Pós-Graduação em Ciência da Computação

"Modelos para Planejamento de Redes Convergentes Considerando a Integração de Aspectos de Infraestrutura e de Negócios"

Por

Almir Pereira Guimarães

Tese de Doutorado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, DEZEMBRO/2013

Universidade Federal de Pernambuco

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALMIR PEREIRA GUIMARÃES

"Modelos para Planejamento de Redes Convergentes Considerando a Integração de Aspectos de Infraestrutura e de Negócios"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: Prof. Dr. Paulo Romero Martins Maciel

RECIFE, DEZEMBRO/2013

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da Silva, CRB4-1217

Guimarães, Almir Pereira
Modelos para planejamento de redes convergentes considerando a integração de aspectos de infraestrutura e de negócios / Almir Pereira Guimarães. - Recife: O Autor, 2013.
203 p.: il., fig., tab.

Orientador: Paulo Romero Martins Maciel.
Tese (doutorado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2013.

Inclui referências e apêndices.

1. Engenharia da computação. 2. Redes de computadores 3. Dependabilidade. 4. Otimização. I. Maciel, Paulo Romero Martins (orientador). II. Título.

621.39 CDD (23. ed.) MEI2014 – 020

Tese de Doutorado apresentada por **Almir Pereira Guimarães** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Modelos para Planejamento de Redes Convergentes Considerando a Integração de Aspectos de Infraestrutura e de Negócios**" orientada pelo **Prof. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Paulo Roberto Freire Cuna
Centro de Informática / UFPE

_____

Prof. DjamelFawzi Hadj Sadok
Centro de Informática / UFPE

_____

Prof. Stênio Flávio de Lacerda Fernandes
Centro de Informática / UFPE

_____

Prof. Luiz Carlos Pessoa Albini
Departamento de Informática / UFPR

_____

Prof. Artur Ziviani
Laboratório Nacional de Computação Cientifica

Visto e permitida a impressão.
Recife, 9 de dezembro de 2013.

_____

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

A Deus.
À Minha Família.
Aos Meus Amigos.
Ao Prof. Dr. Paulo Romero Martins Maciel, orientador.

AGRADECIMENTOS

Gostaria de agradecer a todos que contribuíram para o desenvolvimento deste trabalho. Ao professor Paulo Maciel, pela orientação, apoio e enorme paciência, elementos essenciais para o desenvolvimento deste trabalho. Também gostaria de agradecer-lhe por todas as oportunidades de crescimento acadêmico e pessoal. Aos professores Artur Ziviani, Djamel Sadok, Luiz Carlos Albini, Paulo Cunha e Stênio Fernandes, por terem aceitado o convite para compor esta banca. Ao professor Virgílio Almeida pela participação na banca da proposta desta tese. A todos do grupo MoDCS (*Modeling of Distributed and Concurrent Systems*) pela amizade e parceria. Faço um agradecimento aos meus irmãos (André Luiz e Mário) e em especial a minha esposa (Kely), minha mãe (Maria José), meu pai (Almir, *In Memorian*), meus filhos (Marcelo e Sofia) e Glória, que de tão preciosos não encontro palavras para descrevê-los. Sem eles, e também sem os meus amigos de coração, não teria tido condições de finalizar esta tese. Foram meu suporte nas horas mais difíceis desta longa jornada. Agradeço, principalmente, a Deus, que colocou todas essas pessoas em meu caminho.

Se você pensar que pode ou que não pode, de qualquer forma, você estará certo.

— HENRY FORD

RESUMO

Nos dias atuais, as redes convergentes têm se tornado uma ferramenta essencial para os indivíduos, empresas e governos e têm mudado significativamente a maneira como eles se relacionam. Além disso, estas redes proporcionam o meio de transporte para uma grande variedade de serviços e desta forma, desempenham uma importância considerável para as atividades diárias dos indivíduos e para as receitas das empresas. Questões relativas à prevenção e tratamento de falhas, bem como a garantia de desempenho satisfatório, são assim, de fundamental importância para a permanência nos negócios por parte das empresas. Desta maneira, o projeto da infraestrutura de redes convergentes deve considerar também os aspectos de negócios. A utilização de modelos formais vem sendo constantemente aplicada em diferentes abordagens para a descrição de sistemas computacionais. Este trabalho propõe conjuntamente modelos, métricas e uma metodologia para proporcionar suporte tanto para a escolha do melhor projeto com relação a uma infraestrutura em particular, quanto para uma análise comparativa e objetiva entre diferentes soluções, considerando-se formalmente aspectos de dependabilidade, desempenho e de negócios. Os modelos de infraestrutura consideraram as vantagens de redes de Petri estocásticas e de diagramas de blocos de confiabilidade em uma abordagem hierárquica. Foram ainda definidas métricas para o suporte à escolha do melhor projeto em conformidade com os negócios da empresa, assim como para proporcionar uma análise comparativa e objetiva entre as diferentes soluções de projetos. Por fim, os resultados mostraram a eficácia da aplicação desta metodologia ao reduzir significativamente o número de projetos analisados.

Palavras-Chave: Dependabilidade; Desempenho; Custo de Infraestrutura; Lucro Líquido; Rede de Petri Estocástica; Cadeia de Markov Baseada em Tempo Contínuo; Diagrama de Blocos de Confiabilidade; Prrojeto de Redes.

ABSTRACT

Nowadays, converged networks have become an essential tool for individuals, companies and governments and have significantly changed the way they relate. Additionally, these networks provide the means of transport for a wide range of services and thus play considerable importance for the daily activities of individuals and companies for revenue. Issues relating to the prevention and treatment of failures as well as the guarantee of satisfactory performance, are thus of paramount importance to stay in business by companies. Thus, converged network infrastructure design should also consider the business aspects. The use of formal models has been consistently applied in different approaches to the description of computational systems. This paper proposes jointly models, metrics and methodology to provide support both for the choice of the best design with respect to an infrastructure as for a comparative and objective analysis among different solutions considering aspects of dependability, performance and business. The infrastructure models considered the advantages of stochastic Petri nets and reliability block diagrams on a hierarchical approach. It was also defined metrics to support the choice of the best design in accordance with the company's business as well as to provide an objective and comparative analysis between different design solutions. Finally, the results showed the effectiveness of the application of this methodology to significantly reduce the number of analyzed designs.

Keywords: Dependability; Stochastic Petri Net; Continuous Time Markov Chain; Reliability Importance; Reliability Block Diagram;

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

- **AHC** - Agrupamento hierárquico aglomerativo. Sigla proveniente do inglês *Agglomerative Hierarchical Clustering*;
- **BDIM** - *Business-Driven IT Management*;
- **BORAP** - *Bi-Objective Redundancy Allocation Problem*;
- **bps** - Bits por segundo. Sigla proveniente do inglês *bits per second*;
- **CDF** - Função de distribuição cumulativa. Sigla proveniente do inglês *Cumulative Distribution Function*;
- **CLI** - Interface de linha de comando. Sigla proveniente do inglês *Command Line Interface*;
- **CODEC** - Codificador/Decodificador. Sigla proveniente do inglês *Coder/Decoder*;
- **CQ** - *Custom Queuing*;
- **CTMC** - Cadeia de Markov de tempo contínuo. Sigla proveniente do inglês *Continuous Time Markov Chain*;
- **CV** - Coeficiente de Variação;
- **DTMC** - Cadeia de Markov de tempo discreto. Sigla proveniente do inglês *Discrete Time Markov Chain*;
- **ECS** - *Extended Conflict Set*;
- **EFM** - *Energy Flow Model*;
- **FIFO** - Primeiro a entrar primeiro a sair. Sigla proveniente do inglês *First-In-First-Out*;
- **FT** - Árvore de falhas. Sigla proveniente do inglês *Fault-Tree*;
- **GSPN** - Rede de Petri estocástica generalizada. Sigla proveniente do inglês *Generalized Stochastic Petri Net*;
- **GA** - Algoritmo genético. Sigla proveniente do inglês *Genetic Algorithm*;
- **HGA** - *Hierarchical Genetic Algorithm*;
- **IP** - Protocolo de internet. Sigla proveniente do inglês *Internet Protocol*;

- **ISP** - Provedor de serviços de intrenet. Sigla proveniente do inglês *Internet Service Provider*;
- **Kbps** - Quilobit por segundo. Sigla proveniente do inglês *Kilobit per second*;
- **LAN** - Rede de área local. Sigla proveniente do inglês *Local Area Network*;
- **MA** - Algoritmo memético. Sigla proveniente do inglês *Memetic Algorithm*;
- **MC** - Cadeia de Markov. Sigla proveniente do inglês *Markov Chain*;
- **MPSO** - *Multi-Objective Particle Swarm Optimization*;
- **MTTA** - Tempo médio para ativação. Sigla proveniente do inglês *Mean Time To Activation*;
- **MTTF** - Tempo médio para falha. Sigla proveniente do inglês *Mean Time To Failure*;
- **MTTR** - Tempo médio para reparo. Sigla proveniente do inglês *Mean Time To Repair*;
- **NSGA-II** - *Non-Dominated Sorting Genetic Algorithm*;
- **OSP** - Provedor de serviços online. Sigla proveniente do inglês *Online Service Provider*;
- **PC** - *Personal Computer*;
- **pmf** - Função de probabilidade de massa. Sigla proveniente do inglês *Probability Mass Function*;
- **PN** - Redes de Petri. Sigla proveniente do inglês *Petri Net*;
- **PPP** - Protocolo ponto a ponto. Sigla proveniente do inglês *Point-to-Point Protocol*;
- **pps** - Pacotes por segundo. Sigla proveniente do inglês *Packet Per Second*;
- **PQ** - *Priority Queuing*;
- **PRTG** - *Paessler Router Traffic Grapher*;
- **PSO** - Otimização por enxame de partículas. Sigla proveniente do inglês *Particle Swarm Optimization*;
- **RAP** - Problema de alocação de redundância. Sigla proveniente do inglês *Redundancy Allocation Problem*;
- **RBD** - Diagrama de bloco de confiabilidade. Sigla proveniente do inglês *Reliability Block Diagram*;
- **RG** - Grafo de alcançabilidade. Sigla proveniente do inglês *Reachability Graph*;
- **RMSSTD** - *Root Mean Square Standard Deviation*;

- **RMP** - Problema mestre restrito. Sigla proveniente do inglês *Restricted Master Problem*;
- **RS** - Conjunto de marcações alcançáveis. Sigla proveniente do inglês *Reachability Set*;
- **RSq** - *R Squared*;
- **SaaS** - Software as a Service;
- **SHARPE** - *Symbolic Hierarchical Automated Reliability and Performance Evaluator*;
- **SLA** - Acordo de nível de serviço. Sigla proveniente do inglês *Service Level Agreement*;
- **SLI** - Indicadores do nível de serviço. Sigla proveniente do inglês *Service Level Indicator*;
- **SLM** - Gerenciamento de Níveis de Serviço. Sigla proveniente do inglês *Service Level Management*;
- **SLO** - Objetivos do nível de serviço. Sigla proveniente do inglês *Service Level Objective*;
- **SNMP** - Protocolo de gerenciamento de rede simples. Sigla proveniente do inglês *Simple Network Management Protocol*;
- **SPN** - Rede de Petri estocástica. Sigla proveniente do inglês *Stochastic Petri Net*;
- **TCA** - Custo total de aquisição. Sigla proveniente do inglês *Total Cost of Acquisition*;
- **TED** - Taxa de Entrada de Dados;
- **TFGEN** - *Traffic Generator*;
- **TI** - Tecnologia da informação;
- **TTF** - Tempo para falha. Sigla proveniente do inglês *Time To Failure*;
- **TTR** - Tempo para reparo. Sigla proveniente do inglês *Time To Repair*;
- **TTP** - *Trusted Third Party*;
- **VoIP** - Voz sobre IP. Sigla proveniente do inglês *Voice Over IP*;
- **UTP** - Par trançado não blindado. Sigla proveniente do inglês *Unshielded Twisted Pair*;
- **VM** - Máquina virtual. Sigla proveniente do inglês *Virtual Machine*;
- **VMM** - Monitor de máquina virtual. Sigla proveniente do inglês *Virtual Machine Monitor*;
- **WLAN** - Rede local sem fio. Sigla proveniente do inglês *Wireless Local Area Network*.

Sumário

CAPÍTULO 1

Introdução

1.1 Visão Geral

Desde a última metade do século *XX*, a informática tem penetrado em quase todos os aspectos da nossa sociedade. Como resultado, os sistemas computacionais se tornaram a base de muitas infraestruturas nacionais, tais como: telecomunicações, transportes, abastecimento de água, energia elétrica, serviços financeiros e serviços governamentais.

À medida que os sistemas computacionais permeiam todos os espaços da vida cotidiana das empresas, a necessidade de se avaliar aspectos de desempenho e dependabilidade de tais sistemas assume uma importância crescente. Grande parte desta demanda provém do acentuado nível de exigência dos consumidores, interessados cada vez mais em serviços de melhor qualidade. Esta demanda por qualidade se explica pela responsabilidade que o ser humano deposita nos produtos manufaturados, especificamente nos sistemas computadorizados ou microcomputadorizados [31].

Enquanto razões econômicas forçam o desenvolvimento de novos sistemas computacionais com um número cada vez maior de facilidades, razões de qualidade impõem a necessidade de um funcionamento contínuo por parte destes sistemas. Por sua vez, os custos anuais com paralisações ou mesmo falhas são da ordem de bilhões de dólares devido à dependência cada vez maior das empresas nestes sistemas, em particular, em redes convergentes [50].

Essas redes, através de seu enorme crescimento, têm grande impacto sobre a maneira como

nós vivemos. Consequentemente, eventos como falhas ou mau funcionamento em subsistemas ou componentes de sua infraestrutura tornam-se cada vez mais significativos. Estes eventos podem ocorrer devido a problemas físicos, a exemplo da descontinuidade de enlaces, problemas de software, problemas de fornecimento de energia, incêndios ou terremotos. O seu correto funcionamento é importante por uma simples razão: empresas perdem dinheiro devido a falhas ou mau funcionamento, ocasionando prejuízos de diversas naturezas. Desta forma, uma das metas a ser seguida refere-se à elaboração de um projeto adequado relativo à infraestrutura de redes convergentes.

Para executar um projeto em redes convergentes, métricas técnicas (ex., disponibilidade, confiabilidade, desempenho), tanto quanto seus valores (ex., 99,9% de disponibilidade, 100 pps de vazão), devem ser selecionadas, juntamente com o esforço de se obter um projeto que minimize os custos e atenda aos requisitos das métricas consideradas. A questão mais importante é que a área de negócios não consegue traduzir seus objetivos, definidos através de métricas orientadas a negócios, utilizando apenas métricas técnicas relativas ao projeto. Desta maneira, como existe uma distância semântica entre as métricas orientadas a negócios em relação às métricas técnicas, seus requisitos não são capturados de maneira significativa.

Visto que as redes convergentes são ferramentas para auxiliar empresas a atingirem seus objetivos, é uma falha não considerar formalmente as necessidades de negócios em seus projetos de infraestrutura [85]. Estas necessidades referem-se, por exemplo, ao impacto que a indisponibilidade dos serviços terá sobre as receitas ou sobre os custos decorrentes do não cumprimento de metas acertadas. Outro exemplo seria o impacto que a permuta de um componente da rede teria sobre os custos de infraestrutura, sobre as receitas e, consequentemente, sobre os lucros.

O problema descrito é relevante pois pode acarretar grandes perdas financeiras ao não levar em conta os aspectos de negócios no projeto de infraestrutura. Em nosso trabalho, iremos propor, de maneira integrada, modelos, métricas e uma metodologia para viabilizar tanto a escolha do melhor projeto com relação a uma infraestrutura em particular, quanto para executar

análise comparativa entre as soluções de projeto obtidas a partir de diferentes arquiteturas, considerando-se formalmente os aspectos de dependabilidade, desempenho e de negócios [45].

1.2 Motivação

Devido à importância estratégica, empresas dos mais diferentes setores estão sendo empurradas para uma realidade que as obriga concentrar uma maior quantidade de esforços em questões relativas ao planejamento de suas redes convergentes. Esta preocupação pode ocorrer por exigências contratuais, por força de regulamentações ou pela pressão de seus clientes que irão impactar direta ou indiretamente sobre seus aspectos de negócios. Por sua vez, empresas perdem em média de 2% a 16% de suas receitas devido ao tempo em que suas redes permanecem paradas [49]. Estas perdas variam em função tanto do grau de dependência em relação a estas redes como do setor da economia ao qual as empresas pertencem. A Tabela 1.1 mostra o custo relativo ao tempo de parada, por hora, em diferentes setores da economia americana.

Tabela 1.1: Custo do tempo de parada por hora em diferentes setores da economia

Setor	Aplicação	Custo *Tempo Parada*/Hora (Us$)
Financeiro	Operações de Corretagem	7.840.000
Financeiro	Vendas por Cartão de Crédito	3.160.000
Mídia	*Pay-per-View*	183.000
Varejo	Telecompras (TV)	137.000
Varejo	Vendas por Catálogo	109.000
Transporte	Reservas Aéreas	108.000
Diversão	Tele-Vendas de Ingressos	83.000

Fonte:*Navigating Network Infrastructure Expenditures During Business Transformations*, Nicholas John Lippis III, July 2009

De acordo com a abordagem utilizada em projetos de redes convergentes, centenas de milhares de dólares em gastos financeiros pode ser a diferença entre soluções que levam formalmente em consideração aspectos relativos a negócios em conjunto com aspectos técnicos e

soluções *ad-hoc*, mesmo para infraestruturas de tamanho médio [85]. Desta forma, deve-se considerar que os aspectos relativos à infraestrutura e a negócios sejam avaliados de maneira integrada, permitindo uma análise de diferentes parâmetros de desempenho e dependabilidade em função dos impactos destes sobre os aspectos de negócios das empresas.

Por exemplo, dependendo da adoção ou não de mecanismos de redundância na fase de projeto, podemos ter diferentes resultados para a disponibilidade do sistema, o que pode afetar tanto os custos de infraestrutura como as receitas e multas oriundas dos serviços diretamente relacionados. As políticas de fila suportadas por um dos componentes da infraestrutura pode ter impacto sobre o desempenho de todo o tráfego da rede, afetando tanto os custos de infraestrutura como as receitas obtidas. Desta maneira, decisões na fase de planejamento de redes convergentes afetam diretamente os aspectos de negócios.

Além disso, espera-se também analisar, de maneira isolada, diferentes configurações alternativas a uma configuração adotada como padrão, em termos de equipamentos e mecanismos de redundância, podendo-se estimar o impacto destas alternativas sobre os aspectos de negócios. Os resultados obtidos destas avaliações podem ser utilizados para dar suporte a um processo de tomada de decisão que procure otimizar o custo-benefício entre os aspectos em questão.

Assim sendo, são necessários mecanismos eficazes para uma avaliação dos aspectos de infraestrutura e de negócios de maneira formal e integrada. Devido à complexidade, ou mesmo à impossibilidade de se avaliar os sistemas de outra forma, adota-se a modelagem como estratégia para análise quantitativa do sistema. A utilização tanto de modelagem analítica como de modelagem para simulação da descrição de diferentes aspectos de um sistema deve-se ao fato de o sistema ainda não existir ou pela necessidade, facilidade e custos de criação de diferentes situações e cenários. Estes aspectos podem ser capturados através das métricas oriundas dos modelos propostos. Neste trabalho, foram utilizados diagrama de blocos de confiabilidade (RBD) [84, 95, 103], rede de Petri estocástica (SPN) [7, 69], cadeias de Markov de tempo contínuo (CTMC) [7, 102], junto com modelos para a representação dos aspectos de negócios [85].

Cada uma dessas técnicas possui características que a torna mais ou menos adequada para ser empregada em um determinado tipo de avaliação.

1.3 Definição do Problema

Uma pesquisa exaustiva examinaria dezenas, centenas ou mesmo mihares de projetos de infraestrutura. Cada um destes projetos corresponde a uma combinação específica de parâmetros considerados relativos a cada um dos componentes da infraestrutura. Entretanto, esta solução poderia levar meses para explorar um projeto viável. Para acelerar este tempo, diferentes metododologias têm sido desenvolvidas para reduzir o número de possíveis projetos a serem examinados. O problema com estas abordagens é que os requisitos de negócios não têm sido propriamente capturados em suas soluções, podendo acarretar perdas financeiras por parte das empresas.

A escolha da melhor opção de projeto para uma infraestrutura deve ser orientada pela otimização da relação entre os aspectos de negócios e de infraestrutura em conjunto com técnicas para a redução do número de projetos candidatos. Os aspectos de negócios devem ser relativos aos custos e receitas oriundos da infraestrutura, custos adicionais pelo não cumprimento de metas acertadas etc. Suas estimativas podem ser obtidas através de modelos, que devem estar diretamente relacionados aos aspectos técnicos de redes convergentes, tais como: desempenho e dependabilidade.

1.4 Contribuições

Nos últimos anos diferentes trabalhos têm buscado soluções quase-ótimas de projetos de infraestrutura de sistemas computacionais [10, 30, 81, 88, 109]. Entretanto, nenhuma destas referências considera formalmente aspectos de negócios em suas abordagens. Ao invés disso, esses aspectos são geralmente tratados como restrições para os projetos baseados em métricas técnicas. Por outro lado, nos trabalhos mostrados em [85, 87, 93, 94, 108] são consideradas formalmente métricas técnicas e de negócios para proporcionar suporte à determinação da melhor opção de projeto. No próximo capítulo será realizada uma análise comparativa do nosso trabalho em função de trabalhos correlatos.

Figura 1.1: Estrutura da tese

O principal objetivo de nossa tese é a proposição conjunta de modelos, métricas e uma metodologia para proporcionar suporte tanto para a escolha do melhor projeto com relação a uma infraestrutura em particular quanto para uma análise comparativa e objetiva entre as soluções de projetos de diferentes infraestruturas de redes convergentes, considerando-se de maneira integrada os aspectos de dependabilidade, desempenho e de negócios [45]. Um esquema da estrutura proposta é mostrado na Figura 1.1. Esta estrutura atua sobre os níveis de infraestrutura e de negócios considerando e relacionando seus aspectos de maneira formal e integrada. A união dos aspectos relativos a estes níveis para a escolha do melhor projeto de uma infraestrutura em particular é uma contribuição importante desta tese.

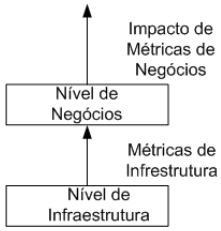


Figura 1.2: Níveis da estrutura

Além disso, embora desenvolvido inicialmente para infraestruturas de redes convergentes, nosso trabalho pode ser adaptado para a busca do melhor projeto de infraestrutura de diferentes tipos de sistemas computacionais. Outras importantes contribuições desta tese podem ser sumarizadas abaixo:

- Concepção de modelos heterogêneos de infraestrutura utilizando SPN, CTMC, RBD e modelos hierárquicos [7]. Além disso, são propostos modelos de negócios que possuem como parâmetros as métricas de infraestrutura (ver Figura 1.2), o que possibilita o cálculo do impacto dos aspectos técnicos sobre os negócios relacionados, tais como: custos de infraestrutura, receitas de infraestrutura, multas etc;

- Definição de métricas para o suporte à escolha do melhor projeto de uma infraestrutura em conformidade com os negócios da empresa. Nossa abordagem propõe métricas que consideram aspectos de infraestrutura e de negócios de maneira conjunta ao invés de separadamente. Estas métricas capturam o impacto dos eventos oriundos da infraestrutura sobre os negócios. Além disso, proporcionam uma análise comparativa e objetiva entre as melhores soluções de projetos de diferentes infraestruturas;

- Definição de uma metodologia baseada em mecanismos tais como: agrupamento hierárquico aglomerativo (AHC) [24], importância para confiabilidade [56] e projeto de experimento fatorial [51]. Esta metodologia tanto reduz as dificuldades para a escolha do melhor projeto, como incrementa a relação entre os aspectos técnicos e de negócios relacionados à infraestrutura de redes convergentes.

1.5 Estrutura do Trabalho

Este trabalho está organizado da seguinte maneira: o próximo capítulo irá mostrar uma visão geral dos trabalhos relacionados, contextualizando a nossa abordagem em relação aos trabalhos existentes. O Capítulo 3 irá apresentar conceitos importantes que serão utilizados em capítulos subsequentes. No Capítulo 4 serão mostrados modelos de dependabilidade e de desempenho para a representação dos mecanismos encontrados em projetos de infraestrutura. O Capítulo 5 detalhará modelos orientados a negócios em redes convergentes. No Capítulo 6 será detalhada a metodologia proposta utilizada no processo de otimização. Serão mostradas as diferentes fases desta metodologia junto com os algoritmos correspondentes às atividades *geração de agrupamentos/definição do número bom de agrupamentos*, *seleção de agrupamento* e *seleção de projeto*. No Capítulo 7 serão definidas quatro arquiteturas junto com seus modelos de dependabilidade e de desempenho. Estes modelos irão proporcionar suporte para diferentes tipos de análises sobre estas infraestruturas que serão detalhadas no capítulo referente ao estudo de diferentes tipos de casos. O Capítulo 8 apresentará quatro diferentes casos para exemplificar a aplicação da estrutura proposta. Serão mostrados tanto casos planejados, a partir das infraestruturas definidas no capítulo anterior, como casos definidos a partir da infraestruturas de redes convergentes reais. Por fim, o Capítulo 9 apresenta as contribuições de nosso trabalho e esboça possibilidades de trabalhos futuros.

CAPÍTULO 2

Trabalhos Relacionados

2.1 Introdução

O processo de projeto/dimensionamento da infraestrutura de sistemas computacionais assume grande importância pelo fato de que estes sistemas são críticos para indivíduos, negócios e governos. Nos últimos anos, uma grande quantidade de conhecimento em diversos aspectos de infraestrutura tem sido proposta com diferentes abordagens e objetivos. Alguns trabalhos consideram uma análise de maneira independente sobre os aspectos de infraestrutura com o objetivo de um melhor dimensionamento operacional. Estes últimos consideram a análise de aspectos relativos à dependabilidade e desempenho utilizando tanto técnicas baseadas em modelagem [14, 23, 57, 96, 104, 112], quanto técnicas baseadas em medição [19, 46, 64]. Eles não abordam de maneira simultânea aspectos de infraestrutura e de negócios. Suas contribuições para nosso trabalho residem nas proposições de diferentes abordagens e técnicas para a análise e dimensionamento de aspectos de dependabilidade e de desempenho. Outros trabalhos utilizam aspectos técnicos juntamente com aspectos de negócios visando um aperfeiçoamento do processo para a escolha do projeto quase-ótimo da infraestrutura destes sistemas.

2.2 Trabalhos Focados em Infraestrutura e Negócios

Alguns trabalhos vêm sendo propostos considerando os aspectos de infraestrutura e de negócios para a escolha do projeto ótimo ou quase-ótimo [10, 13, 30, 81, 85, 87, 88, 93, 94, 106, 108, 109].

Inicialmente, o trabalho mostrado em [87] apresenta uma abordagem para a escolha da melhor opção de projeto de redes considerando formalmente aspectos de infraestrutura, tais como disponibilidade e aspectos orientados a negócios, a exemplo de custos de infraestrutura. Foram construídos modelos analíticos para o cálculo da disponibilidade de diferentes arquiteturas. Este trabalho apresenta uma importante contribuição ao propor métricas que integram formalmente os aspectos de infraestrutura e de negócios, a fim de executar uma análise comparativa entre diversas opções de projetos. Baseado nestas variáveis, são fornecidas recomendações para auxiliar os projetistas na seleção da melhor solução. Entretanto, o trabalho em [87] apresenta limitações tanto para um número crescente de projetos candidatos quanto para infraestruturas complexas. Além disso, não são discutidas questões relativas a diferentes métricas de negócios, tais como receitas, multas, assim como questões relativas ao desempenho destas redes.

Nas próximas seções iremos mostrar diferentes áreas de pesquisa que tratam questões relativas a projetos de infraestrutura de sistemas computacionais.

2.2.1 Problema de Alocação de Redundância - RAP

O problema de alocação de redundância (RAP) é um dos mais importantes problemas de otimização no que diz respeito à melhoria da confiabilidade de sistemas em sua fase de projeto [106]. Ele tem atraído muitos pesquisadores devido à criticidade da confiabilidade para vários tipos de sistemas, tais como sistemas elétricos, sistemas mecânicos e sistemas computacionais

[54, 55].

Nas últimas décadas, tem havido uma série de estudos com diferentes abordagens para o RAP na busca da solução ótima ou quase-ótima para o projeto de diferentes tipos de sistemas, que podem ser aplicados a sistemas computacionais [10, 13, 30, 81, 88, 106, 109].

Inicialmente, o trabalho mostrado em [10] propõe uma abordagem baseada em decomposição para a resolução do problema de alocação de redundância (RAP) com multiobjetivos em sistemas série-paralelo. Esta abordagem decompõe o problema original em vários subproblemas com multiobjetivos e sistematicamente combina as soluções.

O trabalho em [13] foca o problema de alocação de redundância (RAP) e desenvolve um RAP com dois objetivos (BORAP). Este modelo de RAP inclui sistemas série-paralelo não reparáveis em que a estratégia de redundância é considerada como uma variável de decisão para cada subsistema individualmente. As funções objetivo do modelo são (1) maximizar a confiabilidade do sistema e (2) minimizar o custo do sistema. Para a resolução do modelo, dois algoritmos meta heurísticos com multiobjetivos denominados de algoritmo Genético de Ordenação não dominante (NSGA-II) e otimização por enxame de partículas (MOPSO) são propostos. Além disso, o desempenho dos algoritmos é analisado e conclusões são demonstradas.

O trabalho em [30] aborda o problema de alocação de redundância de sistemas em série através de um algoritmo meta heurístico denominado de otimização por enxame de partículas (PSO). Por seu lado, PSO é um ramo inteligência artificial que otimiza um problema iterativamente ao tentar melhorar a solução candidata com respeito a uma dada medida.

O trabalho mostrado em [81] apresenta um novo algoritmo baseado em uma abordagem de otimização híbrida (*probabilistic solution discovery* e simulação de Monte Carlo) para a resolução de problemas de alocação de confiabilidade em redes, considerando a comunicação entre todos os nós. Este problema de otimização considera a minimização do custo do pro-

jeto de rede que constitui uma restrição sobre a confiabilidade, assumindo que a rede possui um número conhecido de componentes funcionalmente equivalentes (com diferentes especificações de desempenho).

Em [88] é apresentada uma abordagem híbrida baseada em algoritmo genético (GA) e algoritmo de otimização por enxame de partículas (PSO) para o problema de alocação de redundância em sistemas em série, série-paralelo e *bridge*. Esta abordagem maximiza a confiabilidade do sistema enquanto minimiza o custo, o peso e o volume do sistema, sob restrições não-lineares.

O trabalho em [106] investiga o problema de alocação de redundância considerando sistemas multiníveis. Ele propõe uma abordagem meta heurística baseada em uma nova versão do algoritmo memético (MA). Estudos experimentais em dois exemplos demonstram que a versão de MA proposta superou o algoritmo genético hierárquico (HGA) proposto em [53].

Por fim, o trabalho em [109] mostra um método de otimização para problemas de alocação de redundância em sistemas série-paralelo que está baseado na técnica de geração de colunas considerando diferentes tipos de componentes em cada subsistema. A solução proposta está baseada na formulação de um problema mestre restrito (RMP) e a geração de possíveis soluções através de um conjunto de subproblemas.

Embora reduzam o número de projetos candidatos para a obtenção do projeto ótimo ou quase-ótimo, os trabalhos apresentados nesta área de pesquisa não consideram formalmente as métricas de negócios em conjunto com métricas técnicas. Ao invés disso, eles tratam as métricas de negócios como restrições nos projetos baseados em métricas técnicas, tal como a confiabilidade. Além disso, questões de desempenho não são tratadas por estes trabalhos.

Uma nova área de pesquisa estende a gerência de serviços através da adoção de métricas de negócio em adição às métricas técnicas convencionais. Esta área é denominada de *BDIM* [61].

2.2.2 Gerência de TI Orientada a Negócios - BDIM

Com relação à BDIM, [86] apresenta a seguinte definição: "*BDIM é a aplicação de um conjunto de modelos, práticas, técnicas e ferramentas a fim de mapear e quantitativamente avaliar dependências entre soluções de TI e desempenho dos negócios e utilizando esta avaliação para aperfeiçoar a qualidade das soluções de tecnologia da informação em relação aos serviços e os resultados de negócios relacionados*".

Alguns trabalhos importantes em BDIM incluem aplicações para as áreas de gerenciamento de serviços (SLM) [8, 26, 58, 60] e de planejamento de infraestrutura de sistemas computacionais [85, 93, 94, 108].

Devido ao foco de nosso trabalho, iremos considerar os trabalhos da área de planejamento de infraestrutura de sistemas computacionais [85, 93, 94, 108].

Inicialmente, o trabalho mostrado em [85] trata de uma metodologia para a otimização do projeto de um *data center* (DC, centro de dados)[1] [4]. Esta metodologia avalia e compara diferentes projetos utilizando métricas orientadas a negócios (custos de infraestrutura, perdas financeiras) através de modelos analíticos. Ela proporciona a infraestrutura ótima que minimiza perda de negócios e a soma de provisionamentos que são decorrentes de falhas e degradação de desempenho. É mostrada uma visão estruturada em três níveis (ver Figura 2.1) para o projeto de uma infraestrutura de TI. O nível mais baixo mostra a perspectiva relativa à infraestrutura (Componentes de TI/Nível de Serviços de TI). São obtidas métricas tais como a disponibilidade e o tempo de resposta para os serviços. No nível mais alto são estimadas as métricas de negócios. O nível de Processos de Negócios faz a ligação entre os dois níveis.

O trabalho mostrado em [93] propõe uma arquitetura de computação em nuvem sob o ponto de vista de BDIM. Uma metodologia para o projeto otimizado da infraestrutura de computação em nuvem é concebida, em que o número de servidores, roteadores e a largura de banda de

[1] Por todo nosso trabalho será utilizado o termo *data center*

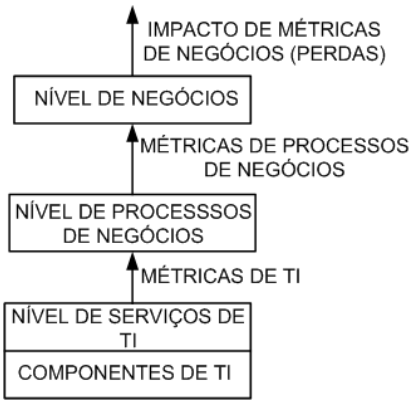


Figura 2.1: Níveis do modelo

comunicação podem ser calculados através da consideração tanto dos custos de infraestrutura quanto de perdas ocorridas pela violação do SLA [9]. Foram construídos modelos analíticos para quantificar tanto aspectos técnicos, quanto os aspectos orientados a negócios.

Em [94] é proposta uma abordagem para o projeto de infraestrutura utilizando os conceitos de BDIM, em um provedor de serviços de videoconferência. De acordo com esta abordagem, o número de servidores, roteadores e enlaces de comunicação são especificados a fim de minimizar tanto os custos da infraestrutura como as perdas de negócios relacionadas. Esta abordagem captura a relação entre os aspectos de infraestutura e de negócios através da modelagem da disponibilidade, do atraso da transmissão e diferentes instâncias de SLA.

Por fim, o trabalho em [108] propõe uma metodologia que busca obter o projeto ótimo de infraestrutura para provedores de serviços SaaS, baseada em BDIM. Este projeto minimiza os custos mensais de infraestrutura e de perda de negócios devido a uma infraestrutura imperfeita (indisponibilidade dos serviços, atraso no tempo de resposta etc.).

Estes trabalhos buscam selecionar o projeto ótimo ou quase-ótimo da infraestrutura de sistemas computacionais através de abordagens baseadas em modelos analíticos que relacionam formalmente aspectos técnicos e de negócios. Entretanto, eles apresentam limitações tanto em relação à redução do número de projetos candidatos como em relação a uma comparação objetiva entre diferentes soluções de projetos.

2.2.3 Análise Comparativa entre Trabalhos Relacionados

Os trabalhos existentes na literatura para a seleção do projeto ótimo ou quase-ótimo de uma infraestrutura de sistemas computacionais não consideram, de maneira integrada, mecanismos para a redução do número de projetos candidatos junto com mecanismos que relacionem formalmente os aspectos técnicos e de negócios. Em geral, estes trabalhos adotam mecanismos para a redução do número de projetos candidatos, tais como os trabalhos relativos à área de pesquisa para a alocação de redundância, ou mecanismos que relacionam formalmente os aspectos técnicos e de negócios, a exemplo dos trabalhos relativos à gerência de TI orientada a negócios. Por sua vez, o trabalho mostrado em [87] propõe métricas que integram explicitamente os aspectos de infraestrutura e de negócios para a escolha do melhor projeto de uma infraestrutura.

Esta tese está diretamente relacionada à área de projeto de infraestrutura. Ela estende nossos trabalhos anteriores apresentados em [37, 38, 39, 40, 41, 42, 43, 44, 45, 67, 72, 79], consolidando e generalizando seus resultados. Seu principal objetivo é a elaboração de modelos, métricas e de uma metodologia para proporcionar suporte tanto para a seleção do melhor projeto a partir de uma infraestrutura em particular quanto para uma análise comparativa e objetiva entre as soluções de projetos de diferentes infraestruturas de redes convergentes, considerando-se formalmente aspectos técnicos e de negócios. Diferentemente dos trabalhos relacionados neste capítulo, o nosso trabalho integra uma metodologia para a escolha do melhor projeto, reduzindo o número de projetos candidatos, modelos analíticos, relacionando formalmente aspectos técnicos e de negócios e métricas que relacionam explicitamente estes últimos aspectos.

Tabela 2.1 mostra uma comparação de nosso trabalho com os principais trabalhos voltados à determinação do projeto ótimo ou quase-ótimo de infraestruturas de sistemas computacionais, que consideram formalmente aspectos técnicos e de negócios.

Tabela 2.1: Comparação entre alguns trabalhos relacionados.

	Red._Proj._Candidatos	Modelos_Dep.	Modelos_Neg.	Modelos_Desemp.	Traf._Multim.	Métricas_Infr./Neg.
Nosso Projeto	S	S	S	S	S	S
Reference[85]	N	S	S	S	N	N
Reference[87]	N	S	N	N	N	S
Reference[93]	N	S	S	N	N	N
Reference[94]	N	S	S	S	S	N
Reference[108]	N	S	S	S	N	N

2.3 Considerações Finais

Neste capítulo foram apresentados trabalhos que mostraram uma grande quantidade de conhecimento em aspectos de infraestrutura de sistemas computacionais, com diferentes abordagens e objetivos. Inicialmente foram mencionados trabalhos que consideram uma análise independente sobre aspectos técnicos (desempenho e dependabilidade), mas que contribuiram na proposição de diferentes abordagens para a análise dos aspectos de infraestrutura. Então, foram descritos trabalhos que possuem por foco um aperfeiçoamento do processo de otimização para a escolha do projeto quase-ótimo destas infraestruturas. Nesta abordagem são considerados aspectos técnicos e aspectos de negócios. As diferentes abordagens utilizam tanto técnicas baseadas em modelagem, através de modelos analíticos (RBD, FT, SPN e CTMC) e simuladores, quanto técnicas baseadas em medições diretas.

CAPÍTULO 3

Fundamentação Teórica

3.1 Redes Convergentes

Cada um dos três séculos anteriores foi dominado por uma única tecnologia. O Século XVIII foi a época dos grandes sistemas mecânicos que acompanharam a Revolução Industrial. O Século XIX foi a era das máquinas a vapor. As principais conquistas tecnológicas do Século XX se deram no campo da aquisição, do processamento e da distribuição de informações. Entre vários desenvolvimentos, vimos a instalação das redes de telefonia em escala mundial, a invenção e a proliferação do rádio e da televisão, o nascimento e o crescimento sem precedentes da indústria de transmissão de dados e o lançamento dos satélites de comunicação. Como resultado deste enorme progresso tecnológico, ocorreu uma tendência para a convergência dos diferentes tipos de mídias utilizando uma única infraestrutura de rede.

Com relação às redes convergentes, estas são redes que possuem uma infraestrutura comum para o transporte simultâneo de dados, voz, imagens e vídeo. O processo de convergência tem ocorrido rapidamente integrando serviços que, anteriormente, requeriam equipamentos, canais de comunicação e padrões independentes.

3.2 Exigências de Projetos em Redes Convergentes

Desempenho, dependabilidade e os aspectos de negócios associados são características de grande importância em projetos de redes convergentes [34]. Existe um compromisso entre os diversos tipos de medidas do sistema com relação a estas três categorias. Projetando um sistema para atender simultaneamente e de maneira integrada as características de infraestrutura, como desempenho e dependabilidade, em conjunto com os aspectos de negócios, não é uma tarefa fácil. Estas características são fundamentalmente díspares: desempenho, dependabilidade e custos posicionam o projeto de redes convergentes em diferentes direções [34]. Por exemplo, um desempenho elevado poderá não acarretar em uma dependabilidade elevada e vice-versa.

A importância estratégica de um bom projeto de redes convergentes se mostra através do grau de dependência por parte das empresas dos serviços suportados por estas redes [34]. Deve-se levar em consideração que estes serviços estão relacionados aos aspectos de negócios de redes convergentes. Desta maneira, um bom projeto não será naturalmente o de maior custo, mas sim o que consegue dimensionar de maneira satisfatória as três principais características consideradas. Por exemplo, a utilização de componentes com maiores custos de aquisição poderá não atingir os níveis de dependabilidade que seriam atingidos através de componentes com custos menores utilizando diferentes mecanismos de redundância. O maior desafio dos projetistas é entender melhor estas inter-relações, de maneira a atingir um nível que elas atendam de maneira satisfatória as suas exigências.

A seguir, iremos detalhar os aspectos de negócios de infraestrutura que foram considerados na escolha do melhor projeto de uma infraestrutura de redes convergentes.

3.2.1 Aspectos de Negócios

Com relação aos aspectos de negócios relacionados às redes convergentes, iremos mostrar uma visão geral sobre os principais elementos que impactam sobre os custos de infraestrutura junto com os principais modelos de receitas e de multas que podem ser utilizados por provedores de serviços de comunicação [1].

Custos de Infraestrutura

Dependendo do segmento de atuação das empresas detentoras das redes convergentes, seus custos de infraestrutura são principalmente concentrados em algum(ns) dos elementos funcionais listados abaixo:

- Equipamentos Terminais;
- Rede de Acesso;
- Comutação;
- Enlaces de Transmissão/Longa Distância;
- Servidores;
- Dispositivos de Armazenamento.

É interessante frisar que estes elementos não representam todos os custos de infraestrutura destas redes, entretanto, representam uma parte significativa. Pode-se incorporar um ou mais destes elementos funcionais aos custos em função do segmento de atuação.

Os equipamentos terminais referem-se aos terminais de usuário que são ligados à rede, tais como: telefones, equipamentos de acesso à rede de dados, faxes, computadores, centrais

telefônicas etc. Entretanto, muitas vezes os clientes estão autorizados a adquirir seus próprios terminais, a partir do operador de rede ou de outros fornecedores. Esses terminais são de propriedade dos clientes e, em geral, não são uma parte do custo de infraestrutura. Isso permite que usuários se tornem mais independentes do operador. Além disso, os requisitos do operador de rede para os investimentos em infraestrutura são reduzidos.

Com relação às redes de acesso local, estas conectam os clientes com as redes nacionais e internacionais. A forma mais comum de acesso é através de cabos de par trançado a partir do terminal do usuário até o comutador local. Cabos coaxiais proporcionam uma maior capacidade de banda passante em relação a cabos de par trançado. Fibra ótica são utilizadas para certos serviços de banda larga, entretanto, são principalmente utilizadas na rede de comutação, onde sua maior capacidade de transmissão pode ser utilizada de maneira mais eficiente. Investimentos em cabeamento para redes de acesso local são investimentos de longo prazo, com uma vida útil de aproximadamente 20 anos [25]. Uma vez que os investimentos na rede de acesso em uma determinada área são feitas, o lucro deve ser gerado por meio de serviços de comunicação prestados a clientes.

Com relação à comutação, os componentes que a constituem são comutadores de grande capacidade que movimentam *tráfego* (informação) ao longo dos enlaces de transmissão/longa distância. Estes comutadores recebem as informações a partir dos enlaces conectados, passando-as para um outro enlace de transmissão. Comutadores que executam funções de roteamento, decidindo qual direção deve passar a informação, são denominados de roteadores.

Com relação aos enlaces de transmissão/longa distância, os equipamentos que os constituem incluem cabos, enlaces de rádios e satélites, que interligam roteadores de trânsito, assim como transmissores, repetidores etc.

Os dois últimos elementos, servidores e dispositivos de armazenamento, são largamente encontrados na infraestrutura de TI de *data centers*. Servidores são computadores com alta

capacidade de processamento e armazenagem que têm por função disponibilizar serviços, arquivos ou aplicações numa rede. É a opção preferida entre sites de alto tráfego, portais de conteúdo muito visitados ou por empresas que queiram hospedar suas aplicações aliando máximo desempenho com maior segurança e sigilo da informação hospedada. Podem estar localizados em um *data center* ou em um ambiente de uma rede de área local (LAN).

Por fim, dispositivos de armazenamento são aqueles capazes de armazenar informações (dados) para posterior consulta ou uso. Sua função é guardar a informação, processá-la ou ambas as coisas. Quando somente guarda informação, é chamado mídia de armazenamento; os que processam informações podem acessar mídia de gravação portátil, bem como podem ter um componente permanente, cuja função é o armazenamento e recuperação de dados.

Modelos de Receita

Modelos de receita descrevem as formas como as empresas geram receitas através de uma variedade de processos. Os principais modelos de receitas que podem ser utilizados por provedores de serviços de comunicação são descritos abaixo [1]:

- Publicidade: No modelo baseado em publicidade, a empresa fornece aos usuários finais conteúdo, subsidiado ou gratuito, serviços, ou mesmo produtos que atraem visitantes. Este modelo refere-se à publicidade como uma fonte de receita em si. Existem duas maneiras em que um modelo baseado em publicidade pode ser bem sucedido: a primeira é baseada na obtenção do maior público possível; a segunda baseia-se na obtenção de um público altamente especializado;

- Comissão: O modelo baseado em comissão é geralmente associado a empresas que agem como intermediárias sobre as transações efetuadas entre duas partes. Neste modelo, as taxas cobradas sobre as transações constituem a principal fonte de receita das empresas;

- Assinatura: Neste modelo, a empresa cobra uma taxa fixa em uma base periódica (como um mês) que qualifica o usuário para uma determinada quantidade de serviço. O usuário paga essa taxa de assinatura caso o serviço seja ou não utilizado;

- Remarcação: Refere-se ao valor acrescentado no processo de venda com relação ao valor de aquisição e, assim, a fonte primária de receita da empresa é através da remarcação. Este modelo tem sido usado tradicionalmente por atacadistas e varejistas e é por isso que alguns pesquisadores o chamam *modelo de comerciante*. Os bens podem ser vendidos por preços de tabela ou através de leilões;

- Produção: Neste modelo, fabricantes tentam alcançar os usuários finais ou clientes diretamente. Desta forma, eles podem fazer economia sobre os custos e melhor atender clientes descobrindo diretamente o que eles desejam. A maioria dos fornecedores de hardware e de software se enquadram neste modelo;

- Taxa-por-serviço: Neste modelo, as empresas arrecadam conforme o consumo. As atividades são medidas e os usuários pagam pelos serviços que consomem. Desta forma, os clientes pagam apenas pelos serviços realmente utilizados. Não existe assinatura básica para amortecer a empresa se a utilização cair;

- Encaminhamento: Neste modelo, empresas dependem de taxas para direcionar visitantes para outras empresas. Esta taxa é muitas vezes uma percentagem das receitas de uma eventual venda ou também pode ser uma taxa fixa por cada venda efetuada. A taxa fixa pode ser recolhida se um pedido for efetuado, pode ser recolhida independente se um pedido é efetuado ou não, ou pode ser recolhida cada vez que uma vantagem é gerada.

Os provedores de serviços de comunicação, em seus diferentes segmentos, podem gerar receitas principalmente a partir de um modelo baseado em *taxa-por-serviço* ou a partir de um modelo baseado em *assinatura*. Entretanto, de acordo com o segmento, podem também ser considerados diferentes modelos.

Modelos de Multa

Uma das principais questões que um provedor de serviços e um cliente devem acertar durante a negociação de um contrato é o esquema de penalidade a ser adotado. O uso de cláusulas de multas em contratos leva a duas importantes questões: que tipos de cláusulas de multas podem ser utilizadas e em quais partes elas podem ser incluídas em contratos. Além disso, dependendo da importância do item violado e das consequências desta violação, o provedor pode tentar evitar ou obter uma diminuição monetária da penalidade perante o cliente. Uma cláusula de multa em um contrato poderá consistir de uma das seguintes opções [82]:

- Uma redução no pagamento acertado pela utilização do serviço, isto é, uma sanção financeira;
- Redução no preço para o cliente, junto com compensação adicional para qualquer interação subsequente;
- Redução do uso do provedor de serviço por parte do consumidor;
- Um decréscimo na reputação do provedor e subsequente propagação deste valor para outros clientes.

Na fase de negociação do contrato, o cliente e o provedor podem acertar sobre uma sanção financeira direta. Normalmente, o valor a ser pago depende do valor da perda sofrida pelo cliente através da violação, e se acertado, uma quantia em dinheiro tem de ser paga como *multa* pelo comportamento indesejado. De outra forma, as partes podem escolher um "acordo de pagamento por não cumprimento de desempenho", ou seja, um montante fixo de dinheiro que terá de ser pago à não execução do serviço como acordado, independentemente do fato de que nenhuma perda financeira tenha sido sofrida pelo cliente. O provedor de serviço pode depositar o montante negociado em juízo, com um terceiro fidedigno (TTP, em inglês *trusted third party*),

que atua como mediador. Após a conclusão bem-sucedida da prestação do serviço (com base no contrato), o TTP retorna o depósito para o prestador de serviços. Caso contrário, o cliente recebe o depósito como compensação pela violação de SLA.

Outra possibilidade é que um cliente reduz seu uso de serviços de um provedor que violou o contrato. Se a situação econômica do cliente é forte o suficiente, isso pode ser uma estratégia bastante válida. Um outro tipo de cláusula de multa pode ser decorrente de uma mudança na reputação do provedor. Em tal sistema, a reputação de prestadores de serviços que violem os contratos vai cair. Neste caso, cuidado especial deve ser tomado para que a reputação de um prestador de serviços esteja corretamente determinada.

3.2.2 Aspectos Técnicos

Com relação aos aspectos técnicos relacionados às redes convergentes, a seguir iremos mostrar os principais atributos de dependabilidade e as políticas de filas *Custom Queuing* (CQ)[17], *Priority Queuing* (PQ)[16, 35] e FIFO [7].

Dependabilidade

Nos dias atuais, sistemas de computação têm sido empregados para o controle de uma enorme variedade de aplicações, desde aplicações domésticas até sistemas de satélite, nas quais as exigências de dependabilidade variam muito. Em sistemas militares, bancários, telecomunicações, controle de tráfego aéreo, serviços públicos de saúde, entre outros, o mau funcionamento pode causar grandes perdas econômicas ou mesmo vidas humanas.

Com relação às redes convergentes, com a crescente dependência das atividades sociais e econômicas em sistemas de informação, existe um grande interesse com relação aos aspectos

de dependabilidade das redes subjacentes a estes sistemas. Este interesse é devido à presença de falhas que podem ser originadas a partir de diversas fontes, tais como falhas de hardware, bugs de software, ataques mal-intencionados, erros humanos de operação/manutenção e desastres naturais. A possibilidade de evitar falhas que podem colocar os serviços em risco é um direcionamento que deve ser sempre seguido.

[5] proporcionou duas definições alternativas para dependabilidade (i) *a habilidade de proporcionar serviços que podem ser legitimamente confiáveis*; (ii) *a habilidade para evitar falhas de serviço que são mais frequentes e mais graves do que é aceitável*. As exigências de dependabilidade englobam os conceitos de disponibilidade, confiabilidade, integridade, mantenabilidade e segurança (*safety*) [5]. Então, definiremos os principais atributos de dependabilidade.

Para o cálculo da disponibilidade (A, em inglês Availability) de determinado dispositivo ou sistema, é necessário o cálculo do tempo para falha (TTF) e o tempo para reparo (TTR). Considerando-se que o tempo de atividade e inatividade não estão disponíveis, a opção mais acessível é a média. Neste caso, as métricas comumente adotadas são o tempo médio para falha (MTTF) e o tempo médio para reparo (MTTR). No entanto, se existe também interesse na variação da disponibilidade, o desvio padrão do tempo de falha (*sd*(TTF)), e seu respectivo desvio padrão do tempo de reparo (*sd*(TTR)) permitem uma estimativa da variação da disponibilidade.

O MTTF é proporcionado pelo fabricante e representa o tempo médio para a falha. O MTTR está diretamente relacionado à política de manutenção adotada pelas empresas, que irá depender da criticidade de cada componente em relação ao sistema. A disponibilidade é obtida pela análise ou simulação do estado estacionário. Ela é dada pela seguinte equação [84]:

$$A = \frac{MTTF}{MTTF + MTTR} \tag{3.1}$$

A disponibilidade em *estado estacionário* é uma medida mais fácil de calcular, mas faltam informações importantes que são capturadas pela disponibilidade transiente, especialmente em

relação a valores menores de t. Outro conceito útil é a disponibilidade instantânea, que indica a disponibilidade para um determinado instante (t), definida por [84]:

$$A(t) = \frac{\mu}{\mu+\lambda} + \frac{\lambda}{\mu+\lambda} e^{-(\mu+\lambda)t} \tag{3.2}$$

Onde μ indica a taxa de reparo (1/MTTR), e λ corresponde à taxa de falha (1/MTTF). Considera-se que os tempos de falha e de reparo seguem distribuições exponenciais.

Uma forma típica da função $A(t)$ para um sistema reparável é mostrada na Figura 3.1. Seu valor é 1 no instante inicial e decresce tendendo a um valor limite conhecido como *disponibilidade em estado estacionário*.

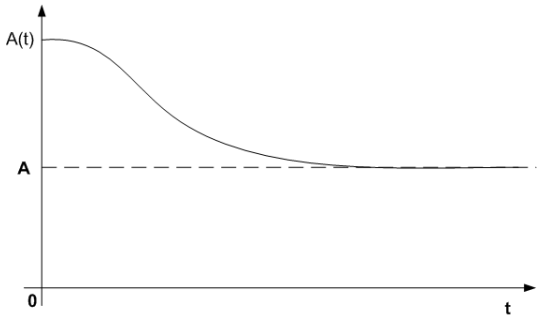


Figura 3.1: Função disponibilidade

A confiabilidade (R, em inglês Reliability) de um componente ou sistema é sua habilidade de funcionar corretamente durante um período de tempo especificado [84]. Este atributo mede a correta continuidade de um serviço, a partir de um tempo de referência. Sob o ponto de vista matemático, a confiabilidade, $R(t)$, de um sistema S pode ser expressa por [84]:

$$R(t) = Pr(\text{S esteja completamente funcional em [0,t]}) \tag{3.3}$$

Seja X a variável aleatória representando o tempo para falha de um sistema e seja F(t) a função de distribuição cumulativa (CDF) da variável X. Desta maneira, a confiabilidade do

sistema é dada por [84]:

$$R(t) = Pr(X > t)) = 1 - F(t) \tag{3.4}$$

Após o cálculo da confiabilidade, podemos calcular o MTTF de um sistema através da seguinte Equação [63]:

$$MTTF = \int_0^\infty R(t) \times dt \tag{3.5}$$

Levando-se em consideração a indisponibilidade (UA, em inglês Unavailability) ser definida por UA = 1 - A, e a Equação 3.1, a seguinte Equação é derivada:

$$MTTR = MTTF \times \frac{UA}{A} \tag{3.6}$$

Se um sistema não é reparável, a definição de $A(t)$ é equivalente à definição de confiabilidade, $R(t)$. Em geral, a disponibilidade e confiabilidade são conceitos relacionados, mas diferem no sentido de que o primeiro pode considerar a manutenção de componentes que falharam [22].

Segurança (*safety*) é definida em [5] como a ausência de consequências catastróficas para o usuário(s) e o meio ambiente. Já a mantenabilidade é definida em [22] através de um índice quantitativo para indicar a probabilidade de que um sistema poderá ser restaurado rapidamente para seu estado operacional. Da mesma maneira que confiabilidade, mantenabilidade é também um parâmetro probabilístico. Desta maneira, a mantenabilidade pode ser expressa como a probabilidade de restauração dentro de um período de tempo especificado.

Políticas de Filas

Neste trabalho, iremos mostrar as políticas de filas *FIFO*, *Custom Queuing* e *Priority Queuing*, que podem ser aplicadas a uma interface de um componente. Diferentemente da política *FIFO*, as políticas *Custom Queuing* e Priority Queuing suportam diferentes classes de tráfego em suas diferentes filas.

FIFO

A política *FIFO* não traz o conceito de classes de tráfego (ver Figura 3.2). Nesta política, os pacotes são transmitidos na mesma ordem de chegada à interface, ou seja, o primeiro que chega é o primeiro a sair. O *FIFO* é o método mais rápido de enfileiramento e pode ser o mais efetivo para enlaces de banda larga com pequeno retardo e congestionamento mínimo [100].

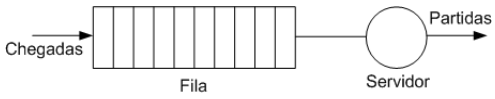


Figura 3.2: Política de fila FIFO [100]

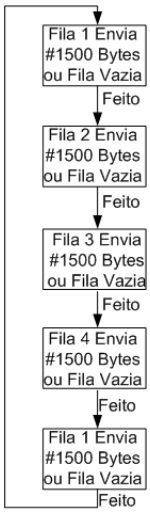


Figura 3.3: Política de fila Custom Queuing - Processamento com cinco filas [100]

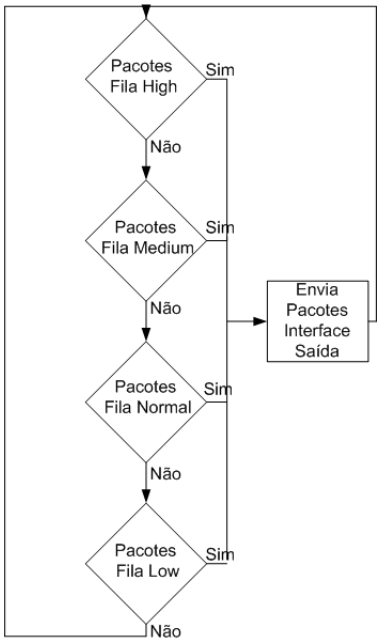


Figura 3.4: Política de fila Priority Queuing [100]

Custom Queuing

A política *Custom Queuing* (ver Figura 3.3) funciona da seguinte maneira: existem até 16 (dezesseis) possíveis filas para os dados, filas de 1 a 16, e uma fila utilizada pelo sistema, fila 0. As filas são tratadas uma a uma utilizando-se o algoritmo *round-robin* [98]. O administrador da rede poderá especificar a quantidade, em bytes, que será processada em cada fila, de maneira que uma fila poderá tratar vários pacotes por vez. A quantidade padrão de *bytes* que poderá ser ser varrida é de 1,500 bytes. O tamanho padrão destas filas é de 20 (Vinte) pacotes.

Priority Queuing

A política *Priority Queuing* (ver Figura 3.4) funciona da seguinte maneira: existem até 4 (quatro) filas, *High*, *Medium*, *Normal* e *Low*, que funcionam assegurando a maior prioridade para a fila *High*, depois para a *Medium* e então *Normal* e por fim *Low*. Os pacotes que estão na fila *Medium* apenas serão servidos depois que todos da fila *High* forem servidos; os pacotes na fila

Normal, quando os da fila *Medium* o forem e assim por diante na ordem de prioridade estabelecida. O tamanho padrão da fila *High* é de 20 pacotes, da fila *Medium* é de 40 pacotes, da fila *Normal* é de 60 pacotes e da fila *Low* é de 80 pacotes. Estes valores podem ser modificados.

3.3 Importância para Confiabilidade

Uma das finalidades da análise da confiabilidade de sistemas é identificar as fraquezas e quantificar o impacto de falhas de componentes. A abordagem conhecida como *importância para confiabilidade* (*Birnbaum importance* ou *B-Importance*) [56] é utilizada para este propósito. Valores de *importância para confiabilidade* são de grande importância no estabelecimento de priorizações de ações relacionadas a esforços de melhorias no projeto de um sistema ou através de sugestões para a operação do sistema de maneira mais eficiente.

Podemos dizer que a *importância para confiabilidade* de um componente i é igual à quantidade de aumento (no tempo t) na confiabilidade do sistema quando a confiabilidade do componente i é aumentada por uma unidade [56], podendo ser entendida como o impacto que um componente representa na confiabilidade do sistema. A *importância para confiabilidade* de um componente i é definida por [29]:

$$I_i^B(t) = \frac{\partial R_s(t)}{\partial R_i(t)} \tag{3.7}$$

onde I_i^B é a *importância para confiabilidade* (ou importância de Birnbaum) do componente i; $R_i(t)$ é a confiabilidade do componente i; R_s é a confiabilidade de todo o sistema e t é o tempo considerado para obtenção da confiabilidade.

3.4 Agrupamento Hierárquico

Uma abordagem hierárquica cria agrupamentos de objetos de maneira recursiva. Agrupamento hierárquico pode ser subdividido em métodos aglomerativos e divisivos [24]. Os métodos aglomerativos procedem por meio de sucessivas fusões dos n objetos, formando $n-1$, ..., $n-k$ agrupamentos, até reunir todos os objetos em um único grupo. Com relação aos métodos divisivos, a ideia é partir de um único grupo e por meio de divisões sucessivas obter vários outros subgrupos.

Com o objetivo de calcular a eficiência no resultado da configuração dos agrupamentos, alguns índices de validade são introduzidos. Algoritmos de agrupamento hierárquico utilizam os índices RMSSTD ou RSq [83]. Particularmente, este trabalho utiliza o índice RMSSTD, descrito na Equação 3.8, que é a variância dos agrupamentos, sendo utilizado para o cálculo da homogeneidade dos agrupamentos, de modo que o menor valor de RMSSTD significa uma melhor separação de agrupamentos.

$$RMSSTD = \sqrt[2]{\frac{\sum_{\substack{i=1\ldots n_{ag}\\ j=1\ldots d}} \sum_{k=1}^{n_{ij}} (x_k - \bar{x}_{ij})^2}{\sum_{\substack{i=1\ldots n_{ag}\\ j=1\ldots d}} (n_{ij} - 1)}} \tag{3.8}$$

onde:

- n_{ag}: É o número de agrupamentos;
- d: É o número de dimensões;
- $\bar{x}_{ij}$: É o valor esperado no agrupamento i e dimensão j;
- n_{ij}: É o número de elementos no agrupamento i, dimensão j.

3.5 Projeto de Experimento Fatorial

Outra técnica bem conhecida refere-se ao *projeto de experimento fatorial* [51]. Esta técnica habilita encontrar o efeito de cada fator dentro de um experimento, incluindo a interação entre eles sobre uma medida de interesse. Em um projeto que adota a técnica *projeto de experimento fatorial completo*, cada combinação possível entre seus diversos fatores e carga de trabalho é examinada. Em um dado sistema, que tem sua medida de interesse afetada por k fatores (componentes), e cada fator tem n níveis possíveis (opções de valores), o número de experimentos (combinações) deve ser igual a n^k. Este estudo abrangente normalmente consome substancial quantidade de tempo e de recursos [51].

[51] considera o projeto de experimento fatorial 2^k como uma abordagem viável, na qual apenas dois níveis são avaliados para cada fator. Como normalmente o efeito de um fator é unidirecinal, uma boa análise inicial poderá ser feita através da consideração no experimento dos níveis mínimo e máximo dos fatores em pauta. Após encontrar quais fatores são relevantes para a medida de interesse, a lista de fatores poderá ser reduzida substancialmente e torna-se viável para se tentar analisar mais níveis por fator.

Por fim, em algumas ocasiões, o número de experimentos exigidos para um projeto de experimento fatorial completo é muito grande. Isto pode acontecer se o número de fatores ou o número de níveis for grande. Nesses casos, um projeto de experimento fracional fatorial poderá também ser utilizado, o qual exige uma quantidade menor de experimentos. Por exemplo, um projeto 2^{k-p} permite a análise de k fatores, considerando 2 níveis, com apenas 2^{k-p} experimentos.

3.6 Acordo de Nível de Serviço

O acordo de nível de serviço (SLA) é um documento que define o nível de prestação de serviços entre uma empresa e um cliente [9]. O SLA define certos indicadores do nível de serviço (SLI) e configura limites para eles através de objetivos do nível de serviço (SLO). Este documento descreve os serviços a serem contratados e as taxas a serem alcançadas para o cumprimento de todos os compromissos acordados [97].

Um contrato do tipo SLA define classes operacionais que atendam às expectativas de clientes em termos de disponibilidade, desempenho, prioridades etc. Multas são impostas quando as expectativas não são alcançadas. O cumprimento destas expectativas está atrelado a indicadores automatizados para coleta e monitoramento dos itens do contrato. Estes indicadores devem incluir meios de segurança e auditoria que agreguem confiabilidade ao indicador. Eles devem ser disponibilizados tanto para o cliente quanto para o fornecedor do contrato. Os mecanismos de monitoramento dos indicadores podem estar implantados tanto no cliente quanto no fornecedor.

3.7 Conceitos de Modelagem

Nesta seção iremos introduzir os métodos de modelagem utilizados para representação de aspectos de infraestrutura, os quais incluem os modelos baseados em espaço de estados e modelos combinatoriais.

3.7.1 Modelos Baseados em Espaço de Estados

Modelos baseados em espaço de estados são do tipo que se utilizam de processos estocásticos para modelar os estados e as transições entre estados de um sistema em relação ao tempo. Modelos baseados em espaço de estados são geralmente capazes de capturar a interação e interdependência entre subsistemas e entre componentes dentro dos subsistemas, sendo ao mesmo tempo, relativamente rápidos para construir e fácil para verificar. Os modelos mais comumente utilizados são os listados abaixo.

Rede de Petri Estocástica

A teoria inicial das redes de Petri foi apresentada na tese de doutoramento *Kommunikation mit Automaten* do Dr. C.A. Petri em 1962 na faculdade de Matemática e Física da Universidade Darmstadf, na então Alemanha Ocidental [62]. Redes de Petri [77] é uma técnica de especificação de sistemas que possibilita uma representação matemática e possui mecanismos de análise poderosos, que permitem a verificação de propriedades e a verificação da corretude do sistema especificado. Usando-se redes de Petri pode-se modelar sistemas paralelos, concorrentes, assíncronos e não determinísticos[18]. As aplicações das Redes de Petri podem se dar em muitas áreas (sistemas de manufatura, desenvolvimento de software, sistemas administrativos, entre outros).

A introdução de conceitos de tempo em modelos de redes de Petri foi proposta por Merlin [74], Ramchandani [80] e Sifakis [89] sob distintos pontos-de-vista. Florin [28] tanto quanto Molloy [75] propuseram modelos de redes de Petri nos quais a temporização estocástica foi considerada. Estes trabalhos abriram a possibilidade de conexão da teoria de redes de Petri com a modelagem estocástica. Nos dias atuais, estes modelos, tanto quanto suas extensões são genericamente denominadas de redes de Petri estocásticas (SPN).

Existem diferentes maneiras de representar as SPNs. Este trabalho adota a definição mostrada em [32], na qual a SPN é uma 9-tupla, SPN = (P, T, I, O, H, Π, G, M_0, $Atts$) onde:

- $P = \{p_1, p_2, ..., p_n\}$ é o conjunto finito de lugares;
- $T = \{t_1, t_2, ..., t_n\}$ é o conjunto finito de transições;
- $I \in (\mathbb{N}^n \rightarrow \mathbb{N})^{nxm}$ é a matriz que representa a multiplicidade dos arcos de entrada (que podem ser dependentes de marcações), onde a entrada i_{jk} de I mostra a multiplicidade do arco (que pode ser dependente da marcação) do lugar p_j para a transição t_k. [$A \subseteq (PXT) \cup (TXP)$ - conjunto de arcos] ;
- $O \in (\mathbb{N}^n \rightarrow \mathbb{N})^{nxm}$ é a matriz que representa a multiplicidade dos arcos de saída (que podem ser dependentes de marcações). Entrada o_{jk} de O mostra a multiplicidade do arco (que pode ser dependente da marcação) da transição t_j para o lugar p_k;
- $H \in (\mathbb{N}^n \rightarrow \mathbb{N})^{nxm}$ é a matriz que representa os arcos inibidores (que podem ser dependentes de marcações). Entrada h_{jk} de H mostra a multiplicidade do arco inibidor (que pode ser dependente da marcação) do lugar p_j para a transição t_k;
- $\Pi \in \mathbb{N}^m$ é o vetor que designa um nível de prioridade para cada transição;
- $G \in (\mathbb{N}^n \rightarrow \{true, false\})^m$ é o vetor que associa uma condição de guarda relacionada à marcação do lugar a cada transição;
- $M_0 \in \mathbb{N}^n$ é o vetor que associa uma marcação inicial de cada lugar (estado inicial);
- $Atts = (Dist, W, Markdep, Policy, Concurrency)^m$ inclui o conjunto de atributos para transições, onde:
  - $Dist \in \mathbb{N}^m \rightarrow \mathscr{F}$. É uma função de distribuição de probabilidade associada ao tempo de uma transição (esta distribuição pode ser dependente de marcação) (o domínio de $\mathscr{F}$ é $[0, \infty)$);

- $W \in \mathbb{N}^m \rightarrow \mathbb{R}^+$ É uma função peso (esta distribuição pode ser dependente de marcação);
- $Markdep \in \{constant, enabdep\}$, onde a distribuição de probabilidade associada ao tempo de uma transição pode ser independente (constante) ou dependente da marcação (enabdep - a distribuição depende da condição de habilitação atual);
- $Policy \in \{prd, prs\}$. É a política de memória adotada pela transição (*prd* - *preemptive repeat different*, valor padrão, de significado idêntico à *race enabling policy*; *prs* - preemptive resume, corresponde ao *age memory policy*);
- $Concurrency \in \{ss, is\}$ É o grau de concorrência das transições, onde *ss* representa a semântica de *single-server* e *is* representa a semântica de *infinity server*.

Em muitas circunstâncias pode ser adequado representar a marcação inicial como um mapeamento do conjunto de lugares para os números naturais (m_0: P $\rightarrow \mathbb{N}$), onde $m_0(p_i)$ denota a marcação inicial do lugar p_i. $m(p_i)$ denota uma marcação alcançável do lugar p_i. Neste trabalho, a notação $\#p_i$ tem sido adotado para representar $m(p_i)$.

Com relação às políticas de memória, duas alternativas básicas são possíveis na mudança de marcação [68, 69]

- *continue*: as temporizações associadas às transições mantêm os valores presentes que continuarão a serem decrementados nas próximas marcações;
- *restart*: as temporizações associadas às transições são reiniciadas, ou seja, os valores presentes são descartados e novos valores serão gerados quando necessário.

Diferentes comportamentos provenientes dos sistemas reais definem diferentes combinações dos mecanismos *continue* e *restart* para as transições que não puderam disparar:

- *Resampling*: a cada disparo de toda e qualquer transição do modelo, todos os temporizadores existentes são reiniciados (Restart) e, sendo assim, não há memória. O temporizador de cada transição será reiniciado sempre que a transição tornar-se habilitada;

- *Enabling memory*: a cada disparo de transição, os temporizadores das transições que estavam desabilitadas são reiniciados, enquanto que os temporizadores das transições que estavam habilitadas mantêm o valor atual (Continue). Assim que estas transições tornarem-se habilitadas novamente, seus temporizadores continuam do ponto em que foram parados. Uma variável (enabling memory variable) mede o tempo que a transição passou habilitada desde o último instante de tempo em que ela se tornou habilitada;

- *Age memory*: após cada disparo, os temporizadores de todas as transições mantêm seus valores atuais (Continue). Uma memória do passado é mantida por uma variável (*age memory variable*) associada a cada transição temporizada. Esta variável contabiliza o tempo gasto na atividade modelada pela transição, medindo o tempo cumulativo de habilitação, desde o instante do seu último disparo.

Quanto à semântica de temporização, as transições temporizadas com grau de habilitação maior do que um (1) podem ser caracterizadas da seguinte forma: *single server*, *multiple server* e *infinite server* [68].

- *single server*: as marcações são processadas serialmente. Após o primeiro disparo da transição temporizada, o temporizador é reiniciado como se a transição temporizada tivesse sido habilitada novamente;

- *multiple server*: as marcações são processadas com um grau máximo K de paralelismo. Caso o grau de habilitação seja maior do que K, não será criado nenhum novo temporizador para processar o tempo para o novo disparo até que o grau de habilitação tenha diminuído abaixo de K;

- *infinite server*: o valor de K é infinito, todas as marcações são processadas em paralelo e as temporizações associadas são decrementadas a zero em paralelo.

Por sua vez, nos modelos SPN, as transições são disparadas obedecendo à semântica interleaving de ações [68]. Essa semântica define que as transições são disparadas uma a uma, mesmo que o estado compreenda transições imediatas não conflitantes.

Conjunto das Marcações Alcançáveis e Grafo de Alcançabilidade

O disparo de uma transição altera a marcação da rede de acordo com sua marcação atual e correspondente estrutura. O conjunto com todas as marcações que podem ser alcançadas a partir das possíveis sequências disparáveis de uma rede de Petri, a partir de uma marcação inicial, é denominado conjunto de marcações alcançáveis (RS). Já o grafo de alcançabilidade (RG) de uma rede de Petri é um grafo no qual seus vértices representam as marcações alcançáveis de RS e os arcos que interconectam estes vértices representam o disparo de cada transição.

Definição 1: O conjunto de marcações alcançáveis de uma rede de Petri qualquer, que possua uma marcação inicial M_0, é representado por RS(M_0) e é definido como o menor conjunto de marcações onde:

- $M_0 \in \mathrm{RS}(M_0)$;
- $\mathrm{M'} \in \mathrm{RS}(M_0) \wedge \exists t_k \in \mathrm{T} : \mathrm{M'}[t_k > \mathrm{M''} \Rightarrow \mathrm{M''} \in \mathrm{RS}(M_0)$.

Definição 2: O grafo das marcações alcançáveis ou grafo de alcançabilidade (RG) de uma rede de Petri com uma marcação inicial M_0 é um multigrafo dirigido cujos nós é o conjunto RS. Seja M_0 o nó inicial do grafo e A é seu conjunto de arcos que é definido como segue:

- $\mathrm{A} \subseteq \mathrm{RS} \times \mathrm{RS} \times \mathrm{T}$;

- < M', M", t > $\in$ A $\Leftrightarrow$ M'[t > M".

Utilizamos a notação < M', M", t > para indicar que um arco dirigido conecta o nó correspondente à marcação M' para o nó correspondente à marcação M" e um rótulo t é associado com o arco.

Os modelos SPN podem ser usados para análise de desempenho e dependabilidade de sistemas, visto que permitem a descrição das atividades através de grafos de alcançabilidade correspondentes. Esses grafos podem ser convertidos em modelos Markovianos, que são utilizados para avaliação quantitativa do sistema analisado.

Devido à propriedade de *memoryless*, ausência de memória, da distribuição exponencial associada aos atrasos de disparo das transições temporizadas, o grafo de alcançabilidade de uma SPN limitada é isomórfico a uma cadeia de Markov finita [77]. Uma CTMC associada com a SPN marcada é obtida através da aplicação das seguintes regras [68]:

1. O espaço de estados da CTMC S = $\{s_i\}$ corresponde ao conjunto das marcações alcançáveis RS(M_0) da rede de Petri associada com a SPN ($M_i \leftrightarrow s_i$);

2. A taxa de transição do estado s_i (correspondente à marcação M_i) para o estado s_j (M_j) é obtida como a soma das taxas de disparo das transições que são habilitadas em M_i, cujos disparos levaram à marcação M_j.

Aproximação por Fases

Muitos estudos proporcionam meios para a adaptação de modelos Markovianos (aproximação) [20] a fim de permitir avaliações numéricas visando uma negociação do custo da complexidade. Testes do tipo *Goodness-of-fit* podem ser aplicados para encontrar a distribuição que melhor se adequa para a amostra de dados. Infelizmente, a distribuição encontrada pode ser muito com-

plexa e quando forem consideradas simultaneamente as variáveis estocásticas que são levadas em consideração, poderia acarretar um processo estocástico bastante complexo.

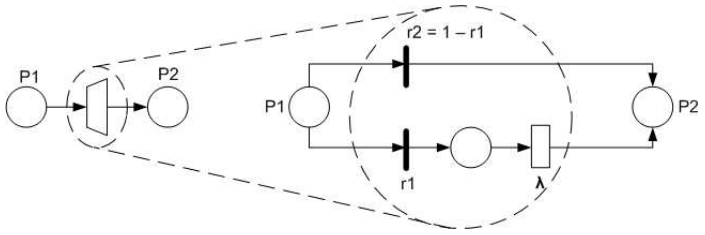


Figura 3.5: Modelo hiperexponencial

Um abordagem comum para a representação das distribuições genéricas de variáveis aleatórias, incluindo aquelas obtidas através de medições diretas (distribuições empíricas), é a representação destas variáveis através da combinação de variáveis aleatórias exponenciais (distribuição expolinomial) [7, 32]. Um método que considera distribuição expolinomial de variáveis aleatórias é baseada em *moment-matching*. O *moment-matching* apresentado em [20] leva em consideração que distribuições Hipoexponencial e de Erlang possuem um atraso médio (μ) maior que o desvio padrão σ ($\mu > \sigma$) e distribuição Hiperexponencial possui $\mu < \sigma$, com o objetivo de representar uma atividade com uma distribuição genérica de atraso como uma sub-rede tipo Erlang ou Hiperexponencial. Pode-se notar que nos casos em que estas distribuições têm $\mu = \sigma$ são equivalentes a uma distribuição exponencial com parâmetro igual a $\frac{1}{\mu}$. Por esta razão, de acordo com o coeficiente de variação (CV) associado ao atraso (*delay*) da atividade, um modelo de sub-rede apropriado pode ser escolhido. Para cada sub-rede (ver Figuras 3.5 e 3.6), um conjunto de parâmetros deve ser configurado para igualar os dois primeiros momentos da distribuição. Em outras palavras, a distribuição do atraso associado (pode também ter sido obtida através de um processo de medição) com a atividade em questão é igualada ao primeiro e segundo momentos da distribuição expolinomial. De acordo com este método, uma atividade com $\mu < \sigma$ é aproximada a uma distribuição Hiperexponencial de duas fases com os parâmetros:

$$r1 = \frac{2\mu^2}{(\mu^2 + \sigma^2)} \tag{3.9}$$

$$r2 = 1 - r1 \tag{3.10}$$

$$\lambda = \frac{2\mu}{\mu^2 + \sigma^2} \tag{3.11}$$

onde λ é a taxa associada à fase 1, r1 é a probabilidade relacionada a esta fase e r2 é a probabilidade designada para a fase 2. Neste modelo, a taxa designada para a fase 2 é assumida ser infinita , isto é, o atraso médio é zero.

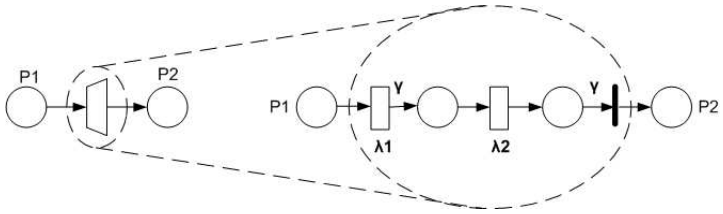


Figura 3.6: Modelo hipoexponencial

Atividades com CVs menor do que um (1) podem ser mapeadas tanto para sub-redes do tipo Hipoexponencial ou Erlang. Se $\frac{\mu}{\sigma} \notin \mathbb{N}, \frac{\mu}{\sigma} \neq 1, (\mu, \sigma \neq 0)$, a atividade é representada por uma distribuição Hipoexponencial com parâmetros $\lambda 1, \lambda 2$ (taxa exponenciais); e γ, o inteiro representando o número de fases com taxa igual a $\lambda 2$, ao passo que o número de fases com taxa igual a $\lambda 1$ é um. Em outras palavras, a sub-rede correspondente é composta de duas transições exponenciais e uma transição imediata. O atraso médio designado para a transição exponencial t1 é igual a $\mu 1 (\lambda 1 = 1/\mu 1)$ e o respectivo atraso médio designado para a transição exponencial t2 é $\mu 2 (\lambda 2 = 1/\mu 2)$. O valor de γ é considerado como a multiplicidade designada para o arco de saída da transição t1, assim como o valor da multiplicidade do arco de entrada da transição imediata t3 (ver Figura 3.6). Estes parâmetros são calculados pela seguinte expressão:

$$(\mu/\sigma)^2 - 1 \leq \gamma < (\mu/\sigma)^2 \tag{3.12}$$

$$\lambda 1 = 1/\mu 1 \tag{3.13}$$

$$\lambda 2 = 1/\mu 2 \tag{3.14}$$

onde

$$\mu 1 = \frac{\mu \pm \sqrt{\gamma(\gamma+1)\sigma^2 - \gamma\mu^2}}{\gamma+1} \tag{3.15}$$

$$\mu 2 = \frac{\gamma\mu \pm \sqrt{\gamma(\gamma+1)\sigma^2 - \gamma\mu^2}}{\gamma+1} \tag{3.16}$$

Quando o inverso do coeficiente de variação é um número inteiro e diferente de um, se $\frac{\mu}{\sigma} \in \mathbb{N}$ e $\frac{\mu}{\sigma} \neq 1$ (considerando $\mu, \sigma \neq 0$), a atividade é representada por uma sub-rede tipo Erlang com dois parâmetros, $\gamma = (\frac{\mu}{\sigma})^2$, um inteiro representando o número de fases desta distribuição, e $\mu 1 = \mu/\gamma$, aonde $\mu 1 (1/\lambda 1)$, o valor do atraso médio de cada fase. Erlang é um caso particular de Hipoexponencial, na qual a taxa de cada fase individual possui o mesmo valor.

O principal problema com relação a modelos baseados em espaço de estados é conhecido como explosão do espaço de estados [3], quando o tamanho do espaço de estados cresce muito e os recursos de memória não são suficientes para resolver o modelo. Realmente, em tais casos, simulação pode ser o tratamento mais adequado.

Cadeias de Markov

Um processo estocástico é definido como uma família de variáveis aleatórias $\{X_t : t \in T\}$ onde cada variável aleatória X_t é indexada pelo parâmetro $t \in T$, o qual é normalmente denominado de parâmetro de tempo se $T \subseteq R_+ = [0,\infty)$, isto é, T está no conjunto de números reais não negativos. O conjunto de todos os possíveis valores de X_t (para cada $t \in T$) é conhecido como o espaço de estados S do processo estocástico [7].

Se o espaço de estados de um processo estocástico é discreto, então ele é denominado de *processo de estado-discreto*, frequentemente referido como um *chain*. Alternativamente, se o espaço de estados é contínuo, então temos um *processo de estado-contínuo*. De maneira similar, se o conjunto T é discreto, então temos um processo de *tempo-discreto*; do contrário, temos um processo de *tempo-contínuo*.

Seja $\Pr\{k\}$ a probabilidade de um dado evento k ocorrer. Um processo de Markov é um processo estocástico em que $\Pr\{X_{t_{n+1}} \leq s_{i+1}\}$ depende apenas do último valor X_{t_n}, para todo $t_{n+1} > t_n > t_{n-1} ... > t_0 = 0$ e todo $s_i \in S$. Esta é a chamada propriedade de Markov [48], que significa que a evolução futura do processo de Markov é totalmente descrita pelo estado atual e é independente de estados passados (ausência de memória) [48]. Os processos de Markov devem seu nome ao matemático russo A.A. Markov que estudou profundamente esta classe de processos estocásticos.

Um processo Markoviano é dito ser uma cadeia de Markov quando as variáveis aleatórias X_t estão definidas em um espaço de estados discreto. Neste trabalho, consideramos apenas os processos Markovianos definidos sobre um espaço de estados discreto, conhecidos como cadeias de Markov (MC). Uma cadeia de Markov é descrita por uma sequência de variáveis aleatórias discretas, X_{t_n}, em que t_n pode assumir valores discretos ou contínuos.

A Figura 3.7 mostra um modelo simples de Markov que representa o comportamento diário das condições meteorológicas em uma cidade. Existe apenas três estados, de maneira que o

espaço de estados é definido como: S = {Ensolarado, Chuvoso, Nublado}. De acordo com a propriedade de Markov, dado que em um momento t_n as condições meteorológicas correspondem ao estado ensolarado, a probabilidade de transição para o estado chuvoso no momento t_{n+1} não depedende sobre quais estados foram visitados antes. O mesmo raciocínio é verdade para todos os outros estados e transições. De maneira semelhante, o tempo gasto no estado atual não influencia a probabilidade de transição.

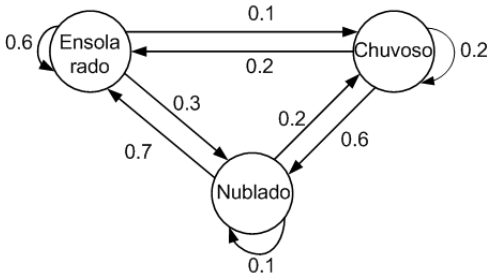


Figura 3.7: Modelo simples de Markov

As cadeias de Markov de tempo contínuo são chamadas CTMC e as de tempo discreto são chamadas DTMC [7, 12, 20, 102]. Nas DMTCs, as transições entre os estados da MC ocorrem apenas em pontos discretos de tempo. Já nas CTMCs, estas transições ocorrem em qualquer instante de tempo. CTMC e DTMC proporcionam diferentes abordagens de modelagens, embora relacionadas, cada uma tendo seu próprio domínio de aplicação. A distribuição geométrica para o caso das DTMCs e a exponencial para o caso das CTMCs apresentam a propriedade de *ausência de memória*.

Segundo [7], quando tratando com DTMC, o processo de transição de um-passo (em inglês *one-step*) do estado i para o estado j no tempo n, $p_{ij}(n)$ possui uma função de probabilidade de massa (pmf) que pode ser escrita com a seguinte notação:

$$p_{ij}(n) = P(X_{n+1} = s_{n+1} = j | X_n = i) \tag{3.17}$$

Dado que para cada estado $i \in S$, $\sum_j p_{ij} = 1$ e $0 \leq p_{ij} \leq 1$, uma matriz de transição estocás-

tica P é utilizada para sumarizar todas a probabilidades de transições de um DTMC. Para o DTMC da Figura 3.7, considerando um espaço de estados equivalente S = {0, 1, 2}, a matriz P é:

$$\mathrm{P}=\begin{pmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.7 & 0.1 & 0.2 \\ 0.2 & 0.6 & 0.2 \end{pmatrix}$$

A matriz de transição tem um papel muito importante no cálculo do vetor de probabilidade de estado $\underline{\pi}$. Este vetor detém a informação sobre a probabilidade de um sistema estar em um dado estado, tanto em um número específico de n passos (ver Equação 3.18), ou em estado-estacionário, quando $n \longrightarrow \infty$ (ver Equação 3.19). É importante enfatizar a necessidade do prévio conhecimento das probabilidades do estado inicial, $\underline{\pi}(0)$, em caso de probabilidades dependentes do tempo, ao passo que probabilidades em estado estacionário não dependem da condição inicial do sistema. A partir do vetor de probabilidade de estado, quase todas as principais métricas podem ser derivadas, dependendo do sistema que é representado.

$$\underline{\pi}(n) = \underline{\pi}(0)P^n \tag{3.18}$$

$$\underline{\pi} = \underline{\pi}P \tag{3.19}$$

Para o caso de modelos em CTMC, a matriz de transição de estados é referenciada como gerador infinitesimal, desde que as transições ocorrem com taxa, ao invés de probabilidade, devido à natureza deste tipo de modelo. A matriz Q é composta pelos componentes q_{ii} e q_{ij}, aonde $i \neq j$ e $\sum q_{ij}$ = -q_{ii}. Utilizando o modelo de disponibilidade mostrado na Figura 3.8, considerando um espaço de estados S = {ATIVO, INATIVO, REPARO} = {0, 1, 2}, a matriz

Q é dada por:

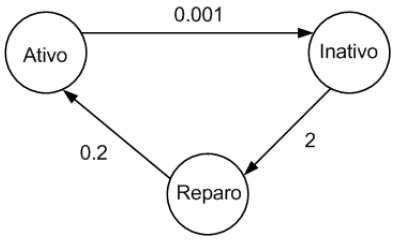


Figura 3.8: CTMC simples

$$\mathrm{Q} = \begin{pmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{pmatrix} = \begin{pmatrix} -0.001 & 0.001 & 0.0 \\ 0.0 & -2 & 2 \\ 0.2 & 0.0 & -0.2 \end{pmatrix}$$

Equação 3.20 e o sistema de Equações 3.21 descrevem, respectivamente, o cálculo do vetor de probabilidades transiente (dependente do tempo) e estacionário.

$$\underline{\pi}'(t) = \underline{\pi}(t) \times Q,\ dado\ \underline{\pi}(0) \tag{3.20}$$

$$\underline{\pi} \times Q = 0, \sum_{i \in S} \pi_i = 1 \tag{3.21}$$

Explicações detalhadas de como obter estas equações podem ser encontradas em [7].

3.7.2 Modelos Combinatoriais

Modelos combinatoriais representam os eventos de falha do sistema através da combinação de cada estado de seus componentes individuais. São normalmente utilizados na análise de confiabilidade para sistemas não reparáveis. Entretanto, estes métodos podem também ser aplicados para análise da disponibilidade para sistemas reparáveis. Iremos mostrar um tipo de modelo combinatorial denominado de Diagrama de Blocos de Confiabilidade.

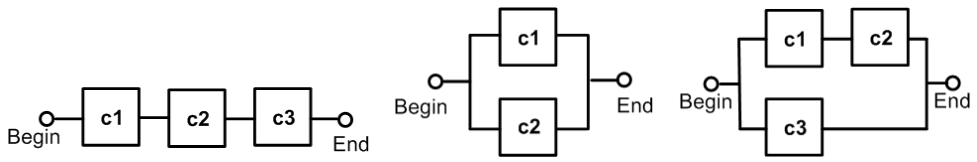
Diagramas de Blocos de Confiabilidade

O diagrama de blocos de confiabilidade (RBD) [65, 76, 84, 95, 103] é uma das técnicas mais usadas para a análise de confiabilidade de sistemas. É considerado um modelo tipo combinatorial, não baseado em espaço de estados. Embora RBD tenha sido inicialmente proposto como um modelo para o cálculo da confiabilidade, ele pode ser utilizado para o cálculo de outras métricas de dependabilidade, tais como disponibilidade e mantenabilidade.

Esta técnica não necessariamente representa como os componentes estão fisicamente conectados no sistema. É utilizada para representar a estrutura lógica de um sistema com relação à forma de como a confiabilidade de seus componentes afeta a confiabilidade do sistema.

Por outro lado, se o sistema executa mais do que uma função (operação), deve ser definido um modelo para a representação de cada uma de suas funções. Desta maneira, um sistema sendo operacional para uma determinada função, poderá não estar operacional para uma outra função.

Os componentes são representados por blocos e seus nomes podem ser fornecidos no próprio bloco. Em um modelo de diagrama de blocos de confiabilidade, os componentes são representados como blocos utilizando as seguinte composições: série, paralelo, ponte, blocos *k-out-of-n*, ou ainda, uma combinação das abordagens anteriores [56].



a) Série b) Paralelo c) Série-Paralelo

Figura 3.9: Estruturas básicas

A Figura 3.9 mostra três exemplos, em que os blocos são organizados em série (Figura 3.6(a)), paralelo (Figura 3.6(b)) e série/paralelo (Figura 3.6(c)).

Inicialmente, assumindo uma estrutura com n componentes em série, a sua confiabilidade (disponibilidade) [56] é dada por:

$$P_s(t) = \Pi_{i=1}^{n} P_i(t) \tag{3.22}$$

onde $P_i(t)$ é a confiabilidade ou a disponibilidade de um componente i. O funcionamento de cada componente é essencial para o funcionamento do sistema. Caso um único componente falhe, todo o sistema irá parar [56].

Para uma estrutura em paralelo, ao menos um dos componentes devem estar operacional para que todo o sistema esteja operacional. Levando-se em consideração um sistema com n componentes, a sua confiabilidade (disponibilidade) é dada por:

$$P_p(t) = 1 - \Pi_{i=1}^{n} (1 - P_i(t)) \tag{3.23}$$

onde $P_i(t)$ é a confiabilidade ou disponibilidade de um componente i.

Com o objetivo de calcular a confiabilidade (disponibilidade) de uma estrutura do tipo série/paralelo, os resultados obtidos através dos componentes em série devem ser combinados e colocados em equações referentes aos componentes em paralelo. Para outros exemplos e equações de forma fechada, o leitor deve fazer referência ao trabalho mostrado em [56].

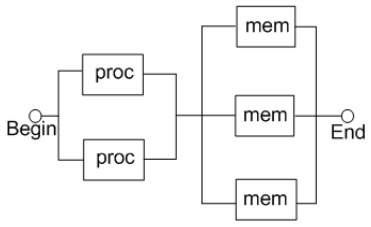


Figura 3.10: Modelo RBD

Por outro lado, a Figura 3.10 mostra um exemplo de RBD que modela um multiprocessador tolerante a falhas com múltiplos módulos de memória compartilhados, garantindo que o sistema

seja operacional se ao menos um processador e um módulo de memória esteja operacional.

Funções Estruturais RBD

Funções estruturais RBD são funções matemáticas discretas que relacionam o estado de funcionamento dos componentes de um determinado sistema com o estado de funcionamento do sistema. Seja x_i a variável que representa o estado do componente i para $1 \leq i \leq n$. Onde n é o número de componentes do sistema. Então o estado x_i do componente i é dado por:

$$x_i = \begin{cases} 1, & \text{se o componente } i \text{ está funcionando} \\ 0, & \text{se o componente } i \text{ está falho} \end{cases}$$

O vetor x = (x_1, x_2, ... x_n) representa o estado de todos os componentes. O estado do sistema é determinado pelos estados dos componentes. Seja Φ a variável que representa o estado do sistema como um todo.

$$\Phi = \begin{cases} 1, & \text{se o sistema } i \text{ está funcionando} \\ 0, & \text{se o sistema } i \text{ está falho} \end{cases}$$

Se o estado de todos os componentes é conhecido, então o estado do sistema também é conhecido. Assim, o estado do sistema é uma função determinística do estado de todos os componentes.

$$\Phi = \Phi(x) = \Phi(x_1, x_2, ...x_n) \tag{3.24}$$

onde Φ(x) é a função estrutural do sistema. As regras de formação das funções estruturais são mostradas a seguir:

Componentes em Série

Sejam n componentes x_1, x_2, ..., x_n em série, a função estrutural Φ desses componentes é representada por:

$$\Phi(x) = \Pi_{i=1}^{n} x_i = min\{x_1, x_2, ...x_n\} \tag{3.25}$$

Componentes em Paralelo

Sejam n componentes x_1, x_2, ..., x_n em paralelo, a função estrutural Φ desses componentes é representada por:

$$\Phi(x) = 1 - \Pi_{i=1}^{n} (1 - x_i) = max\{x_1, x_2, ...x_n\} \tag{3.26}$$

O leitor pode verificar mais detalhes sobre funções estruturais em [56].

3.8 Considerações Finais

O propósito deste capítulo foi o de rever conceitos importantes que serão utilizados no restante deste trabalho. Referências adicionais estão disponíveis para o leitor. Inicialmente, este capítulo apresentou conceitos de redes convergentes e suas exigências de projeto. Foram também definidos os aspectos de negócios (Custos de Infraestrutura, Modelos de Receitas e Modelos de Multas) e de infraestrutura (Dependabilidade e Políticas de Filas) que são levados em consideração na escolha do melhor projeto de uma infraestrutura de redes convergentes. As técnicas utilizadas na metodologia proposta tais como: Importância para Confiabilidade, Projeto de Ex-

perimento Fatorial e Agrupamento Hierárquico, junto com a definição de Acordo de Nível de Serviço, foram expostas neste capítulo. Por fim, foram também apresentados os mecanismos de modelagem utilizados neste trabalho, modelos não baseados em espaço de estados (RBD) e modelos baseados em espaço de estados (SPN e CTMC).

CAPÍTULO 4

Modelos de Infraestrutura de Redes Convergentes

Este capítulo apresenta modelos de base do tipo SPN, CTMC e RBD adotados para representar e proporcionar informações com relação aos aspectos de infraestrutura de redes convergentes. Estes modelos são suficientemente genéricos para representar uma grande variedade de mecanismos encontrados em projetos reais. Estas informações podem também oferecer suporte aos processos de tomadas de decisões por parte de administradores e projetistas com relação a questões de planejamento de capacidade.

Considerações sobre a seleção de modelos e a estratégia de modelagem adotada são inicialmente apresentadas. Então, são mostrados os modelos de dependabilidade correspondentes a um componente simples, às partes funcionais de um componente, a mecanismos de redundância de grande relevância e às principais estruturas básicas encontradas em redes convergentes. Com relação aos mecanismos de redundância, são apresentados modelos correspondentes às técnicas de *warm standby* e *cold standby*[56]. No que diz respeito às estruturas básicas existentes em redes convergentes, são mostrados modelos referentes às estruturas tipo série, paralelo e série/paralelo. Por fim, são expostos modelos de desempenho através da representação das políticas de filas *FIFO*, *Custom Queuing* e *Priority Queuing* aplicadas sobre uma interface de um componente.

4.1 Seleção de Modelos

Uma grande variedade de diferentes modelos analíticos está disponível. Cada tipo de modelo possui seus pontos fortes e fracos em termos de acessibilidade, facilidade de construção, eficácia, precisão de algoritmos de solução e facilidade de acesso às ferramentas de software [84].

Redes de Petri estocásticas [32] e cadeias de Markov [102] proporcionam grande flexibilidade para a modelagem de aspectos de desempenho, dependabilidade, além da combinação de desempenho e dependabilidade (performabilidade). Entetanto, estes modelos frequentemente se deparam com problemas relacionados ao tamanho do espaço de estados (denominados de problemas de *largeness*) para sistemas computacionais com grande número de componentes, tornando inviável, em alguns casos, a aplicação destes modelos.

Por outro lado, modelos combinatoriais são simples, fáceis de serem entendidos e seus métodos de solução têm sido extensivamente estudados [84]. Em redes convergentes, como os componentes são distribuídos geograficamente, o comportamento de falha destes componentes são considerados independentes entre si. Os modelos combinatoriais podem explorar esta independência para permitir uma representação eficiente, evitando problemas de crescimento demasiado do espaço de estados, enfrentado pelos modelos baseados em estados. Entretanto, eles não podem facilmente considerar a dependência entre os componentes e não são adequados à avaliação de desempenho [84].

A familiaridade com diferentes tipos de modelos e o tipo de medida que é necessária pode facilitar a escolha do modelo que melhor se adequa a um determinado sistema. É também possível a utilização de diferentes tipos de modelos de maneira hierárquica em diferentes níveis do sistema.

4.2 Estratégia de Modelagem

Este trabalho adota uma estratégia de modelagem hierárquica [7] com relação aos modelos de infraestrutura, considerando as vantagens e possibilidades tanto de modelos combinatoriais quanto de modelos baseados em espaço de estados. Tal abordagem é adotada quando ocorre o problema da explosão de espaço de estados[3].

Como exemplo, modelos baseados em estados serão adotados para representar algum subsistema (parte do sistema) caso ocorram dependências dinâmicas entre os componentes em seu interior. Os resultados dos subsistemas podem ser utilizados como parâmetros para o modelo de nível mais alto (sistema), representado por um modelo combinatorial. É importante afirmar que os subsistemas não necessitam adotar o mesmo modelo. Alguns subsistemas podem utilizar modelos baseados em estados e outros, modelos combinatoriais.

4.3 Modelos de Dependabilidade

Para a análise de dependabilidade, modelos SPN, CTMC e RBD são utilizados. A partir destes modelos, podemos tanto analisar o comportamento de diferentes arquiteturas/componentes, em termos de confiabilidade e disponibilidade, como obter métricas que serão utilizadas como parâmetros de entrada para os modelos orientados a negócios mostrados no Capítulo 5. Em todo nosso trabalho, P{ } denota a função probabilidade.

4.3.1 Componente Simples

Figura 4.1 representa o modelo SPN de disponibilidade/confiabilidade de um único componente. Este componente é caracterizado pela ausência de redundância, isto é, o componente poderá estar apenas em dois estados, ativo (funcionando) ou inativo (defeito). Ele possui dois parâmetros, a saber: MTTF e MTTR. Transição X_MTTR (neste trabalho, rótulo *X* é instanciado de acordo com o nome do componente) possui a seguinte expressão correspondente ao atraso (parâmetro *delay*): IF #RelFlag=1: 10^{n2} ELSE Y; onde Y representa o parâmetro MTTR do componente. O parâmetro MTTF representa o *atraso* associado à transição X_MTTF. Ambas as transições possuem uma distribuição exponencial (exp) e uma semântica de disparo do tipo *single server* (ss).

O lugar RelFlag modela a avaliação de disponibilidade/confiabilidade. Se o número de marcas (#) no lugar RelFlag é igual a 0 (#RelFlag=0), a avaliação refere-se à disponibilidade. De outra maneira, a avaliação refere-se à confiabilidade. Essa abordagem nos permite parametrizar o modelo e auxiliar a avaliação do sistema, considerando a reparação (ex., disponibilidade) ou não (ex., confiabilidade).

4.3.2 Partes Funcionais de um Componente

Diferentemente da abordagem anterior, um componente pode ser representado por diversas partes funcionais. Diferentes mecanismos de modelagem podem ser utilizados. Como estamos considerando partes funcionais que são independentes em relação à falha e ao reparo, adotamos modelos combinatoriais do tipo RBD.

Por exemplo, vamos considerar um roteador de alto desempenho, constituído pelas seguintes partes funcionais: chassis/placa mãe, fontes de potência, placa de processamento/sistema ope-

[2]Neste trabalho foi adotado o valor $n = 9$ para representar um *atraso* ilimitado.

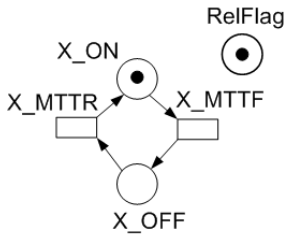


Figura 4.1: Modelo disponibilidade/confiabilidade de um único componente

Figura 4.2: Partes funcionais de um roteador

racional e placas de interface. Para que o sistema funcione, é necessário que o chassis/placa-mãe, a placa processadora/sistema operacional e as placa de interfaces estejam funcionando, assim como pelo menos uma das duas fontes de potência esteja operacional (ver Figura 4.2). O detalhamento de cada uma das partes irá depender do nível de abstração desejado.

4.3.3 Mecanismos de Redundância

Um tipo de redundância bastante utilizado em projetos de sistemas computacionais é denominado de redundância *standby* ou em *espera* [56]. Neste caso, apenas um único componente está ativo e operacional. Um ou mais componentes adicionais podem ser colocados no sistema, mas em condição de *espera*. Um mecanismo de detecção e de comutação é utilizado para monitorar o componente ativo e operacional. Quando este componente principal falhar, um dos componentes em *espera* é levado ao estado ativo e operacional através do mecanismo de comutação. Nesta seção, iremos detalhar modelos para os seguintes tipos de redundância em *espera*: *cold standby* e *warm standby*.

Cold Standby - Modelos

Neste mecanismo, os componentes em *espera* não poderão falhar enquanto estiverem nesta condição [56]. Por sua vez, o componente ativo possui uma taxa de falha constante igual a λ. Esta abordagem é caracterizada pela detecção de falhas junto com um mecanismo de ativação. Devido à interação entre os componentes e dependência de considerações de tempo, são utilizados modelos baseados em espaço de estados, particularmente SPN e CTMC.

Modelo de Dependabilidade - SPN

O modelo correspondente é mostrado na Figura 4.3. Lugares X_ON, X_OFF, X_Sp_ON e X_Sp_OFF representam os estados de atividade e inatividade dos seguintes componentes: principal e em espera. Caso ocorra uma falha do componente principal, a transição X_ACTSp é habilitada. Seu atraso (*delay*) representa o tempo de detecção da falha e de ativação do componente em espera. Este período é denominado de MTTA. Transição imediata X_DCTSp representa o retorno ao estado normal de operação após uma falha.

Transições X_MTTR, X_Sp_MTTR, X_ACTSp possuem a seguinte expressão correspondente ao atraso: IF #RelFlag = 1: 10^n ELSE Y; onde Y representa ou o valor de MTTA da transição X_ACTSp ou o parâmetro MTTR do componente correspondente. O parâmetro MTTF é atribuído às transições X_MTTF e X_Sp_MTTF. É importante observar que os valores de MTTF e MTTR associados ao componente principal podem ser diferente dos valores associados ao componente em espera. As transições temporizadas têm tempo exponencialmente distribuído (exp) e semântica de disparo do tipo *single server* (ss). Conforme previamente descrito, o lugar RelFlag permite a avaliação de confiabilidade ou disponibilidade.

Este modelo permite o cálculo da disponibilidade do sistema através da expressão: P{(#X_ON = 1) OR (#X_Sp_ON = 1)}. O cálculo da confiabilidade do sistema pode ser calculado pela

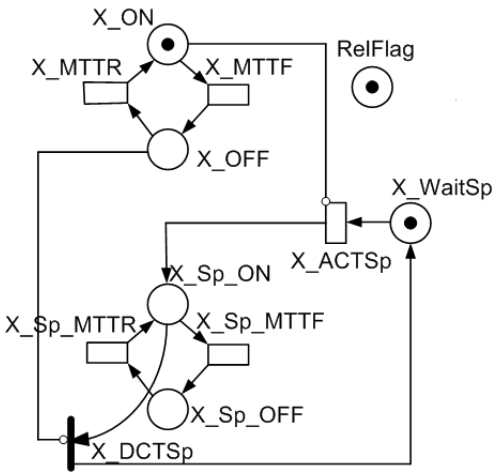


Figura 4.3: Modelo *Cold Standby* - SPN

seguinte expressão: P{(#X_ON = 1)}.

Modelo de Disponibilidade - CTMC

A Figura 4.4 representa dois componentes, principal e de reposição, em uma abordagem de redundância chamada *cold standby* utilizando cadeia de Markov em tempo contínuo (CTMC). As taxas de falha e de reparo de cada componente são representadas por λ_X e μ_X, onde $X \in \{P,R\}$. A taxa α_R é o inverso do tempo onde está incluído tanto o tempo de detecção da falha quanto o tempo de comutação do componente de reposição. Para efeito de simplicidade, foram feitas algumas suposições. A primeira é que existe uma prioridade com relação ao reparo do componente principal em relação ao componente de reposição. Outra suposição é que no tempo de reparo de cada componente podem estar incluídos quaisquer tempos adicionais a fim de que o sistema retorne ao seu estado normal de operação. Desta forma, a taxa de reparo, que é o inverso do tempo de reparo, incorpora esta consideração. Por fim, uma última suposição é que durante o processo de comutação nenhum tipo de falha ou reparo poderá ocorrer.

Neste modelo, o estado de atividade é denotado pelo rótulo U e o estado de inatividade ou falha é representado pelo rótulo D. O rótulo W (do inglês WAIT) representa uma condição de

Tabela 4.1: Estados do modelo CTMC - *Cold Standby*

Estado	Descrição
UW	O Sistema está ativo (UP)
DW	Estado de Inatividade (DOWN), Componente Principal Falhou, Processo de Inicialização
DU	Sistema está ativo (UP), Componente Principal falhou
DD	Estado de Inatividade (DOWN), Componentes Principal e de Reposição falharam
UD	Sistema está ativo (UP), Componente de Reposição falhou

espera, que indica o estado no qual o componente não está sendo utilizado, mas está pronto para entrar em operação normal após sua ativação. O estado do sistema é definido pela sequência de rótulos, representando o componente principal (P) e o de reposição (R), Estado = {P,R}, onde P = {U,D} e R = {U,W,D}. Desta maneira, o estado UW indica que o componente principal está em atividade e o componente de reparo está em estado de *espera*. Nos estados sombreados, o sistema está em estado de inatividade ou de falha. Tabela 4.1 mostra uma descrição de cada estado assumido pelo sistema.

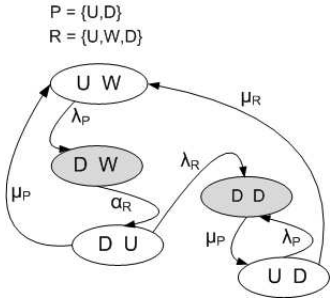


Figura 4.4: Modelo *Cold Standby* - CTMC

Warm Standby - Modelos

Neste mecanismo, tanto o componente principal quanto os componentes em *espera* podem falhar [56]. Entretanto, os componentes em *espera* possuem uma taxa de falha constante, φ, que é menor do que a taxa de falha relacionada ao componente principal, λ, $0 \leq \varphi \leq \lambda$. Considera-

se que os componentes em *espera*, quando em atividade, possuem uma taxa de falha igual a λ. Esta abordagem é caracterizada pela detecção de falhas junto com um mecanismo de ativação. Estamos assumindo que as taxas de falhas entre estes componentes são independentes. De maneira semelhante aos argumentos apresentados na abordagem *cold standby*, nesta abordagem são utilizados modelos baseados em espaço de estados, particularmente SPN e CTMC.

Modelo de Dependabilidade - SPN

Este modelo (ver Figura 4.5) inclui seis lugares relevantes, a saber, X_ON e X_OFF junto com os pares correspondentes que representam os estados atividade e inatividade dos componentes. Caso o componente principal falhe, a transição X_ACTSp é habilitada. Seu atraso representa o tempo de detecção da falha e de ativação do componente em espera. Este período é denominado de MTTA. Lugares X_Sp_ON e X_OSp_ON representam o componente em espera de X nos estados não-operacional e operacional respectivamente. O componente em espera inicia em um estado não operacional. Transições X_ACTSp, X_MTTR, X_Sp_MTTR e X_OSp_MTTR possuem a seguinte expressão correspondente ao atraso: IF #RelFlag=1: 10^n ELSE Y; onde Y representa ou valor de MTTA da transição X_ACTSp ou o parâmetro MTTR de cada componente. Por sua vez, o parâmetro MTTF de cada componente, principal e em espera, é atribuído às transições X_MTTF e X_OSp_MTTF. É feita a suposição de que a transição X_Sp_MTTF possui um atraso 50% mais alto do que transição X_OSp_MTTF. As transições temporizadas possuem uma distribuição exponencial (exp) e uma semântica de disparo do tipo *single server* (ss).

Conforme previamente descrito, o lugar RelFlag permite a avaliação das métricas de confiabilidade ou disponibilidade e a transição imediata X_DCTSp representa o retorno ao funcionamento normal após uma falha. Este modelo permite o cálculo da disponibilidade do sistema através da seguinte expressão: P{(#X_ON=1)OR(#X_OSp_ON=1)}. O cálculo da confiabili-

dade do sistema pode ser calculado pela expressão: P{(X_ON=1)}.

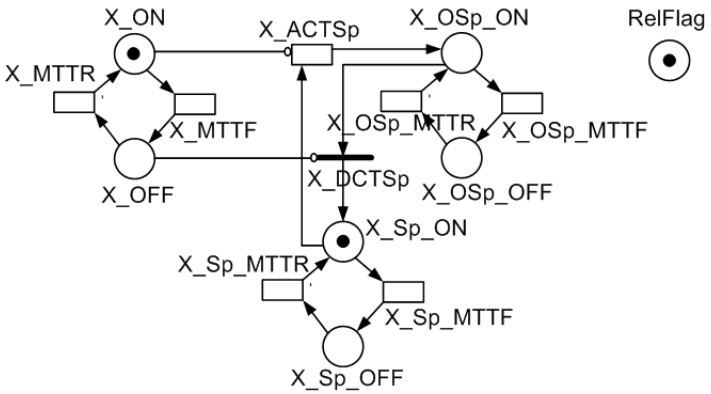


Figura 4.5: Modelo *Warm Standby* - SPN

Modelo de Disponibilidade - CTMC

A Figura 4.6 representa dois componentes, principal e de reposição, utilizando cadeia de Markov em tempo contínuo (CTMC). Neste modelo, os rótulos U, D e W representam os estados de atividade, inatividade e de *espera* respectivamente.

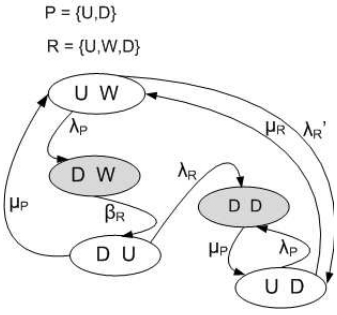


Figura 4.6: Modelo *Warm Standby* - CTMC

As transições de falha têm taxas λ_X e as transições de reparo têm taxas μ_X, onde $X \in \{P,R\}$ representa cada componente do sistema. É feita a suposição de que a taxa de falha do componente de reposição na condição de espera (W) será 50% menor que a taxa de falha do componente de reposição na condição de atividade (U). Estas taxas são representadas por λ'_R e λ_R respectivamente. A taxa β_R é o inverso do tempo onde está incluído tanto o tempo de

Tabela 4.2: Estados do modelo CTMC - *Warm Standby*

Estado	Descrição
UW	O Sistema está ativo (UP)
DW	Estado de Inatividade (DOWN), Componente Principal Falhou, Processo de ativação
DU	Sistema está ativo (UP), Componentes Principal falhou
DD	Estado de Inatividade (DOWN), Componentes Principal e de Reposição falharam
UD	Sistema está ativo (UP), Componente de Reposição falhou

detecção da falha quanto o tempo de ativação do componente de reposição. Com relação ao estado do sistema, é seguida a mesma representação mostrada no modelo CTMC (*cold standy*) anterior. De maneira semelhante, são também utilizadas as mesmas suposições efetuadas nesse modelo. A Tabela 4.2 mostra uma descrição de cada estado assumido pelo sistema.

4.3.4 Estruturas Básicas

As estruturas comumente encontradas em topologias de redes convergentes são as do tipo série, paralelo e série/paralelo [78]. Estas estruturas podem ser representadas por modelos RBD mostrados na Figura 3.9. Nesta seção, iremos mostrar alguns exemplos de topologias em série, em paralelo e em série/paralelo junto com seus respectivos modelos RBD.

Estruturas em Série e em Paralelo

Para que a rede esteja funcionando em uma topologia em série, é necessário que cada componente da rede esteja em funcionamento. Como exemplo, Figura 4.7 representa, através de um modelo RBD, uma topologia em série (ver Figura 4.8). Nesta topologia, os componentes R_0, R_1 e R_2 estão interligados através dos enlaces L_0 e L_1.

A Figura 4.9 mostra um modelo RBD que representa uma rede cujos enlaces entre os

Figura 4.7: Modelo RBD - Topologia em série

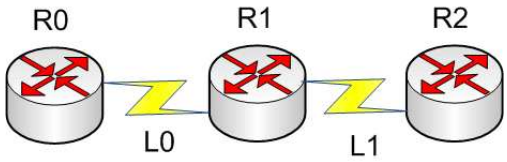


Figura 4.8: Exemplo de topologia em série

roteadores R_0 e R_1 estão estruturados em paralelo (ver Figura 4.10). Para que a rede esteja operacional, é necessário que pelo menos um dos componentes em paralelo (L_0, L_1 e L_2), junto com os roteadores R_0 e R_1, estejam funcionando.

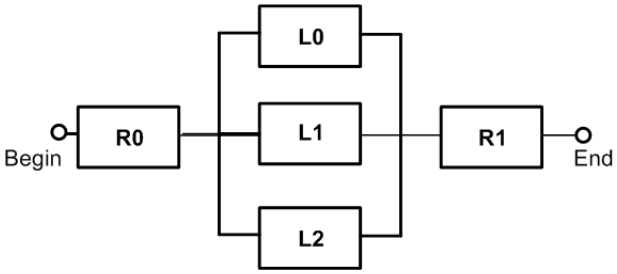


Figura 4.9: Modelo RBD - Topologia em paralelo

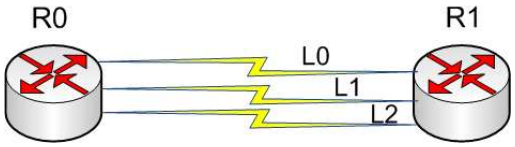


Figura 4.10: Exemplo de topologia em paralelo

A seguir, será mostrado um exemplo de topologia em série/paralelo.

Estrutura em Série/Paralelo

O modelo de dependabilidade mostrado na Figura 4.11 representa uma estrutura série/paralelo (ver Figura 4.12). Nesta rede, os componentes L_3, R_1, L_1, R_2 e L_5 estão em paralelo com os

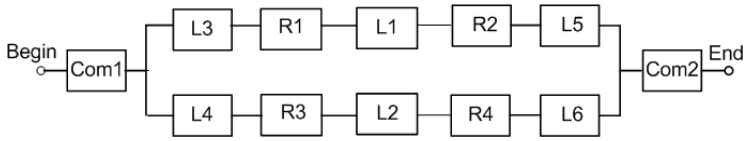


Figura 4.11: Modelo RBD - Topologia em série/paralelo

componentes L_4, R_3, L_2, R_4 e L_6. Por sua vez, estes componentes estão em série como os componentes Com_1 e Com_2.

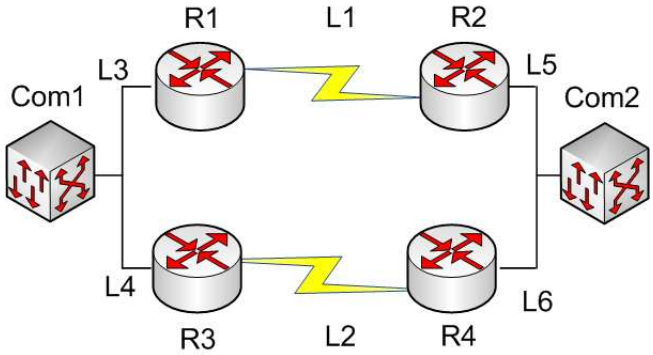


Figura 4.12: Exemplo de topologia em série/paralelo

4.4 Modelos de Desempenho

Será utilizada uma abordagem baseada em modelos SPN para a representação das políticas de filas *Custom Queuing*, *Priority Queuing* e FIFO sobre uma interface de um componente. A utilização do SPN ocorre tanto pela necessidade de representação de questões de tempo e de dependência dinâmica entre seus componentes quanto pela inviabilidade de utilização de CTMC devido ao grande número de estados.

A partir destes modelos, podemos analisar o comportamento de diferentes classes de tráfego e obter métricas que serão utilizadas como parâmetros de entrada para os modelos de negócios mostrados no Capítulo 5.

4.4.1 Custom Queuing e Priority Queuing - Modelo SPN

O modelo mostrado na Figura 4.13 foi construído para a representação das políticas de enfileiramento *Custom Queuing* e *Priority Queuing* sobre uma interface serial. Seu objetivo é o de avaliar o desempenho em termos de vazão e descarte de pacotes, por classe de tráfego, junto com o tamanho médio de cada uma das filas. Este modelo representa o tráfego de voz e de dados. Entretanto, modificações podem ser efetuadas para a representação de outras classes de tráfego.

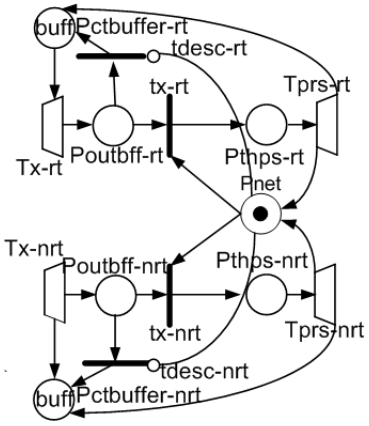


Figura 4.13: Modelo para políticas de enfileiramento - CQ e PQ

Neste modelo, os elementos de maior importância são: os parâmetros peso (*weight*) e prioridade (*priority*) das transições imediatas t*x-rt* e t*x-nrt* junto com o lugar P*net* e as correspondentes funções de guarda (em inglês, *enabling function*) das transições imediatas t*desc-rt* e t*desc-nrt*. Os parâmetros relacionados a estas transições imediatas são apresentados na Tabela 4.3.

Para a política de fila *Custom Queueing*, o parâmetro peso, representado pelas variáveis *w1* e *w2*, indica a probabilidade de disparo de transições habilitadas simultaneamente em um ECS (*Extended Conflict Set*). Desta forma, podemos representar a influência de diferentes recursos de transmissão proporcionados a cada uma das classes de tráfego e efetuar um estudo desta influência sobre as métricas de desempenho estudadas.

Tabela 4.3: Parâmetros das transições imediatas

Transição	Função Guarda	Prioridade	Peso	P. de Enfileiramento
t*x-rt*	–	5	*w1*	Custom Queuing
t*x-nrt*	–	5	*w2*	Custom Queuing
t*desc-rt*	#P*outbff-rt*$>$ *n1*	4	1	Custom Queuing
t*desc-nrt*	#P*outbff-nrt* $>$ *n2*	4	1	Custom Queuing
t*x-rt*	–	*p1*	*1*	Priority Queuing
t*x-nrt*	–	*p2*	*1*	Priority Queuing
t*desc-rt*	#P*outbff-rt*$>$ *n3*	4	1	Priority Queuing
t*desc-nrt*	#P*outbff-nrt* $>$ *n4*	4	1	Priority Queuing

Para a política de filas *Priority Queuing*, o parâmetro prioridade, representado pelas variáveis *p1* and *p2*, determina o disparo da transição com mais alto valor, ao invés de outras transições habilitadas simultaneamente dentro de um ECS (ver Tabela 4.3). De maneira semelhante, podemos representar a atribuição das diferentes classes de tráfego às diferentes filas para realizar um estudo da influência destas modificações sobre as métricas de desempenho estudadas.

Com relação ao lugar P*net*, este modela a transmissão serial. Caso o número de marcas nos lugares P*outbff-rt* e P*outbff-nrt* seja maior que os tamanhos de fila padrão (representado pelas variáveis *n1* e *n2*), então as transições imediatas t*desc-rt* e t*desc-nrt* disparam. Esta condição é representada através da função de guarda designada para estas transições conforme mostrado na Tabela 4.3.

O parâmetro tempo associado às transições estocásticas genéricas T*x-rt* e T*x-nrt* representam o tempo entre pacotes relativo aos tráfegos de voz e de dados, respectivamente. Por fim, o tempo associado às transições estocásticas genéricas T*prs-rt* e T*prs-nrt* representam o atraso sofrido por um pacote de voz e de dados em uma interface. Estes tempos são representados pelas variáveis *vs* e *ds*.

A Tabela 4.4 detalha todas as métricas suportadas pelo modelo proposto na Figura 4.13, junto com as expressões lógicas correspondentes. As taxas de saída, em pps, são computadas

Tabela 4.4: Métricas avaliadas - Custom Queuing e Priority Queuing

Métrica	Equação
VSV	(P{#Pthps-rt>0}*(1/*vs*))
VSD	(P{#Pthps-nrt>0}*(1/*ds*))
PDV	((I_{vnom})-((P{#P*thps-rt*>0})*(1/*vs*)))*time
PDD	((I_{dnom})-((P{#P*thps-nrt*>0})*(1/*ds*)))*time
TFV	E{#Poutbff-rt}
TFD	E{#Poutbff-nrt}

através das métricas *VSV* (*Vazão Saída Voz*) e *VSD* (*Vazão Saída Dados*). Os números de pacotes descartados de voz e de dados, métricas *PDV* (*Pacotes Descartados Voz*) e *PDD* (*Pacotes Descartados Dados*), são obtidos considerando a diferença entre as taxas de entrada nominal e as respectivas taxas de saída multiplicadas por um período de tempo específico. As taxas de entrada nominal de voz e de dados (I_{vnom} e I_{dnom}), em pps, são calculadas através do inverso dos atrasos associados às transições T*x-rt* e T*x-nrt*, respectivamente. Por fim, as métricas *TFV* (*Tamanho Fila Voz*) e *TFD* (*Tamanho Fila Dados*) computam o tamanho médio das filas de saída correspondentes aos tráfegos de voz e de dados.

Este modelo pode ser refinado através de combinações de variáveis aleatórias exponenciais (distribuição expolinomial) [20, 32]. Um método que considera distribuição expolinomial de variáveis aleatórias é baseado em *moment-matching* [20], conforme descrito na Seção 3.7.1 (Aproximação por Fases). Seção 7.3.2 detalhará a aplicação do processo de refinamento sobre um modelo de desempenho proposto.

4.4.2 FIFO - Modelo SPN

O modelo mostrado na Figura 4.14 foi construído para a representação da política de enfileiramento *FIFO* sobre uma interface. Neste modelo, o atraso (*delay*) associado à transição temporizada $T_{TxEntrada}$ representa o tempo entre os pacotes do tráfego de entrada na interface. A transição T_{Int} representa o atraso sofrido por um pacote na interface. Este tempo é representado

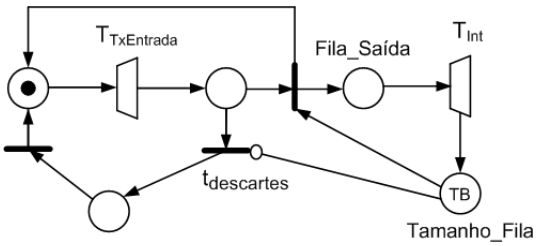


Figura 4.14: Modelo de política de enfileiramento - FIFO

Tabela 4.5: Métricas avaliadas - FIFO

Métrica	Equação
VS	(P{#Fila_Saida>0}*(1/*pp*))
TF	E{#Fila_Saida}

pela variável *pp*. Lugar Fila_Saída representa a fila de saída da interface. Variável *TB*, associada ao lugar Tamanho_Fila, representa o número máximo de pacotes suportados por esta fila. O disparo da transição imediata $t_{descartes}$ representa o descarte de um pacote devido ao tamanho da fila ter alcançado seu limite máximo (#Tamanho_Fila=0).

A Tabela 4.5 mostra as métricas suportadas pelo modelo mostrado na Figura 4.14 junto com as expressões lógicas correspondentes. A métrica VS (*Vazão de Saída*) mostra a vazão de saída resultante. O tamanho médio da fila de saída é calculado através da métrica TF (*Tamanho Fila*).

Este modelo poderá também ser refinado através da representação das transições estocásticas genéricas por distribuições expolinomiais através do método *moment-matching* [20].

4.5 Considerações Finais

Este capítulo detalhou os modelos SPN, CTMC e RBD adotados para representar e proporcionar informações com relação aos aspectos de infraestrutura de redes convergentes. Foram mostrados modelos de dependabilidade e de desempenho junto com suas correspondentes

métricas. Com relação aos modelos de dependabilidade, foram mostrados modelos correspondentes a um componente simples, às partes funcionais de um componente, aos principais mecanismos de redundância (*cold standby* e *warm standby*) e às estruturas mais comumente encontradas em topologias de redes convergentes (*série*, *paralelo* e *série-paralelo*). Foram também mostrados modelos de desempenho correspondentes às políticas de filas *Priority Queuing*, *Custom Queuing* e *FIFO*.

CAPÍTULO 5

Modelos de Negócios

Este capítulo irá apresentar os modelos analíticos de *custo de infraestrutura*, *receita de infraestrutura*, *multa*, *lucro líquido*, *lucro líquido adicional por unidade monetária gasta* e *variação do tempo de parada por unidade monetária gasta* referentes às classes de redes convergentes consideradas neste trabalho. Estes modelos, em conjunto com os de infraestrutura e a metodologia proposta, proporcionam suporte para o processo de otimização da infraestrutura de redes convergentes.

Os modelos analíticos possuem parâmetros relacionados a aspectos da infraestrutura, como disponibilidade e vazão (ver Figura 1.2). Desta maneira, é possível capturar o impacto sobre os negócios de eventos oriundos da infraestrutura. As métricas de negócios obtidas são facilmente compreensíveis por parte de gerentes e executivos, que por sua vez, buscam constantemente melhorar o desempenho financeiro de uma empresa.

A partir dos modelos analíticos propostos, pode-se responder questões tais como as seguintes:

- Qual o impacto de falhas e da degradação de desempenho sobre as métricas de negócios?
- Qual o impacto sobre as receitas oriundas da infraestrutura se o principal componente da rede estiver *fora de operação*?
- Qual impaco sobre os níveis de serviços oferecidos se for utilizado um mecanismo de redundância alternativo em um ponto do sistema?
- Qual impacto sobre receitas oriundas da infraestrutura decorrente do tratamento dispensado às diferentes classes de tráfego?

As métricas de negócios permitem contabilizar o impacto da dependabilidade e do desempenho de redes convergentes sobre seus aspectos financeiros. A contribuição de cada fator pode ser combinada, oferecendo uma maneira de unir considerações de infraestrutura e de negócios sob um único problema de projeto. O fato das métricas de negócio serem numéricas permite-nos formular problemas de projeto utilizando métricas de negócios conjuntamente com métricas técnicas em funções objetivo (ver Seção 5.3).

5.1 Descrição do Ambiente

Para facilitar tanto o processo de escolha do melhor projeto como uma análise comparativa entre as soluções de projeto obtidas, são consideradas duas classes de redes convergentes. Inicialmente, consideramos o caso em que as redes convergentes não geram diretamente receitas, a partir de seus serviços, às empresas que as detêm. O lucro líquido resultante destas empresas não é significativamente afetado pela disponibilidade destas redes. Embora esta classe de redes convergentes seja importante para o funcionamento das empresas, ela não interfere diretamente sobre seus processos vitais. Posteriormente, consideramos redes convergentes que geram diretamente receitas a partir de seus serviços. Esta classe interfere sobre os processos vitais das empresas, que, são diretamente afetados pela disponibilidade destas redes. Como exemplo, podemos mencionar redes de provedores de serviços de comunicação.

Os provedores de serviços de comunicação podem ser divididos em vários segmentos [1]: provedores de serviços de backbone, provedores de serviços de internet/provedores de serviços online e provedores correspondentes ao *último quilômetro*, ou provedor *Last Mile*.

Provedores de serviços de backbone referem-se às empresas que mantêm seus próprios enlaces de comunicação. As empresas neste segmento controlam enlaces de longas distâncias de âmbito regional, nacional ou mesmo internacional e estão também aptas a manusear um

grande volume de tráfego digital dentro de suas infraestruturas.

Um provedor de serviços de internet (ISP) proporciona acesso à internet para clientes individuais e empresas [21]. Os ISPs possuem seus próprios servidores, comutadores, roteadores e softwares. Por outro lado, um provedor de serviços online (OSP) [108] é uma entidade que oferece serviços, tais como serviços bancários, compras, armazenamento, entretenimento etc, através da internet. Um recurso de grande importância para a gestão, armazenamento e disseminação destes serviços é conhecido como *data center*, que, dependendo da natureza dos serviços oferecidos, pode ser de uso privado ou público. A Figura 5.1 ilustra a arquitetura de um OSP e sua relação com os ISPs [108].

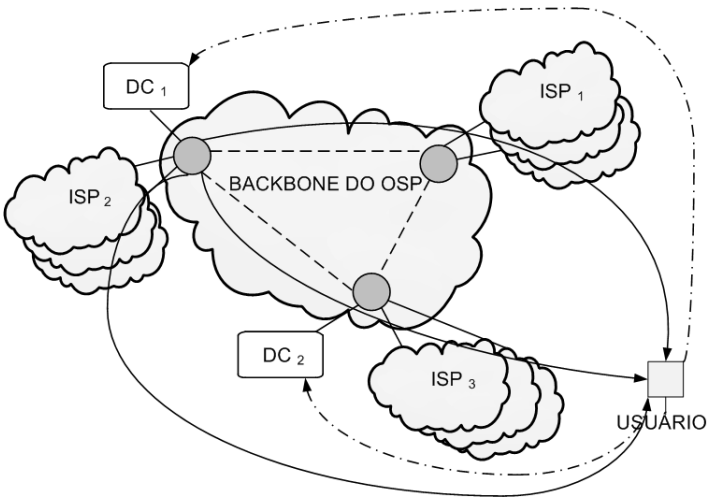


Figura 5.1: Arquitetura de rede típica de um grande OSP

Em geral, as empresas de comércio eletrônico, agregadoras de conteúdo, provedoras de serviços e empresas intermediárias para transações comerciais podem estar *"hospedadas"* em uma infraestrutura de um *data center*. As empresas de comércio eletrônico trocam "*produtos reais*"por "*dinheiro*"através de canais *online*. As empresas agregadoras de conteúdo englobam as de mídia e provedoras de conteúdo; as empresas provedoras de serviços obtêm seus lucros com a venda de seus serviços ou seus conhecimentos. Elas podem incluir suporte a serviços de consultoria e terceirização. Empresas intermediárias com relação a transações comerciais proporcionam um ambiente para o comércio entre as diferentes partes envolvidas e também

definem as regras conjuntas do mercado.

Por fim, o provedor correspondente ao último quilômetro proporciona e mantém conexões físicas (p. ex, conexões sem-fio, telefone, *cable modem*) aos clientes individuais e empresas. As empresas pertencentes a este segmento podem proporcionar serviços entre si ou para clientes individuais. Com a evolução das tecnologias, os provedores correspondentes ao último quilômetro podem incorporar clientes tanto fixos quanto móveis.

5.2 Modelos Analíticos Associados a Redes Convergentes

Nesta seção serão mostrados modelos analíticos para o cálculo das métricas de negócios. Serão consideradas métricas relativas aos *custo de infraestrutura* (ICust), *receita de infraestrutura* (Rc), *multa*(M_{Tot}), *lucro líquido (Lc)*, *lucro líquido adicional por unidade monetária gasta (ALc)* e *variação do tempo de parada por unidade monetária gasta (VTp)*. As métricas *ALc* e *VTp* possibilitam uma análise comparativa entre diversas soluções de projetos e são úteis ao processo de tomada de decisão.

5.2.1 Custo de Infraestrutura

A métrica custo de infraestrutura é derivada da métrica proposta em [85]. Para determinar o custo de infraestrutura é necessário um modelo que represente os recursos de hardware e de software da infraestrutura. Cada *l-ésimo* componente, tal como um roteador, consiste de um conjunto de recursos, tal que $C_l = \{r_1, r_2, ..., r_n\}$. Como um exemplo, um roteador consiste de dois recursos, hardware e sistema operacional. Cada *k-ésima* classe de componentes, isto é, classe de roteadores, classe de comutadores, consiste de componentes similares, tal que $CL_k = \{C_1, C_2, ..., C_l\}$. RC é o conjunto de classes de componentes, tal que RC = $\{CL_1, CL_2, ..., CL_k\}$

(ver Figura 5.2). Cada recurso possui uma taxa de custo (custo por unidade de tempo) de $r_{k,l,n}$. Se uma taxa de custo variável ($r_{k,l,n}$) for considerada, o custo de infraestrutura é dado pela Equação 5.1:

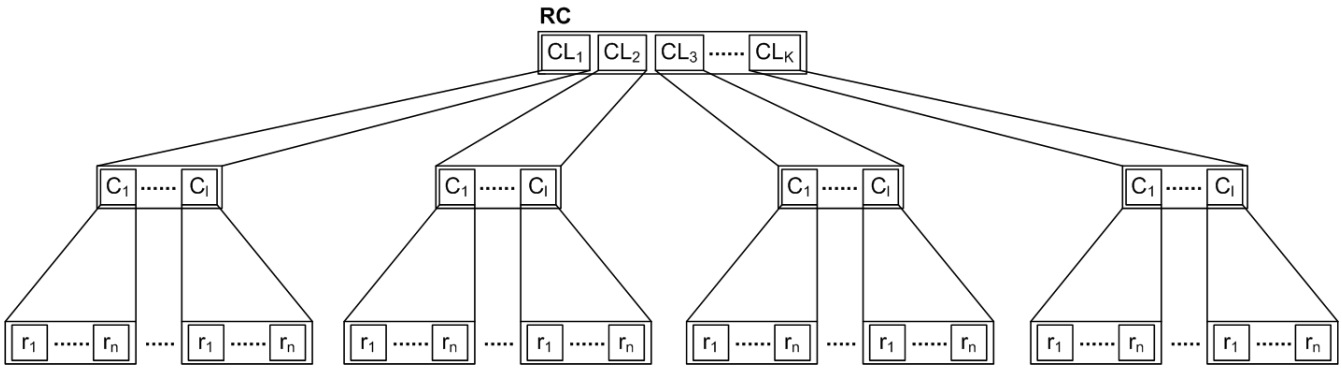


Figura 5.2: Entidades do modelo custo de infraestrutura

$$ICust(t) = \int_0^t (\sum_{k=1}^{|RC|} \sum_{l=1}^{|CL_k|} \sum_{n=1}^{|C_l|} r_{k,l,n}(\tau)) \times d\tau \tag{5.1}$$

Se a taxa de custo for constante, o custo da infraestrutura considerando-se o período de tempo T pode ser calculado como a soma dos custos individuais de todos os componentes (ver Equação 5.2).

$$ICust(T) = (\sum_{k=1}^{|RC|} \sum_{l=1}^{|CL_k|} \sum_{n=1}^{|C_l|} r_{k,l,n}) \times T \tag{5.2}$$

5.2.2 Receita de Infraestrutura

Com relação às receitas, este estudo considerou receitas da perspectiva de um provedor de serviços de comunicação [1]. Nesta classe de redes, podemos ter um modelo de receitas baseado na alocação de recursos para os diferentes tipos de serviços de um cliente individual ou de uma empresa (*taxa-por-serviço*) ou um modelo baseado em assinatura. Inicialmente, o modelo de *taxa-por-serviço* proposto considera apenas um cliente junto com os serviços aloca-

dos em função da demanda de tráfego. Uma generalização para um maior número de clientes é direta.

O modelo proposto calcula as receitas de cada tipo de serviço. Ele considera características tais como: disponibilidade do sistema (A), vazão ($thps$) e valor financeiro (vf) associados ao serviço j. Se uma vazão variável for considerada, a receita de infraestrutura é dada pela Equação 5.3:

$$Rc(t,A,thps,vf,k) = \int_0^t (\sum_{j=1}^{k} A(\tau) \times thps_j(\tau) \times vf_j) d\tau \tag{5.3}$$

Agora, uma vazão média e disponibilidade em estado estacionário são assumidos. A receita oriunda da infraestrutura, considerando-se o período de tempo T, pode ser calculada (ver Equação 5.4).

$$Rc(T,A,thps,vf,k) = (\sum_{j=1}^{k} A \times thps_j \times vf_j) \times T \tag{5.4}$$

Finalmente, as receitas podem ser também baseadas em um modelo de *assinatura*. Neste modelo, cada cliente gera uma receita fixa em um período de tempo determinado (tipicamente 1 mês ou 1 ano). A Equação 5.5 considera x classes de assinatura (C_1, C_2, ..., C_x) em um determinado período de tempo.

$$Rc(C,n,x) = \sum_{i=1}^{x} n_i \times C_i \tag{5.5}$$

- $thps$: Vazão média associada a um serviço, em pacotes por segundo (pps);
- $thps(\tau)$: Vazão instantânea associada a um serviço, em pacotes por segundo (pps);

- A: Disponibilidade em estado estacionário;
- $A(\tau)$: Disponibilidade instantânea;
- T: Período de tempo;
- vf: Valor financeiro associado a um serviço j, em Us$/pacote;
- k: Número de Serviços;
- C: Valor financeiro associado a uma classe de assinatura i em período de tempo, em Us$;
- n: Número de clientes em cada classe i;
- x: Número de classes de assinatura.

5.2.3 Multa

Este estudo considerou multas da perspectiva de um provedor de serviços de comunicação [1]. Estas multas são baseadas na redução do pagamento acertado pela utilização do serviço, isto é, uma sanção financeira [82]. O cálculo baseia-se no tempo de interrupção do sistema e o custo, por serviço j, está de acordo com o nível de serviço oferecido (ver Equação 5.6) sobre um período de tempo T. Com relação à função I, a Equação 5.7 mostra seus diferentes valores em função do nível de serviço oferecido. Diferentes limites, com relação à disponibilidade, devem ser definidos para cada nível de serviço.

Como um exemplo, considerando que a disponibilidade calculada é maior que um limite definido em contrato (th_1), não haverá multa e o índice I_1 assume um valor igual a zero.

$$M_j(T,A,I) = \begin{cases} 0 & \text{para } A \geq th_1 \\ T \times I_2 \times (1-A) & \text{para } th_2 \leq A < th_1 \\ ..... & ..... \\ T \times I_n \times (1-A) & \text{para } th_n \leq A < th_{n-1} \end{cases} \tag{5.6}$$

$$I(A) = \begin{cases} I_1 & \text{para } A \geq th_1 \\ I_2 & \text{para } th_2 \leq A < th_1 \\ ..... & ..... \\ I_n & \text{para } th_n \leq A < th_{n-1} \end{cases} \tag{5.7}$$

O custo total da multa sobre um período de tempo T pode ser calculado como a soma dos custos individuais para cada serviço j (ver Equação 5.8).

$$M_{Tot}(T,A,I) = \sum_{j=1}^{k} M_j \tag{5.8}$$

- th_i: Valor do limite no nível de serviço i, tal que $0 \leq th_i \leq 1$;
- T: Período de Tempo;
- A: Disponibilidade em estado estacionário do sistema no período de tempo;
- I: Índice cujo valor depende do nível de serviço i.

5.2.4 Lucro Líquido

O lucro líquido é um benefício financeiro que acontece quando o valor da receita obtida a partir de uma atividade de negócios ultrapassa os custos para manter esta atividade. De maneira semelhante aos modelos de Receita de Infraestrutura e Multa, este modelo é considerado a

partir de uma perspectiva de um provedor de serviços de comunicação. O lucro líquido é obtido utilizando a Equação 5.9 no caso de receitas geradas a partir de um modelo de *taxa-por-serviço*. O lucro líquido é obtido utilizando a Equação 5.10 no caso de receitas geradas a partir do modelo de *assinatura*.

$$Lc(T,A,thps,vf,I) = R_c(T,A,thps,vf) - (M_{Tot}(T,A,I) + ICust(T)) \tag{5.9}$$

$$Lc(T,A,I,C,n,x) = R_c(C,n,x) - (M_{Tot}(T,A,I) + ICust(T)) \tag{5.10}$$

5.2.5 Lucro Líquido Adicional por Unidade Monetária Gasta (ALc)

Esta métrica auxilia na escolha do projeto mais adequado considerando a classe de redes convergentes que gera diretamente receitas a partir de seus serviços. ALc é o lucro líquido adicional por unidade monetária gasta entre duas soluções de projetos. Embora relacione apenas aspectos de negócios, esta métrica embute aspectos de infraestrutura (a disponibilidade está embutida na métrica Lucro Líquido). Se a nova solução (cenário *j*) resulta em um valor de ALc maior que um (> 1) em relação à solução original (cenário *i*), a solução deverá ser selecionada. É também útil para proporcionar análise comparativa entre diferentes arquiteturas/cenários. Por exemplo, se o valor resultante de ALc entre o cenário *i* e o cenário *j* é maior que o valor de ALc entre o cenário *i* e o cenário *k*, será melhor permutar para o cenário *j*, a partir do cenário *i*, pois o lucro líquido adicional por unidade monetária gasta é maior para esta opção. A Equação 5.11 mostra a expressão relativa a este modelo.

$$ALc = \Delta Lc(T,A,thps,vf,I)/\Delta ICust(T) \tag{5.11}$$

- $\Delta Lc(T,A,thps,vf,I)$: Lucro Líquido adicional entre duas soluções de projeto, i e j, que possuem diferentes custos de infraestrutura, onde $Lc_i(T,A,thps,vf,I)$ é o lucro da solução de projeto i e $Lc_j(T,A,thps,vf,I)$ é o lucro para a solução de projeto j.

- $\Delta ICust(T)$: Diferença em custo de infraestrutura entre duas soluções de projeto, i e j, onde $ICust_i(T)$ é o custo de infraestrutura da solução de projeto i e $ICust_j(T)$ é o custo de infraestrutura para a solução de projeto j.

5.2.6 Variação de Tempo de Parada por Unidade Monetária Gasta (VTp)

Esta métrica auxilia na escolha do projeto mais adequado considerando redes que não geram diretamente receitas a partir de seus serviços. VTp é a variação do tempo de parada, correspondente à disponibilidade adicionalmente obtida, por unidade monetária gasta entre duas soluções de projetos. Esta métrica também é útil à análise comparativa entre diferentes arquiteturas/cenários. Por exemplo, se o VTp entre o cenário i e o cenário j é maior que o VTp entre o cenário i e o cenário k, será melhor permutar para o cenário j, a partir do cenário i, pois a variação do tempo de parada, correspondente à disponibilidade adicionalmente obtida, por unidade monetária gasta é maior para esta opção. A Equação 5.12 mostra a expressão correspondente.

$$VTp = \Delta D(A,T)/\Delta ICust(T) \tag{5.12}$$

- $\Delta D(A,T)$: Variação do tempo de parada entre duas soluções de projetos i e j, que possuem diferentes custos de infraestrutura, onde $D_i(A,T)$ é o tempo de parada para a solução de projeto i e $D_j(A,T)$ é o tempo de parada para a solução de projeto j.

- $\Delta ICust(T)$: Diferença em custo de infraestrutura entre duas soluções de projetos i e j, onde $ICust_i(T)$ é o custo de infraestrutura para a solução de projeto i e $ICust_j(T)$ é o custo de infraestrutura para solução de projeto j.

5.3 Funções Objetivo

Com relação ao problema da escolha do melhor projeto de uma infraestrutura, quando levamos em consideração a classe de redes convergentes que gera diretamente receitas a partir de seus serviços, será considerada a função objetivo mostrada abaixo. Esta função deve ser maximizada a fim de obter a melhor solução dentro do espaço de projetos candidatos. Os projetos candidatos são analisados na atividade Seleção de Projeto da metodologia proposta (ver Seção 6.3.3). Os parâmetros da função objetivo são os valores normalizados (ver Equação 6.1) das métricas de disponibilidade (nA) e de lucro líquido (nLc). Como restrição, os valores de nA e de nLc devem estar compreendidos entre 0 e 1.

Maximizar: F(nA, nLc)

Satisfazendo:

$0 \leq nA_i \leq 1$

$0 \leq nLc_i \leq 1$

Em que:

nA, nLc - Variáveis de Projeto. Valores normalizados de A e Lc.

F(nA,nLc) - Função Objetivo. Maior distância Euclidiana de (nA,nLc) para (0,0).

Por outro lado, quando levamos em consideração a classe de redes convergentes que não gera diretamente receitas a partir de seus serviços, será considerada a função objetivo mostrada abaixo. Esta função deve ser minimizada a fim de se obter a melhor solução dentro do espaço de projetos candidatos. De maneira semelhante, os projetos candidatos são analisados na atividade Seleção de Projeto da metodologia proposta. Os parâmetros da função objetivo são os valores normalizados (ver Equação 6.1) das métricas de indisponibilidade (nUA) e de custo de infraestrutura (nICust). Como restrição, os valores de nUA e de nICust devem estar compreendidos entre 0 e 1.

Minimizar: F(nUA, nICust)

Satisfazendo:

$0 \leq nUA_i \leq 1$

$0 \leq nICust_i \leq 1$

Em que:

nUA, nICust - Variáveis de Projeto. Valores normalizados de UA e ICust.

F(nUA,nICust) - Função Objetivo. Menor distância Euclidiana de (nUA,nICust) para (0,0).

5.4 Modelos de Negócio

Estratégia de comércio refere-se à estratégia que identifica os clientes base ou a população servida por uma empresa. As relações comerciais, considerando a venda de bens e de serviços, podem acontecer em várias direções entre clientes individuais, empresas e governos sobre a internet ou através da utilização de redes privadas. Neste trabalho, em virtude do escopo envolvido, tratamos apenas com as relações do tipo B2C (*Business-to-Customer*), B2B (*Business-to-Business*) e B2G (*Business-to-Governement*), que são detalhadas abaixo:

- B2C: Neste tipo de negócio, as empresas vendem bens e serviços aos consumidores;
- B2B: Neste tipo de negócio, empresas compram e vendem bens e serviços para outras empresas;
- B2G: Neste tipo de negócio, as empresas vendem bens e serviços aos governos.

No trabalho mostrado em [2] é proposto um modelo de negócios de dois lados, em que um provedor poderá prestar serviços tanto a um cliente individual, B2C, quanto a uma empresa privada ou pública, B2B e/ou B2G, de maneira a rentabilizar as novas oportunidades das mídias

digitais. A rede envolvida pertence à classe de redes que gera diretamente receitas, multas e lucros a partir dos serviços oferecidos em sua infraestrutura. Os diferentes tipos de serviços podem ser proporcionados de acordo com a classe do cliente, originando uma maior variedade relativa às fontes de receitas.

A Equação 5.4 proporciona suporte para diferentes tipos de serviços, considerando uma vazão média e disponibilidade em estado estacionário, de um cliente. Esta equação baseia-se em um modelo de receita de *taxa-por-serviço*. Diferentes modelos de receita, como por exemplo modelo baseado em *assinatura* (ver Equação 5.5), poderiam ser considerados. Estas equações podem ser facilmente adaptadas para proporcionar suporte a diferentes classes de clientes, com diferentes tipos de serviços em cada uma das classes.

A Equação 5.2 mostra o custo de infraestrutura envolvido para este provedor. A Equação 5.8 detalha a multa total a ser paga devido ao caso de as metas não serem integralmente cumpridas. Por fim, as Equações 5.9 e 5.10 calculam o lucro líquidos originado por estas redes.

Desta forma, os modelos propostos podem ser também utilizados para proporcionar suporte a modelos de negócios, para viabilizar a escolha do projeto de uma infraestrutura e para proporcionar uma análise comparativa entre soluções obtidas.

5.5 Considerações Finais

Este capítulo detalhou os modelos analíticos para o cálculo das métricas de negócios de redes convergentes. Foram apresentados modelos analíticos de *custo de infraestrutura*, *receita de infraestrutura*, *multa*, *lucro líquido*, *lucro líquido adicional por unidade monetária gasta* e *variação do tempo de parada por unidade monetária gasta*. Para proporcionar suporte à escolha do melhor projeto de uma infraestrutura, foram detalhadas duas classes de redes convergentes.

Para a classe que não gera diretamente receitas a partir de seus serviços, são consideradas as métricas de *custo de infraestrutura* (ICust) e de indisponibilidade (UA). Para uma comparação objetiva entre diferentes projetos, a métrica de *variação do tempo de parada por unidade monetária gasta* (VTp) foi proposta. Por outro lado, com relação à classe de redes convergentes que gera diretamente receitas partir de seus serviços, são consideradas as métricas de *lucro líquido* (Lc) e de disponibilidade (A). Para uma comparação objetiva entre diferentes projetos, a métrica de *lucro líquido adicional por unidade monetária gasta* (ALc) foi proposta.

CAPÍTULO 6

Metodologia Utilizada no Processo de Otimização

O presente capítulo detalha a metodologia proposta para o processo de escolha do melhor projeto de uma infraestrutura de redes convergentes. Esta metodologia tem por objetivo reduzir a complexidade do processo de seleção do melhor projeto, bem como aperfeiçoar a relação entre aspectos técnicos e de negócios relacionados a uma infraestrutura de redes convergentes. O projeto escolhido corresponde ao ponto, dentro do espaço de projetos candidatos, que representa um compromisso de equlíbrio entre as métricas técnicas e de negócios. A metodologia é composta de três fases, a saber: Análise do Problema, Modelagem do Sistema e Seleção de Projeto. Figura 6.1 detalha esta sequência de fases.

Por sua vez, a Figura 6.2 mostra a sequência de atividades que deve ser executada quando a metodologia for aplicada. Estas atividades, em suas correspondentes fases, serão detalhadas nas próximas seções.

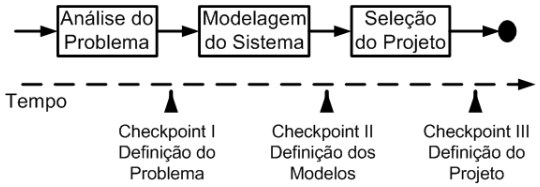


Figura 6.1: Fases da metodologia proposta e seus checkpoints

6.1 Fase I: Análise do Problema

O objetivo desta fase é determinar o problema junto com seu escopo em termos de aspectos de infraestrutura e de negócios. Esta fase é composta da atividade Definição do Problema. Ao fim

desta atividade três documentos são produzidos, a saber: Diagrama do Problema, Diagrama de Escopo e as Métricas consideradas.

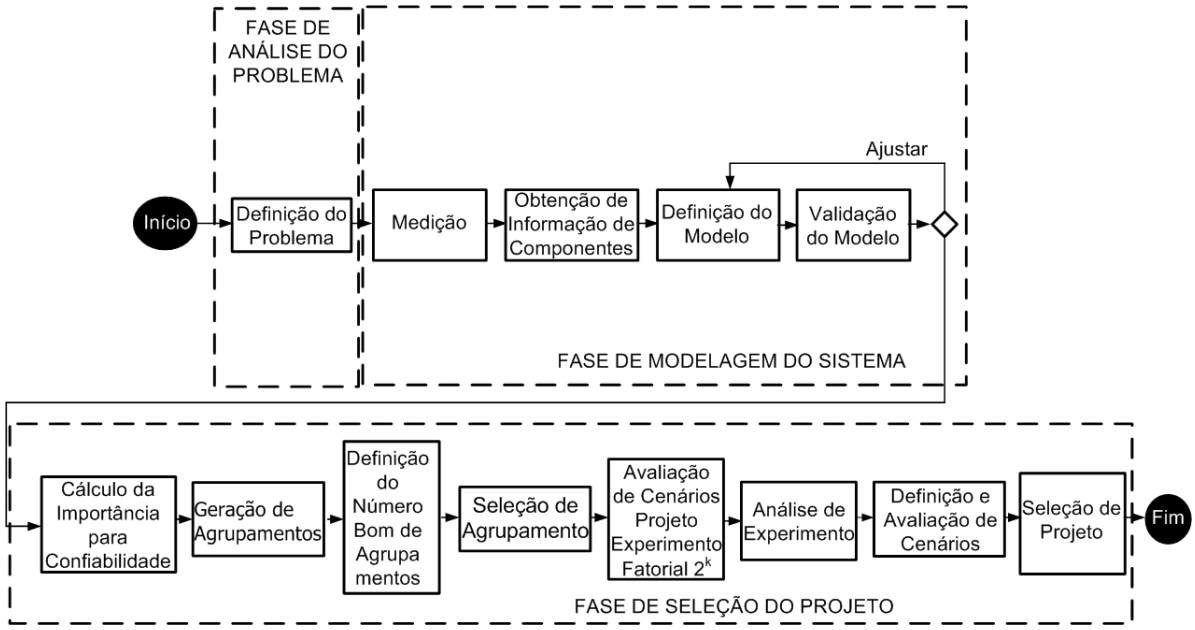


Figura 6.2: Metodologia proposta

Diagrama do Problema é um documento que contém aspectos da infraestrutura do sistema sob análise tais como: sua topologia, interconexão e dependência entre os componentes, definição dos componentes e configurações de hardware (ver Apêndice D). O segundo documento, o Diagrama de Escopo (ver Apêndice D), define quais partes da infraestrutura do sistema são consideradas com respeito a aspectos de desempenho e dependabilidade, junto com a definição de quais documentos de negócios são também considerados. Finalmente, o terceiro documento, define o conjunto de métricas, relacionadas à disponibilidade, confiabilidade, desempenho e custos, utilizadas para a análise do sistema.

É importante notar que a atividade Definição do Problema é executada de maneira iterativa através da definição do Diagrama do Problema, Diagrama de Escopo e de Métricas e finaliza quando os objetivos da avaliação tornam-se claros.

6.2 Fase II: Modelagem do Sistema

Os principais produtos gerados nesta fase são os modelos de infraestrutura e de negócios. Esta fase é composta de quatro atividades, a saber: Medição, Obtenção de Informação de Componentes, Definição do Modelo, Validação do Modelo. Informações sobre os componentes são obtidas nas atividades Medição e Obtenção de Informação de Componentes. Estas informações serão utilizadas como parâmetros de entrada para os modelos criados.

Com relação à medição, o sistema deve ser isolado e programas que não estão diretamente relacionados ao ambiente devem ter suas execuções terminadas a fim de que as medidas obtidas não sejam influenciadas por fatores externos. Ferramentas para geração de carga e para monitoração do sistema devem ser adotadas (ver Apêndice A). Informações podem ser obtidas dos componentes do sistema a partir de interfaces dos sistemas operacionais correspondentes, que coletam dados relativos às diferentes classes de tráfego. Com relação à atividade Obtenção de Informação de Componentes, será necessária a obtenção de informações oriundas tanto dos fabricanes dos componentes, tais como MTTF, como das empresas responsáveis pelas políticas de reparo. Particularmente, com relação a componentes como enlaces, o MTTF é fornecido pela empresa detentora deste recurso, através de uma base de informações.

A atividade Definição do Modelo refere-se à criação de modelos para o sistema sob análise. Foram utilizados modelos do tipo SPN, CTMC e RBD, além de expressões analíticas. Esta atividade é executada pela composição de cada componente do sistema de acordo com regras específicas e pelo mapeamento das métricas desejadas.

A atividade Validação do Modelo compara os resultados obtidos a partir de cada modelo construído tanto com os resultados de um outro modelo semelhante como com medições diretas, obtidas através de ambientes construídos com uma finalidade específica, e realiza ajustes quando necessário. O fim desta fase é alcançado quando cada modelo proporciona resultados

com exatidão apropriada.

6.3 Fase III: Seleção do Projeto

Esta fase é responsável pela seleção do melhor projeto de uma infraestrutura. Devido a sua importância, Apêndice E ilustra as principais atividades desta fase. Ela é composta das seguintes atividades: Cálculo da Importância para Confiabilidade, Geração de Agrupamentos, Definição do Número Bom de Agrupamentos, Seleção de Agrupamento, Avaliação de Cenários Projeto de Experimento Fatorial 2^k, Análise de Experimento, Definição e Avaliação de Cenários, Seleção de Projeto.

O principal produto desta fase é a definição de qual opção de projeto deve ser implementada. A atividade Cálculo da Importância para Confiabilidade calcula o índice *Importância para Confiabilidade* de cada componente a partir de uma configuração definida pelo projetista e tomada por base. A atividade Geração de Agrupamentos cria agrupamentos utilizando a abordagem agrupamento hierárquico aglomerativo (AHC) [24]. Dentre os métodos aglomerativos, estão o método do vizinho mais próximo, o método do vizinho mais distante, método da ligação média, método de centróide e o método de Ward [107]. Neste trabalho será utilizado o método de centróide [24].

Para agrupar componentes semelhantes são utilizados os atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente*. Os atributos *custo do componente* e *vazão do componente* são obtidos a partir do fabricante ou a partir do provedor de serviços. A atividade Definição do Número Bom de Agrupamentos seleciona a melhor configuração de agrupamentos. Esta atividade utiliza o índice RMSSTD [83].

Após a determinação do valor do índice RMSSTD em função do número de agrupamen-

tos (Ag) em cada passo do algoritmo hierárquico [47], estes valores serão normalizados. A quantidade de agrupamentos referente ao passo i, que está relacionada à menor distância Euclidiana dos valores normalizados de RMSSTD (nRMSSTD) e do número correspondente de agrupamentos (nAg), para a origem, indica o número bom de agrupamentos. A partir desta quantidade, não se verificam maiores declínios nos valores do RMSSTD, que por sua vez, computa a homogeneidade dos agrupamentos formados.

A partir do número bom de agrupamentos, a atividade Seleção de Agrupamento escolhe o agrupamento com o mais alto valor de média aritmética (índice I_c) dentre os agrupamentos. Para o cálculo desta média aritmética em cada agrupamento, considera-se a média aritmética ponderada de cada um dos componente do agrupamento. O cálculo da média aritmética ponderada utiliza os valores normalizados dos atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente*, junto com os respectivos *pesos* (w_1, w_2 e w_3). Os valores dos pesos devem priorizar a escolha do agrupamento cujos componentes possuem maior impacto sobre a confiabilidade do sistema.

Considerando os componentes representados no agrupamento selecionado, projeto de experimento fatorial 2^K [51] (k é o número total de componentes no agrupamento selecionado) é analisado. A atividade Análise de Experimento escolhe apenas os componentes do agrupamento selecionado com maior impacto sobre a disponibilidade do sistema com o objetivo de reduzir este número. Estes componentes são então adotados para a avaliação dos projetos candidatos.

A atividade Definição e Avaliação de Cenários produz um conjunto de cenários que são representados por diferentes combinações de parâmetros de entrada, relacionados aos modelos de dependabilidade, desempenho e de negócios. Estes cenários são construídos com base nos componentes selecionados na atividade anterior, considerando suas diferentes opções (valores). Após a etapa de definição, os cenários serão avaliados com base nos valores de métricas associadas. Os valores de métricas serão normalizados em cada cenário i.

Com relação à atividade Seleção de Projeto, duas abordagens são consideradas. Na primeira abordagem, se a rede convergente gera diretamente receitas a partir de seus serviços, as métricas de *Disponibilidade* (nA) e *Lucro_Líquido* (nLc) são consideradas. O projeto correspondente ao cenário i com a maior distância Euclidiana de (nA_i, nLc_i) para a origem $(0,0)$ é escolhido. Na segunda abordagem, se a rede convergente em questão não gera diretamente receitas a partir de seus serviços, as métricas de *Indisponibilidade* (nUA) e *Custo de Infraestrutura* (nICust) são consideradas. O projeto correspondente ao cenário i com a menor distância Euclidiana de $(nUA_i, nICust_i)$ para a origem $(0,0)$ é escolhido.

Devido à importância e complexidade associada, a próxima etapa mostra uma visão detalhada das atividades Geração de Agrupamentos e Definição do Número Bom de Agrupamentos. Posteriormente, é mostrada uma visão detalhada das atividade Seleção de Agrupamento e Seleção de Projeto respectivamente.

6.3.1 Geração de Agrupamentos-Definição do Número Bom de Agrupamentos

Estas atividades (ver Algoritmo 1) têm por objetivo determinar o número bom de agrupamentos, Nc. Na atividade Geração de Agrupamentos, a abordagem agrupamento hierárquico aglomerativo (AHC) foi aplicada para agrupar componentes similares da rede utilizando os atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente* considerando uma configuração específica. O índice RMSSTD é calculado em cada passo t do algoritmo hierárquico [47] em função do número de agrupamentos (Ag). Este valores serão normalizados utilizando a Equação 6.1:

$$nm_i = (m_i - m_{mn})/(m_{mx} - m_{mn}) \tag{6.1}$$

As notações utilizadas nesta equação são mostradas abaixo:

Algorithm 1 Calcula Número Bom de Agrupamentos

Input: CP #Conjunto de todos os Componentes#
dist(c1,c2) #Uma Função Distância#
Output: Nc #Número Bom de Agrupamentos#
int i, t, dm, Nc;
int k; #Número total de Componentes#
int p; #Número de Componentes no interior de um Agrupamento#
CP $\longrightarrow \mathbb{R}^3$;
CP = $\{cp_i \mid cp_i{=}(ri_i, cust_i, thp_i)\}$;
CR $\longrightarrow \mathbb{R}^2$;
CR = $\{cr_t \mid cr_t{=}(RMSSTD_t, Ag_t)\}$
C #Conjunto de Agrupamentos#
C = $\{c_i | c_i = \{cp_1, ..., cp_p\}\}$
#Cada componente cp_i de CP é colocado em um Agrupamento c_i#
for i=1 **to** k **do**
$c_i = \{cp_i\}$
end for
C = $\{c_1, ..., c_k\}$
while C.size ≥ 1 **do**
$t \longleftarrow C.size$
$RMSSTD_t = \sqrt[2]{\frac{\sum_{i=1...C.size} \sum_{\substack{j=1...d \\ k=1}}^{n_{ij}} (x_k - \bar{x}_{ij})^2}{\sum_{i=1...C.size} \sum_{j=1...d} (n_{ij}-1)}}$
$cr_t = (RMSSTD_t, Ag_t)$
if t = 1 **then**
Exit
end if
(c_{min1}, c_{min2}) = min $dist(c_i, c_j)$ **for all** $c_i, c_j \in C$
Remove c_{min1} and c_{min2} **from C**
add $\{c_{min1}, c_{min2}\}$ **to C**
end while
#Normalizar CR#
for t=1 **to** k **do**
NCR = $\{ncr_t = (nRMSSTD_t, nAg_t) | ncr_t = (\frac{RMSSTD_t - RMSSTD_{mn}}{RMSSTD_{mx} - RMSSTD_{mn}}, \frac{Ag_t - Ag_{mn}}{Ag_{mx} - Ag_{mn}})\}$
end for
#Escolhendo o Número Bom de Agrupamentos (Menor Distância Euclidiana)#
t $\longleftarrow$ 1;
$d_t = \sqrt{(nRMSSTD_t)^2 + (nAg_t)^2}$;
$dm \longleftarrow d_t$;
Nc $\longleftarrow$ t;
for t=2 **to** k **do**
$d_t = \sqrt{(nRMSSTD_t)^2 + (nAg_t)^2}$;
if $d_t < dm$ **then**
$dm \longleftarrow d_t$;
$Nc \longleftarrow t$;
else
$dm \longleftarrow dm$;
end if
end for
Return Nc;

- nm_i: *i-ésima* medida normalizada, tal que $0 \leq nm_i \leq 1$;
- m_i: *i-ésima* medida obtida sem normalização, tal que $m_i \in \mathbb{R}$;
- m_{mn}: Valor Mínimo da medida, tal que $m_{mn} \in \mathbb{R}$;
- m_{mx}: Valor Máximo da medida, tal que $m_{mx} \in \mathbb{R}$;

A quantidade de agrupamentos Nc que corresponde à menor distância Euclidiana de ($nRMSSTD_t$,nAg_t) para a origem, indica o número bom de agrupamentos. O valor de Nc será utilizado na próxima atividade, Seleção de Agrupamento.

Neste algoritmo, CP representa o conjunto de todos os componentes da rede e C representa o conjunto de agrupamentos. As notações utilizadas neste e nos próximos algorítmos são descritas abaixo:

- Nc: Número Bom de Agrupamentos, tal que Nc $\in \mathbb{Z}_+^*$;
- n_{ij}: Número de elementos no agrupamento i, dimensão j, tal que i $\in \mathbb{Z}_+^*$ e j $\in \mathbb{Z}_+^*$;
- $\bar{x}_{ij}$: Valor Esperado no agrupamento i e dimensão j;
- x_k: Elemento na posição k;
- RMSSTD, Ag: Corresponde aos valores do índice RMSSTD e do número associado de agrupamentos (Ag);
- nRMSSTD, nAg: Corresponde aos valores normalizados do índice RMSSTD (nRMSSTD) e do número associado de agrupamentos (nAg);
- ri, cust, thp: Corresponde aos valores dos atributos importância para confiabilidade (ri), custo do componente (cust) e vazão do componente (thp) de cada componente i;

- nri, ncust, nthp: Corresponde aos valores normalizados dos atributos importância para confiabilidade (nri), custo do componente (ncust) e vazão do componente (nthp) de cada componente *i*;
- mx_{ri}, mx_{cust}, mx_{thp}: Valores máximos correspondentes aos atributos importância para confiabilidade, custo do componente e vazão do componente considerando o conjunto de todos os componentes;
- d: Número de dimensões;
- Ic: Índice Ic de um Agrupamento.

6.3.2 Seleção de Agrupamento

Nesta atividade, o agrupamento selecionado representa os componentes com maior impacto sobre a confiabilidade do sistema (ver Algoritmo 2). Os pesos *w1*, *w2* e *w3*, correspondentes aos atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente*, devem assumir valores que priorizem a escolha deste agrupamento. O agrupamento selecionado possui o valor mais alto do índice I_c. Inicialmente, os atributos dos componentes de CP são normalizados, resultando no conjunto NCP. Então, o índice I_c é calculado em cada agrupamento, considerando o número bom de agrupamentos, Nc, calculado na atividade anterior. O Algoritmo 2 foi implementado e poderá executar esta atividade de maneira automatizada (ver Apêndice C).

6.3.3 Seleção de Projeto

Esta atividade tem por objetivo oferecer suporte ao processo de seleção do melhor projeto de uma infraestrutura em função de métricas técnicas e de negócios (ver Algoritmo 3).

Algorithm 2 Seleção de Agrupamento

Input: Nc #Número Bom de Agrupamentos#
Output: SC #Agrupamento Selecionado#
int i, g, p;
float Ic;
P $\longrightarrow \mathbb{N}^{*Nc}$; #Número de componentes em cada agrupamento *i*#
P = $\{p_1, ..., p_i\}$;
CP $\longrightarrow \mathbb{R}^3$;
CP = $\{cp_i | cp_i{=}(ri_i, cost_i, thp_i)\}$;
NCP $\longrightarrow \mathbb{R}^3$;
NCP = $\{ncp_i | ncp_i{=}(nri_i, ncost_i, nthp_i)\ \}$;
int k; #Número de Componentes#
$c_i = \{cp_1, ..., cp_{p_i}\}$; #Agrupamento *i*#
g $\longleftarrow$ 1;
$Ic \longleftarrow 0$;
#Normalizar CP#
for $cp_i \in CP$ **do**
  NCP = $\{ncp_i | ncp_i = (\frac{ri_i}{mx_{ri}}, \frac{cust_i}{mx_{cust}}, \frac{thp_i}{mx_{thp}})\}$
end for
#Selecione o Agrupamento Apropriado#
for i = 1 **to** Nc **do**
  $Ic_i = \frac{\sum_{m=1}^{p_i} \frac{((w_1 \times nri_m)+(w_2 \times ncost_m)+(w_3 \times nthp_m))}{\sum_{k=1}^{d} w_k}}{p_i}$
  if $Ic_i > Ic$ **then**
    $Ic \longleftarrow Ic_i$
    $g \longleftarrow i$
  end if
end for
SC = c_g;
$c_g = \{cp_1, ..., cp_p\}$
Return (SC); #Retorna o agrupamento selecionado com *p* componentes#

Diferentes cenários são construídos com base em diferentes combinações de parâmetros de entrada relacionados aos modelos de infraestrutura e de negócios. Cada cenário *i*, que representa um projeto candidato, possui diferentes métricas associadas. O conjunto de cenários construídos, SS = $\{ss_i | ss_i = (A_i(sc'), UA_i(sc'), Pf_i(sc'), ICust_i(sc'))\}$, é baseado nos componentes do conjunto SC', $SC' \subseteq SC$ e SC = $\{cp_1, ..., cp_p\}$. Por seu turno, SC' é obtido através da análise da função projeto de experimento fatorial 2^K, considerando-se SC. Esta análise seleciona componentes representados no agrupamento selecionado SC com maior impacto sobre a

disponibilidade do sistema. Este algoritmo foi implementado e poderá executar suas atividades de maneira automatizada (ver Apêndice C).

Os valores das métricas são, então, normalizados em cada cenário construído, utilizando a Equação 6.1, resultando no conjunto NSS.

Então, a escolha do melhor projeto de uma infraestrutura será realizada em função da abordagem selecionada. Para a primeira abordagem, as métricas normalizadas de disponibilidade (nA) e Lucro_Líquido (nLc) são consideradas. O projeto correspondente ao cenário com a maior distância Euclidiana de (nA_i, nLc_i) para a origem é escolhido. Na segunda abordagem, as métricas de indisponibilidade (nUA) e custo de infraestrutura (nICust) são consideradas. O projeto correspondente ao cenário com a menor distância Euclidiana de $(nUA_i, nICust_i)$ para a origem é selecionado. Este algoritmo pode ser adaptado para tratar com outras abordagens. Além disso, é importante observar que apenas uma das abordagens pode ser usada de cada vez, neste algoritmo.

6.4 Considerações Finais

Este capítulo detalhou a metodologia proposta a ser utilizada no processo de otimização para a escolha do melhor projeto de uma infraestrutura. A metodologia proporciona suporte para a escolha do melhor projeto considerando, simultaneamente, aspectos de desempenho, dependabilidade e de negócios. Foi detalhada cada uma de suas fases e a sequência de atividades correspondentes a estas fases. Além disso, devido à importância e complexidade, foram especificadas as atividades *geração de agrupamentos/definição do número bom de agrupamentos*, *seleção de agrupamento* e *seleção de projeto* através de seus respectivos algoritmos. Os algoritmos das atividades *seleção de agrupamento* e *seleção de projeto* foram adicionalmente implementados, possibilitanto suas execuções de maneira automática.

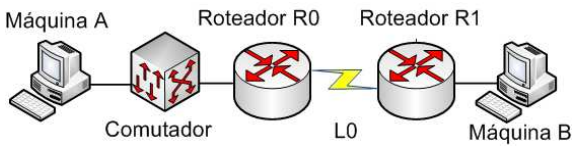
Algorithm 3 Seleção de Projeto

Input: $SS = \{ss_i | ss_i = \{((A_i(sc'), UA_i(sc'), Pf_i(sc'), ICost_i(sc'))\}$; #Conjunto de Cenários#
Output: SD = (m_1, m_2) #Projeto Selecionado#
int i, dl, l, dg, g;
int n; #Número de Opções#
#Primeira Abordagem m1=A, m2=Lc; Segunda Abordagem m1=UA, m2=ICust.#
$SS = \{ss_i | ss_i = (m1_i, m2_i)\}$;
normalize $ss_i, \forall i$;
for i = 1 **to** $n^{|SC'|}$ **do**
  $NSS = \{nss_i | nss_i = (nm1_i, nm2_i)\}$;
  $nss_i = ((\frac{m1_i - m1_{mn}}{m1_{mx} - m1_{mn}}), (\frac{m2_i - m2_{mn}}{m2_{mx} - m2_{mn}}))$;
end for
#Escolhendo o Melhor Projeto (Maior Distância Euclidiana)#
$dg \longleftarrow 0$;
$g \longleftarrow 1$;
for i=1 **to** $n^{|SC'|}$ **do**
  $d_i = \sqrt{(nA_i)^2 + (nLc_i)^2}$;
  if $d_i > dg$ **then**
    $dg \longleftarrow d_i$;
    $g \longleftarrow i$;
  else
    $dg \longleftarrow dg$;
  end if
end for
Projeto Selecionado - $(nA_g), (nLc_g)$;
#Escolhendo o Melhor Projeto (Menor Distância Euclidiana)#
$d_1 = \sqrt{(nUA_1)^2 + (nICust_1)^2}$;
$dl \longleftarrow d_1$;
l $\longleftarrow$ 1;
for i=2 **to** $n^{|SC'|}$ **do**
  $d_i = \sqrt{(nUA_i)^2 + (nICust_i)^2}$;
  if $d_i < dl$ **then**
    $dl \longleftarrow d_i$;
    $l \longleftarrow i$;
  else
    $dl \longleftarrow dl$;
  end if
end for
Projeto Selecionado - $(nUA_l), (nICust_l)$;

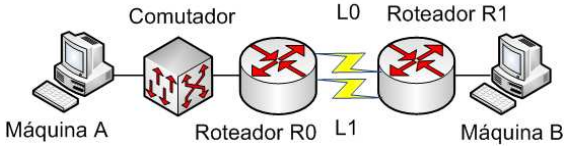
CAPÍTULO 7

Arquiteturas e Modelos

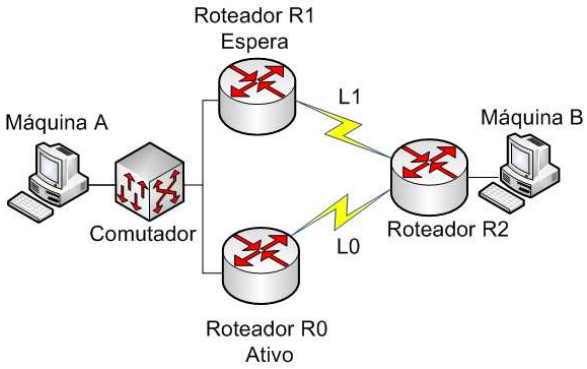
Neste capítulo iremos apresentar quatro arquiteturas junto com seus modelos de dependabilidade e de desempenho a serem utilizados nas Seções 8.1 e 8.2.



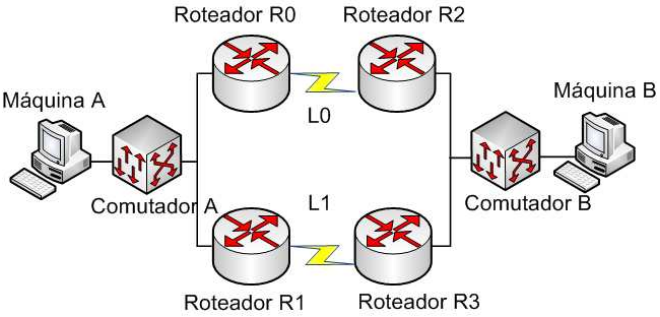
a) Arquitetura 1



b) Arquitetura 2



c) Arquitetura 3



d) Arquitetura 4

Figura 7.1: Arquiteturas 1, 2, 3 e 4

Estas arquiteturas (ver Figura 7.1) possuem uma ordem ascendente de redundância. Os respectivos modelos proporcionam suporte tanto para análises de dependabilidade e de desempenho sobre as infraestruturas, quanto para avaliação da metodologia proposta.

7.1 Arquiteturas

A Arquitetura 1 (ver Figura 7.1(a)) é composta de duas máquinas (máquina *A* e máquina *B*), dois roteadores (R0 e R1), um enlace (L0[3]) e um comutador. Esta arquitetura não possui redundância de componentes. Se um dos componentes falhar, todo o sistema cairá.

A Arquitetura 2 (ver Figura 7.1(b)) é composta de duas máquinas (máquina *A* e máquina *B*), dois roteadores (R0 e R1), dois enlaces redundantes (L0 e L1) e um comutador. Esta arquitetura considera os enlaces organizados na abordagem de redundância denominada de *warm standby*. Quando o enlace primário falhar, o enlace redundante (L1) irá assumir o papel do enlace primário. Após a restauração do enlace primário, o sistema retorna para sua condição inicial. Caso um dos roteadores (R0 ou R1) ou ambos os enlaces falhem, todo o sistema cairá.

A Arquitetura 3 (ver Figura 7.1(c)) é composta de duas máquinas (máquina *A* e máquina *B*), três roteadores (R0, R1 e R2), dois enlaces (L0 e L1) e um comutador. Esta arquitetura considera roteadores e enlaces organizados na abordagem de redundância denominada de *cold standby*. Quando pelo menos um dos componentes primários (R0 e L0) falhar, os componentes redundantes (R1 e L1) assumem o papel dos componentes primários. Após a restauração, o sistema retorna a sua condição inicial. Todo o sistema irá falhar quando o roteador R2 ou pelo menos um dos componentes primários e redundantes falharem.

A Arquitetura 4 (ver Figura 7.1(d)) é composta de duas máquinas (máquina *A* e máquina *B*), quatro roteadores (R0, R1, R2 e R3), dois enlaces (L0 e L1) e dois comutadores. Esta arquitetura considera roteadores e enlaces estruturados segundo o mecanismo de redundância denominada de *cold standby*. Quando pelo menos um dos componentes primários (R0, L0 e R2) falhar, os componentes em espera (R1, L1 e R3) assumem o papel dos componentes primários. Após a restauração, o sistema retorna a sua condição inicial. Todo o sistema irá falhar quando pelo menos um dos componentes primários e um dos componentes redundantes

[3]Os enlaces considerados nas arquiteturas 1, 2, 3 e 4 são de 128 Kbps e utilizam o protocolo PPP.

falhar.

7.2 Modelos de Dependabilidade

Para análise de dependabilidade destas arquiteturas, são propostos modelos do tipo RBD e SPN. Nestes modelos, serão considerados apenas os componentes representando os enlaces de WAN e os roteadores. Com relação aos modelos SPN, as transições temporizadas possuem uma distribuição exponencial (exp) e possuem uma semântica de disparo do tipo *single server* (ss).

7.2.1 Arquitetura 1 - Dependabilidade RBD

Este modelo representa uma estrutura em série composta de três componentes independentes apresentados na Figura 7.2. Os blocos representam os roteadores (R0 e R1) e o enlace (L0). Suponha que os tempos de falha e de reparo destes componentes obedecem a uma distribuição exponencial. A disponibilidade em estado estacionário para este sistema é calculada através da Equação 7.1 mostrada abaixo:

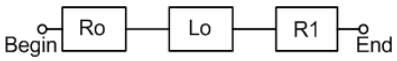


Figura 7.2: Arquitetura 1 - RBD

$$A_s = \frac{\mu_{R_0}}{\mu_{R_0} + \lambda_{R_0}} \times \frac{\mu_{L_0}}{\mu_{L_0} + \lambda_{L_0}} \times \frac{\mu_{R_1}}{\mu_{R_1} + \lambda_{R_1}} \tag{7.1}$$

onde λ_x e μ_x representam as taxas de falha e de reparo do componente x, x $\in \{R_0, L_0, R_1\}$. Por sua vez, a confiabilidade do sistema é calculada através da Equação 7.2 mostrada abaixo:

$$R_s(t) = \Pi_{i=1}^{n} R_i(t) \tag{7.2}$$

onde $R_i(t) = e^{(-\lambda_i \times t)}$ é a confiabilidade do componente i.

7.2.2 Arquitetura 2 - Dependabilidade SPN

Figura 7.3 apresenta o modelo de dependabilidade da arquitetura 2 que representa aspectos de tolerância a falhas baseados na abordagem de redundância *warm standby* (ver Figura 4.5). O modelo inclui dez lugares relevantes, a saber, R0_ON, R0_OFF junto com os pares correspondentes que representam os estados atividade e inatividade dos demais componentes (L0, L1 e R1). L0 representa o componente principal e Lugares L1_Sp_ON junto com L1_OSp_ON representam o componente redundante de L0 nos estados não operacional e operacional, respectivamente.

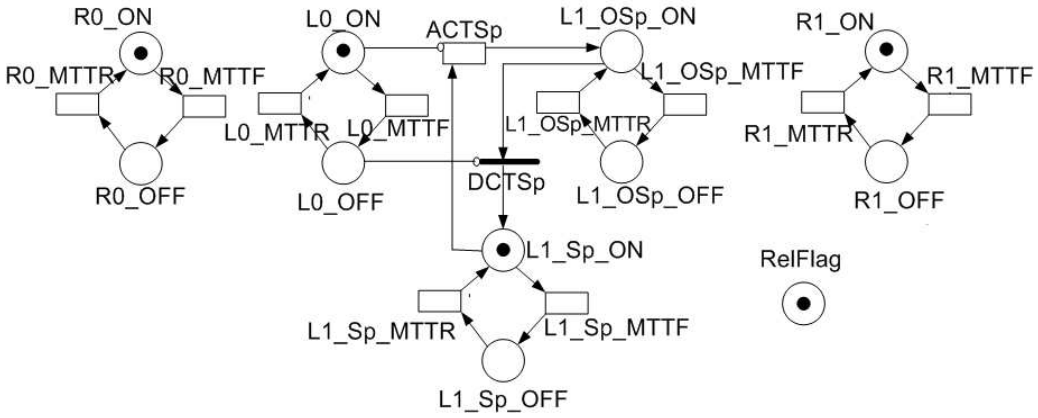


Figura 7.3: Arquitetura 2 - SPN

Conforme descrito na Seção 4.3.1, o lugar RelFlag premite a avaliação das métricas de confiabilidade e disponibilidade. Por sua vez, a transição ACTSp representa o tempo, em horas, de detecção da falha e de ativação do componente em espera (MTTA); a transição imediata DCTSp representa o retorno ao estado normal de operação. Transições ACTSp, R0_MTTR,

L0_MTTR, L1_Sp_MTTR, L1_OSp_MTTR e R1_MTTR possuem a seguinte expressão para o *atraso*: IF #RelFlag = 1 : 10^n ELSE Y; onde Y é o valor de MTTA da transição ACTSp ou o parâmetro MTTR de cada componente. Além disso, o parâmetro MTTF associado a cada componente é atribuído às transições R0_MTTF, R1_MTTF, L0_MTTF e L1_OSp_MTTF. É feita a suposição de que a transição L1_Sp_MTTF possui um valor de *atraso* 50% maior em relação à transição L1_OSp_MTTF.

Este modelo poderá computar a disponibilidade do sistema através da seguinte expressão: P{(#R0_ON = 1) AND (#L0_ON = 1 OR #L1_OSp_ON = 1) AND (#R1_ON = 1)}; a confiabilidade do sistema poderá ser calculada através da expressão: P{((#R0_ON = 1) AND (#L0_ON = 1) AND (#R1_ON = 1))}.

7.2.3 Arquitetura 3 - Dependabilidade SPN

A Figura 7.4 apresenta o modelo de dependabilidade da arquitetura 3. Ele é baseado no modelo SPN da Figura 4.3. Para os estados de atividade e inatividade dos componentes (R0, L0, R1, L1 e R2), é seguida a mesma representação mostrada no modelo SPN no qual ele é baseado. Conforme descrito anteriormente, o lugar RelFlag premite a avaliação das métricas de confiabilidade e disponibilidade.

O *atraso* associado às transições R0_ACTSp e L0_ACTSp representa o tempo, em horas, de detecção da falha e de ativação dos componentes em espera (MTTA); a transição imediata DCTSp representa o retorno ao estado normal de operação.

O parâmetro MTTF de cada componente é associado às transições R0_MTTF, L0_MTTF, R1_MTTF, L1_MTTF e R2_MTTF. Transições X_MTTR e X_ACTSp possuem a seguinte expressão para o *atraso*: IF #RelFlag = 1: 10^n ELSE Y; onde Y representa o valor de MTTA das transições R0_ACTSp e L0_ACTSp, ou o parâmetro MTTR, de cada componente, atribuído

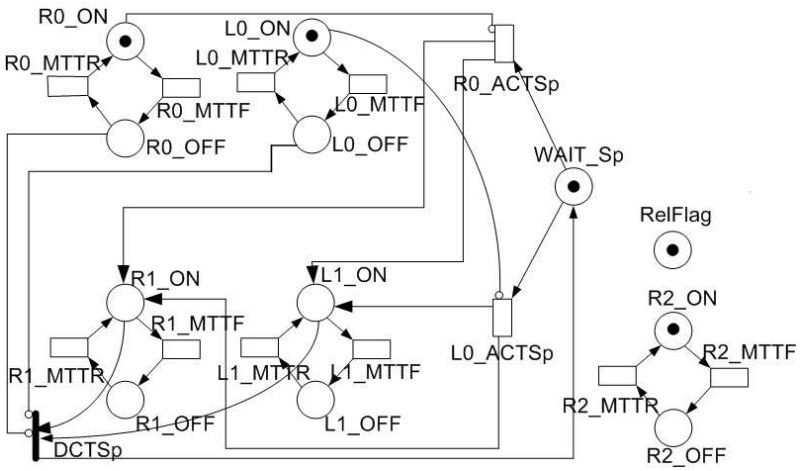


Figura 7.4: Arquitetura 3 - SPN

às transições R0_MTTR, L0_MTTR, R1_MTTR, L1_MTTR e R2_MTTR. Além disso, a expressão P{((#R0_ON = 1 AND #L0_ON = 1) OR (#R1_ON = 1 AND #L1_ON = 1)) AND #R2_ON = 1} denota a disponibilidade do sistema; a expressão P{((#R0_ON = 1 AND #L0_ON = 1) AND #R2_ON = 1)} calcula a confiabilidade do sistema.

7.2.4 Arquitetura 4 - Dependabilidade SPN

A Figura 7.5 apresenta o modelo de dependabilidade da arquitetura 4 que representa aspectos de tolerância a falhas baseados no mecanismo de redundância *cold standby* (ver Figura 4.3). O modelo inclui doze lugares relevantes, a saber, R0_ON, R0_OFF junto com os pares correspondentes que representam os estados atividade e inatividade dos demais componentes (L0, R2, R1, L1 e R3). O lugar RelFlag permite a avaliação das métricas de confiabilidade e disponibilidade.

Caso ocorra uma falha em um dos componentes principais (R0, L0, R2), a correspondente transição X_ACTSp é habilitada. O *atraso* associado às transições R0_ACTSp, L0_ACTSp e R2_ACTSp representa o tempo, em horas, de detecção da falha e de ativação dos componentes em espera (MTTA); a transição imediata DCTSp representa o retorno ao estado normal de

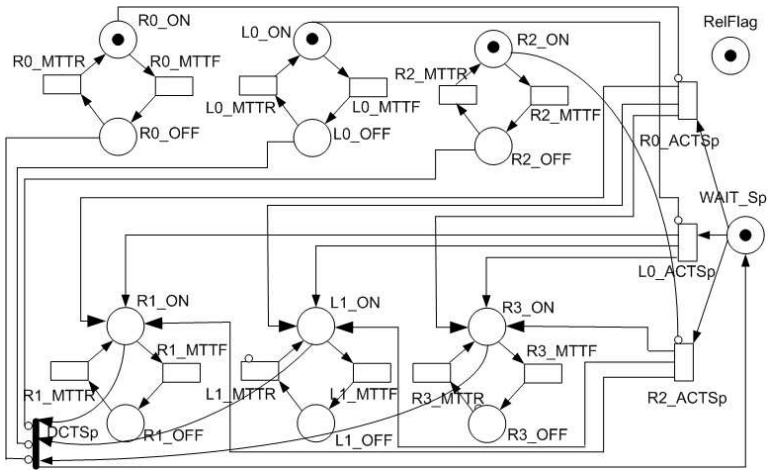


Figura 7.5: Arquitetura 4 - SPN.

operação.

O parâmetro MTTF de cada componente é atribuído às transições R0_MTTF, L0_MTTF, R2_MTTF, R1_MTTF, L1_MTTF e R3_MTTF. Transições X_MTTR e X_ACTSp possuem a seguinte expressão para o *atraso*: IF #RelFlag = 1: 10^n ELSE Y; aonde Y representa o valor de MTTA das transições R0_ACTSp, L0_ACTSp e R2_ACTSp, ou o parâmetro MTTR das transições R0_MTTR, L0_MTTR, R2_MTTR, R1_MTTR, L1_MTTR e R3_MTTR. A expressão P{((#R0_ON = 1 AND #L0_ON = 1 AND #R2_ON = 1) OR (#R1_ON = 1 AND #L1_ON = 1 AND #R3_ON = 1))} denota a disponibilidade do sistema; a expressão P{(#R0_ON = 1) AND (#L0_ON = 1) AND (#R2_ON = 1)} calcula a confiabilidade do sistema.

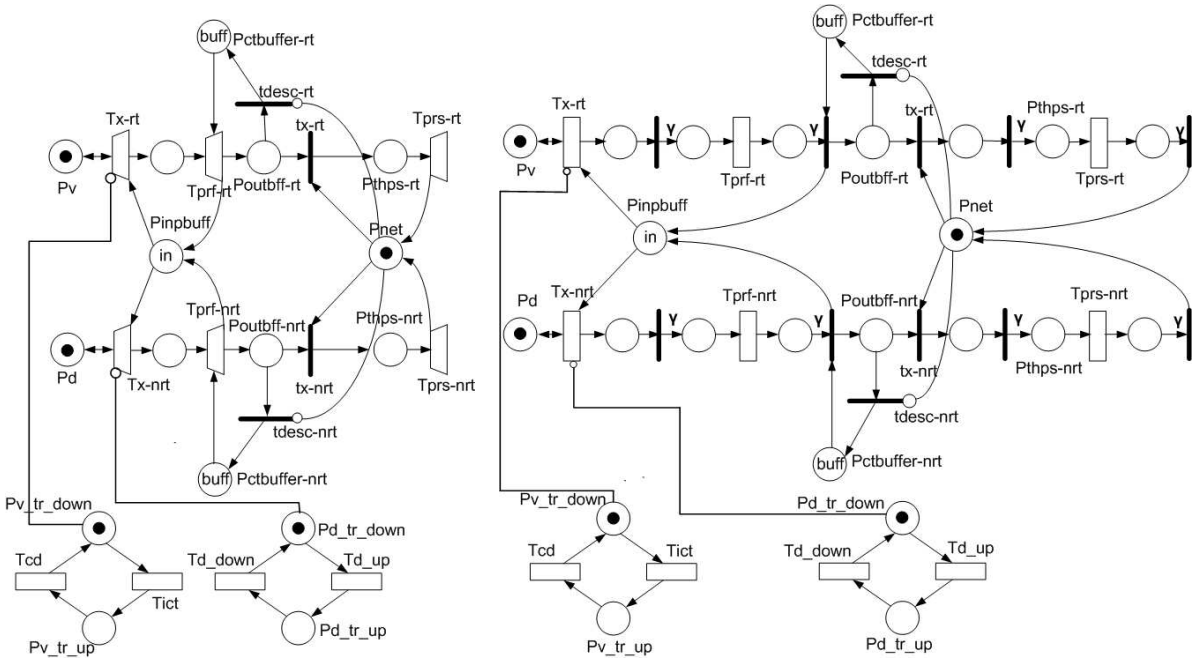
7.3 Modelos de Desempenho

Conforme mencionado na Seção 4.4, iremos utilizar modelos SPN para a avaliação de desempenho. Esta seção define um modelo SPN de desempenho, que está baseado no modelo mostrado na Seção 4.4.1, que poderá ser utilizado em cada uma das arquiteturas apresentadas. Inicialmente é mostrado o modelo abstrato e sua versão refinada. A validação referente à versão

refinada é detalhada no Apêndice B. Com relação a estes modelos, as transições temporizadas possuem uma distribuição exponencial (exp) e possuem uma semântica de disparo do tipo *single server* (ss).

7.3.1 Modelo Abstrato de Desempenho

O processo de construção do modelo de desempenho se inicia com a elaboração de seu modelo abstrato. O modelo proposto é mostrado na Figura 7.6(a). Ele proporciona suporte para diferentes tipos de análises considerando os tráfegos de dados e de voz. Extensões deste modelo para o suporte a diferentes tipos de tráfego são realizadas de maneira direta.



a) Modelo abstrato b) Modelo refinado

Figura 7.6: Modelos de desempenho

Os lugares P_v e P_d representam as aplicações na máquina A que geram tráfegos de voz e de dados. A transição estocástica genérica T*x-rt* representa o tempo entre pacotes de voz, que

varia de acordo com o CODEC utilizado. O lugar P*v_tr_up* representa o estado de transmissão de voz (#P*v_tr_up*=1). Por sua vez, o lugar P*v_tr_down* representa o estado sem transmissão de tráfego de voz. Transições exponenciais T*ict* e T*cd* representam, respectivamente, o início e o término de uma chamada de voz.

Além disso, a transição estocástica genérica T*x-nrt* representa o tempo entre pacotes de dados. O lugar P*d_tr_up* representa o estado de transmissão de dados (#P*d_tr_up* = 1). Nos períodos de silêncio, não há transmissão de dados (#P*d_tr_down* = 1). Transições exponenciais T*d_down* e T*d_up* representam, respectivamente, o início e o término de uma transmissão de dados.

O parâmetro atraso (*delay*) associado às transições estocásticas genéricas T*prf-rt* e T*prf-nrt* representa o atraso sofrido por um pacote de voz e por um pacote de dados na interface de entrada do roteador de entrada. Estes tempos são representados pelas variáveis *vf* e *df*. Os parâmetros atraso associados às transições estocásticas genéricas T*prs-rt* e T*prs-nrt* representam, respectivamente, o atraso sofrido por um pacote de voz e o atraso sofrido por um pacote de dados na interface de saída do roteador de entrada. Estes tempos são representados pelas variáveis *vs* e *ds*, respectivamente. Com relação ao roteador de entrada, o tamanho máximo, em pacotes, da fila na interface de entrada é representado pela variável *in* (lugar P*inpbuff*). Além disso, o lugar P*ctbuffer* representa o controle de acesso para a fila de saída no roteador de entrada. A variável correspondente *buff* (ver Figura 7.6(a)) é colocada para um valor muito alto de modo a proporcionar a obtenção de um modelo limitado. Neste modelo, as transições estocásticas genéricas possuem semântica do tipo *single-server* e são representadas graficamente por um trapézio.

Para a representação da política de enfileiramento *custom queuing* na interface de saída do roteador de entrada, o parâmetro peso (*weight*) associado às transições t*x-rt* e t*x-nrt*, representado pelas variáveis *w1* e *w2*, indica a probabilidade de disparo das transições imediatas habilitadas simultaneamente no ECS.

Tabela 7.1: Parâmetros das transições imediatas t*x-nrt*, t*x-rt*, t*desc-nrt* e t*desc-rt*

Transição	Função de Guarda	Prioridade	Peso	P. Enfileiramento
t*x-rt*	–	5	*w1*	Custom Queuing
t*x-nrt*	–	5	*w2*	Custom Queuing
t*desc-rt*	#P*outbff-nrt* $>$ *n1*	4	1	
t*desc-nrt*	#P*outbff-rt* $>$ *n2*	4	1	

Se o número de marcas nos lugares P*outbff-rt* e P*outbff-nrt* for maior que o tamanho padrão das filas correspondentes (variáveis *n1* e *n2*), então as transições imediatas t*desc-rt* e t*desc-nrt* disparam. Isto é modelado pela *função de guarda* (*enabling function*) designada para estas transições (ver Tabela 7.1).

7.3.2 Modelo Refinado de Desempenho

A Figura 7.6(b) mostra o modelo refinado de desempenho correspondente à política de enfileiramento *Custom Queuing*. Esta variante do modelo abstrato apresentado considera o refinamento de transições estocásticas genéricas com o objetivo de satisfazer o primeiro e o segundo momentos [20](ver Seção 3.7.1 - Aproximação por Fases). Para uma representação adequada das atividades representadas pelas transições T*x-rt* e T*x-nrt*, deverá ocorrer a substituição destas transições por transições exponenciais. Por outro lado, com relação às transições T*prf-rt*, T*prf-nrt*, T*prs-rt* e T*prs-nrt* deverá ocorrer a substituição destas transições por *s-transitions* do tipo Hipoexponencial, de acordo com os valores da média, desvio padrão e coeficiente de variação calculados para cada uma destas transições. Contudo, como o número de fases aumentou para um valor demasiado em todas as *s-transitions*, tem-se o problema da explosão do espaço de estados. Assumindo que o modelo de Erlang é um caso particular de um modelo Hipoexponencial, onde cada fase individual tem o mesmo valor, nós adotamos um pequeno número de estágios com *s-transitions* do tipo Erlang, com resultados muito próximos aos obtidos diretamente do sistema. É adotado um número de fases igual a quatro (valor de γ na Figura 7.6(b)).

Neste modelo (ver Figura 7.6(b)), as transições temporizadas possuem uma distribuição exponencial (exp) e possuem uma semântica de disparo do tipo *single server* (ss). Como parâmetros de entrada nós temos: atraso (*delay*) associado às transições T*x-rt* e T*x-nrt*, representando os tempos entre pacotes de voz e de dados, respectivamente; o atraso associado às transições T*prf-rt* e T*prf-nrt*, representando o atraso sofrido por um pacote de voz e por um pacote de dados na interface de entrada do roteador de entrada, variáveis *vf* e *df*; o atraso associado às transições T*prs-rt* e T*prs-nrt*, representando o atraso sofrido por um pacote de voz e por um pacote de dados na interface de saída do roteador de entrada, variáveis *vs* e *ds*; o parâmetro peso (*weight*), associado às transições t*x-rt* e t*x-nrt* (variáveis *w1* e *w2*), representando os recursos proporcionados a cada classe de tráfego; variáveis *n1* e *n2*, associadas às funções de guarda, representando o tamanho padrão das filas na interface de saída.

Tabela 7.2: Métricas avaliadas - Custom Queuing

Métrica	Equação
VSV	((P{#P*thps-rt*>0} × (1/vs))/γ)
VSD	((P{#P*thps-nrt*>0} × (1/ds))/γ)
PDV	((I_{vnom})-((P{#P*thps-rt*>0} × (1/vs))/γ) × *time*)
PDD	((I_{dnom})-((P{#P*thps-nrt*>0} × (1/ds))/γ) × *time*)
TFV	E{#Poutbff-rt}
TFD	E{#Poutbff-nrt}

A Tabela 7.2 detalha suas métricas, junto com as correspondentes expressões lógicas. As taxas de saída, em pps, são computadas através das métricas *VSV* (*Vazão Saída Voz*) e *VSD* (*Vazão Saída Dados*). São mostradas as expressões para as métricas Pacotes Descartados Voz (*PDV*) e Pacotes Descartados Dados (*PDD*) em um período de tempo específico, variável *time*. As taxas de entrada nominal de voz e de dados (I_{vnom} e I_{dnom}), em pps, são calculadas através do inverso dos atrasos associados às transições T*x-rt* e T*x-nrt*, respectivamente. Por fim, as métricas *TFV* (*Tamanho Fila Voz*) e *TFD* (*Tamanho Fila Dados*) calculam o tamanho médio, das filas de saída correspondentes aos tráfegos de voz e de dados.

7.4 Considerações Finais

Neste capítulo foram definidas quatro arquiteturas e seus modelos de dependabilidade e de desempenho. Foram utilizados tanto modelos baseados em espaço de estados, tais como SPN, quanto modelos modelos combinatoriais, tais como RBD. Estas arquiteturas e seus modelos correspondentes irão proporcionar suporte tanto para diferentes tipos de análises sobre as infraestruturas, quanto para avaliação da metodologia proposta.

CAPÍTULO 8

Estudo de Casos

Este capítulo é dividido em quatro etapas. Na primeira etapa, são analisados aspectos de desempenho e dependabilidade das arquiteturas apresentadas no capítulo anterior. Na segunda etapa, estas arquiteturas são utilizadas para proporcionar suporte a diferentes situações com o objetivo de avaliar a metodologia proposta. São consideradas as classes de redes convergentes detalhadas no Capítulo 5. Em cada classe, a melhor solução do projeto de infraestrutura em cada arquitetura avaliada, é determinada tanto para a abordagem em que a metodologia proposta não é aplicada, quanto para a abordagem considerando a sua utlização. As demais etapas, terceira e quarta, apresentam a aplicação desta metodologia, junto com a métrica VTp e os modelos de dependabilidade propostos, em uma rede real de uma empresa pública e de uma empresa privada, respectivamente. Nestas etapas, foram executadas análises comparativa entre as melhores soluções de projetos de infraestrutura das diferentes arquiteturas.

8.1 Caso I - Análise de Aspectos de Infraestrutura

É importante lembrar que este estudo pode ser considerado como o primeiro passo para um processo de planejamento de capacidade. São mostradas diferentes abordagens. O objetivo da primeira abordagem é o de analisar o comportamento de diferentes classes de tráfego, através da alocação dinâmica de recursos de transmissão em uma interface. Foi utilizado o modelo refinado de desempenho (ver Figura 7.6(b)) que pode ser aplicado a cada uma das arquiteturas apresentadas no Capítulo 7 (*testbeds*) para a obtenção, através de simulação, dos valores médios

das métricas de *vazão de saída de voz* e *vazão de saída de dados* em um determinado cenário. Ele pode representar tanto a política de enfileiramento *Custom Queuing* quanto a política de enfileiramento *Priority Queuing* na interface de saída (serial) do roteador de entrada. Conforme descrito no Apêndice B, a escolha desta interface para ser representada através deste modelo deve-se ao fato de que é neste ponto onde existe o gargalo da rede (disputa aos recursos de acesso à rede WAN). Nesta abordagem, foi considerada a representação da política de enfileiramento *Custom Queuing*.

Os diferentes recursos de transmissão são representados através do parâmetro peso (*weight*) designado para as transições t*x-rt* e t*x-nrt* (ver Tabela 7.1), as quais são relacionadas aos tráfegos de voz e de dados, respectivamente. Os pesos designados (variáveis *w1* e *w2*) em cada cenário investigado foram, respectivamente: cenário 1 (0,3 e 0,7); cenário 2 (0,4 e 0,6); cenário 3 (0,6 e 0,4); e cenário 4 (0,7 e 0,3). Estes valores foram escolhido para representar diferentes situações de alocação de recursos de transmissão.

Sob condições semelhantes, foram também executadas medições diretas em um dos sistemas. Estas medições foram efetuadas em paralelo às análises do modelo de desempenho. Foi considerada uma taxa de transmissão de dados de 50 Kbps (64 pps). Esta taxa de transmissão de dados em conjunto com a taxa de transmissão de voz (CODEC G.711, 33.3 pps) [99] alcançam um valor próximo ao limite de carga suportado pelo enlace serial (PPP), de 128 kbps.

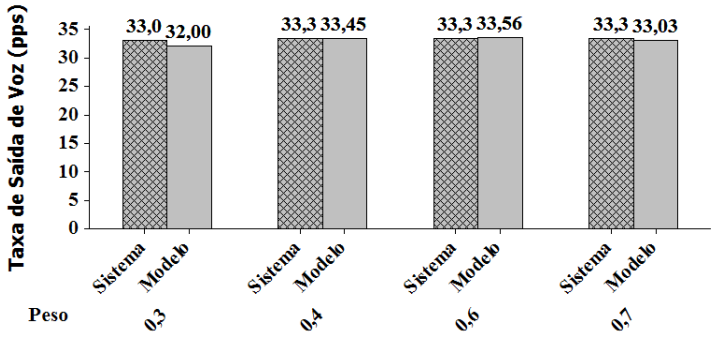


Figura 8.1: Taxa de saída de voz x Peso

As Figuras 8.1 e 8.2 mostram as taxas de saída de voz e de dados levando-se em conta os valores obtidos através de medição no próprio sistema (*Sistema*) e os valores obtidos por meio do modelo de desempenho (*Modelo*). A Figura 8.1 mostra a taxa de saída de voz em função dos recursos alocados. Os resultados indicam que apenas o peso de 0,3 acarreta um leve decréscimo na taxa de saída de voz. Com relação à taxa de saída de dados, a Figura 8.2 mostra que apenas em pesos iguais ou maiores que 0,7 não ocorrem perdas de pacotes. Baseado nestes resultados, o comportamento do tráfego de voz é comparado com comportamento do tráfego de dados. Após análise, os resultados mostraram que o tráfego de voz é pouco afetado pelo tráfego de dados em função da alocação de recursos. Além disso, este modelo também permite comparar o tráfego de voz com diferentes CODECs e o tráfego de dados com diferentes características, sendo possível a obtenção de diferentes resultados.

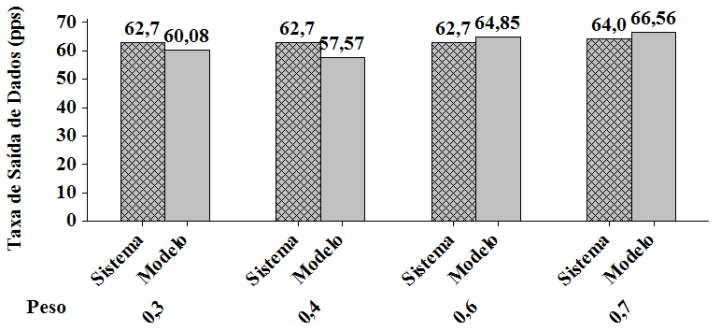


Figura 8.2: Taxa de saída de dados x Peso

Uma outra abordagem, que pode ser executada com modelos de dependabilidade propostos no Capítulo 7, refere-se ao estudo da disponibilidade de cada uma das arquiteturas tomadas como base tanto em função da variação do MTTF do enlace (ver Figura 8.3) quanto em função da variação do MTTR de cada um de seus componente (ver Figura 8.4). Para esta análise, foram utilizados roteadores com MTTF de 131.000h [45]. Para os valores de MTTA, foi utilizado um valor de 0,0027 [4] horas para a arquitetura 2 e um valor de 0,0416 [5] horas para a arquitetura 3 [43].

[4]Tempo médio de ativação do componente em espera no mecanismo de redundância *warm standby*.
[5]Tempo médio de ativação do componente em espera no mecanismo de redundância *cold standby*.

Inicialmente, foi escolhida a variação do MTTF do enlace em virtude de sua frequência de falhas ser maior que a dos roteadores. Foi considerado um valor de MTTR de doze (12) horas para todos os componentes (roteadores e enlaces) [72]. A Figura 8.3 mostra uma maior sensibilidade da arquitetura 1 (sem redundância) em função da variação do MTTF do enlace. Com relação às arquiteturas 2, 3 e 4, os mecanismos de redundância adotados fazem estas arquiteturas menos sensíveis à variação do MTTF destes componentes.

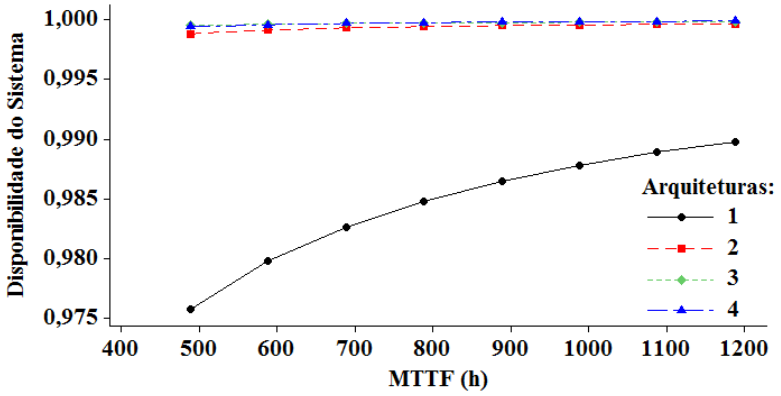


Figura 8.3: Disponibilidade do sistema de acordo com o MTTF do enlace

Então, o MTTR foi utilizado para esta análise pelo fato de que ele está intimamente relacionado com a política de manutenção adotada e, como consequência, com a disponibilidade do sistema. Figura 8.4 mostra a variação da disponibilidade do sistema em função do MTTR de cada componente para as diferentes arquiteturas (arquiteturas 1, 2, 3 e 4). Devido à ausência de redundância, a Arquitetura 1 apresenta uma maior sensibilidade à variação do MTTR. Foi considerado um MTTF de 1.188 h para os enlaces de WAN [43].

Por fim, uma outra abordagem relaciona os aspectos de dependabilidade com aspectos de negócios para a análise de cenários em função da disponibilidade e do custo total de aquisição (TCA). Foi considerada a arquitetura 3, mostrada no capítulo anterior. Esta arquitetura foi escolhida devido às suas peculiaridades em termos de componentes redundantes (R0/L0 e R1/L1) e componentes não redundantes (R2), além de seu mecanismo de redundância *cold standby*. Para proporcionar suporte a esta análise, foi considerado o modelo de dependabilidade SPN

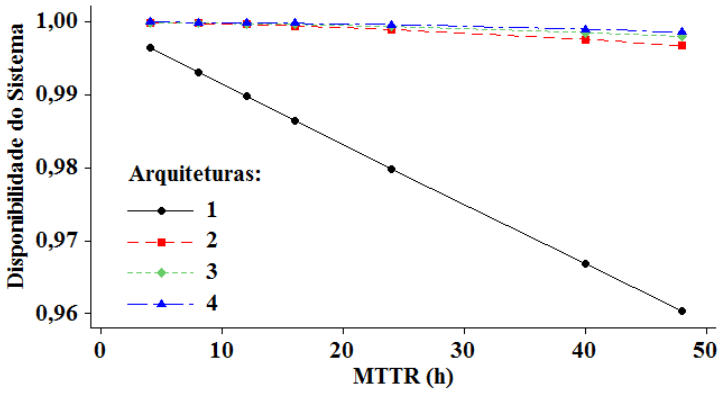


Figura 8.4: Disponibilidade do sistema de acordo com o MTTR dos componentes

proposto (ver Figura 7.4). Então, vamos analisar diferentes cenários, sendo que cada cenário corresponde a uma combinação com diferentes opções de equipamentos de rede (ver Tabela 8.1[6]), para a seleção do melhor cenário em termos de disponibilidade e TCA.

Tabela 8.1: Opções de equipamentos

Equipamento	MTTF (h)	TCA
Opção 1	131.000	Us$ 1.900
Opção 2	105.000	Us$ 1.095
Opção 3	68.000	Us$ 895

Tabela 8.2: Disponibilidade - Arquitetura 3

Cenário	MTTF R0	MTTF R1	MTTF R2	Disp.%	T.de Parada	TCA Us$
1	68.000h	68.000h	105.000h	99,975	2,208h	2.885
2	105.000h	105.000h	68.000h	99,969	2,742h	3.085
3	131.000h	131.000h	105.000h	99,975	2,199h	4.895
4	131.000h	131.000h	68.000h	99,969	2,742h	4.695
5	105.000h	105.000h	131.000h	99,977	2,0148h	4.090
6	68.000h	68.000h	68.000h	99,969	2,751h	2.685
7	105.000h	105.000h	105.000h	99,975	2,199h	3.285

Tabela 8.2[7] mostra os cenários definidos. Foi considerado um MTTF de 1.188 h para os enlaces de WAN e um valor de MTTA de 0,0416 h. Com relação ao MTTR, foi considerado um

[6]Os valores apresentados são baseados em informações fornecidas pelo fabricante.
[7]O tempo de parada mostrado nesta tabela corresponde ao período de 8.760 horas ou um (01) ano.

valor de 12h para todos os componentes considerados. Nestes cenários são considerados apenas os custos de aquisição relacionados aos roteadores (enlaces de WAN e comutadores não são considerados). Todos os cenários apresentam um valor muito próximo de disponibilidade com diferentes TCAs. Os cenários 1, 2, 6 e 7 apresentam os menores valores de TCA. Em particular, o cenário 6 apresenta o TCA mais baixo. É interessante observar o cenário 6 que apresenta um TCA de Us$ 2.685 e uma disponibilidade de 99,969 % e cenário 4, que apresenta um TCA de Us$ 4.695 e uma disponibilidade de 99,969 %. Estes cenários apresentam o mesmo valor de disponibilidade com os respectivos TCAs, relacionados aos componentes, significativamente diferentes.

8.2 Caso II - Avaliação da Metodologia

O objetivo desta etapa é o de avaliar a metodologia proposta. Para isto, iremos inicialmente mostrar a aplicação desta metodologia para a escolha do melhor projeto em cada uma das arquiteturas mostradas no capítulo anterior. Então, iremos escolher o melhor projeto de cada uma destas arquiteturas sem considerar a aplicação desta metodologia. Desta forma, podemos analisar sua utilização e vantagens.

Inicialmente, iremos considerar a aplicação desta metodologia. Após a definição dos modelos (ver Seções 7.2 e 7.3) e seguindo a fase de Seleção do Projeto (ver Seção 6.3.3), o atributo *importância para confiabilidade* foi calculado para cada componente considerando um tempo igual a 8.760 horas (ver Tabela 8.3). Então, a abordagem de agrupamento hierárquico aglomerativo foi aplicada. Os atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente* foram considerados a partir de uma configuração específica de componentes.

A Tabela 8.4 mostra os valores do índice RMSSTD em função do número de agrupamentos em cada passo *i* do algoritmo de agrupamento hierárquico aglomerativo em cada uma das

Tabela 8.3: Valores de importância para confiabilidade - Arquiteturas 1, 2, 3 e 4

Arquitetura	I^B_{L0}	I^B_{L1}	I^B_{R0}	I^B_{R1}	I^B_{R2}	I^B_{R3}
1	1,00	–	1,71E-5	–	1,82E-5	–
2	1,00	0,99	1,71E-5	–	1,82E-5	–
3	1,00	0,94	1,68E-5	0,00	1,68E-5	–
4	0,84	0,82	1,40E-5	0,00	1,50E-5	0,0

arquiteturas. Estes valores são normalizados utilizando a Equação 6.1. Então, o número de agrupamentos correspondente à menor distância Euclidiana entre os valores normalizados de RMSSTD e o correspondente número de agrupamentos ($nRMSSTD_i$,nAg_i), para a origem (0,0), indica o número bom de agrupamentos. De acordo com as Tabelas 8.4a, 8.4b, 8.4c e 8.4d, este número, em cada uma das arquiteturas, é igual a dois.

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	5.379	0,00	1,00	1,00
2	737	0,50	0,14	0,52
3	0	1,00	0,00	1,00

(a) Menor distância Euclidiana - Arquitetura 1

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	4730	0,00	1,00	1,00
2	1809	0,33	0,38	0,51
3	737	0,67	0,16	0,68
4	0	1,00	0,00	1,00

(b) Menor distância Euclidiana - Arquitetura 2

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	5146	0,00	1,00	1,00
2	1497	0,25	0,29	0,38
3	602	0,50	0,12	0,51
4	0	0,75	0,00	0,75
5	0	1,00	0,00	1,00

(c) Menor distância Euclidiana - Arquitetura 3

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	4192	0,00	1,00	1,00
2	1331	0,20	0,32	0,38
3	602	0,40	0,14	0,42
4	100	0,60	0,02	0,60
5	0	0,80	0,00	0,80
6	0	1,00	0,00	1,00

(d) Menor distância Euclidiana - Arquitetura 4

Tabela 8.4: Menor distância Euclidiana - Arquiteturas 1, 2, 3 e 4

Após a normalização dos atributos de cada um dos componentes, o índice I_c é calculado (ver Tabela 8.5). Por simplicidade, esta tabela omite detalhes dos agrupamentos. As variáveis w_1, w_2 e w_3 assumem os valores de 0,9, 0,05 e 0,05. Estes valores proporcionam suporte para a escolha do agrupamento cujos componentes possuem maior impacto sobre a confiabilidade do sistema.

Tabela 8.5: Valores do índice I_c - Arquiteturas 1, 2, 3 e 4

Arquitetura/Agrupamentos	Agrupamento1/I_c	Agrupamento2/I_c	Agrupamento Selecionado
1/2	0,05	0,95	Agrupamento2
2/2	0,05	0,94	Agrupamento2
3/2	0,05	0,91	Agrupamento2
4/2	0,05	0,94	Agrupamento2

Considerando os componentes representados no agrupamento selecionado, o projeto de experimento fatorial 2^k é analisado. Então, a atividade Análise de Experimento escolhe apenas os componentes com maior impacto sobre a disponibilidade do sistema (ver Tabela 8.6). Na atividade Definição e Avaliação de Cenários, estes componentes proporcionam suporte à construção de cenários, que são representados pela combinação de suas diferentes opções como parâmetros de entrada relativos aos modelos de infraestrutura e de negócios. A Tabela 8.7 mostra a opções correspondentes aos componentes selecionados (L0 e L1). Os cenários construídos serão avaliados baseados nos valores das métricas associadas. Estes valores de métricas são normalizados em cada cenário.

Tabela 8.6: Valores de projeto de experimento fatorial 2^k - Arquiteturas 1, 2, 3 e 4

Arquitetura	L0	L1	R0	R1	R2	R3	L0L1	Componentes Selecionados
1	99,91%	–	–	–	–	–	–	L0
2	41,53%	30,52%	–	–	–	–	28,04%	L0-L1
3	50,12%	26,14%	–	–	–	–	23,81%	L0-L1
4	41,04%	30,80%	–	–	–	–	28,15%	L0-L1

Tabela 8.7: SLA e custos - Enlaces

SLA	Disp.(%)	MTTF (h)	Custo Operacional(Us$/ano)	Vazão Máxima (pps)
SLA I	95,0	152	9.600	250
SLA II	98,0	392	14.400	250
SLA III	99,0	792	15.600	250
SLA IV	99,9	7.992	24.000	250

Inicialmente, para a classe de redes convergentes que gera diretamente receitas a partir de seus serviços, as métricas normalizadas de disponibilidade (nA) e lucro líquido (nLc) são

consideradas. Então, o projeto correspondente ao cenário *i* com a maior distância Euclidiana de (nA_i, nLc_i) para a origem (0,0) é escolhido. Para os cálculos da receita de infraestrutura, da multa e do lucro líquido foram utilizadas as Equações 5.4 [8], 5.8 e 5.9, respectivamente. Para o cálculo do custo de infraestrutura foi utilizada a Equação 5.2.

Por outro lado, para a classe de redes convergentes que não gera diretamente receitas a partir de seus serviços, as métricas normalizadas de indisponibilidade (nUA) e custo da infraestrutura (nICust) são consideradas. O projeto correspondente ao cenário *i* com a menor distância Euclidiana de $(nUA_i, nICust_i)$ para a origem (0,0) é escolhido. Para o cálculo do custo de infraestrutura foi utilizada a Equação 5.2.

Para o caso em que a metodologia proposta não for aplicada, os cenários serão construídos considerando todos os componentes com suas correspondentes opções [9] levando-se em conta as classes de redes convergentes descritas no Capítulo 5. As métricas consideradas serão as mesmas adotadas quando da aplicação de nossa metodologia. Cada cenário representa uma combinação diferente de parâmetros de entrada, relacionados aos modelos de infraestrutura e de negócios.

Tabela 8.8: Resultados da avaliação da metodologia utilizando a primeira abordagem

Arq./Opções	L.líquido(Us$) (N.Met.)	L.líquido(Us$) (Met.)	E.Relativo(%)	Cenários - N. Met.	Cenários - Met.
1/3	4.650.105	4.648.593	0,03	27	3
2/3	4.668.714	4.667.595	0,02	81	9
2/4	4.668.713	4.667.595	0,02	256	16
3/3	4.674.874	4.673.763	0,02	243	9
4/3	4.674.179	4.672.138	0,04	729	9

Então, os projetos escolhidos, sem normalização, em cada uma das arquiteturas são mostrados nas Tabelas 8.8 e 8.9. Com relação à classe de redes convergentes que gera diretamente

[8] Com relação a este modelo, a vazão calculada, a partir do modelo de desempenho, do tráfego de voz e de dados é de 29,89 pps e 0,55 pps, respectivamente

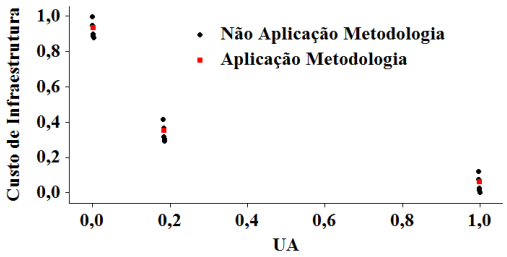
[9] O MTTF, TCA e vazão das diferentes opções de roteadores utilizadas nesta etapa são, respectivamente: Opção 1, 68.000 horas, Us$ 895 e 7.000 pps; Opção 2, 105.000 horas, Us$ 1.095 e 8.000 pps; Opção 3, 131.000 horas, Us$ 1.900 e 8.500 pps; Opção 4, 169.000 horas, Us$ 4.000 e 12.000 pps. Os valores apresentados são baseados em informações fornecidas pelo fabricante.

Tabela 8.9: Resultados da avaliação da metodologia utilizando a segunda abordagem

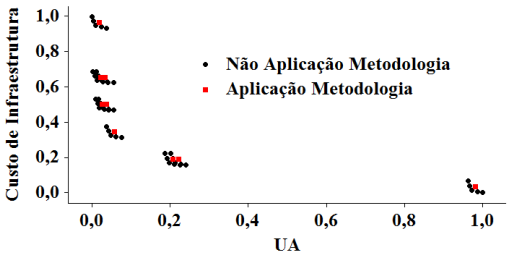
Arq./Opções	ICust(Us$) (N. Met.)	ICust(Us$) (Met.)	E.Relativo(%)	Cenários - N. Met.	Cenários - Met.
1/3	16.190	17.195	6,21	27	3
2/3	26.190	26.795	2,31	81	9
2/4	27.390	27.995	2,23	256	16
3/3	26.885	28.695	6,7	243	9
4/3	28.780	30.790	6,90	729	9

receitas a partir de seus serviços, a Tabela 8.8 mostra os valores da métrica lucro líquido (Lc) correspondentes aos projetos selecionados em cada arquitetura, tanto para o caso de utilização da metodologia (Met.) como para o caso de não utilização da metodologia proposta (N. Met.). Particularmente, na arquitetura 2, a melhor opção de projeto foi escolhida considerando 3 e 4 opções para cada um dos componentes selecionados na atividade Análise de Experimento de nossa metodologia. Os resultados obtidos através da aplicação da metodologia são muito próximos aos resultados computados levando-se em consideração todos os componentes da arquitetura. O máximo erro relativo entre os melhores projetos para esta classe de redes convergentes é de 0,04 %.

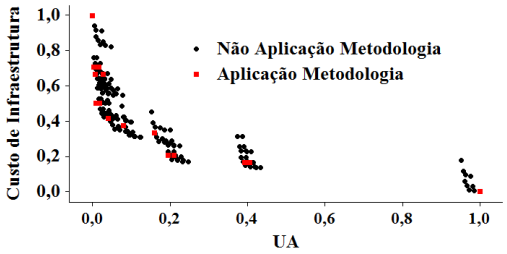
Com relação à classe de redes convergentes que não gera diretamente receitas a partir de seus serviços, a Tabela 8.9 mostra os valores da métrica custo de infraestrutura (ICust) correspondentes aos projetos selecionados em cada arquitetura, tanto para o caso de utilização da metodologia (Met.) como para o caso de não utilização de nossa metodologia (N. Met.). Na arquitetuta 2, da mesma maneira que na primeira abordagem, a melhor opção de projeto foi escolhida considerando 3 e 4 opções para cada um dos componentes selecionados na atividade Análise de Experimento de nossa metodologia. De forma semelhante à abordagem anterior, os resultados obtidos através da aplicação da metodologia são muito próximos aos resultados computados levando-se em consideração todos os componentes em cada arquitetura. O máximo erro relativo entre os melhores projetos é de 6,9 %. A Figura 8.5 mostra os valores normalizados de custo de infraestrutura (ICust) em função da indisponibilidade (UA), em cada



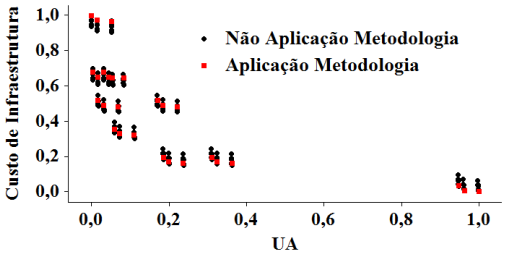
a) Custo de Infraestrutura x UA - Arquitetura 1



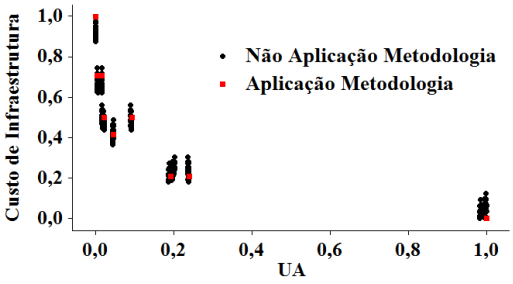
b) Custo de infraestrutura x UA - Arquitetura 2 (3 Opções)



c) Custo de infraestrutura x UA - Arquitetura 2 (4 Opções)



d) Custo de infraestrutura x UA - Arquitetura 3



e) Custo de infraestrutura x UA - Arquitetura 4

Figura 8.5: Custo de infraestrutura x UA - Arquiteturas 1, 2, 3 e 4

um dos cenários construídos, considerando os casos da adoção ou não da metodologia proposta, em cada uma das arquiteturas. Os projetos selecionados em cada uma das arquiteturas são apresentados na Tabela 8.9.

Outro aspecto a ser realçado é o pequeno número de cenários construídos através da aplicação de nossa metodologia (Cenários - Met.) em comparação com o número de cenários que

foram necessários serem construídos não aplicando a nossa metodologia (Cenários - N. Met.). Estes fatos reforçam nossos argumentos de que esta metodologia reduz significativamente a complexidade da escolha do melhor projeto de uma infraestrutura de redes convergentes (ver Tabelas 8.8 e 8.9).

8.3 Caso III - Aplicação da Metodologia

Este estudo considera a rede de computadores de uma instituição federal de educação superior de ensino e de pesquisa. Esta rede não gera diretamente receitas e lucros financeiros, a partir de seus serviços. Desta maneira, as métricas de indisponibilidade (UA) e custo de infraestrutura (ICust) são consideradas em seu projeto de infraestrutura. Para o cálculo do custo de infraestrutura foi utilizada a Equação 5.2[10]. A metodologia proposta, juntamente com a métrica VTp, irá proporcionar suporte a uma análise comparativa entre soluções de projetos de diferentes arquiteturas. A Figura 8.6 mostra a primeira arquitetura (Arquitetura A11) que consiste de um nó central, CORE, ao qual os outros nós estão conectados. O nó central, os nós folha e os enlaces entre eles formam uma topologia estrela. Seu modelo de dependabilidade (confiabilidade e disponibilidade) é hierárquico. O nível superior é um RBD. Cada nó é modelado como um bloco, estruturado em série com os demais blocos. Os modelos do nível mais baixo são do tipo SPN ou RBD.

Os parâmetros de entrada para estes modelos são mostrados na Tabela 8.10. Estes valores estão baseados em [78] e foram utilizados para calcular as métricas de dependabilidade. Nesta tabela, (C) representa o componente nó central, CORE, e (NF) representa um nó folha. Enlace tipo I representa os enlaces de *fibra*; enlace tipo II representa enlaces de *radio*; enlace tipo III representa os enlaces que utilizam *cordões ópticos (em inglês, Patchcord)*. O mo-

[10] Os valores considerados são baseados em informações fornecidas pelo fabricante

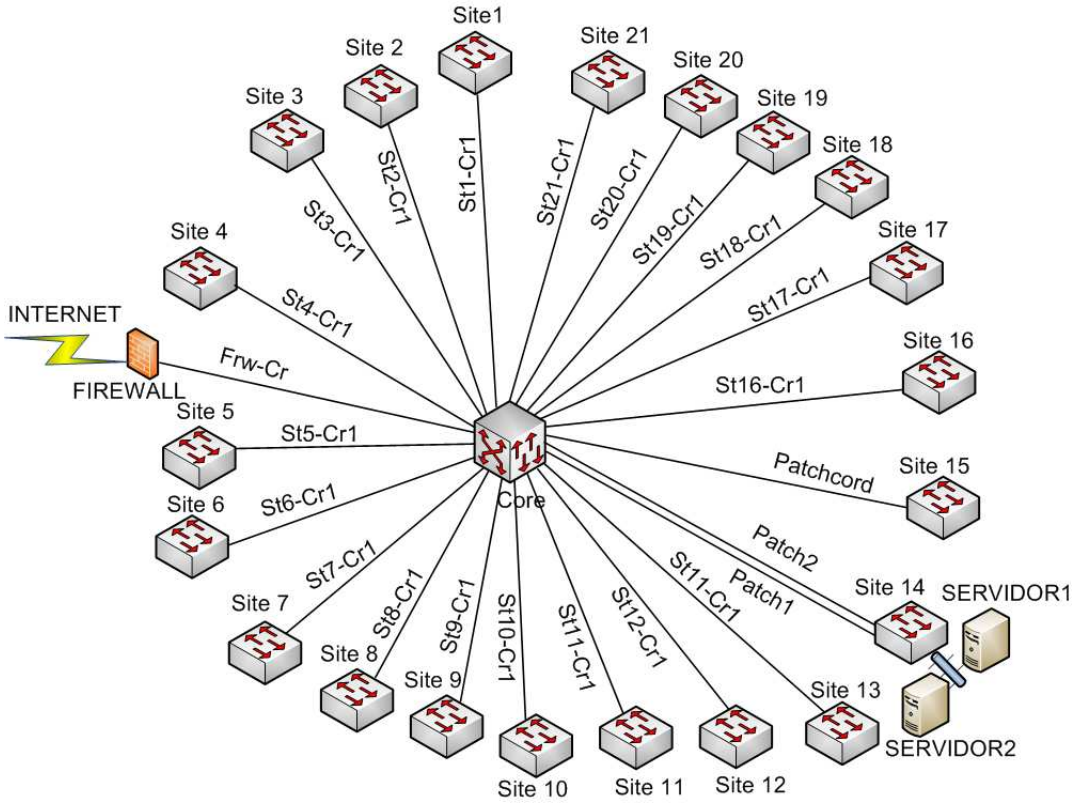


Figura 8.6: Arquitetura A11

delo de dependabilidade (confiabilidade e disponibilidade) do nó central corresponde a uma estrutura série/paralelo (ver Seção 4.3.4 - Estruturas Série/Paralelo). Por sua vez, o modelo de dependabilidade (confiabilidade e disponibilidade) de um servidor/firewall e de um nó folha corresponde a uma estrutura em série. Por fim, os componentes *Patch1* e *Patch2* (Patch (1,2)) consideram uma redundância de enlaces baseada em uma abordagem denominada de *warm standby* (ver Figura 4.5).

O valor da disponibilidade calculada desta rede é de $91,47\%$ - uma baixa disponibilidade. Considerando os componentes mais importantes em termos de confiabilidade na arquitetura A11 (ver Tabela 8.11[11]), duas diferentes arquiteturas foram propostas. Inicialmente, uma arquitetura (Arquitetura A21) foi proposta considerando redundância no componente St1-Cr1

[11] Foi considerado um tempo igual a 8.760 horas para o cálculo do atributo importância para confiabilidade. Por simplicidade, serão mostrados apenas os componentes com os maiores valores deste atributo.

Tabela 8.10: Componentes da rede

Componente	MTTF	MTTR	A(%)	Un	Componente	MTTF	MTTR	A(%)	Un
F._Potência(C)	746.249h	1,00h	99,99	2	P.Interface(C)	195.240h	4,00h	99,99	1
P.Proc./S.Op.(C)	9.041h	0,50h	99,99	2	Chassis/P.Mãe(C)	182.207h	4,00h	99,99	1
Core	94.194h	3,99h	99,99	1	Nó Folha(NF)	16.667h	4,08h	99,97	1
Chassis/P.Mãe(NF)	300.000h	24,00h	99,99	1	S.Op.(NF)	20.000h	0,10h	99,99	1
F._Potêencia(NF)	150.000h	24,00h	99,98	1	Servidor(Firewall)	17.247h	0,60h	99,99	1
Enlace-Tipo_I	59.988h	12,00h	99,98	1	Enlace-Tipo_II	294h	6,00h	98,00	1
Enlace-Tipo_III	280.000h	4,00h	99,99	1					

(ver Figura 8.6). Outra arquitetura (arquitetura A31) foi proposta considerando redundância nos componentes St1-Cr1 e St2-Cr1. Os respectivos modelos de dependabilidade (confiabilidade e disponibilidade) também adotam uma abordagem hierárquica. No nível superior, tem-se um modelo do tipo RBD e no nível mais baixo, tem-se modelos do tipo SPN ou RBD. Os componentes redundantes Patch1/Patch2 e St1-Cr1/St1-Cr2 na arquitetura A21 e Patch1/Patch2, St1-Cr1/St1-Cr2 e St2-Cr1/St2-Cr2 na arquitetura A31, consideram o mecanismo *warm standby* (ver Figura 4.5).

Após a definição dos modelos de dependabilidade e da utilização destes modelos com o intuito de determinar o atributo *importância para confiabilidade* de cada componente em cada arquitetura, iremos seguir as demais atividade da fase de Seleção do Projeto da metodologia proposta.

Tabela 8.11: Valores de importância para confiabilidade - Arquiteturas A11, A21 e A31

Arquitetura A11		Arquitetura A21		Arquitetura A31	
Imp._Conf.	Valor	Imp._Conf.	Valor	Imp._Conf.	Valor
$I^B_{St1-Cr1}$	1,00	$I^B_{St1-Cr1}$	0,49	$I^B_{St1-Cr1}$	0,49
$I^B_{St2-Cr1}$	1,00	$I^B_{St1-Cr2}$	0,49	$I^B_{St1-Cr2}$	0,49
$I^B_{St3-Cr1}$	1,00	$I^B_{St2-Cr1}$	1,00	$I^B_{St2-Cr1}$	0,49
$I^B_{St4-Cr1}$	1,00	$I^B_{St3-Cr1}$	1,00	$I^B_{St2-Cr2}$	0,49
$I^B_{Server1}$	1,91E-13	$I^B_{St4-Cr1}$	1,00	$I^B_{St3-Cr1}$	1,00
$I^B_{Server2}$	1,91E-13	$I^B_{Server1}$	1,91E-13	$I^B_{St4-Cr1}$	1,00
I^B_{Core}	1,33E-13	I^B_{Core}	1,26E-13	I^B_{Core}	1,26E-13

Então, a abordagem de agrupamento foi aplicada na atividade Geração de Agrupamentos.

Os atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente* foram considerados a partir de uma configuração específica de componentes. Então, o valor do índice RMSSTD é calculado em função do número de agrupamentos em cada passo *i* do algoritmo de agrupamento hierárquico em cada uma das arquiteturas (ver Tabela 8.12). Estes valores são normalizados utilizando a Equação 6.1. O número de agrupamentos correspondente à menor distância Euclidiana dos valores normalizados para a origem indica o número bom de agrupamentos. De acordo com as Tabelas 8.12a, 8.12b e 8.12c, a quantidade de agrupamentos selecionada em cada arquitetuta é de quatro agrupamentos.

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	179288	0,00	1,00	1,00
2	112226	0,14	0,63	0,64
3	66131	0,29	0,37	0,47
4	20305	0,43	0,11	0,44
5	8787	0,57	0,05	0,57
6	2546	0,71	0,01	0,71
7	0	0,86	0,00	0,86
8	0	1,00	0,00	1,00

(a) Menor distância Euclidiana - Arquitetura 11

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	173745	0,00	1,00	1,00
2	121525	0,13	0,70	0,71
3	65665	0,25	0,38	0,45
4	20469	0,38	0,12	0,39
5	9158	0,50	0,05	0,50
6	2516	0,63	0,01	0,63
7	0	0,75	0,00	0,75
8	0	0,88	0,00	0,88
9	0	1,00	0,00	1,00

(b) Menor distância Euclidiana - Arquitetura 21

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	182872	0,00	1,00	1,00
2	116546	0,13	0,64	0,65
3	64795	0,25	0,35	0,43
4	20290	0,38	0,11	0,39
5	9331	0,50	0,05	0,50
6	2487	0,63	0,01	0,63
7	0	0,75	0,00	0,75
8	0	0,88	0,00	0,88
9	0	1,00	0,00	1,00

(c) Menor distância Euclidiana - Arquitetura 31

Tabela 8.12: Menor distância Euclidiana - Arquiteturas 11, 21 e 31

Após a normalização dos atributos dos componentes, o índice I_c é calculado (ver Tabela 8.13). Para os pesos w_1, w_2 e w_3 utilizamos os valores de 0,9, 0,05 e 0,05 respectivamente.

Tabela 8.13: Valores do índice I_c - Arquiteturas A11, A21 e A31

Arq./Agrup.	Agrupamento1/I_c	Agrupamento2/I_c	Agrupamento3/I_c	Agrupamento4/I_c	Agrupamento Selecionado
A11/4	0,54	0,03	0,02	0,05	Agrupamento1
A21/4	0,48	0,03	0,02	0,04	Agrupamento1
A31/4	0,43	0,03	0,02	0,04	Agrupamento1

Estes valores oferecem suporte para a escolha do agrupamento cujos componentes possuem maior impacto sobre a confiabilidade do sistema. Por simplicidade, esta tabela omite detalhes dos agrupamentos.

Tabela 8.14: Valores de projeto de experimento fatorial 2^k - Arquiteturas A11, A21 e A31

Arq.	St1-Cr1	St1-Cr2	St2-Cr1	St2-Cr2	St3-Cr1	St4-Cr1	Servidor1	Servidor2	Firewall	Componentes Selecionados
A11	24,99%	–	24,99%	–	24,99%	24,99%	≈ 0	≈ 0	≈ 0	St1-Cr1,St2-Cr1,St3-Cr1,St4-Cr1
A21	≈ 0	≈ 0	33,34%	–	33,34%	33,34%	≈ 0	≈ 0	≈ 0	St2-Cr1,St3-Cr1,St4-Cr1
A31	≈ 0	≈ 0	≈ 0	≈ 0	49,99%	49,99%	≈ 0	≈ 0	≈ 0	St3-Cr1,St4-Cr1

O projeto de experimento fatorial 2^K é analisado considerando os componentes no agrupamento selecionado. Então, a atividade Análise de Experimento escolhe apenas os componentes com maior impacto sobre a disponibilidade em cada arquitetura (ver Tabela 8.14). Para a arquitetura A11, enlaces St1-Cr1, St2-Cr1, St3-Cr1 e St4-Cr1 têm igualmente a maior importância sobre a disponibilidade do sistema. Por sua vez, St2-Cr1, St3-Cr1 e St4-Cr1 para a arquitetura A21 junto com St3-Cr1 e St4-Cr1 para a arquitetura A31 são os mais importantes para a disponibilidade destes sistemas. Na atividade Definição e Avaliação de Cenários, os componentes escolhidos proporcionam suporte à construção de diferentes cenários, que são representados pela combinação de suas diferente opções como parâmetros de entrada considerando os modelos de infraestrutura e de negócios. Tabela 8.15 mostra as diferentes opções correspondentes aos componentes selecionados.

A Figura 8.7 mostra os valores normalizados das métricas UA e ICust obtidos a partir dos cenários construídos. O projeto correspondente ao cenário com a menor distância Euclidiana é escolhido. A solução da arquitetura A11 possui indisponibilidade de 0,0427 (4,27%) a um

Tabela 8.15: SLA e custos - Enlaces tipo II

SLA	Disp.(%)	MTTF(h)	Custo Operational(Us$/ano)	Vazão(pps)
I	98,00	294	100.000	19.531
II	99,00	1.194	150.000	19.531
III	99,90	5.994	187.500	19.531

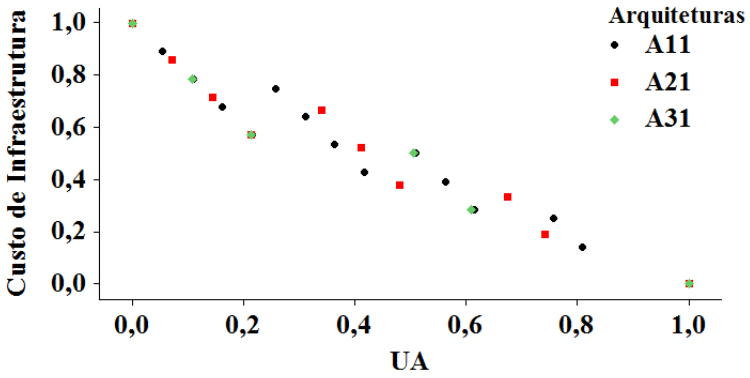


Figura 8.7: Custo de infraestrutura x UA - Arquiteturas A11, A21 e A31

custo de infraestrutura de Us$ 1.116.000. Na arquitetura A21, a solução encontrada tem indisponibilidade de 0,0235 (2,35%) a um custo de infraestrutura de Us$ 1.216.000. Por fim, a solução selecionada na arquitetura A31 tem indisponibilidade de 0,0189 (1,89%) a um custo de infraestrutura de Us$ 1.266.000. A Tabela 8.16 mostra as soluções selecionadas para as diferentes arquiteturas analisadas.

Então, podemos utilizar a métrica VTp para comparar cenários de diferentes arquiteturas. Por exemplo, considerando a melhor solução em cada arquitetura, se alguém escolhe a arquitetura A21 a partir da arquitetura A11, paga-se Us$100.000 a mais ($\Delta ICust$) e será obtido uma variação do *tempo de parada* de 605.556 segundos (ΔD). Se alguém escolhe arquitetura A31 sobre arquitetura A11, paga-se Us$150.000 a mais ($\Delta ICust$) e será obtido uma variação do *tempo de parada* de 750.556 segundos (ΔD). Com relação à métrica VTp, a arquitetura A21 tem um incremento de 6,05 segundos por unidade monetária gasta sobre a arquitetura A11. De maneira semelhante, a arquitetura A31 tem um incremento de 5,01 segundos por unidade monetária gasta sobre a arquitetura A11. Pode-se notar, considerando-se a métrica VTp, que a

mudança a partir da arquitetura A11 para a arquitetura A21 será mais atrativa em comparação com a mudança da arquitetura A11 para a arquitetura A31.

Tabela 8.16: Soluções de cada arquitetura

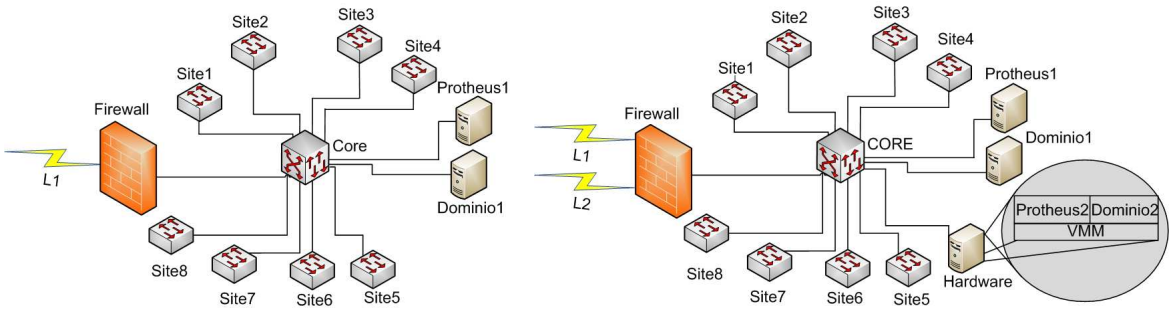
Arquitetura	Indisponibilidade	Custo de Infraestrutura
A11	4,27%	Us$ 1.116.000
A21	2,35%	Us$ 1.216.000
A31	1,89%	Us$ 1.266.000

8.4 Caso IV - Aplicação da Metodologia

Agora, iremos considerar uma rede de computadores de uma indústria de alimentos localizada no nordeste do Brasil. Esta rede não gera diretamente receitas e lucros. Desta forma, as métricas de indisponibilidade (UA) e custo de infraestrutura (ICust) são consideradas com relação à escolha do melhor projeto. Para o cálculo do custo de infraestrutura foi utilizada a Equação 5.2[12]. A Figura 8.9(a) mostra a arquitetura base (arquitetura A12) que consiste de um nó central, CORE, ao qual todos os outros nós estão conectados. O nó central, os nós folha e os enlaces entre eles formam uma topologia estrela. De maneira semelhante ao estudo de caso anterior, o modelo de dependabilidade (confiabilidade/disponibilidade) desta arquitetura adota uma abordagem hierárquica, onde o nível superior é do tipo RBD e os modelos do nível mais baixo são do tipo SPN ou RBD. Os enlaces UTP que interconectam os nós folha ao nó central não foram considerados no processso de modelagem.

Os parâmetros de entrada para estes modelos são mostrados na Tabela 8.17. Estes valores foram baseados em [52, 78] e foram utilizados para calcular as métricas de dependabilidade. Nesta tabela, (C) representa o componente nó central, CORE, e (NF) representa um nó folha. O modelo de dependabilidade (confiabilidade/disponibilidade) do nó central corresponde a uma

[12]Os valores apresentados são baseados em informações fornecidas pelo fabricante



a) Arquitetura A12 b) Arquiteturas A22 e A32

Figura 8.8: Arquiteturas A12, A22 e A32

estrutura série/paralelo (ver Seção 4.3.4 - Estruturas Série/Paralelo). Os modelos de dependabilidade (confiabilidade/disponibilidade) de um nó folha e de um servidor/firewall correspondem a uma estrutura em série.

Tabela 8.17: Componentes da rede

Componente	MTTF	MTTR	Un	Componente	MTTF	MTTR	Un
F._Potência(C)	746.249h	4,00h	2	P.Interface(C)	195.240h	4,00h	1
P.Proc./S.Op.(C)	9.041h	0,50h	2	Chassis/P.Mãe(C)	182.207h	4,00h	1
Core	94.194h	3,99h	1	Nó_Folha(NF)	16.667h	1,08h	1
Chassis/P.Mãe(NF)	300.000h	6,00h	1	S.Op.(NF)	20.000h	0,10h	1
F._Potência(NF)	150.000h	6,00h	1	Servidor	1.414h	0,99h	1
Firewall	15.465h	0,98h	1				

A arquitetura A12 tem uma disponibilidade de $99,29\%$. De acordo com a classificação para disponibilidade proposta em [36], pode-se verificar que esta arquitetura está entre as classes *gerenciada* e *bem-gerenciada*. Contudo, como aplicações *online* e de missão crítica não podem tolerar esta disponibilidade, novas arquiteturas foram propostas. Considerando os componentes mais importantes em termos de confiabilidade (ver Tabela 8.18[13]) na arquitetura A12, duas arquiteturas diferentes foram, então, propostas (ver Figura 8.9(b)). Uma nova arquitetura (arquitetura A22) considera uma redundância no componente *L1*. Outra arquitetura (arquite-

[13]Foi considerado um tempo igual a 8.760 horas para o cálculo do atributo importância para confiabilidade. Por simplicidade, serão mostrados apenas os componentes com os maiores valores deste atributo. Nesta tabela, *Nó-Folha* representa os componentes *Site1*, *Site2*, *Site3*, *Site4*, *Site5*, *Site6*, *Site7* e *Site8*.

tura A32) foi proposta considerando redundância nos componentes *L1*, *Protheus1* e *Dominio1*. Nesta arquitetura foi utilizado um servidor de virtualização com duas máquinas virtuais (componentes *Protheus2* e *Dominio2*).

Tabela 8.18: Valores de importância para confiabilidade - Arquiteturas A12, A22 e A32

Arquitetura A12		Arquitetura A22		Arquitetura A32	
Imp._Conf.	Valor	Imp._Conf.	Valor	Imp._Conf.	Valor
Enlace_L1	1,00	Enlace_L1	1,00	Enlace_L1	1,00
Protheus1	8,14E-3	Enlace_L2	0,99	Enlace_L2	0,99
Domain1	8,14E-3	Protheus1	8,14E-3	Protheus1	4,10E-3
Firewall	2,90E-5	Domain1	8,14E-3	Domain1	4,10E-3
Nó_Folha	2,80E-5	Firewall	2,90E-5	Protheus2	4,10E-3
Core	1,80E-5	Nó_Folha	2,80E-5	Domain2	4,10E-3
–	–	Core	1,80E-5	Nó_Folha	2,90E-5
–	–	–	–	Core	1,80E-5

De forma semelhante à arquitetura A12, os modelos de dependabilidade das arquiteturas A22 e A32 adotam uma abordagem hierárquica na qual o modelo de nível superior é do tipo RBD e os modelos do nível mais baixo são do tipo SPN ou RBD. Os componentes redundantes *L1*/*L2* nas arquiteturas A22 e A32 consideram o mecanismo de redundância *warm standby* (ver Figura 4.5). Na arquitetuta A32, este também é o mecanismo adotado para a redundância entre um servidor convencional (*Protheus1* ou *Dominio1*) e uma máquina virtual (*Protheus2* ou *Dominio2*) no servidor de virtualização. Figura 8.9 mostra o modelo SPN que representa este mecanismo.

Este modelo inclui dez lugares relevantes, a saber, Serv_ON e Serv_OFF junto com os pares correspondentes que representam os estados atividade e inatividade dos demais componentes. Caso o servidor convencional (Serv) falhe, a transição VM_ActSt é habilitada. Seu atraso representa o tempo de detecção da falha e de ativação de uma máquina virtual (VM) que está inicialmente no estado não operacional (StVM_ON). Este período é denominado de MTTA. Lugares StVM_ON e OpVM_ON representam uma máquina virtual nos estados não-operacional e operacional respectivamente. Transição imediata X_DCTSp representa o retorno

ao estado normal de operação.

As transições VM_ActSt, Hw_MTTR, VMM_MTTR, StVM_MTTR, OpVM_MTTR e Serv_MTTR possuem a seguinte expressão para o atraso (*delay*): IF #RelFlag = 1 : 10^n ELSE Y; onde Y representa o valor de MTTA da transição VM_ActSp ou o parâmetro MTTR do correspondente componente. O parâmetro MTTF associado a cada componente é atribuído às transições Hw_MTTF, VMM_MTTF, Serv_MTTF e OpVM_MTTF. A transição StVM_MTTF possui um valor de *atraso* 50% maior em relação à transição OpVM_MTTF, por suposição. Com relação ao lugar RelFlag, caso a condição #RelFlag = 1 seja verdadeira, então a avaliação refere-se à confiabilidade; de modo contrário, a avaliação refere-se à disponibilidade. As transições temporizadas possuem uma distribuição exponencial (exp) e uma semântica de disparo do tipo *single server* (ss).

A disponibilidade do sistema pode ser calculada pela seguinte expressão: P{(#Serv_ON = 1) OR (#Hw_ON = 1 AND #VMM_ON = 1 AND #OpVM_ON = 1)}. Por sua vez, a confiabilidade do sistema pode ser calculada pela expressão: P{#Serv_ON = 1}.

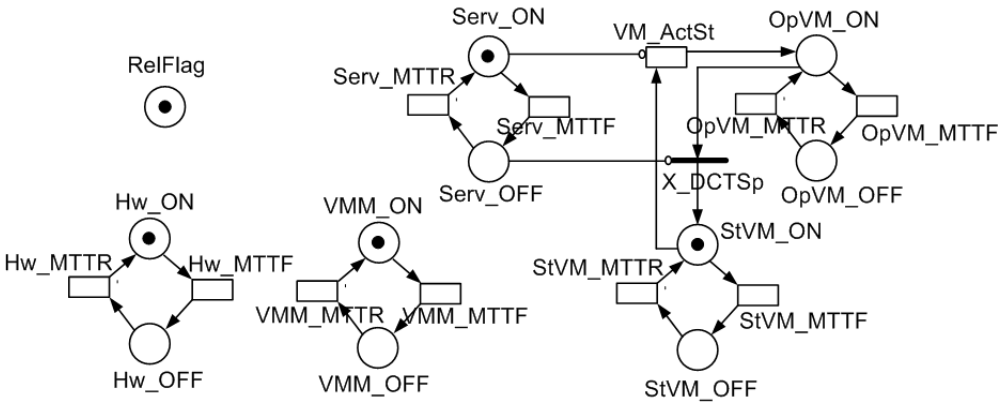


Figura 8.9: Modelo SPN - Redundância de servidores

Após a definição dos modelos de dependabilidade e a determinação do atributo de importância para confiabilidade dos componentes nas diferentes arquiteturas, foi então executada a atividade de Geração de Agrupamentos.

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	27.563	0,00	1,00	1,00
2	13.424	0,25	0,49	0,55
3	2.572	0,50	0,09	0,51
4	1.149	0,75	0,04	0,75
5	0	1,00	0,00	1,00

(a) Menor distância Euclidiana - Arquitetura 12

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	27.531	0,00	1,00	1,00
2	12.237	0,20	0,44	0,49
3	2.350	0,40	0,09	0,41
4	1.476	0,60	0,05	0,60
5	220	0,80	0,01	0,80
6	0	1,00	0,00	1,00

(b) Menor distância Euclidiana - Arquitetura 22

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	26.856	0,00	1,00	1,00
2	10.798	0,17	0,40	0,44
3	3.625	0,33	0,13	0,36
4	1.991	0,50	0,07	0,51
5	957	0,67	0,04	0,67
6	198	0,83	0,01	0,83
7	0	1,00	0,00	1,00

(c) Menor distância Euclidiana - Arquitetura 32

Tabela 8.19: Menor distância Euclidiana - Arquiteturas 12, 22 e 32

Nesta atividade foi aplicada a abordagem de agrupamento hierárquico aglomerativo. Os atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente* foram considerados a partir de uma configuração específica de componentes para cada arquitetura. São também calculados os valores do índice RMSSTD em função do número de agrupamentos em cada passo *i* desta abordagem para cada uma das arquiteturas (ver Tabela 8.19). Adicionalmente, esta tabela também mostra os valores normalizados do índice RMSSTD e do número de agrupamentos, utilizando a Equação 6.1. O número de agrupamentos associado à menor distância Euclidiana dos valores normalizados para a origem indica um número bom de agrupamentos. De acordo com as Tabelas 8.19a, 8.19b e 8.19c, o número bom de agrupamentos em cada arquitetura é três.

A atividade Seleção de Agrupamento irá escolher o agrupamento com mais alto valor do índice I_c, a partir do número bom de agrupamentos calculado na atividade anterior. Utilizamos

Tabela 8.20: Valores do índice I_c - Arquiteturas A12, A22 e A32

Arch./Numb._Clusters	Cluster1/I_c	Cluster2/I_c	Cluster3/I_c	Selected Cluster
A12/3	0,23	0,07	0,20	Cluster1
A22/3	0,28	0,07	0,20	Cluster1
A32/3	0,25	0,08	0,20	Cluster1

os valores de 0,90, 0,05 e 0,05 para os pesos w_1, w_2 e w_3, respectivamente. Estes valores favorecem a escolha do agrupamento cujos componentes possuem maior influência sobre o aspecto de confiabilidade do sistema. Após a normalização dos atributos de cada um dos componentes, o índice I_c é calculado (ver Tabela 8.20). Por simplicidade, esta tabela omite detalhes dos agrupamentos.

Tabela 8.21: Valores de projeto de experimento fatorial 2^k - Arquiteturas A12, A22 e A32

Arq.	EnlaceL1	EnlaceL2	Protheus1	Dominio1	Firewall	Protheus2	Dominio2	Componentes Selecionados
A12	99,00%	–	0,46%	0,46%	6,92E-3%	–	–	Enlace_L1
A22	0,50%	0,40%	49,20%	49,20%	0,70%	–	–	Protheus1, Dominio1
A32	30,40%	24,90%	0,03%	0,03%	44,50%	8,00E-4%	8,00E-4%	EnlaceL1, EnlaceL2, Firewall

Então, considerando os componentes representados no agrupamento selecionado, o projeto de experimento fatorial 2^K é analisado. A atividade Análise de Experimento escolhe apenas os componentes do agrupamento selecionado que possuem maior influência sobre a disponibilidade em cada arquitetura (ver Tabela 8.21). Para arquitetura A12, enlace L1 possui a maior influência sobre a disponibilidade deste sistema. Por sua vez, Protheus1 e Dominio1 para a arquitetura A22 junto com Firewall, enlace L1 e enlace L2 para a arquitetura A32 possuem a maior influência sobre a disponibilidade destes sistemas. Na atividade Definição e Avaliação de Cenários, os componentes escolhidos na atividade anterior proporcionam suporte à construção de diferentes cenários, que são descritos pela combinação de suas diferentes opções (ver Tabela 8.22) como parâmetros de entrada com relação aos modelos de infraestrutura e de negócios. Os cenários serão analisados baseados nos valores das métricas relacionadas. Estes valores são

normalizados em cada cenário *i*.

Tabela 8.22: Opções dos componentes selecionados

Opções	Enlaces (L1 e L2)			Servidores(Protheus1 e Dominio1)			Firewall		
	MTTF(h)	MTTR(h)	Custo(Us$)	MTTF(h)	MTTR(h)	Custo(Us$)	MTTF(h)	MTTR(h)	Custo(Us$)
1^a	396	4,00	12.000/year	1.399	0,99	5.000	65.891	4,00	40.000
2^a	796	4,00	18.000/year	2.778	0,98	15.000	15.465	0,98	15.500
3^a	3.996	4,00	24.000/year	2.815	0,99	25.000	8.717	0,60	5.500

A Figura 8.10 mostra as métricas normalizadas de indisponibilidade e custo de infraestrutura obtidas com base nos cenários construídos. O projeto associado ao cenário com a menor distância Euclidiana será escolhido. A Tabela 8.23 mostra as soluções selecionadas para as diferentes arquiteturas analisadas.

Tabela 8.23: Soluções de cada arquitetura

Arquitetura	Indisponibilidade	Custo de Infraestrutura
A12	0,7%	Us$ 246.600
A22	0,14%	Us$ 258.600
A32	0,064%	Us$ 280.100

A métrica VTp pode ser utilizada para comparar soluções selecionadas das diferentes arquiteturas. Por exemplo, se alguém escolhe a arquitetura A22 a partir da arquitetura A12, será pago Us$12.000 a mais ($\Delta ICust$) e obtido a variação do *tempo de parada*, correspondente à disponibilidade adicional, de 177.887 segundos (ΔD). Por outro lado, se alguém escolhe a arquitetura A32 a partir da arquitetura A12, será pago Us$ 33.500 a mais ($\Delta ICust$) e obtida a variação do *tempo de parada*, correspondente à disponibilidade adicional, de 201.060 segundos (ΔD). Considerando a métrica VTp, a arquitetura A22 tem um incremento de 14,82 segundos por unidade monetária gasta sobre a arquitetura A12. Por sua vez, a arquitetura A32 possui um incremento de 6,01 segundos por unidade monetária gasta sobre a arquitetura A12. É interessante notar que a mudança da A12 para a arquitetura A22 possui um maior valor de VTp em comparação com a mudança da arquitetura A12 para a arquitetura A32. Pode-se notar, considerando-se a métrica VTp, que a mudança a partir da arquitetura A12 para a arquitetura

A22 será mais atrativa em comparação com a mudança da arquitetura A12 para a arquitetura A32.

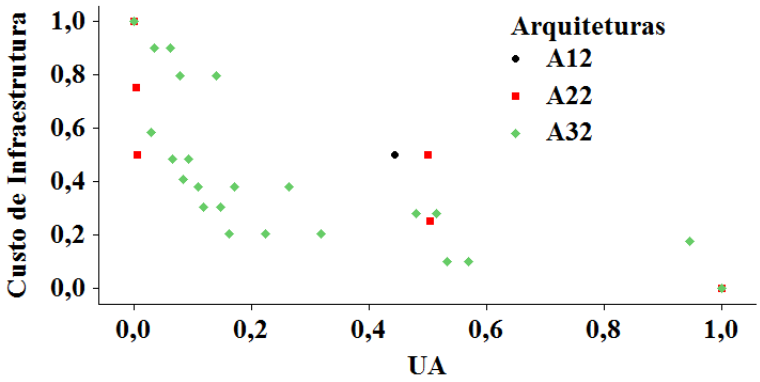


Figura 8.10: Custo de Infraestrutura x UA - Arquiteturas A12, A22 e A32

8.5 Considerações Finais

Neste capítulo, foram definidos quatro casos a fim de mostrar diferentes abordagens utilizando a estrutura mostrada nesta tese. Foram mostrados tanto casos planejados como casos definidos a partir da infraestruturas de redes convergentes reais. Para os casos planejados, foram utilizadas as quatro arquiteturas apresentadas no capítulo anterior, a fim de se fazer uma análise de aspectos da infraestrutura e da metodologia proposta. Por fim, a metodologia proposta foi aplicada para situações reais tanto em uma empresa pública quanto em uma uma empresa privada.

CAPÍTULO 9

Conclusão

9.1 Contribuições

Nos dias atuais, as redes convergentes são críticas para o sucesso dos negócios, tanto grande quanto pequeno. Um funcionamento correto por parte destas redes é importante por uma razão simples: as empresas perdem dinheiro devido a qualquer tipo de falha ou mau funcionamento, ocasionando prejuízos de diversas naturezas. Estas perdas variam em função do grau de dependência em relação a estas redes e do setor da economia ao qual a empresa pertence. Desta maneira, um dos objetivos a serem seguidos pelas empresas refere-se a um dimensionamento correto da infraestrutura de suas redes convergentes, de maneira a satisfazer tanto suas necessidades técnicas quanto suas necessidades baseadas em negócios.

Os trabalhos apresentados na área de projeto de infraestrutura de redes convergente geralmente não consideram de maneira formal as métricas de negócios. Ao invés disso, eles tratam estas métricas como restrições a projetos baseados em métricas técnicas. As técnicas utilizadas nestas abordagens podem ser classificadas como métodos heurísticos, métodos metaheurísticos, abordagens híbridas etc. Existem ainda trabalhos que relacionam formalmente aspectos de infraestrutura e de negócios com o objetivo de incrementar esta relação considerando o projeto de uma infraestrutura. Entretanto, estes últimos estão focados nas áreas de gestão de provedores de serviços online e não implementam mecanismos para a redução do número de projetos candidatos.

Neste contexto, esta tese apresenta uma estrutura composta por modelos, métricas e uma

metodologia para a seleção do melhor projeto a partir de uma infraestrutura em particular e para análise comparativa e objetiva entre as soluções de projetos de diferentes infraestruturas de redes convergentes. A estrutura proposta integra os aspectos técnicos e de negócios de maneira formal.

Esta estrutura pode ser ainda utilizada para vários tipos de análises sobre os aspectos financeiros como exemplo: pode-se avaliar o impacto sobre os negócios na troca ou falha de um dos componentes da infraestrutura, na adoção de diferentes mecanismos de redundância ou mesmo na priorização de determinado tipo de tráfego. Desta maneira, esta estrutura pode ser também utilizada em um processo de tomada de decisão. Finalmente, embora desenvolvida para infraestrutura de redes convergentes, nosso trabalho pode ser facilmente adaptado para a solução de projeto de diferentes tipos de sistemas computacionais.

Foram propostos modelos de negócios tais como: custo de infraestrutura, receita de infraestrutura, multa e lucro líquido. Estes modelos podem capturar diretamente o impacto de aspectos da infraestrutura sobre os aspectos de negócios. Com relação aos aspectos de infraestrutura, foram criados modelos heterogêneos. Foi adotada abordagens baseada em espaço de estados, tais como SPN e CTMC, não baseada em espaço de estados, a exemplo de RBD, e uma abordagem hierárquica. Os modelos de infraestrutura criados são genéricos bastante para representar uma grande variedade de mecanismos e cenários encontrados em projetos reais.

Foram construídos sistemas, *testbeds*, que são ambientes baseados em diversas ferramentas, construídos para uma finalidade específica. Estes sistemas foram construídos para proporcionar suporte à criação dos modelos de desempenho e de dependabilidade. Com relação aos modelos de dependabilidade, foram coletados dados de ativação relativos aos diferentes mecanismos de redundância mostrados, *warm standby* e *cold standby*. Para os modelos de desempenho, foram coletados dados relativos às diferentes políticas de enfileiramento (*Custom Queuing*, *Priority Queuing* e *FIFO*), características e tipos de tráfego, variável/constante e dados/voz/vídeo.

Métricas foram também definidas para dar suporte à escolha do melhor projeto de uma infraestrutura. Com relação às métricas de infraestrutura pode-se mencionar: *vazão de saída*, *número de pacotes descartados*, *tamanho médio da fila*, disponibilidade e confiabilidade. Com relação às métricas orientadas a negócios, pode-se destacar: *lucro líquido*, *custo de infraestrutura*, *receita de infraestrutura*, *multa*, *lucro líquido adicional por unidade monetária gasta (ALc)* e *variação de tempo de parada por unidade monetária gasta (VTp)*. Particularmente, as métricas ALc e VTp, além de integrar direta ou indiretamente aspectos de infrestrutura e de negócios, podem ser utilizadas para realizar uma comparação objetiva entre as soluções selecionadas em diversas arquiteturas.

A metodologia proposta é baseada na utilização de mecanismos tais como: agrupamento hierárquico aglomerativo (AHC), importância para confiabilidade e projeto de experimento fatorial, os quais visam reduzir o número de projetos candidatos e proporcionar suporte à escolha do melhor projeto da infraestrutura de redes convergentes. Ela considera tanto uma classe de redes convergentes que gera diretamente receitas a partir dos serviços oferecidos, quanto uma outra classe de redes que não gera diretamente, a partir de seus serviços, receitas às empresas que as detêm. Na fase de Seleção do Projeto desta metodologia, as atividades *Geração de Agrupamentos*, *Definição do Número Bom de Agrupamentos*, *Seleção de Agrupamento* e *Seleção de Projeto* foram adicionalmente especificadas através de algoritmos. O algoritmo 1 detalhou as atividades *Geração de Agrupamentos* e *Definição do Número Bom de Agrupamentos*. Este algoritmo é executado de maneira semiautomática. O algoritmo 2, detalhando a atividade *Seleção de Agrupamento*, é executado de maneira automatizada. O código correspondente à implementação deste algoritmo pode ser encontrado no Apêndice C. Por fim, o algoritmo 3, detalhando a atividade *Seleção de Projeto*, é também executado de maneira automatizada. Seu código pode ser encontrado no Apêndice C.

Com o intuito de avaliar e aplicar a estrutura proposta, nesta tese foram definidos diferentes casos. O primeiro caso realizou um estudo para o planejamento de capacidade em redes con-

vergentes com a utilização dos modelos de desempenho e de dependabilidade das arquiteturas tomadas por base. Foi feito um estudo para a alocação de recursos das diferentes classes de tráfego utilizando a política de enfileiramento *Custom Queuing*, assim como uma análise da disponibilidade do sistema em função de métricas como MTTF e MTTR.

O segundo caso avaliou a metodologia proposta, a partir das arquiteturas tomadas como base. Os resultados mostraram a eficácia desta metodologia para reduzir significativamente o número de projetos candidatos e selecionar o melhor projeto. A diferença entre os resultados obtidos considerando as situações de aplicação ou não desta metodologia foi muito pequena. Para a classe de redes convergentes que gera diretamente receitas a partir de seus serviços, o erro relativo máximo entre os melhores projetos foi de 0,04 %. Por outro lado, para a classe de redes convergentes que não gera diretamente receitas a partir de seus serviços, o erro relativo máximo foi de 6,9 % entre os projetos.

O terceiro e o quarto casos fizeram a aplicação desta metodologia, juntamente com os respectivos modelos e métricas, em redes convergentes de uma empresa pública e de uma privada. Foram encontrados os melhores projetos em diferentes arquiteturas e efetuada uma comparação objetiva entre estes projetos, utilizando a métrica VTp.

9.2 Trabalhos Futuros

Embora esta tese aborde diretamente questões sobre o processo de otimização de infraestruturas de redes convergentes, existem muitas possibilidades para melhorar algumas limitações e estender o trabalho atual. Os itens a seguir resumem algumas possibilidades:

- Realizar estudos que permitam uma análise detalhada do impacto da infraestrutura de potência sobre a disponibilidade destas redes;

- Realizar estudos para incluir a infraestrutura de potência na escolha do melhor projeto de redes convergentes;
- Realizar estudos que permitam uma análise do impacto de diferentes políticas de manutenção, corretiva e preventiva, sobre a disponibilidade destas redes. Em cada uma destas políticas, podem também ser considerados diferentes níveis de manutenção;
- Realizar estudos que permitam uma análise do impacto do estado atual do sistema operacional de seus componentes sobre a disponibilidade destas redes. Para efeito comparativo, será utilizada a técnica de *rejuvenescimento de software*;
- Incorporar custos operacionais, tais como: consumo de energia, manutenção, pessoal etc. Além disso, também tratar os diferentes tipos de custos relacionados às falhas de redes convergentes, tais como: produtividade reduzida, perda de dados, imagem pública comprometida etc. Estes custos podem variar em função da área de atuação da empresa;
- Considerar políticas de roteamento e técnicas de engenharia de tráfego no estudo da disponibilidade de redes convergentes;
- Considerar o projeto de infraestrutura de um *data center* em função de diferentes aspectos de negócios;
- Adicionar novas métricas relacionando questões de infraestrutura e de negócios. Por exemplo, as métricas propostas (ALc e VTp) não consideram diretamente custos relacionados à manutenção ou de violação do nível de serviço contratado, em função de aspectos técnicos;
- Incluir o índice importância estrutural [56] na metodologia proposta, correspondente à atividade *geração de agrupamentos*, e efetuar uma análise de seus resultados em comparação com os atuais.

Referências Bibliográficas

[1] AFUAH, A.; TUCCI, C.L. Internet Business Models and Strategies. 2^a Edição. McGraw-Hill/Irwin. 2003.

[2] AGUSLEO, H.; UY, J.; IZDEBSKI, L.; PUOPOLO, S.; SHEN, D. Exploring Two-Sided Business Models for Service Providers. Cisco Internet Business Solutions Group (IBSG). 2010.

[3] ANTTI, V. The State Explosion Problem. In: REISIG, W.; ROZENBERG, G. Lectures on Petri Nets I: Basic Models, Advances in Petri Nets. Springer Berlin Heidelberg, 1998. p. 429-528.

[4] ARREGOCES, M.; PORTOLANI, M. Data Center Fundamentals. USA: Cisco Press. 2003.

[5] AVIZIENIS, A.; LAPRIE, J.C.; RANDELL, B.; LANDWEHR, C. Basic concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, p. 11-33, 2004.

[6] BALBO, G. Introduction to Stochastic Petri Nets. European Educational Forum: School on Formal Methods and Performance Analysis, 2001. p. 84-155.

[7] BOLCH, G.; GREINER, S.; DE MEER, H.; TRIVEDI, K.S. Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. 2^a Edição. John Wiley & Sons. 2006.

[8] BOZDOGAN, C.; ZINCIR-HEYWOOD, A.N.; GOKCEN, Y. Automatic Optimization for a Clustering Based Approach to Support IT Management. International Workshop on Business-driven IT Management (BDIM), 2013. p. 1233-1236.

[9] BUCO, M.J.; CHANG, R.N.; LUAN, L.Z.; WARD, C.; WOLF, J.L.; YU, P.S. Utility Computing SLA Management Based Upon Business Objectives. IBM Systems Journal, Vol. 43, p. 159-178, 2004.

[10] CAO, D.; MURAT, A.; CHINNAM, R.B. Efficient Exact Optimization of Multi-Objective Redundancy Allocation Problems in Series-Parallel Systems. Reliability Engineering and System Safety, Vol. 111, p. 154-163, 2013.

[11] CASE, J.; FEDOR, M.; SCHOFFSTALL, M.; DAVIN, J. A Simple Network Management Protocol (SNMP). IETF - RFC 1157. 1990.

[12] CASSANDRAS, C.; LAFORTUNE S. Introduction to Discrete Event Systems. Massachusetts: Kluwern Academic Publishers. 1999.

[13] CHAMBARI, A.; RAHMATI, S.H.A.; NAJAFI, A.A., KARIMI, A. A Bi-Objective Model to Optimize Reliability and Cost of System with a Choice of Redundancy Strategies. Journal Computers & Industrial Engineering, Vol. 63, p. 109-119, 2012.

[14] CHU, C.H.K.; PANT H.; RICHMAN, S.H.; WU, P. Enterprise VoIP Reliability. International Telecommunications Network Strategy and Planning Symposium, 2007. p. 1-6.

[15] CIARDO, G.; BLAKEMORE, A.; CHIMENTO, P.F.; MUPPALA, J.; TRIVEDI, K.S. Automated Generation and Analysis of Markov Reward Models Using Stochastic Reward Nets. In: MEYER, C.; PLEMMONS, R.J. Linear Algebra, Markov Chains, and Queuing Models. IMA Volumes in Mathematics and its Applications. Alemanha: Springer-Verlag, 1993. Vol. 48. p. 145-191.

[16] CISCO SYSTEMS. Quality of Service Solutions Configuration Guide. Disponível em: http://www.cisco.com/c/en/us/td/ios/12_2/qos/configuration/guide/fqos_c/qcfconmg.html. Acesso em: Setembro/2013.

[17] CHEN, X.; WANG, C.; XUAN, D.; LI, Z.; MIN, Y.; ZHAO, W. Survey on QoS Management of VoIP. International Conference on Computer Networks and Mobile Computing (ICCNMC), 2003. p. 69-77.

[18] COURVOISIER, M.; VALETTE R. Systèmes de Commande en Temps Réel. SCM Editions, 1980.

[19] CRUZ, H.T.; ROMAN, D.T. Traffic Analysis for IP Telephony. International Conference on Electrical and Electronics Engineering (ICEEE), 2005. p. 136-139.

[20] DESROCHERS, A.A.; AL-JAAR, R.Y. Applications of Petri Nets in Manufacturing Systems: Modeling, Control and Performance Analysis. New York: IEEE Control Systems Society Press, 1995.

[21] DOVERSPIKE, R.; RAMAKRISHNAN, K.K.; CHASE, C. Structural Overview of ISP Networks. In: KALMANEK, C.; MISRA, S.; YANG, R. (eds). Guide to Reliable Internet Services and Applications. Springer, 2010.

[22] EBELING, C. An Introduction to Reliability and Maintainability Engineering. Waveland Press. 1997.

[23] EPIPHANIOU, G.; MAPLE, C.; SANT, P.; REEVE, M. Affects of Queuing Mechanisms on RTP Traffic: Comparative Analysis of Jitter, End-to-End Delay and Packet Loss. Annual conference on Access, Reliability and Security (ARES), 2010. p. 33-40.

[24] EVERITT, B.S.; LANDAU, S.; LEESE, M.; STAHL, D. Cluster Analysis. 5^a Edição. Willey Series in Probability and statistics, John Wiley & Sons Ltd, 2011.

[25] FALCH, M. Cost and Demand Characteristics of Telecom Networks. In: MELODY, W. Telecom Reform: Principles, Policies and Regulatory Practices. Schultz DocuCenter, 2001.

[26] FITÓ, J.O.; MACÍAS, M.; JULIA, F.; GUITART, J. Business-Driven IT Management for Cloud Computing Providers. IEEE International Conference on Cloud Computing Technology and Science, 2012. p. 193-200.

[27] FLORIN, G.; NATKIN, S. Evaluation Based Upon Stochastic Petri Nets of the Maximum Throughput of a Full Duplex Protocol. Workshop on Application and Theory of Petri Nets, 1982. p. 280-288.

[28] FLORIN, G.; NATKIN, S. Matrix Product Form Solution for Closed Synchronized Queueing Networks. International Workshop on Petri Nets and Performance Models, 1989. p. 29-39.

[29] FRICKS, M.F.; TRIVEDI, K.S. Importance Analysis with Markov Chains. Proceedings of the Annual Reliability and Maintainability Symposium, 2003. p. 89-95.

[30] GARG, H.; SHARMA, S.P. Multi-Objective Reliability-Redundancy Allocation Problem Using Particle Swarm Optimization. Journal of Computers & Industrial Engineering, Vol. 64, No. 1, p. 247-255, 2013.

[31] GEFFROY, J.C.; MOTET, G. Design of Dependable Computing Systems. Kluwer Academic Publishers. 2002.

[32] GERMAN, R. Performance Analysis of Communicating Systems - Modeling with Non-Markovian Stochastic Petri Nets. John Wiley & Sons. 2000.

[33] GERMAN, R.; KELLING, Ch.; ZIMMERMANN, A.; HOMMEL, G. TimeNET - A Toolkit for Evaluating Stochastic Petri Nets with Non-Exponential Firing Times. Performance Evaluation, Vol. 24, No 1, p. 69-87, 1995.

[34] GOETTGE, R.T.; PALCZAK, C.; BREHM, E.W. QUEST - A Tool for Automated Performance, Dependability and Cost Tradeoff Support. International Symposium and Workshop on Systems Engineering of Computer Based Systems, 1995. p. 351-357.

[35] GOSPODINOV, M. The Effects of Different Queuing Disciplines Over FTP, Video and VoIP Performance. International Conference on Computer Systems and Technologies, 2004. p. 1-5.

[36] GRAY, J.; SIEWIOREK, D.P. High-Availability Computer Systems. IEEE Computer, Vol. 24, p. 39-48, 1991.

[37] GUIMARÃES, A.P.; MACIEL, P.R.M.; MATIAS JÚNIOR, R. Quantitative Analysis of Performability in Voice and Data Networks. IEEE International Conference on Systems, Man and Cybernetics (SMC), 2010. p. 761-768.

[38] GUIMARÂES, A.P; MACIEL, P.R.M.; MATIAS JÚNIOR, R. Dependability and Performability Modeling of Voice and Data Networks. IEEE Latin-American Conference on Communications (LATINCOM), 2010. p. 1-6.

[39] GUIMARÃES, A.P.; MACIEL P.R.M.; MATIAS JÚNIOR, R. Quantitative Analysis of Dependability and Performability in Voice and Data Networks. International Conference on Computer Science and Information Technology (CCSIT), 2011. p. 302-312.

[40] GUIMArÃES, A.P.; MACIEL P.R.M.; MATOS JÚNIOR, R.S.; CAMBOIM, K. Impact Analysis of Availability on Computer Networks Infrastructure. International Conference on Information and Computer Networks (ICICN), 2011. p. 171-175.

[41] GUIMARÃES, A.P.; MACIEL, P.R.M.; MATOS JÚNIOR, R.S.; CAMBOIM, K. Dependability Analysis in Redundant Communication Networks Using Reliability Importance. International Conference on Information and Network Technology, 2011. p. 12-17.

[42] GUIMARÃES, A.P.; MACIEL, P.R.M.; OLIVEIRA, H.M.N.; BARROS, R. Availability Analysis of Redundant Computer Networks: a Strategy Based on Reliability Importance. IEEE International Conference on Communication Software and Networks (ICCSN), 2011. p. 328-332.

[43] GUIMARÃES, A.P.; MACIEL, P.R.M. Supporting Infrastructure Optimization of Converged Networks. IEEE Latin-American Conference on Communications (LATINCOM), 2011. p. 1-6.

[44] GUIMARÃES, A.P.; MACIEL, P.R.M. Infrastructure Modeling of Converged Networks for Business-Oriented Metrics Evaluation. IEEE International Conference on Systems, Man and Cybernetics (SMC), 2012. p. 1274-1279.

[45] GUIMARÃES, A.P.; MACIEL, P.R.M.; MATIAS JÚNIOR, R. An Analytical Modeling Framework to Evaluate Converged Networks Through Business-Oriented Metrics. Reliability Engineering and System Safety, Vol. 118, p. 81-92, 2013.

[46] HAGHANI, E.; DE, S.; ANSARI, N. On Modeling VoIP Traffic in Broadband Networks. IEEE Global Telecommunications Conference (GLOBECOM), 2007. p. 1922-1926.

[47] HAMILTON, H. (n.d.). Knowledge Discovery in Databases. Disponível em: http://www2.cs.uregina.ca/∼dbd/cs831/index.html. Acesso em: Outubro/2012.

[48] HAVERKORT, B.R. Markovian Models for Performance and Dependability Evaluation. EEF/Euro Summer School on Trends in Computer Science Bergen Dal, 2001. p. 38-83.

[49] INFONETICS RESEARCH REPORT. The Costs of Enterprise Downtime: North American Vertical Markets. INFONETICS RESEARCH. Disponível em: http://www.optrics.com/emprisa_networks/2005_UPNA05_DWN_ToC_Excerpts.pdf. Acesso em: Julho/2011.

[50] JONES, C.; RANDELL, B. Dependable Pervasive Systems. Technical report, University Newcastle upon Tyne, Cyber Trust & Crime Prevention Project, 2004.

[51] JAIN, R. The Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurement Simulation and Modeling. John Wiley & Sons. 1991.

[52] KIM, D.S.; MACHITA, F.; TRIVEDI, K.S. Availability Modeling and Analysis of a Virtualized System. IEEE Pacific Rim International Symposium on Dependable Computing, 2009. p. 365-371.

[53] KUMAR, R.; IZUI, K.; MASATAKA, Y.; NISHIWAKI, S. Multilevel Redundancy Allocation Optimization Using Hierarchical Genetic Algorithm. IEEE Transactions on Reliability, Vol. 57, No. 4, p. 650-661, 2008.

[54] KUO, W.; PRASAD, V.R. An Annotated Overview of System-Reliability Optimization. IEEE Transactions on Reliability, Vol. 49, No 2, p. 176-187, 2000.

[55] KUO, W.; WAN, R. Recent Advances in Optimal Reliability Allocation. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, Vol. 37, No 2, p. 143-156, 2007.

[56] KUO, W.; ZUO, M.J. Optimal Reliability Modeling: Principles and Applications. John Wiley & Sons, Inc. 2003.

[57] LANUS, M.; YIN, L.; TRIVEDI, K.S. Hierarchical Composition and Aggregation of State-Based Availability and Performability Models. IEEE Transactions on Reliability, Vol. 52, No. 1, p. 44-52, 2003.

[58] LIMA, A.; SAUVÉ, J.; SOUZA, N. Capturing the Quality and Business Value of IT Services Using a Business-Driven Model. IEEE Transactions on Network and Service Management, Vol. 9, No. 4, p. 421-432, 2012.

[59] LINDEMANN, C. Performance Modelling with Deterministic and Stochastic Petri Nets. John Wiley and Sons. 1998.

[60] LIU, X.; ZHAN, Z. The Optimal Design of Service Level Agreement in IAAS Based on BDIM. International Conference on Graphic and Image Processing, 2013. Vol. 8768.

[61] MACHIRAJU, V.; BARTOLINI, C.; CASATI, F. Technologies for Business-Driven IT Management. In: CAVEDON, L.; MAAMAR, Z.; MARTIN, D.; BENATALLAH, B. Extending Web Services Technologies: the Use of Multi-Agent Approaches. Kluwer Academic, 2005. p. 1-28.

[62] MACIEL, P.R.M.; LINS, R.; CUNHA, P. Introdução às Redes de Petri e Aplicações. Sociedade Brasileira de Computação, 1996.

[63] MACIEL, P.R.M.; TRIVEDI, K.S.; MATIAS JÚNIOR, R.; KIM, D.S. Dependability Modeling. In: CARDELLINI, V.; CASALICCHIO, E.; CASTELO BRANCO, K.R.L.J.; ESTRELLA, J.C.; MONACO, F.J. (eds). Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions. IGI Global, 2011. p. 53-97.

[64] MAGRO, J.C. Estudo da Qualidade de Voz em Redes IP. Campinas, 2005. Dissertação de Mestrado - Universidade Estadual de Campinas.

[65] MALHOTRA, M.; TRIVEDI, K.S. Power-Hierarchy of Dependability-Model Types. IEEE Transactions on Reliability, Vol. 43, No. 3, p. 493-502, 1994.

[66] MALHOTRA, M.; TRIVEDI, K.S. Dependability Modeling Using Petri-Net Based Models. IEEE Transactions on Reliability, Vol. 44, No. 3, p. 428-440, 1995.

[67] MARINHO, M.; MACIEL, P.R.M.; SOUSA, E.; MACIEL, T.; GUIMARÃES, A.P. Stochastic Model for Performace Evaluation of Test Planning. IEEE International Conference on Systems, Man and Cybernetics (SMC), 2010. p. 3690-3697.

[68] MARSAN, M.A.; BALBO, G.; CONTE, G.; DONATELLI, S.; FRANCESCHINIS, G. Modelling with Generalized Stochastic Petri Nets. Series in Parallel Computing. John Wiley and Sons. 1995.

[69] MARSAN, M.A.; BOBBIO, A.; DONATELLI, S. Petri Nets in Performance Analysis: An Introduction. In: REISIG, W.; ROZENBERG, G. Lectures on Petri Nets I: Basic Models, Advanced in Petri Nets. Springer Berlin Heidelberg, 1996. p.211-256.

[70] MARSAN, M.A.; CHIOLA, G. On Petri Nets with Deterministic and Exponentially Distributed Firing Times. In: ROZENBERG, G. Advances in Petri Nets 1987. Springer Berlin Heidelberg, 1987. p. 132-145.

[71] MARSAN, M.A.; CONTE, G.; BALBO, G. A Class of Generalized Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems. ACM Transactions on Computer Systems, Vol. 2, No. 2, p. 93-122, 1984.

[72] MATOS JÚNIOR, R.S.; GUIMARÃES, A.P.; CAMBOIM, K.; MACIEL, P.R.M. Sensitivity Analysis of Availability of Redundancy in Computer Networks. International Conference on Communication Theory, Reliability and Quality of Service, 2011, p. 115-121.

[73] MENASCÉ, D.A.; ALMEIDA, V.A.; DOWDY, L.W. Performance by Design: Computer Capacity Planning by Example. USA: Prentice Hall Inc. 2004.

[74] MERLIN, P.M.; FARBER, D.J. Recoverability of Communication Protocols: Implications of a Theoretical Study. IEEE Transactions on Communications, Vol. 24, No. 9, p. 1036-1043, 1976.

[75] MOLLOY, M.K. On the Integration of Delay and Throughput Measures in Distributed Processing Models. Los Angeles, 1981. PhD thesis - University of California.

[76] MUPPALA, J.; FRICKS, R.; TRIVEDI, K.S. Techniques for System Dependability Evaluation. In: GRASSMAN, W.K. (editor). Computational Probability. USA: Springer, 2000. p. 445-480.

[77] MURATA, T. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, No. 4, p. 541-580, 1989.

[78] OGGERINO, C. High Availability Network Fundamentals. Cisco Press. 2001.

[79] QUEIROZ, M.; MACIEL, P.R.M.; SILVA, B.; GUIMARÃES, A. Performability Evaluation of Emergency System. IEEE International Conference on Systems, Man and Cybernetics (SMC), 2013. p. 4794-4799.

[80] RAMCHANDANI, C. Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. Cambridge, MA, 1974. PhD Thesis - Massachusetts Institute of Technology.

[81] RAMIREZ-MARQUEZ, J.E.; BOCCO, C.M. All-Terminal Network Reliability Optimization via Probabilistic Solution Discovery. Reliability Engineering and System Safety, Vol. 93, No. 11, p. 1689-1697, 2008.

[82] RANA, O.; WARNIER, M.; QUILLINAN, T.B.; BRAZIER, F.; COJOCARASU, D. Managing Violations in Service Level Agreements. In: TALIA, D.; YAHYAPOUR, R.; ZIEGLER, W. (eds). Grid Middleware and Services: Challenges and Solutions. Springer, 2008. p. 349-358.

[83] RUJASIRI, P.; CHOMTEE, B. Comparison of Clustering Techniques for Cluster Analysis. Kasetsart Journal (Natural Science), Vol. 43, No. 2, p. 378-388, 2009.

[84] SAHNER, R.; TRIVEDI, K. S.; PULIAFITO, A. Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Package. Kluwer Academic Publishers. 1995.

[85] SAUVÉ, J.; MOURA, A.; MARQUES, F. Business-Driven Design of Infrastructures for IT Services. Journal of Network and Systems Management, Vol. 17, No. 4, p. 422-456, 2009.

[86] SAUVÉ, J.; MOURA, A.; SAMPAIO, M.; JORNADA, J.; RADZIUK, E. An Introductory Overview and Survey of Business-Driven IT Management. IEEE/IFIP International Workshop on Business-Driven IT Management, 2006. p. 1-10.

[87] SEMAAN, G. Designing Networks with the Optimal Availability. Conference on Optical Fiber Communications/National Fiber Optic Engineers, 2008. p. 1-6.

[88] SHEIKHALISHAHI, M.; EBRAHIMIPOUR, V.; SHIRI, H.; ZAMAN, H.; JEIHOONIAN, M. A Hybrid GA-PSO Approach for Reliability Optimization in Redundancy Allocation Problem. The International Journal of Advanced Manufacturing Technology, Vol. 68, p. 317-338, 2013.

[89] SIFAKIS, J. Use of Petri Nets for Performance Evaluation. International Symposium on Measuring, Modelling and Evaluating Computer Systems, 1977. p. 75-93.

[90] SILVA, B.; MACIEL, P.R.M. Mercury Tool. Disponível em: http://www.cin.ufpe.br/∼bs/MercuryTool/mercury.html. Acesso em: Setembro/2013.

[91] SILVA, B.; MACIEL, P.R.M.; TAVARES, E.; ARAÚJO, C.; CALLOU, G.; SOUSA, E.; ROSA, N.; MARWAH, M.; SHARMA, R.; SHAH A.; CHRISTIAN, T.; PIRES, J.P. ASTRO: A tool for Dependability Evaluation of Data Center Infrastructures. IEEE International Conference on Systems Man and Cybernetics (SMC), 2010. p. 783-790.

[92] SIMPSON, W. The Point-to-Point Protocol (PPP). IETF - RFC 1661. 1994.

[93] SHI, H.; ZHAN, Z. An Optimal Infrastructure Design Method of Cloud Computing Services from the BDIM Perspective. Conference on Computational Intelligence and Industrial Applications, 2009. p. 393-396.

[94] SHI, H.; ZHAN, Z.; WANG, D. BDIM-Based Optimal Design of Videoconferencing Service Infrastructure in Multi-SLA Environments. Asia-Pacific Network Operations and Management Symposium (APNOMS), 2009. p. 251-260.

[95] SHOOMAN, M. The equivalence of Reliability Diagram and Fault-Tree Analysis. IEEE Transactions on Reliability, Vol. 19, No. 2, p. 74-75, 1970.

[96] SMITH, W.E.; TRIVEDI, K.S.; TOMEK, L.A.; ACKARET, J. Availability Analysis of Blade Server Systems. IBM Systems Journal, Vol 47, p. 621-640, 2010.

[97] STURM, R.; MORRIS, W.; JANDER, M. Foundations of Service Level Management. Sams Publishing, 2000.

[98] TANENBAUM, A.S. Modern Operating Systems. 3^a Edição. Pearson Prentice Hall. 2008.

[99] ITU-T. Pulse Code Modulation (PCM) of Voice Frequencies. Recommendation G.711. 1988.

[100] THOMAS II, T.M.; QUIGGLE, A. BCRAN - Building Cisco Remote Access Networks. USA: McGraw-Hill. 2000.

[101] TRIOLA, M.F. Statdisk Student Laboratory Manual and Workbook to Accompany the Triola Statistics Series. Addison-Wesley Longman. 2004.

[102] TRIVEDI, K.S. Probability and Statistics with Reliability, Queuing and Computer Science Applications. John Wiley and Sons. 2001.

[103] TRIVEDI, K. S.; HUNTER, S.; GARG, S.; FRICKS, R. Reliability Analysis Techniques Explored Through a Communication Network Example. International Workshop on Computer-Aided Design, Test, and Evaluation for Dependability, 1996. p. 110-120.

[104] TRIVEDI, K.S.; KIM, D.S.; ROY, A.; MEDHI, D. Dependability and Security Models. Design of Reliable Communication Networks (DRCN), 2009. p. 11-20.

[105] TUKEY, J.W. Exploratory Data Analysis. Addison-Wesley Publishing Company. 1977.

[106] WANG, Z.; TANG, K.; YAO, X. A Memetic Algorithm for Multi-Level Redundancy Allocation. IEEE Transactions on Reliability, Vol. 59. No. 4. p. 754-764, 2010.

[107] WARD, J.H. Hierarquical Grouping to Optimize an Objective Function. Journal of the American Statistical Association, Vol. 58, No. 301, p. 236-244, 1963.

[108] ZHANG, Z.; ZHANG, M.; GREENBERG, A., HU, Y.C., MAHAJAN, R.; CHRISTIAN, B. Optimizing Cost and Performance in Online Service Provider Networks. Symposium on Networked Systems Design and Implementation (NSDI), 2010. p. 33-48.

[109] ZIA, L.; COIT, D.W. Redundancy Allocation for Series-Parallel Systems Using a Column Generation Approach. IEEE Transactions on Reliability, Vol. 59, No. 4, p. 706-717, 2010.

[110] ZIMMERMANN, A.; GERMAN, R.; FREIHEIT, J.; HOMMEL, G. TimeNET 3.0 Tool Description. International Conference on Petri Nets and Performance Models (PNPM). 1999.

[111] ZIMMERMANN, A. TimeNET: A Software Tool for the Performability Evaluation with Stochastic Petri Nets. TU Berlin. 2001.

[112] ZOU, W.; JANIC, M.; KOOIJ, R.; KUIPERS, F. On the availability of networks. Broad Band Europe. 2007.

APÊNDICE A

Ferramental Utilizado

A.1 Visão Geral

Diversas ferramentas são utilizadas neste trabalho. Iremos dividir suas descrições considerando as atividades de medição, definição do modelo, validação do modelo, seleção de agrupamento e seleção de projeto correspondentes às fases de *modelagem do sistema* e *seleção do projeto* da metodologia proposta (ver Capítulo 6). Inicialmente, iremos mostrar as ferramentas correspondente à atividade de *Medição*.

A.1.1 Ferramentas para Medição

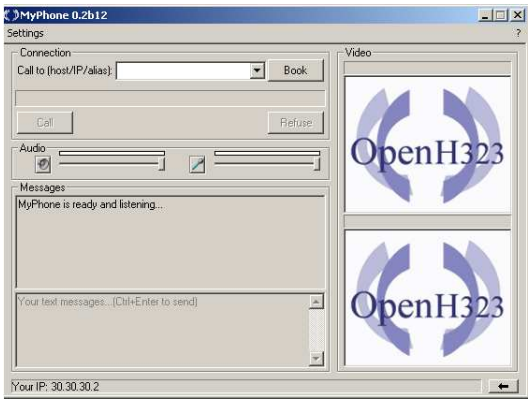


Figura A.1: MyPhone

Nesta etapa, foram adotados o pacote de software MyPhone[1] (ver Figura A.1) e o TFGEN[2]

[1] myphone.sourceforge.net
[2] www.st.rim.or.jp/∼yumo/pub/tfgen.html

(ver Figura A.2) para geração de tráfego de voz, vídeo e de dados. O software *MyPhone* gera tráfego de voz e de vídeo. Este software atua numa arquitetura *ponto-a-ponto* entre duas máquinas e pode selecionar diferentes *CODECs* para as conexões de voz e de vídeo.

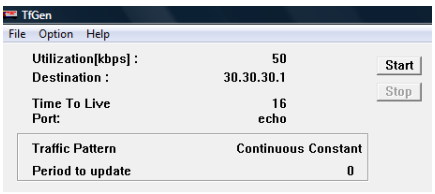


Figura A.2: TFGEN

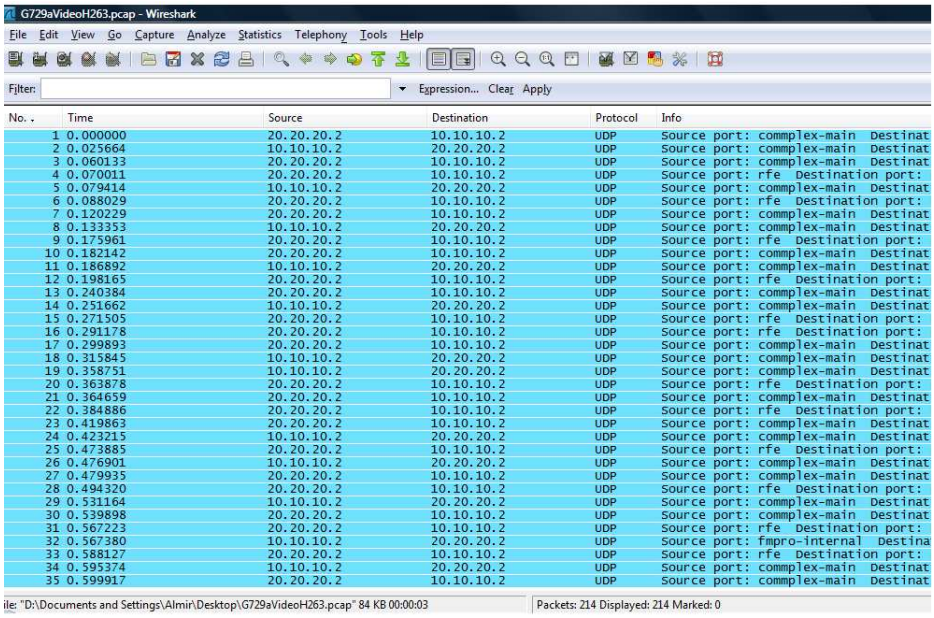
Por outro lado, o software *TFGEN* gera tráfego de dados. Neste software podem ser selecionadas as taxas de transmissão (em *bits por segundo - bps*), a interface de destino e o padrão de tráfego adotado. Os tráfegos variam de acordo com os cenários criados.

Como ferramentas de monitoração, foram utilizados o *WireShark*[3] (ver Figura A.3) e o *PRTG*[4], (ver Figura A.4). O *Wireshark* coleta todos os pacotes de dados, voz e vídeo que atravessam a interface selecionada (ver Figura A.3(a)). Podem ser ainda ativados filtros para seleção de pacotes de acordo com o protocolo, endereço de origem, endereço de destino etc. Estas informações servem para analisar o tamanho dos pacotes e dos cabeçalhos de cada tipo de tráfego.

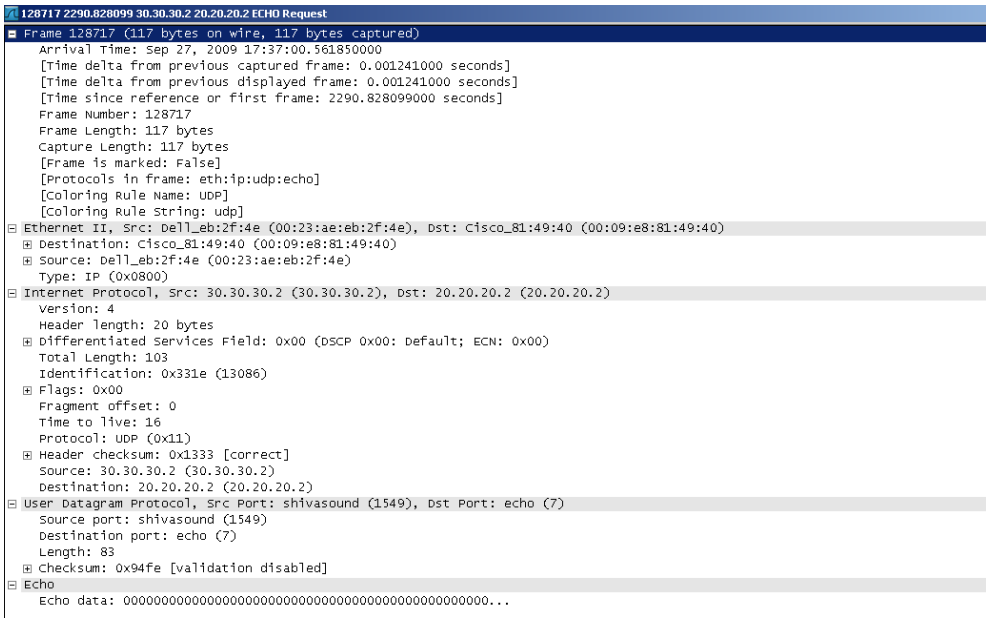
O *PRTG* coleta dados através do protocolo *SNMP* [11]. São disponibilizados gráficos (ver Figura A.4(a)) e tabelas (ver Figura A.4(b)) em tempo real relativos à evolução dos tráfego de dados, voz e vídeo. Estes gráficos e tabelas fornecem a taxa, em *bps*, dos tráfegos sobre a interface analisada.

Informações podem ser também coletadas diretamente a partir dos componentes (ver Figura A.5), baseada em uma interface de linha de comando (*CLI*). Nesta figura, são coletadas informação tais como: a vazão dos tráfegos, em *pps* e em *bps*, número de pacotes descartados e

[3]www.wireshark.org
[4]http://www.paessler.com/prtg

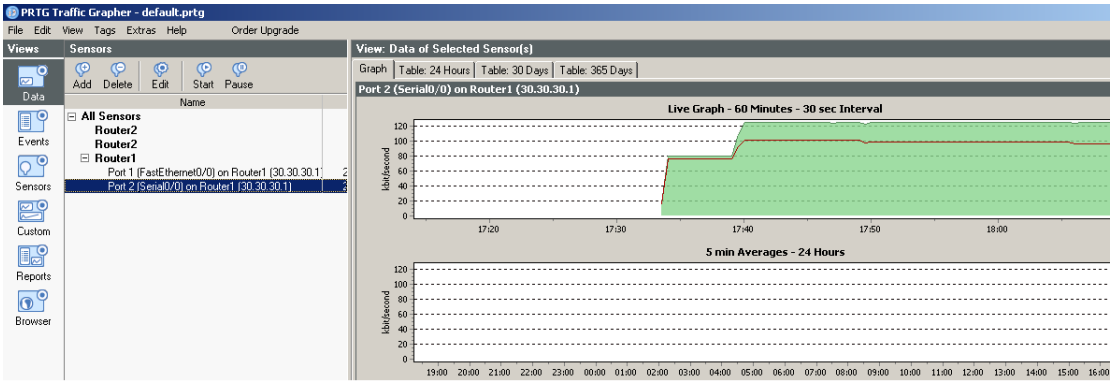


a) Wireshark



b) Wireshark - Detalhamento dos pacotes

Figura A.3: Wireshark



a) Gráficos

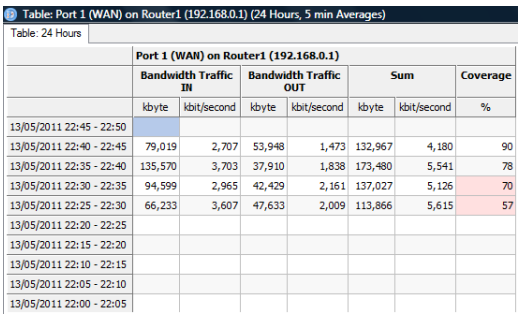


Table: Port 1 (WAN) on Router1 (192.168.0.1) (24 Hours, 5 min Averages)

Table: 24 Hours

	Port 1 (WAN) on Router1 (192.168.0.1)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
13/05/2011 22:45 - 22:50							
13/05/2011 22:40 - 22:45	79,019	2,707	53,948	1,473	132,967	4,180	90
13/05/2011 22:35 - 22:40	135,570	3,703	37,910	1,838	173,480	5,541	78
13/05/2011 22:30 - 22:35	94,599	2,965	42,429	2,161	137,027	5,126	70
13/05/2011 22:25 - 22:30	66,233	3,607	47,633	2,009	113,866	5,615	57
13/05/2011 22:20 - 22:25							
13/05/2011 22:15 - 22:20							
13/05/2011 22:10 - 22:15							
13/05/2011 22:05 - 22:10							
13/05/2011 22:00 - 22:05							

b) Tabelas

Figura A.4: PRTG

```
R0#show interface serial 0
Serial0 is up, line protocol is up, Internet address is 10.10.10.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,  reliability 255/255,
txload 12/255, rxload 12/255,  Encapsulation HDLC, loopback not set
Keepalive set (10 sec)  Last clearing of "show interface" counters 0:11:04
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Output queue (queue priority: size/max/drops):
Queueing strategy: priority-list 1
high: 0/20/0, medium: 0/40/0, normal: 0/60/0, low: 0/80/0
5 minute input rate 76000 bits/sec, 33 packets/sec
5 minute output rate 75000 bits/sec, 33 packets/sec
```

Figura A.5: CLI do roteador

número médio de pacotes em uma fila nas interfaces de entrada e de saída dos componentes correspondendo aos tráfegos de dados, voz e de vídeo. Os valores das taxas de transmissão nas interfaces, em *bps*, obtidos através do *PRTG*, são também comparados aos mesmos valores obtidos através da CLI.

```
1   hostname R0
2   interface Loopback0
3   ip address 20.20.20.1 255.255.255.0
4   interface FastEthernet0
5   ip address 2.0.0.1 255.0.0.0
6   interface Serial0
7   ip address 10.10.10.1 255.255.255.0
8   priority-group 1
9   router rip
10  network 2.0.0.0
11  network 10.0.0.0
12  network 20.0.0.0
13  access-list 102 permit udp any any range 5000 6000
14  access-list 103 permit tcp any any eq 1720
15  queue-list 1 protocol ip 1 list 102
16  queue-list 1 protocol ip 1 list 103
17  queue-list 1 interface FastEthernet0 2
18  queue-list 1 queue 1 byte-count 192
19  queue-list 1 queue 2 byte-count 82
20  priority-list 1 protocol ip high list 102
21  priority-list 1 protocol ip high list 103
22  priority-list 1 default medium
23  snmp-server community public RO
24  snmp-server community private RW
25  voice-port 2/0
26  voice-port 2/1
27  dial-peer voice 1 pots
28    destination-pattern 55
29    port 2/0
30  dial-peer voice 2 voip
31    destination-pattern 77
32     session target ipv4:30.30.30.1
33  line con 0
34  line vty 0 4
35  login
```

Figura A.6: Configuração do roteador R0

Por fim, Figura A.6 mostra a configuração da política de fila *Priority Queuing* correspondente ao roteador R0 (ver Figura 7.1). A política de fila será aplicada na interface interligada ao enlace de rede WAN, interface serial 0, que corresponde ao enlace com menor quantidade de recursos. Esta política faz com que o tráfego de dados não interfira na qualidade associada aos tráfegos de voz e de vídeo. Nesta configuração, são também mostrados alguns comandos relativos à política de fila *Custom Queuing*. Entretanto, esta política não está habilitada para esta configuração.

A.1.2 Ferramentas para Modelagem

As ferramentas correspondentes à atividade de *Definição do Modelo* serão mostradas nesta seção.

Ferramenta TimeNET

TimeNet [110, 111, 33] (ver Figura A.7) é uma ferramenta de modelagem desenvolvida pela Technische Universitat Berlin. Esta ferramenta suporta modelos SPN [32], tais como: GSPN [71], DSPN [70] e EDSPN [59]. Este projeto foi motivado pela necessidade de softwares poderosos para a avaliação eficiente de redes de Petri temporizadas com arbitrários atrasos de disparo.

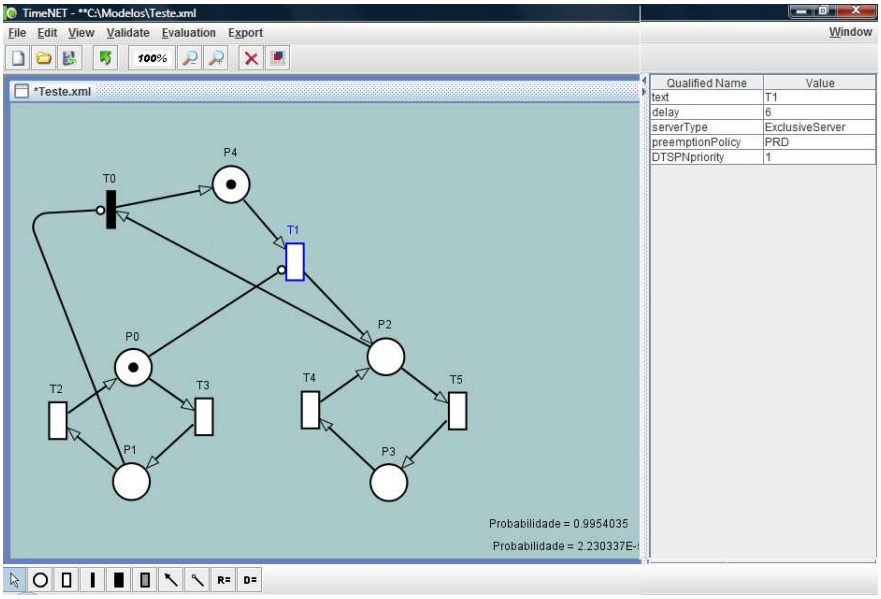


Figura A.7: TimeNet

A ferramenta proporciona uma interface gráfica e é especialmente adaptada para a análise de redes de Petri determinísticas e estocásticas.

Ferramenta SHARPE

A ferramenta SHARPE [84] (ver Figura A.8) proporciona tanto suporte para muitos tipos de modelos como mecanismos flexíveis para a combinação de resultados a fim de que os modelos possam ser utilizados em combinações hierárquicas. SHARPE permite seus usuários construir

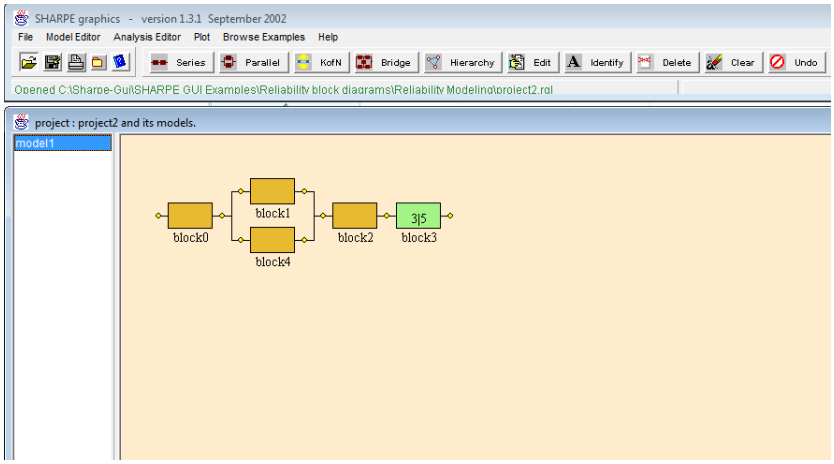


Figura A.8: SHARPE

e analisar modelos de desempenho, dependabilidade (confiabilidade/disponibilidade) e performabilidade. Ele proporciona acesso direto e completo para os modelos sem fazer qualquer suposição sobre o domínio da aplicação.

Ferramenta Mercury

Figura A.9: Ferramenta Mercury

Mercury [90, 91] (ver Figura A.9) tem sido desenvolvido pelo grupo de pesquisa *MODCS* (Modeling of Distributed and Concurrent Systems). Ele foi concebido para proporcionar suporte à avaliação de modelos de desempenho e dependabilidade e é genérico o suficiente para também permitir a avaliação da sustentabilidade de sistemas em geral. A ferramenta possui uma interface gráfica para a modelagem e avaliação de SPN, RBD, EFM e CTMC. EFM foi proposto para o cálculo da sustentabilidade e estimativas de custo de infraestruturas de refrigeração e de potência em *data centers*.

A.1.3 Ferramentas para Validação

A ferramenta utilizada para a atividade de *Validação de Modelos* é mostrada nesta seção.

Statdisk

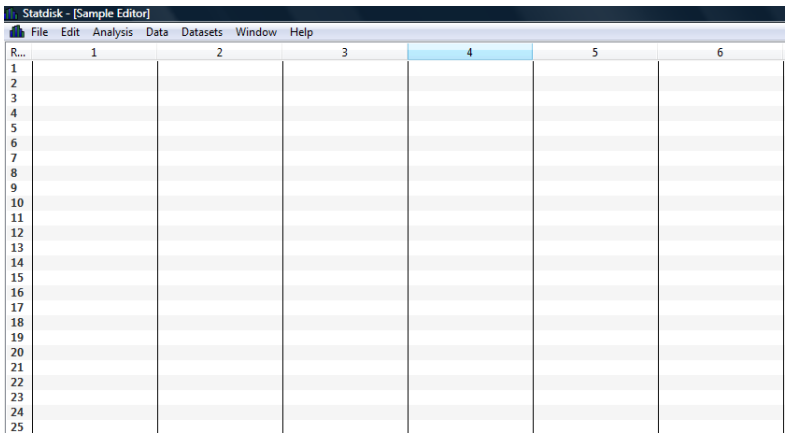


Figura A.10: Ferramenta Statdisk

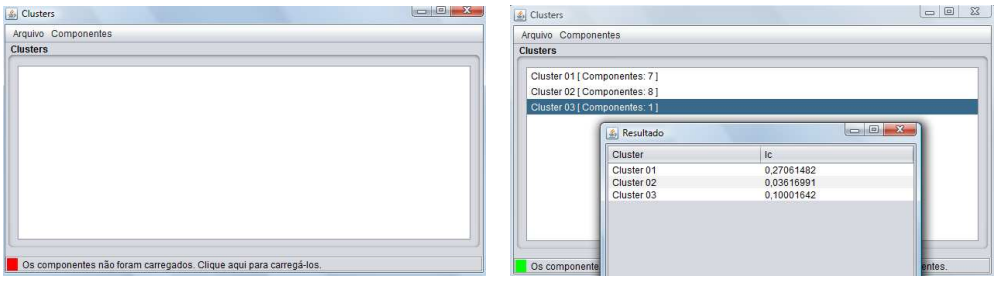
Para a validação dos resultados, foi utilizada a ferramenta Statdisk [101], versão 11.0.1. Statdisk (ver Figura A.10) é um pacote de software com diversos recursos de análise estatística. Através da utilização do método *Mean-Matched Pairs* é possível determinar que o zero (0) é parte do intervalo de confiança das métricas analisadas com um nível de confiança de 95%.

Foram comparados os valores obtidos diretamente através de medição e os valores correspondentes obtidos através dos modelos.

A.1.4 Ferramentas na Fase de Seleção do Projeto

Por fim, as ferramentas utilizadas para as atividade de *seleção de agrupamento* e *seleção de projeto* são mostradas nesta seção.

Seleção de Agrupamento - Algoritmo 2

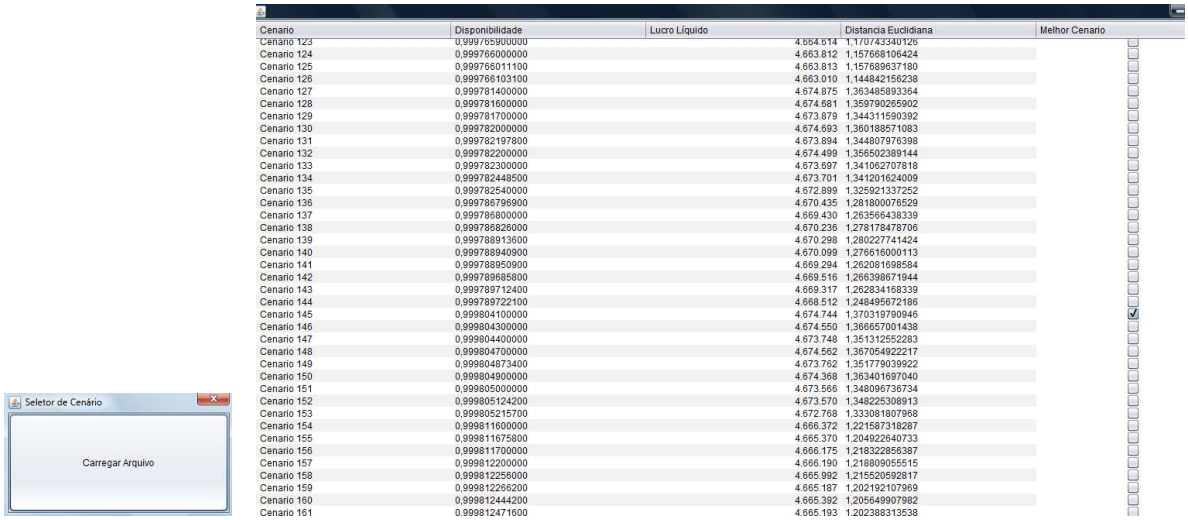


a) Tela Inicial b) Índice I_c

Figura A.11: Seleção de Agrupamento

A atividade *seleção de agrupamento* determina o agrupamento que representa os componentes com maior impacto sobre os aspectos de dependabilidade da infraestrutura. O agrupamento selecionado é aquele que possui o valor mais alto do índice I_c (ver Algoritmo 2). *w1*, *w2* e *w3* correspondem aos atributos *importância para confiabilidade*, *custo do componente* e *vazão do componente*. O Algoritmo 2 foi implementado e o código é mostrado no Apêndice C. A Figura A.11(a) mostra a tela inicial da ferramenta gerada. A Figura A.11(b) mostra a tela com o índice I_c calculado para cada agrupamento.

Seleção de Projeto - Algoritmo 3



a) Tela Inicial b) Projeto Selecionado

Figura A.12: Seleção de Projeto

O propósito da atividade *seleção de projeto* é o de proporcionar suporte ao processo de seleção do melhor projeto em uma infraestrutura de redes convergentes em função de métricas técnicas e de negócios. Esta atividade considera as duas classes de redes convergentes (ver Algoritmo 3). Primeiro, se a rede convergente gera diretamente receitas a partir de seus serviços, as métricas normalizadas de *Disponibilidade* (nA_i) e *Lucro_Líquido* (nLc_i) são consideradas. Segundo, se a rede convergente em questão não gera diretamente receitas a partir de seus serviços, as métricas normalizadas de *Indisponibilidade* (nUA_i) e *Custo de Infraestrutura* ($nICust_i$) são consideradas. O Algoritmo 3 foi implementado e o código é mostrado no Apêndice C. A Figura A.12(a) mostra a tela inicial da ferramenta gerada. A Figura A.12(b) mostra a tela com o projeto selecionado.

APÊNDICE B

Validação do Modelo de Desempenho

É importante ressaltar que o processo de validação, utilizando-se do método *Mean-Matched Pairs*[105], é realizado comparando-se os resultados obtidos a partir do modelo (ver Figura 7.6(b)), através de simulação, com os resultados obtidos a partir de medição direta realizada no próprio sistema.

B.1 Medição Direta - Sistema

A medição das métricas de *vazão de saída*, *número de pacotes descartados* e *tamanho médio da fila* foi efetuada na interface serial de acesso à rede WAN do roteador de entrada, a partir dos tráfegos de dados e de voz oriundos da máquina *A*. Os mesmos resultados são obtidos a partir de qualquer uma das arquiteturas mostradas na Figura 7.6. A escolha da interface serial do roteador de entrada deve-se ao fato de que é neste ponto onde existe o gargalo da rede (disputa aos recursos de acesso à rede WAN), pois o tráfego passa de um enlace *fast-ethernet* com uma taxa de transmissão de 100 Mbps para um enlace *PPP* [92] com uma taxa de transmissão de 128 Kbps. Desta forma, é nesta interface que são aplicadas as políticas de enfileiramento *Custom Queuing* e *Priority Queuing*. Estas políticas são adotadas pelo fato de que possibilitam uma diferenciação dos recursos de acesso à rede a cada classe de tráfego considerada.

Com relação aos dados, foram consideradas diferentes taxas de transmissão (40Kbps, 50Kbps, 60Kbps, 70Kbps e 80Kbps)[5], utilizando o gerador de tráfego TFGEN (ver Apêndice A), com

[5]Todas estas diferentes taxas possuem diferentes tamanhos de pacotes com uma vazão de 64 pps.

um tráfego contínuo e constante. Estas diferentes taxas foram escolhidas para proporcionar uma carga crescente de dados para o enlace serial. O tráfego contínuo e constante representa a vazão média de dados em um período determinado. Por outro lado, com relação ao tráfego de voz, foi também considerada uma transmissão com o CODEC G.711 (33.3 pps) [99].

B.2 Modelo de Desempenho

O modelo mostrado na Figura 7.6(b) pode ser aplicado em cada uma das arquiteturas apresentadas no Capítulo 7 para a obtenção das métricas de *vazão de saída*, *número de pacotes descartados* e *tamanho médio da fila* tanto para dados quanto para voz. Ele pode representar tanto a política de enfileiramento *Custom Queuing* quanto a política de enfileiramento *Priority Queuing* na interface de saída (serial) do roteador de entrada. Sua validação foi efetuada considerando cada uma das políticas de enfileiramento (*Custom Queuing* e *Priority Queuing*).

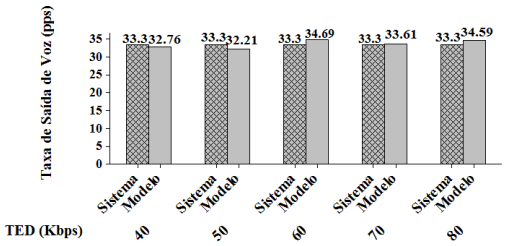
Este modelo representa, além da interface *fast-ethernet* do roteador de entrada, a interface serial de acesso à rede WAN do roteador de entrada. A representação de outros componentes em sua estrutura acarretaria em uma complexidade desnecessária, pois é no roteador de entrada onde as diferentes classes de tráfego disputam os recursos de acesso à rede WAN. Devido à grande flexibilidade por parte da política *Custom Queuing* para a alocação de recursos entre as diferentes classes de tráfego, iremos apenas mostrar os resultados relativos a esta política.

Foram utilizados os valores de 0,75 e 0,25 para as variáveis *w1* e *w2*, além do valor de 20 para as variáveis *n1* e *n2*. Com relação à adoção dos valores 0,75 e 0,25, procuramos priorizar o tráfego de voz (variável *w1*) em relação ao tráfego de dados (variável *w2*), devido ao fato de que o tráfego de voz é mais sensível à falta de recursos em relação a dados. O valor de 20 para as variáveis *n1* e *n2* representa o tamanho padrão das filas de cada classe de tráfego.

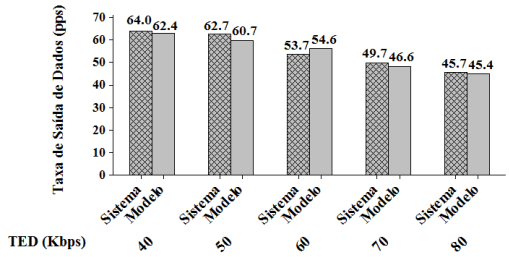
Para as variáveis *in* e *buff*, foram utilizados os valores de 75 e 20.000, respectivamente. O valor de 75, para *in*, representa o tamanho padrão da fila da interface de entrada, *fast-ethernet*, do roteador de entrada. É proporcionado um valor de 20.000 para a variável *buff* de modo a proporcionar a obtenção de um modelo limitado (ver Seção 7.3.1).

Tabela B.1: Mean matched pairs

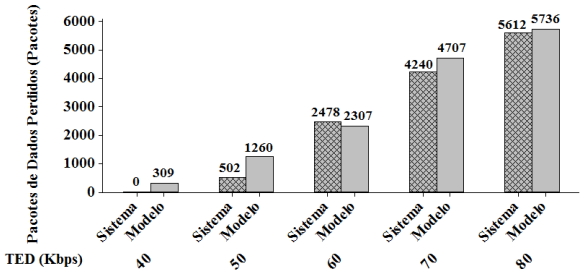
Métrica	Intervalo
Taxa de Saída de Voz	-1,632471 < σ < 1,088471
Taxa de Saída de Dados	-1,795536 < σ < 3,227536
Pacotes de Dados Perdidos	-732,2121 < σ < 137,4121
Taxa Total de saída	-3,144451 < σ < 4,144451



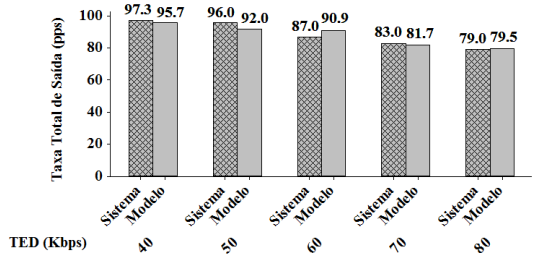
a) Transmissão de pacotes de voz



b) Transmissão de pacotes de dados



c) Pacotes de dados perdidos



d) Taxa de saída total

Figura B.1: Comparação Sistema x Modelo

Para a validação do modelo de desempenho, foram utilizadas as seguintes métricas: *Taxa de Saída de voz*, *Taxa de Saída de Dados*, *Pacote de Dados Perdidos* e *Taxa Total de Saída*. A utilização do método *Mean-Matched Pairs* mostrou que o zero (0) faz parte do intervalo de

confiança estimado, indicando a impossibilidade de se refutar a equivalência entre os resultados do sistema e do modelo (ver Tabela B.1). Como ilustração dos resultados obtidos, as Figuras B.1(a), B.1(b), B.1(c) e B.1(e) mostram uma comparação dos valores medidos a partir do sistema e dos valores obtidos através do modelo refinado. A Figura B.1(a) apresenta a influência do tráfego de dados de entrada (Taxa de Entrada de Dados, TED, em Kbps) sobre a taxa de saída de voz. A Figura B.1(b) mostra a taxa de saída de dados em função do tráfego de entrada de dados. A Figura B.1(c) mostra a influência da taxa de entrada de dados sobre a perda de pacotes. Finalmente, a Figura B.1(d) mostra a influência da variação da taxa de entrada de dados (TED) sobre a taxa de saída total (dados e voz).

APÊNDICE C

Algoritmo 2 e Algoritmo 3 - Códigos

Os algoritmos 2 e 3 foram desenvolvidos em Java, utilizando o NetBeans 7.0.1, com um formato de código fonte/binário: JDK6. Eles foram escritos, compilados e executados no sistema operacional Windows 7, 64bits. Estes algoritmos correspondem às atividades *seleção de agrupamento* e *seleção de projeto* da metodologia proposta.

C.1 Código Algoritmo 2

Módulo Cluster

```
package seletor.de.cluster;


import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;


    public class Cluster {


    private String nome;
    private List<Componente> componentes = new ArrayList<Componente>();
    private double mRI = 0;
```

```
    private double mCusto = 0;
    private double mTHP = 0;

    public Cluster(String nome) {
        this.nome = nome;
    }

    @Override
    public String toString()
        return this.getNome() + "[ Componentes: "+ this.componentes.size() + "]";
    }

    public String getNome() {
        return this.nome;
    }

    public Iterator<Componente> getComponentesIterator()
        return this.componentes.iterator();
    }

    public void setNome(String nome)
        this.nome = nome;
    }

    private void setmCusto(double mCusto)
```

```
        this.mCusto = mCusto;
    }

.....

    public getValor(double_mRI, double_mCusto, double_mTHP, int pesoRI, int pesoCusto, int pesoTHP)
{
        if (this.componentes.isEmpty()) {
            return 0;
    }
    double somaValorComponentes = 0;

.....
```

Cluster Resultado

```
package seletor.de.cluster;

public class ClusterResultado {

    private Cluster cluster;
    private double indice;

    public ClusterResultado(Cluster cluster, double indice)
```

```
        this.cluster = cluster;
        this.indice = indice;
    }

    @Override
    public String toString() {
        return "Cluster: "+ this.cluster + "Indice: "+ this.indice;
    }

    public double getIc() {
        return this.indice;
    }
    public String getNomeCluster() {
        return this.cluster.getNome();
    }
}
```

Componente

```
package seletor.de.cluster;

import java.text.DecimalFormat;

public class Componente {
```

```
    private String nome;
    private double ri;
    private double custo;
    private double thp;

    public Componente(String nome, double ri, double custo, double thp) {
        this.nome = nome;
        this.ri = ri;
        this.custo = custo;
        this.thp = thp;
    }

    public double getCusto() {
        return custo;
    }

    public String getNome() {
        return nome;
    }

    public double getRi() {
        return ri;
    }

    public double getThp() {
```

```
        return thp;
    }

.....

        double valor;
        valor = ((pesoRI * getRi() / mRI) + (pesoCusto * getCusto() / mCusto) +
        (pesoTHP * getThp() / mTHP)) / (pesoRI + pesoCusto + pesoTHP);
        return valor;
    }
}
```

Diálogo

```
package seletor.de.cluster;

import java.awt.Component;
import javax.swing.JOptionPane;

public class Dialogo extends JOptionPane {

        static public void erro(Component parent, String mensagem) {
            showMessageDialog(parent, mensagem, "Erro!", ERROR_MESSAGE);
        }
        static public int pergunta(Component parent, String mensagem) {
```

```
            return showConfirmDialog(parent, mensagem, "Atenção!", YES_NO_OPTION,
            WARNING_MESSAGE);
        }
        static public void aviso(Component parent, String mensagem) {
            showMessageDialog(parent, mensagem, "Atenção!", INFORMATION_MESSAGE);
        }
}
```

C.2 Código Algoritmo 3

Cenário

```
package seletor.de.cenario;

public class Cenario {

    private String nome;
    private double a;
    private double b;
    private double distancia = 0;

    public Cenario(int n, double a, double b) {
        this.nome = "Cenario "+ n;
```

```
        this.a = a;
        this.b = b;
    }
    public String getNome() {
        return this.nome;
    }
    public double getDistanciaEuclidiana() {
        return this.distancia;
    }

    public double calcularDistanciaEuclidiana(double maxA, double minA, double maxB,
    double minB) {
        double denominadorA = (maxA - minA != 0 ? maxA - minA : 1);
        double denominadorB = (maxB - minB != 0 ? maxB - minB : 1);
        this.distancia = Math.sqrt(Math.pow((this.a - minA) / denominadorA, 2) +
        Math.pow((this.b - minB) / denominadorB, 2));
        return this.distancia;
    }
}
```

Diálogo

```
package seletor.de.cenario;

import java.awt.Component;
import javax.swing.JOptionPane;
```

```
public class Dialogo extends JOptionPane {

    static public void erro(Component parent, String mensagem) {
        showMessageDialog(parent, mensagem, "Erro!", ERROR_MESSAGE);
    }
    static public int pergunta(Component parent, String mensagem) {
        return showConfirmDialog(parent, mensagem, "Atenção!", YES_NO_OPTION,
        WARNING_MESSAGE);
    }
    static public void aviso(Component parent, String mensagem) {
        showMessageDialog(parent, mensagem, "Atenção!", INFORMATION_MESSAGE);
    }
}
```

Seletor de Cenário

```
package seletor.de.cenario;

import javax.swing.UIManager;
import seletor.de.cenario.form.GUIBotaoArquivo;

public class SeletorDeCenario {

    public static void main(String args[]) {
```

```
        try {
            for (javax.swing.UIManager info : javax.swing.UIManager.getInstalledLookAndFeels())
{
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.getLogger(GUIBotaoArquivo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.getLogger(GUIBotaoArquivo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.getLogger(GUIBotaoArquivo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.getLogger(GUIBotaoArquivo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }

         Cria e exibe o formulário

        java.awt.EventQueue.invokeLater(new Runnable() {
```

```
        public void run() {
            traduzir();
            new GUIBotaoArquivo().setVisible(true);
        }

        private void traduzir() {
            UIManager.put("FileChooser"String.valueOf(java.awt.event.KeyEvent.VK_E));
            UIManager.put("FileChooser.lookInLabelText", "Examinar em");
            UIManager.put("FileChooser", String.valueOf(java.awt.event.KeyEvent.VK_S));
            UIManager.put("FileChooser", "Salvar em");
            UIManager.put("FileChooser", "Um nível acima");
            UIManager.put("FileChooser", "Um nível acima");
            UIManager.put("FileChooser", "Desktop");
            UIManager.put("FileChooser", "Desktop");
            UIManager.put("FileChooser", "Criar nova pasta");
            UIManager.put("FileChooser", "Criar nova pasta");
            UIManager.put("FileChooser", "Lista");
            UIManager.put("FileChooser", "Lista");
            UIManager.put("FileChooser", "Detalhes");
            UIManager.put("FileChooser", "Detalhes");
.....
            UIManager.put("FileChooser", "Salvar");
            UIManager.put("FileChooser", String.valueOf(java.awt.event.KeyEvent.VK_S));
            UIManager.put("FileChooser", "Salvar");
            UIManager.put("FileChooser", "Alterar");
            UIManager.put("FileChooser", String.valueOf(java.awt.event.KeyEvent.VK_A));
```

```
                UIManager.put("FileChooser", "Alterar");
                UIManager.put("FileChooser", "Ajuda");
                UIManager.put("FileChooser", String.valueOf(java.awt.event.KeyEvent.VK_A));
                UIManager.put("FileChooser", "Ajuda");
                UIManager.put("FileChooser", "Todos os arquivos");
                UIManager.put("OptionPane", "Sim");
                UIManager.put("OptionPane", String.valueOf(java.awt.event.KeyEvent.VK_S));
                UIManager.put("OptionPane", "Sim");
                UIManager.put("OptionPane", "Não");
                UIManager.put("OptionPane", String.valueOf(java.awt.event.KeyEvent.VK_N));
                UIManager.put("OptionPane", "Não");
                UIManager.put("OptionPane", "Cancelar");
                UIManager.put("OptionPane", String.valueOf(java.awt.event.KeyEvent.VK_C));
                UIManager.put("OptionPane", "Cancelar");
                UIManager.put("OptionPane", String.valueOf(java.awt.event.KeyEvent.VK_O));
                UIManager.put("OptionPane", "OK");
            }
        });
    }
}
```

APÊNDICE D

Diagrama do Problema e Diagrama de Escopo

D.1 Diagrama do Problema

Como exemplo de um Diagrama de Problema, Figura D.1 mostra a topologia de uma rede de computadores. O nó central, os nós folha (sites) e os enlaces entre eles formam uma topologia estrela. Figura D.2 mostra uma visão geral da relação lógica entre os componentes desta rede.

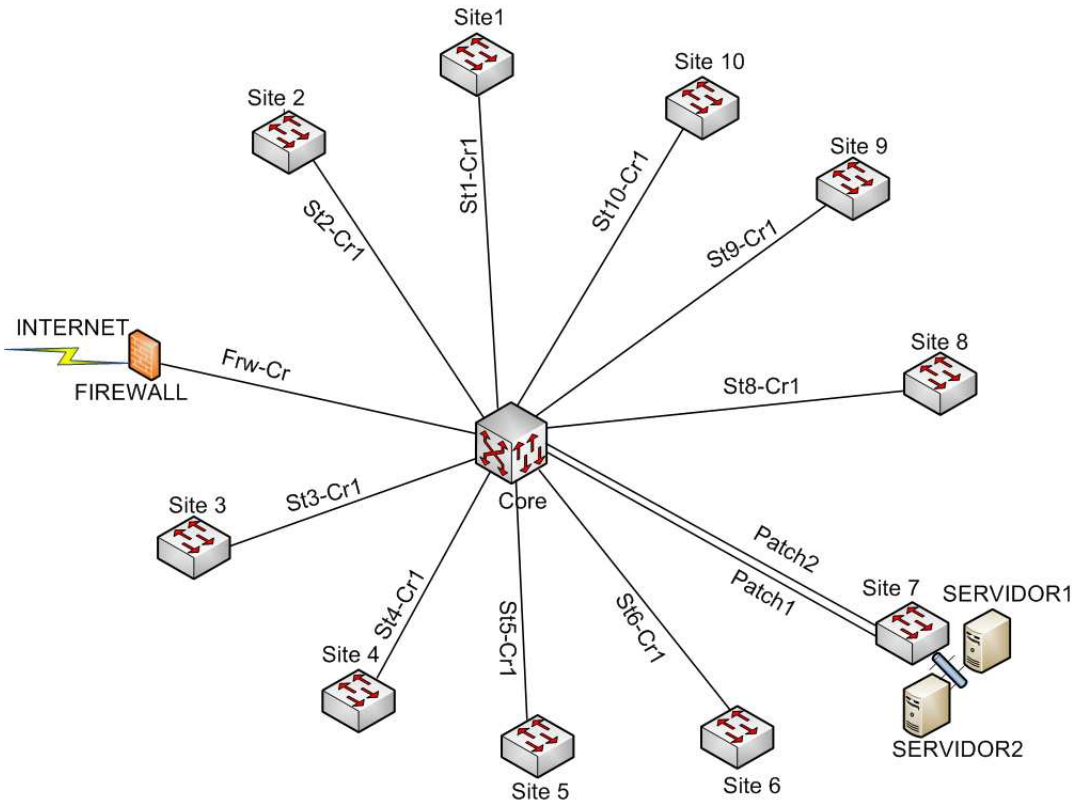


Figura D.1: Topologia da rede

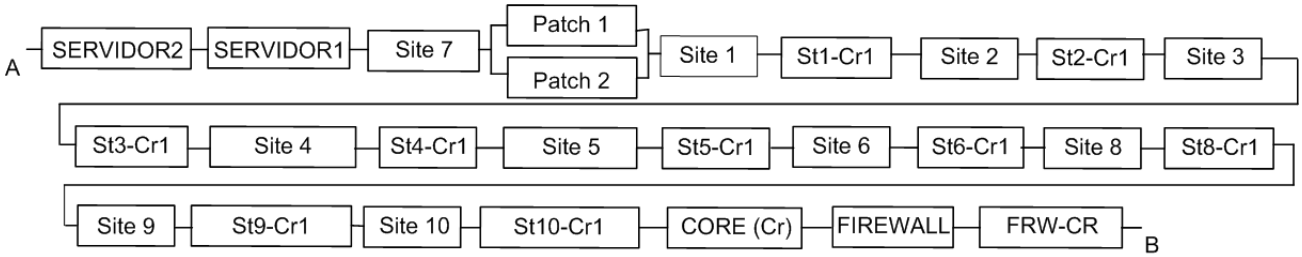


Figura D.2: Visão geral dos componentes da rede

Figura D.3 mostra uma visão geral da relação entre as partes funcionais de um dos componente, CORE. Como o CORE agrega o tráfego de toda a rede, este foi dimensionado com partes redundantes. Ele é constituído por duas fontes de potência e duas placas processadores, junto com seus sistemas operacionais, além do chassis/placa mãe e uma placa de interface. Para que o sistema funcione, é necessário que o chassis/placa mãe e a placa de interface estejam funcionando, assim como pelo menos uma das placas processadoras/sistema operacional e fontes de potência.

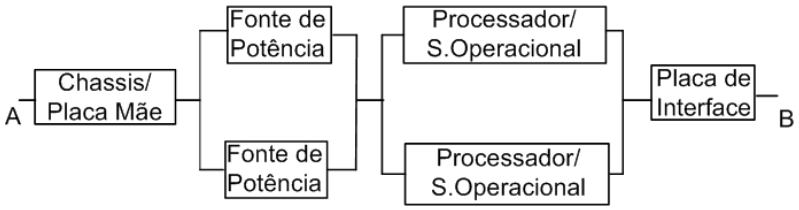


Figura D.3: Visão geral das partes funcionais do componente CORE

Figura D.4 mostra uma visão geral da relação entre as partes funcionais do componente Nó Folha (Site). Como este componente apenas agrega tráfego local, não foi necessário o dimensionamento de partes redundantes. Este componente é constituído por um chassis/placa mãe, por uma fonte de potência e pelo sistema operacional. Caso uma das parte falhe, todo o componente irá falhar.

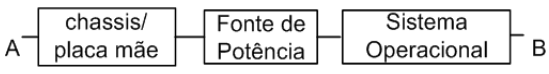


Figura D.4: Visão geral das partes funcionais do componente Nó Folha

Por fim, Figura D.5 mostra uma visão geral da relação entre as partes funcionais dos com-

ponentes servidor/firewall. O *hardware* inclui os seguintes elementos: processador, fonte de potência, cpu, resfriador, memoria e interface de rede.

Figura D.5: Visão geral das partes funcionais dos componentes Servidor/Firewall

D.2 Diagrama do Escopo

Para facilitar a visualização de quais elementos são considerados em relação aos aspectos de infraestrutura e de negócios, dois diagramas de escopo são elaborados.

O primeiro, refere-se à classe de redes convergentes não gera diretamente receitas a partir de seus serviços. Neste diagrama, dependabilidade (confiabilidade e disponibilidade) e custo de infraestrutura impactam diretamente sobre o projeto de uma infraestrutura. Como fator externo, tem-se o ambiente financeiro que poderá impactar sobre os custos da infraestrutura. Figura D.6 representa o Diagrama de Escopo para esta classe de redes.

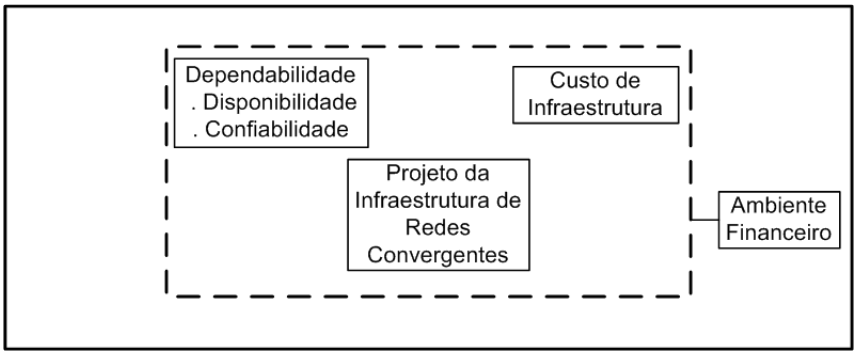


Figura D.6: Diagrama de escopo 1

O segundo Diagrama de Escopo refere-se à classe de redes convergentes que gera diretamente receitas a partir de seus serviços. A dependabilidade (confiabilidade e disponibilidade), desempenho (tamanho da fila, descarte de pacotes e vazão), custo de infraestrutura, receita de

infraestrutura, multa e lucro líquido impactam diretamente sobre o projeto de uma infraestrutura. Dentre os fatores externos considerados neste trabalho podemos mencionar o ambiente financeiro e a política de filas. O diagrama de escopo desta classe de redes é mostrado na Figura D.7.

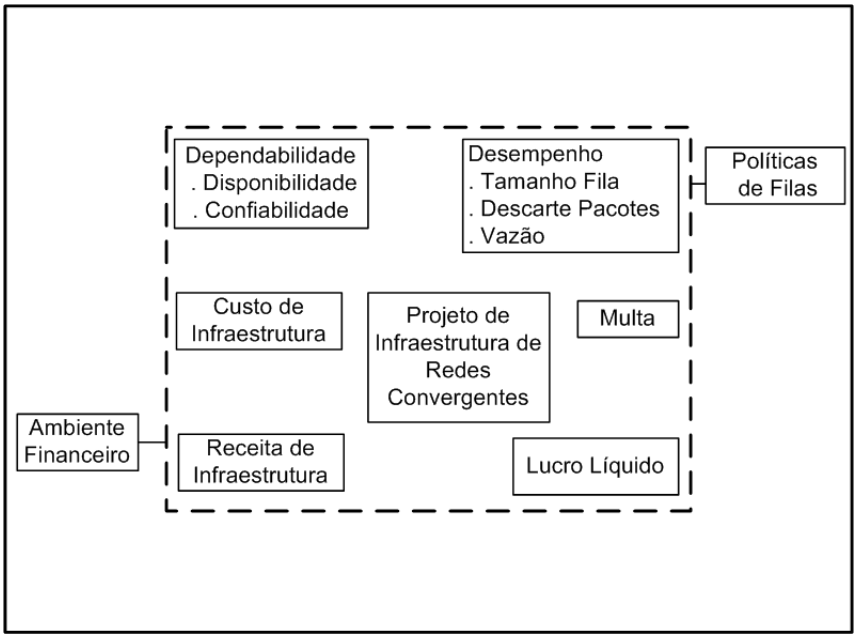


Figura D.7: Diagrama de escopo 2

APÊNDICE E

Ilustração Fase de Seleção do Projeto

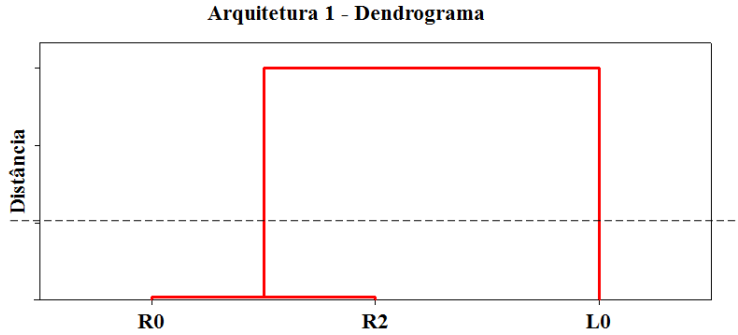
E.1 Descrição

Devido a sua complexidade e importância, este apêndice busca ilustrar as atividades mais importantes da Fase III da metodologia proposta para facilitar a sua compreensão (ver Figura E.1). Estas atividades foram aplicadas considerando a Arquitetura 1 (ver Figura 7.1(a)). A Figura E.1(a) ilustra, através de um dendrogrma, a atividade de Geraçao de Agrupamentos. Foi utilizada a abordagem de agrupamento hierárquico aglomerativo.

Em cada um dos passos da abordagem de agrupamento hierárquico aglomerativo foram determinados os valores de RMSSTD em função do número de agrupamentos. Estes valores são adicionalmente normalizados segundo a Equação 6.1. A quantidade de agrupamentos referente ao passo i, que corresponde à menor distância Euclidiana dos valores normalizados de RMSSTD e do número de agrupamentos, para a origem, indica o número bom de agrupamentos. A Figura E.1(b) ilustra a atividade de Definição do Número Bom de Agrupamentos. Por outro lado, a linha pontilhada na Figura E.1(a) indica o bom número de agrupamentos através do dedrograma.

A Figura E.1(c) ilustra a atividade Seleção de Agrupamento, a partir do número bom de agrupamentos, utilizando o índice I_c. Esta figura mostra que o primeiro agrupamento foi selecionado.

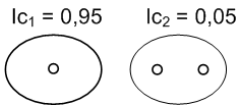
Para uma maior facilidade de compreensão, as atividades Análise de Experimento, Definição e Avaliação de Cenários e Seleção de Projeto são ilustradas conjuntamente utilizando a Figura



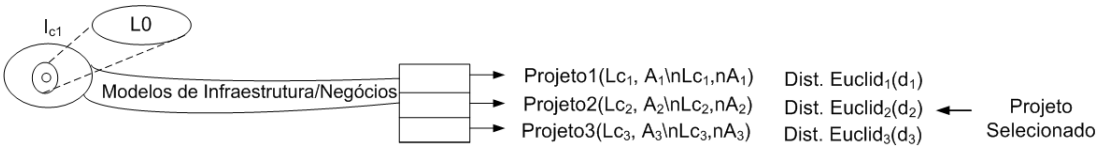
a) Geração de Agrupamentos

N. Agrup.	RMSSTD	Norm. Agrup.	Norm. RMSSTD	Dist. Euclid.
1	5379	0,00	1,00	1,00
2	737	0,50	0,14	0,52
3	0	1,00	0,00	1,00

b) Definição do Número Bom de Agrupamentos



c) Seleção de Agrupamento



d) Análise de Experimento/Definição e Avaliação de Cenários/Seleção de Projeto

Figura E.1: Atvidades Fase III - Metodologia proposta

E.1(d). Após a seleção do agrupamento utilizando o índice I_c na Figura E.1(c), e considerando os componentes representados no agrupamento selecionado (L0), projeto de experimento fatorial 2^K é analisado. Então, a atividade Análise de Experimento escolhe o componente do agrupamento selecionado com maior impacto sobre a disponibilidade do sistema. Neste caso, o componente selecionado foi L0. A atividade Definição e Avaliação de Cenários produz um conjunto de cenários que são representados por diferentes combinações de parâmetros de entrada, relacionados aos modelos de infraestrutura e de negócios. Estes cenários são construídos com base no componente selecionado na atividade anterior, considerando suas diferentes opções. Os cenários serão avaliados com base nos valores de métricas associadas. Os valores

de métricas serão normalizados em cada cenário *i*.

Com relação à atividade Seleção de Projeto, o projeto com maior distância Eucliana ou menor distância Euclidiana será selecionado, como o melhor projeto, em função da abordagem utilizada. A Figura E.1(d) ilustra a abordagem que considera a maior distância Euclidiana.