}$$

Repare que o intervalo de integração da Equação A.1 não é alterado pela substituição de variável, visto que $t = 0$ implica em $x = 0$ e $t \to \infty$ implica em $x \to \infty$. Assim, após a substituição de variável, a Equação A.1 pode ser reescrita como:

$$MTTF \approx \frac{1}{r}\int_0^\infty x^{\left(\frac{1}{r}-1\right)}e^{\frac{-xc}{E[\tau]^r}}\,dx. \tag{A.3}$$

Para resolver a integral da Equação A.3 utiliza-se a função gama definida como [77]:

$$\Gamma(z) = k^z\int_0^\infty x^{z-1}e^{-kx}\,dx, (\Re z > 0, \Re k > 0). \tag{A.4}$$

Para facilitar, rescreve-se a integral da Equação A.4 da seguinte forma:

$$\int_0^\infty x^{z-1}e^{-kx}\,dx = \frac{\Gamma(z)}{k^z}. \tag{A.5}$$

Fazendo $z = \frac{1}{r}$ e $k = \frac{c}{E[\tau]^r}$ na Equação A.5 e multiplicando ambos os lados da equação por $\frac{1}{r}$ tem-se:

$$\frac{1}{r}\int_0^\infty x^{\left(\frac{1}{r}-1\right)} e^{-\frac{xc}{E[\tau]^r}}\, dx = \frac{1}{r}\frac{\Gamma\left(\frac{1}{r}\right)}{\frac{c}{E[\tau]^r}^{\frac{1}{r}}} = \frac{E[\tau]}{r}\sqrt[r]{\frac{1}{c}}\Gamma\left(\frac{1}{r}\right). \tag{A.6}$$

Note que o termo mais à esquerda da Equação A.6 equivale à aproximação do MTTF dada pela Equação A.3. Assim, o MTTF pode ser escrito por:

$$MTTF \approx \frac{E[\tau]}{r}\sqrt[r]{\frac{1}{c}}\Gamma\left(\frac{1}{r}\right). \tag{A.7}$$

Apêndice B

Comparação entre as Equações de MTTF para Falhas de Enlace

Na BCube tem-se $MTTF_{bcube} \approx \frac{E[\tau]}{l+1} \sqrt[l+1]{\frac{1}{|S|}}\Gamma\left(\frac{1}{l+1}\right)$. Assim, mostra-se inicialmente que com uma nova configuração da BCube com $l' = l + 1$ (i.e., mais uma interface de servidor) é possível aumentar o MTTF. Para simplificar, considera-se que $|S|$ é igual para ambas as configurações l e l'. Apesar de isso não ser necessariamente verdade, devido a dependência de $|S|$ em relação à combinação de l e n, é possível ajustar n de forma a permitir um número próximo de servidores utilizando l e l', como visto nas configurações da Tabela 3.1. Inicialmente, é necessário provar que:

$$\frac{E[\tau]}{l'+1} \sqrt[l'+1]{\frac{1}{|S|}}\Gamma\left(\frac{1}{l'+1}\right) > \frac{E[\tau]}{l+1} \sqrt[l+1]{\frac{1}{|S|}}\Gamma\left(\frac{1}{l+1}\right). \tag{B.1}$$

Fazendo $l' = l+1$ e reorganizando os termos, tem-se os seguintes requisitos para a formulação acima ser verdadeira:

$$|S| > \left(\frac{l+2}{l+1}\frac{\Gamma\left(\frac{1}{l+1}\right)}{\Gamma\left(\frac{1}{l+2}\right)}\right)^{(l+1)(l+2)}. \tag{B.2}$$

O termo à direita da Equação B.2 é uma função decrescente de l na região considerada ($l \geq 1$). Assim, provar que a suposição da Equação B.2 é válida para $l = 1$ é suficiente para provar essa formulação para $l > 1$. Fazendo $l = 1$ na Equação B.2, tem-se $|S| > 0,955$, que é verdadeiro para qualquer DC.

Como uma DCell com $l > 1$ possui o mesmo MTTF que uma BCube com o mesmo l, o raciocínio acima é válido para essa topologia. Para a DCell2 ($l = 1$), a equação do MTTF é a mesma que da BCube2 ($l = 1$) exceto que a DCell2 possui o valor $\sqrt{\frac{1}{1,5|S|}}$ ao invés de $\sqrt{\frac{1}{|S|}}$. Consequentemente, o MTTF da BCube2 é maior que o da DCell2. Assim, é possível concluir que a DCell2 possui o MTTF mais baixo entre as topologias centradas em servidor. Assim, para mostrar que o MTTF da

Fat-tree é menor do que o de todas as topologias centradas em servidor, compara-se sua equação com a da DCell2. Assim, é necessário mostrar que:

$$\frac{E[\tau]}{|S|} < \frac{E[\tau]}{2}\sqrt{\frac{1}{1,5|S|}}\Gamma\left(\frac{1}{2}\right). \tag{B.3}$$

A solução para essa equação é $|S| > 1,909$, que é sempre verdadeiro considerando um DC real.

Referências Bibliográficas

[1] *Cisco Data Center Infrastructure 2.5 Design Guide*, Nov. 2007. www.cisco.com/application/pdf/en/us/guest/netsol/ns107/c649/ccmigration_09186a008073377d.pdf - Acessado em Novembro de 2014.

[2] AL-FARES, M., LOUKISSAS, A., VAHDAT, A. "A Scalable, Commodity Data Center Network Architecture". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 63–74, Seattle, EUA, Ago. 2008.

[3] GUO, C., LU, G., LI, D., et al. "BCube: A High Performance, Server-Centric Network Architecture for Modular Data Centers". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 63–74, Barcelona, Espanha, Ago. 2009.

[4] GUO, C., WU, H., TAN, K., et al. "DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 75–86, Seattle, EUA, Ago. 2008.

[5] GREENBERG, A., HAMILTON, J., MALTZ, D. A., et al. "The Cost of a Cloud: Research Problems in Data Center Networks", *SIGCOMM Computer Communnication Review*, v. 39, n. 1, pp. 68–73, 2009.

[6] BAUER, E., ADAMS, R. *Reliability and Availability of Cloud Computing*. Nova Jérsia, EUA, John Wiley & Sons, 2012.

[7] MELO, R., SANTOS, A., NOGUEIRA, M., et al. "Modelagem e Projeto de Redes sem Fio Heterogeñeas Resilientes e Sobreviventes". In: *Minicursos do XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 1–50, Brasília, Brasil, Maio 2013.

[8] *Amazon EC2 Service Level Agreement*. Amazon Web Services, Inc., Jun. 2013. http://aws.amazon.com/ec2-sla/ - Acessado em Novembro de 2014.

[9] GROVER, W. D. *Mesh-Based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*. 1a ed. Nova Jérsia, EUA, Prentice Hall - PTR, 2004.

[10] LAM, C., LIU, H., KOLEY, B., et al. "Fiber Optic Communication Technologies: What's Needed for Datacenter Network Operations", *Communications Magazine, IEEE*, v. 48, n. 7, pp. 32–39, 2010.

[11] ENDO, P. T., PALHARES, A. V. A., PEREIRA, N. N., et al. "Resource Allocation for Distributed Cloud: Concepts and Research Challenges", *IEEE Network*, v. 25, n. 4, pp. 42–46, 2011.

[12] DEVELDER, C., DE LEENHEER, M., DHOEDT, B., et al. "Optical Networks for Grid and Cloud Computing Applications", *Proceedings of the IEEE*, v. 100, n. 5, pp. 1149–1167, 2012.

[13] RAJAGOPALAN, S., CULLY, B., O'CONNOR, R., et al. "SecondSite: Disaster Tolerance As a Service". In: *Proceedings of the ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments (VEE)*, pp. 97–108, Londres, Reino Unido, Mar. 2012.

[14] POPA, L., RATNASAMY, S., IANNACCONE, G., et al. "A Cost Comparison of Datacenter Network Architectures". In: *Proceedings of the International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pp. 16:1–16:12, Filadélfia, EUA, Dez. 2010.

[15] LI, D., GUO, C., WU, H., et al. "Scalable and Cost-Effective Interconnection of Data-Center Servers Using Dual Server Ports", *IEEE/ACM Transactions on Networking*, v. 19, n. 1, pp. 102 –114, 2011.

[16] FERRAZ, L. H. G., MATTOS, D. M. F., DUARTE, O. C. M. B. "A Two-Phase Multipathing Scheme based on Genetic Algorithm for Data Center Networking". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 2311—2316, Austin, EUA, Dez. 2014.

[17] DEVELDER, C., BUYSSE, J., DE LEENHEER, M., et al. "Resilient Network Dimensioning for Optical Grid/Clouds Using Relocation". In: *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 6262–6267, Ottawa, Canadá, Jun. 2012.

[18] XIAO, J., WU, B., JIANG, X., et al. "Data Center Network Placement and Service Protection in All-optical Mesh Networks". In: *Proceedings of the International Conference on the Design of Reliable Communication Networks (DRCN)*, pp. 88–94, Budapeste, Hungria, Mar. 2013.

[19] HABIB, M. F., TORNATORE, M., DE LEENHEER, M., et al. "Design of Disaster-resilient Optical Datacenter Networks", *Journal of Lightwave Technology*, v. 30, n. 16, pp. 2563–2573, 2012.

[20] KACHRIS, C., TOMKOS, I. "A Survey on Optical Interconnects for Data Centers", *IEEE Communications Surveys Tutorials*, v. 14, n. 4, pp. 1021–1036, 2012.

[21] SINGLA, A., HONG, C., POPA, L., et al. "Jellyfish: Networking Data Centers, Randomly". In: *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San José, EUA, Abr. 2012.

[22] CURTIS, A., CARPENTER, T., ELSHEIKH, M., et al. "REWIRE: An Optimization-Based Framework for Unstructured Data Center Network Design". In: *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Orlando, EUA, Mar. 2012.

[23] RAICIU, C., BARRE, S., PLUNTKE, C., et al. "Improving Datacenter Performance and Robustness with Multipath TCP". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 350–361, Ontário, Canadá, Ago. 2011.

[24] MENG, X., PAPPAS, V., ZHANG, L. "Improving the Scalability of Data Center Networks with Traffic-Aware Virtual Machine Placement". In: *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–9, San Diego, EUA, Mar. 2010.

[25] CLOS, C. "A Study of Non-Blocking Switching Networks", *Bell System Technical Journal*, v. 32, n. 2, pp. 406–424, 1953.

[26] GREENBERG, A., HAMILTON, J. R., JAIN, N., et al. "VL2: A Scalable and Flexible Data Center Network". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 51–62, Barcelona, Espanha, Ago. 2009.

[27] GERTZBAKH, I., SHPUNGIN, Y. *Models of network reliability: analysis, combinatorics, and Monte Carlo*. Boca Raton, EUA, CRC Press, 2009.

[28] LIEW, S. C., LU, K. W. "A Framework for Characterizing Disaster-Based Network Survivability", *IEEE Journal on Selected Areas in Communications*, v. 12, n. 1, pp. 52–58, 1994.

[29] COUTO, R. S., CAMPISTA, M. E. M., COSTA, L. H. M. K. "Uma Avaliação da Robustez Intra Data Centers Baseada na Topologia da Rede". In: *Anais do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 610—623, Ouro Preto, Brasil, Maio 2012.

[30] COUTO, R. S., CAMPISTA, M. E. M., COSTA, L. H. M. K. "A Reliability Analysis of Datacenter Topologies". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1890—1895, Anaheim, EUA, Dez. 2012.

[31] HAGBERG, A., SWART, P., S CHULT, D. *Exploring Network Structure, Dynamics, and Function Using NetworkX*. Relatório técnico, Los Alamos National Laboratory, 2008.

[32] EGELAND, G., ENGELSTAD, P. "The availability and Reliability of Wireless Multi-Hop Networks with Stochastic Link Failures", *IEEE Journal on Selected Areas in Communications*, v. 27, n. 7, pp. 1132–1146, 2009.

[33] RAHMAN, M. R., AIB, I., BOUTABA, R. "Survivable Virtual Network Embedding". In: Crovella, M., Feeney, L., Rubenstein, D., et al. (Eds.), *NETWORKING 2010*, v. 6091, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 40–52, 2010.

[34] BARLOW, R., PROSCHAN, F. *Statistical theory of reliability and life testing: probability models*. Nova Iorque, EUA, Holt, Rinehart and Winston, 1975.

[35] GILL, P., JAIN, N., NAGAPPAN, N. "Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 350–361, Ago. 2011.

[36] HEEGAARD, P. E., TRIVEDI, K. S. "Network survivability modeling", *Computer Networks*, v. 53, n. 8, pp. 1215–1234, 2009.

[37] KNIGHT, J. C., STRUNK, E. A., SULLIVAN, K. J. "Towards a rigorous definition of information system survivability". In: *Proceedings of the DARPA Information Survivability Conference and Exposition*, pp. 78–89, Washington, EUA, Abr. 2003.

[38] ALBERT, R., JEONG, H., BARABÁSI, A. "Error and Attack Tolerance of Complex Networks", *Letters to Nature*, v. 406, n. 6794, pp. 378–382, 2000.

[39] COLEMAN, T. F., MORÉ, J. J. "Estimation of Sparse Jacobian Matrices and Graph Coloring Problems", *SIAM journal on Numerical Analysis*, v. 20, n. 1, pp. 187–209, 1983.

[40] NEUMAYER, S., MODIANO, E. "Network Reliability With Geographically Correlated Failures". In: *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–9, San Diego, EUA, Mar. 2010.

[41] TOUCH, J., PERLMAN, R. "Transparent Interconnection of Lots of Links TRILL: Problem and Applicability Statement", *RFC 5556*, Maio 2009.

[42] ALLAN, D., ASHWOOD-SMITH, P., BRAGG, N., et al. "Shortest Path Bridging: Efficient Control of Larger Ethernet Networks", *IEEE Communications Magazine*, v. 48, n. 10, pp. 128–135, 2010.

[43] MUDIGONDA, J., YALAGANDULA, P., AL-FARES, M., et al. "SPAIN: COTS Data-Center Ethernet for Multipathing over Arbitrary Topologies". In: *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 18–18, San José, EUA, Abr. 2010.

[44] BILAL, K., MANZANO, M., KHAN, S., et al. "On the Characterization of the Structural Robustness of Data Center Networks", *IEEE Transactions on Cloud Computing*, v. 1, n. 1, pp. 64–77, 2013.

[45] NI, W., HUANG, C., WU, J. "Provisioning High-Availability Datacenter Networks for Full Bandwidth Communication", *Computer Networks*, v. 68, pp. 71–94, 2014.

[46] VISHWANATH, K. V., NAGAPPAN, N. "Characterizing cloud computing hardware reliability". In: *Proceedings of the ACM Symposium on Cloud Computing (SOCC)*, pp. 193–204, Indianápolis, EUA, Jun. 2010.

[47] CHOLDA, P., TAPOLCAI, J., CINKLER, T., et al. "Quality of resilience as a network reliability characterization tool", *IEEE Network*, v. 23, n. 2, pp. 11–19, 2009.

[48] *Rackspace Cloud Services - Legal Information.* Rackspace, US Inc., Jan. 2014. http://www.rackspace.com/information/legal/cloud/sla - Acessado em Agosto de 2014.

[49] COUTO, R. S., SECCI, S., CAMPISTA, M. E. M., et al. "Network Design Requirements for Disaster Resilience in IaaS Clouds", *IEEE Communications Magazine*, v. 52, n. 10, pp. 52–58, 2014.

[50] WOOD, T., CECCHET, E., RAMAKRISHNAN, K., et al. "Disaster Recovery as a Cloud Service: Economic Benefits & Deployment Challenges". In: *Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*, pp. 1–7, Boston, EUA, Jun. 2010.

[51] CABINET OFFICE. *ITIL®Service Design 2011 Edition*. Norwich, Reino Unido, TSO, 2008.

[52] *ISO/IEC 24762:2008, Information technology - Security techniques - Guidelines for Information and Communications Technology Disaster Recovery Services*. ISO/IEC standard, 2008.

[53] HABIB, M. F., TORNATORE, M., DIKBIYIK, F., et al. "Disaster Survivability in Optical Communication Networks", *Computer Communications*, v. 36, n. 6, pp. 630–644, 2013.

[54] XIAO, J., WEN, H., WU, B., et al. "Joint Design on DCN Placement and Survivable Cloud Service Provision over All-Optical Mesh Networks", *IEEE Transactions on Communications*, v. 62, n. 1, pp. 235–245, 2014.

[55] STANKIEWICZ, R., CHOLDA, P., JAJSZCZYK, A. "QoX: What Is It Really?" *IEEE Communications Magazine*, v. 49, n. 4, pp. 148–158, 2011.

[56] *Rackspace Cloud DNS Overview*. Rackspace, US Inc., Jul. 2012. http://www.rackspace.com/knowledge_center/article/rackspace-cloud-dns-overview - Acessado em Novembro de 2014.

[57] BARI, M., BOUTABA, R., ESTEVES, R., et al. "Data Center Network Virtualization: A Survey", *IEEE Communications Surveys & Tutorials*, v. 15, n. 2, pp. 909–928, 2013.

[58] *The Cloud Calculator*. Intel Corporation, 2013. http://www.thecloudcalculator.com - Acessado em Novembro de 2014.

[59] PIÓRO, M., MEDHI, D. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. São Francisco, EUA, Elsevier, 2004.

[60] COUTO, R. S., SECCI, S., CAMPISTA, M. E. M., et al. "Latência Versus Sobrevivência no Projeto de Centros de Dados Geograficamente Distribuídos". In: *Anais do XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 809—822, Florianópolis, Brasil, Maio 2014.

[61] COUTO, R. S., SECCI, S., CAMPISTA, M. E. M., et al. "Latency Versus Survivability in Geo-Distributed Data Center Design". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1107—1112, Austin, EUA, Dez. 2014.

[62] CISCO. "Cisco Global Cloud Index: Forecast and Methodology, 2012–2017", 2013.

[63] TELIKEPALLI, R., DRWIEGA, T., YAN, J. "Storage Area Network Extension Solutions and Their Performance Assessment", *IEEE Communications Magazine*, v. 42, n. 4, pp. 56–63, 2004.

[64] BODÍK, P., MENACHE, I., CHOWDHURY, M., et al. "Surviving Failures in Bandwidth-constrained Datacenters". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pp. 431–442, Helsínquia, Finlândia, Ago. 2012.

[65] CHEN, Y., JAIN, S., ADHIKARI, V., et al. "A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets". In: *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 1620–1628, Xangai, China, Abr. 2011.

[66] RAMASWAMI, R., SIVARAJAN, K., SASAKI, G. *Optical Networks: A Practical Perspective*. 3a ed. Burlington, EUA, Morgan Kaufmann, 2009.

[67] DEVELDER, C., BUYSSE, J., SHAIKH, A., et al. "Survivable Optical Grid Dimensioning: Anycast Routing with Server and Network Failure Protection". In: *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–5, Quioto, Japão, Jun. 2011.

[68] RAMAMURTHY, S., SAHASRABUDDHE, L., MUKHERJEE, B. "Survivable WDM Mesh Networks", *Journal of Lightwave Technology*, v. 21, n. 4, pp. 870, 2003.

[69] CULLY, B., LEFEBVRE, G., MEYER, D., et al. "Remus: High Availability via Asynchronous Virtual Machine Replication". In: *Proceedings of the*

USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 161–174, São Francisco, EUA, Abr. 2008.

[70] DA SILVA, M. P., KOSLOVSKI, G., OBELHEIRO, R. R. "Uma Análise da Sobrecarga Imposta pelo Mecanismo de Replicação de Máquinas Virtuais Remus". In: *Anais do XV Workshop de Testes e Tolerância a Falhas (WTF)*, pp. 160—173, Florianópolis, Brasil, Maio 2014.

[71] WOOD, T., LAGAR-CAVILLA, H. A., RAMAKRISHNAN, K. K., et al. "PipeCloud: Using Causality to Overcome Speed-of-light Delays in Cloud-based Disaster Recovery". In: *Proceedings of the ACM Symposium on Cloud Computing (SOCC)*, pp. 1–13, Cascais, Portugal, Out. 2011.

[72] MULLER, L. F., OLIVEIRA, R. R., LUIZELLI, M. C., et al. "Survivor: an Enhanced Controller Placement Strategy for Improving SDN Survivability". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1944—1950, Austin, EUA, Dez. 2014.

[73] PIERRE, G., VAN STEEN, M. "Globule: A Collaborative Content Delivery Network", *IEEE Communications Magazine*, v. 44, n. 8, pp. 127–133, 2006.

[74] YAO, J., LU, P., ZHU, Z. "Minimizing Disaster Backup Window for Geo-Distributed Multi-Datacenter Cloud Systems". In: *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 3631–3635, Cidade de Sydney, Austrália, Jun. 2014.

[75] BIANCO, A., GIRAUDO, L., HAY, D. "Optimal Resource Allocation for Disaster Recovery". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–5, Miami, EUA, Dez. 2010.

[76] YU, H., ANAND, V., QIAO, C., et al. "Cost Efficient Design of Survivable Virtual Infrastructure to Recover from Facility node Failures". In: *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–6, Quioto, Japão, Jun. 2011.

[77] ABRAMOWITZ, M., STEGUN, I. A. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. 9a ed. Nova Iorque, EUA, Dover Books on Mathematics, 1970.