



**UNIVERSIDADE FEDERAL DE ALAGOAS
CAMPUS A.C SIMÕES
BACHARELADO EM SISTEMA DE INFORMAÇÕES**

FLAVIO PEREIRA DA SILVA

**Avaliação de Dependabilidade em Redes de Computadores, utilizando os
Mecanismos de Modelagem Analítica e Análise de Sensibilidade: Um estudo de
caso baseado em uma pequena empresa.**

**Maceió
2020**

FLAVIO PEREIRA DA SILVA

Avaliação de Dependabilidade em Redes de Computadores, utilizando os Mecanismos de Modelagem Analítica e Análise de Sensibilidade: Um estudo de caso baseado em uma pequena empresa.

Monografia apresentada ao curso de Sistema de Informações da Universidade Federal de Alagoas-campus A.C. Simões como requisito parcial para obtenção do grau de Bacharel em Sistema de informações.

Orientador (a): Professor Doutor Almir Pereira Guimarães

**Maceió
2020**

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

S586a Silva, Flávio Pereira da.

Avaliação de dependabilidade em redes de computadores utilizando os mecanismos de modelagem analítica e análise de sensibilidade: um estudo de caso baseado em uma pequena empresa / Flávio Pereira da Silva. – 2020.

52 f. : il., figs. e tabs. color.

Orientador: Almir Pereira Guimarães.

Monografia (Trabalho de Conclusão de Curso em Sistemas de Informação: Bacharelado) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2021.

Bibliografia: f. 38-41.

Apêndices: f. 42-52.

1. Redes de computadores. 2. Diagrama de blocos de confiabilidade. 3. Análise de sensibilidade. I. Título.

CDU: 004.7

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus. A ele toda honra e toda glória, nos momentos mais difíceis sei que me carregou no colo, e sentimentos de esperança e fé nunca me faltaram. Gratidão por eu ter o meu caminho abençoado, agradeço por todas as bênçãos, sem sua força nada disso teria sentido.

Aos meus pais Maria Jose Moreno da Silva e Antônio Pereira da Silva que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória.

À minha esposa Joselma Maria de Lima pela compreensão e paciência demonstrada durante o período do projeto.

Agradeço ao meu orientador Dr^o Almir Pereira Guimarães por aceitar conduzir o meu trabalho de pesquisa, e por dedicar seu tempo na construção e elaboração do trabalho apresentado.

RESUMO

Com a crescente dependência das atividades sociais e econômicas em sistemas de informação, existe grande expectativa quanto à disponibilidade, confiabilidade e robustez para as redes de computadores subjacentes a estes sistemas. Essa expectativa é devido à possibilidade de falhas sobre estas redes que podem ser originadas de diversas fontes, tais como: falhas de *hardware*, bugs de *software*, ataques mal-intencionados, erros humanos de operação/manutenção e desastres naturais. Justifica-se a relevância deste tema em virtude da grande importância que as redes de computadores desempenham para o funcionamento dos sistemas vitais das empresas.

O presente trabalho possui por objetivo proporcionar modelos analíticos, utilizando o mecanismo de modelagem diagrama de blocos de confiabilidade, para analisar o nível de disponibilidade de redes de computadores considerando a implantação de redundância em componentes críticos para o funcionamento destas redes. Além disso, é utilizado o índice de análise de sensibilidade para determinar quais parâmetros, relacionados aos seus componentes, possuem maior impacto sobre sua disponibilidade a fim de direcionar as estratégias para o incremento desta métrica. Foi analisada a infraestrutura da rede de computadores original (arquitetura base) e também foram propostas duas arquiteturas adicionais com um incremento no nível de redundância a fim de analisar o impacto de cada um de seus parâmetros sobre a disponibilidade do sistema. O tempo de reparo relacionado ao enlace na arquitetura base, o tempo de reparo relacionado ao servidor na arquitetura 2 e o tempo de reparo relacionado ao host na arquitetura 3 são os parâmetros que mais impactam sobre a disponibilidade de todo o sistema.

Palavras-chave: Dependabilidade; Análise de Sensibilidade; Diagrama de Blocos de Confiabilidade; Redes Corporativas.

ABSTRACT

With the growing dependence on social and economic activities in information systems, there is great expectation regarding the availability, reliability and robustness for the computer networks underlying these systems. This expectation is due to the possibility of failures on these networks that can originate from several sources, such as: hardware failures, software bugs, malicious attacks, human operation / maintenance errors and natural disasters. The relevance of this topic is justified in view of the great importance that computer networks play for the functioning of companies' vital systems.

The present work aims to provide analytical models, using the reliability block diagram modeling mechanism, to analyze the level of availability of computer networks considering the implementation of redundancy in critical components for the functioning of these networks. In addition, the sensitivity analysis index is used to determine which parameters, related to their components, have the greatest impact on their availability in order to direct strategies for increasing this metric. The infrastructure of the original computer network (base architecture) was analyzed and two additional architectures were proposed with an increase in the level of redundancy in order to analyze the impact of each of its parameters on the availability of the system. The repair time related to the link in the base architecture, the repair time related to the server in architecture 2 and the repair time related to the host in architecture 3 are the parameters that most impact on the availability of the entire system.

Keywords: Dependability. Sensitivity Analysis; Reliability Block diagram; Corporate Networks.

LISTA DE FIGURAS

Figura 1- Taxonomia da Dependabilidade.....	7
Figura 2- Estruturas Básicas.....	24
Figura 3- Metodologia Utilizada.....	27
Figura 4- Topologia da Rede Corporativa.....	29
Figura 5 - Modelo de Dependabilidade RBD – Arquitetura A1.....	30
Figura 6 - Modelo de Dependabilidade RBD – Arquitetura A2.....	30
Figura 7- Modelo de Dependabilidade RBD – Arquitetura A3.....	30
Figura 8 - Variação MTTR X Disponibilidade Sistema – Arquitetura 1.....	34
Figura 9 - Variação MTTR X Disponibilidade Sistema – Arquitetura 2.....	34
Figura 10 - Variação MTTR X Disponibilidade Sistema – Arquitetura 3.....	35

LISTA DE TABELAS

Tabela 1 - Parâmetro de Entrada – Modelos RBD.....	31
Tabela 2 – Análise de Sensibilidade Arquitetura 1.....	33
Tabela 3 – Análise de Sensibilidade Arquitetura 2.....	33
Tabela 4 – Análise de Sensibilidade Arquitetura 3.....	34

LISTA DE SIGLAS E ABREVIATURAS

FT: Árvore de Falhas. Sigla proveniente do *inglês* *Fault Tree*.

LAN: Rede de Área Local. Sigla proveniente do *inglês* *Local Area Network*.

MAN: Rede de Área Metropolitana. Sigla proveniente do *inglês* *Metropolitan Area Network*.

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*.

RBD: Diagrama de Blocos de Confiabilidade. Sigla proveniente do *inglês* *Reliability Block Diagram*.

SPN: Redes de Petri Estocásticas. Sigla proveniente do *inglês* *Stochastic Petri Nets*.

TTF: Tempo para falha. Sigla proveniente do *inglês* *Time To Failure*;

TTR: Tempo para reparo. Sigla proveniente do *inglês* *Time To Repair*;

WAN: Rede de Área Geográfica. Sigla proveniente do *inglês* *Wide Area Network*.

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO.....	11
1.1 Visão Geral.....	11
1.2 Motivação.....	12
1.3 Objetivos.....	12
1.3.1 Objetivo Geral.....	13
1.3.2 Objetivos específicos.....	13
1.4 Estrutura do trabalho.....	14
CAPÍTULO 2: FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 Redes de Computadores	15
2.1.1 Classificação de Redes de Computadores.....	16
2.2 Topologias de Rede.....	17
2.3 Dependabilidade.....	18
2.4 Mecanismos de Redundância.....	21
2.5 Modelagem de sistema.....	22
2.5.1 Modelos Combinatoriais.....	23
2.5.1.1 Reliability Block Diagram.....	23
2.6 Análise de Sensibilidade – Diferença Percentual.....	24
CAPÍTULO 3: TRABALHOS RELACIONADOS.....	26
CAPÍTULO 4: Metodologia.....	28
CAPÍTULO 5: Estudo de Caso.....	30
CAPÍTULO 6: CONSIDERAÇÕES FINAIS.....	37
REFERÊNCIAS.....	38
ANEXOS.....	42

1: INTRODUÇÃO

1.1 Visão Geral

Ao se analisar os séculos anteriores, percebe-se que a tecnologia desempenhou um papel fundamental em cada século, determinando a principal forma de produção em cada período. Os sistemas mecânicos predominaram no século XVIII. As máquinas a vapor representaram a principal inovação do século XIX. No século XX, os principais avanços foram relacionados com a aquisição, processamento e disseminação de informações. Dentre as mais notáveis tecnologias deste século pode-se mencionar o rádio, a televisão, os computadores, as redes de computadores e a Internet (MACEDO *et al.*, 2018).

O surgimento das redes de computadores mudou a forma como os sistemas computacionais eram organizados. Inicialmente, os sistemas computacionais eram organizados tendo como base um único computador responsável por executar todo o processamento e armazenamento necessários. Este modelo foi massivamente empregado quando o custo dos computadores era expressivamente alto, dificultado a aquisição de uma grande quantidade destas máquinas por organização (MACEDO *et al.*, 2018).

Segundo as palavras de [Kurose et al, 2010]:

Essa nova forma de pensar os sistemas computacionais abriu as portas para uma nova área, as redes de computadores. Neste momento, era evidente os benefícios da adoção de um único computador para realizar tarefas consideradas onerosas para os seres humanos, tais como a execução de cálculos complexos e o armazenamento de uma vasta quantidade de informação. Se apenas um computador pode gerar muitos benefícios, imagine se vários deles pudessem se comunicar e operar de maneira cooperativa? Pensando dessa forma podemos afirmar que o princípio por trás da criação das redes de computadores foi possibilitar a interconexão entre diversos computadores a fim de trocar dados e explorar os benefícios provenientes.

De acordo com o que foi descrito acima, pode-se afirmar que nos dias atuais, seria praticamente impossível para a sociedade sobreviver sem o suporte das redes de computadores. A cada dia que passa, mais pessoas adquirem dispositivos capazes de se conectar à Internet para usufruir dos mais variados tipos de serviços oferecidos através da rede, tais como serviços de correio eletrônico (e-mail), redes sociais ou serviços de mensagens instantâneas. Estes serviços foram cuidadosamente projetados para potencializar a execução das atividades cotidianas das pessoas de

forma a minimizar os ônus envolvidos nas atividades e melhorar significativamente a sua eficácia [MACEDO *et al.*, 2018].

Diante deste mundo cada vez mais dependente da interconexão das redes de computadores e dos serviços que rodam sobre estas redes, surge a necessidade de se criar infraestruturas de redes de computadores robustas. Então, o estudo da dependabilidade em infraestruturas de redes de computadores de modo a poder mensurar a chance da falha destes sistemas com base em seus componentes e diminuir sua chance de falha mostra-se de grande importância. Embora as falhas não possam ser evitadas, suas consequências, ou seja, parada total ou parcial de sistemas e de seus serviços, podem ser minimizadas através do uso adequado de técnicas de redundância.

Diante destes fatos, o presente trabalho pretende analisar os aspectos de dependabilidade de redes de computadores, tais como sua disponibilidade, através da adoção de mecanismos de modelagem analítica e de análise de sensibilidade junto a diferentes infraestruturas de redes de computadores com diferentes níveis de redundância.

1.2 Motivação

Devido à importância estratégica, empresas dos mais diferentes setores estão sendo empurradas para uma realidade que as obriga a concentrar uma maior quantidade de esforços em questões relativas a aspectos de dependabilidade de suas redes de computadores [Guimarães, 2013].

A avaliação da dependabilidade de redes de computadores pode ser considerada uma atividade essencial para promover a melhoria da qualidade dos serviços prestados e para o planejamento de infraestruturas. Além disso, é possível verificar atributos destas redes que não correspondem a níveis adequados de serviços e, então, propor melhorias para maximizar o tempo de atividade destes sistemas em geral. Segundo [Guimarães, 2013], devido a fatores como por exemplo o custo de configuração, modelos analíticos foram adotados como estratégia para uma análise quantitativa. A utilização deste tipo de modelagem deve-se ao fato de o sistema ainda não existir ou pela necessidade, facilidade e custos de criação de diferentes situações e cenários.

As atividades de melhoria envolvem os meios pelos quais os níveis dos aspectos de dependabilidade podem ser incrementados, destacando-se soluções para prevenção, tolerância e remoção de falhas, que podem incluir elaboração de modelos analíticos, técnicas para a redundância de componentes e verificação de análise de sensibilidade dos parâmetros dos componentes.

Segundo [Camboim, 2012], devido aos inúmeros serviços oferecidos pelas redes de computadores, muitas empresas têm seus negócios baseados em seus serviços tais como: correio eletrônico, mensagens instantâneas, VoIP, videoconferência etc. Desta forma, em caso de sua indisponibilidade, operações podem ser interrompidas completamente, agregando prejuízos de diversas naturezas que podem acarretar graves consequências tanto em aspectos financeiros quanto em aspectos de imagem para as empresas detentoras destas redes. Desta forma, a análise de aspectos de dependabilidade mostram-se de grande importância para manter as redes de computadores em estado operacional com níveis de disponibilidade e confiabilidade adequados.

1.3 Objetivos

1.3.1 Objetivos gerais

Proporcionar uma análise de aspectos de dependabilidade de redes de computadores através da utilização de modelos analíticos e do mecanismo de análise de sensibilidade. Foi utilizado o mecanismo de modelagem de diagrama de blocos de confiabilidade, a fim de quantificar o nível de disponibilidade destas redes e foram também propostos mecanismos de redundância em componentes críticos a fim de incrementar seu nível de disponibilidade.

1.3.2 Objetivos específicos

Como objetivos específicos, o presente trabalho se propõe:

- Criar modelos de RBD de dependabilidade;
- Proporcionar um estudo adicional de aspectos de dependabilidade utilizando o índice de análise de sensibilidade;
- Proporcionar componentes redundantes.

1.4 Estrutura do trabalho

O trabalho por ora apresentado será dividido em cinco (5) capítulos. Além deste capítulo introdutório, o capítulo 2 refere-se a trabalhos relacionados ao tema em questão. O capítulo 3 trata da fundamentação teórica onde foram descritos conceitos fundamentais de rede de computadores, dependabilidade, modelagem, diagrama de blocos de confiabilidade e análise de sensibilidade. O capítulo 4 descreve a metodologia realizada para compor o trabalho, ou seja, compreensão do problema, obtenção de dados, construção de modelos analíticos, análise de cenário e interpretação de resultados. O capítulo 5 trata do estudo de caso detalhando os diversos cenários com análise e interpretação de resultados. Por fim, O capítulo 6 trata das considerações finais que destaca o que foi feito no trabalho assinalando seus pontos fortes e fracos com sugestões para trabalhos futuros.

CAPÍTULO 2: FUNDAMENTAÇÃO TEÓRICA

Este capítulo abordará tópicos sobre uma visão geral das redes de computadores, dependabilidade, mecanismos de redundância, modelagem de sistemas, diagrama de blocos de confiabilidade e análise de sensibilidade.

2.1 Redes de Computadores

Hoje em dia, tanto no ambiente explícito da informática quanto fora dele, todos nós temos contato com algum tipo de rede em maior ou menor grau. As redes de computadores surgiram da necessidade da troca de informações, onde é possível ter acesso a um dado que está fisicamente localizado longe de você [TORRES, 2001].

[TANENBAUM, 1997] enfatiza que a facilidade de se trocar dados e compartilhar periféricos, como impressoras ou scanners, é o motivo básico de uma rede, podendo significar uma redução nos custos dos equipamentos e aumentar a confiabilidade do sistema, pois tem fontes alternativas de fornecimento.

As redes de computadores permitem que as aplicações distribuídas utilizem login remoto, correio eletrônico, transmissão de áudio e vídeo em tempo real (como em uma videoconferência), jogos distribuídos, a World Wide Web (www) e muito mais [KUROSE et al., 2003].

Segundo [FOROUZAN et al., 2013], uma rede de computadores é a interligação de um conjunto de dispositivos capazes de se comunicar. Um dispositivo pode ser um host (ou um sistema final como às vezes é chamado), tal como um grande computador, *desktop*, *laptop*, estação de trabalho, telefone celular ou sistema de segurança. Um dispositivo nessa definição também pode ser um dispositivo de conexão, tal como um roteador, que liga uma rede a outras redes, um *switch* (ou comutador) que liga dispositivos entre si, um *modem* (*modulador-demodulador*) que altera a forma dos dados, e assim por diante. Tais dispositivos em uma rede são conectados utilizando meios de transmissão com ou sem fio, como cabo ou ar.

Segundo [SOARES et al., 1995 apud GOMES, 2005], o sistema de comunicação em uma rede de computadores constitui-se de um arranjo topológico interligando os vários módulos processadores através de enlaces físicos (meios de comunicação) e de um conjunto de regras com o fim de organizar a comunicação (protocolos).

A possibilidade de mesclar informações, comunicação e entretenimento certamente dará origem a uma nova e avançada indústria baseada nas redes de computadores [TANENBAUM, 1997 apud GOMES, 2005].

2.1.1 – Classificação de Redes de Computadores

As redes de computadores podem ser classificadas, baseado em seu alcance, em LAN, MAN e WAN.

Segundo [Tanenbaum, 1996], Redes de Área Local, geralmente denominadas de LANs, são redes privadas dentro de um único prédio ou campus de até alguns metros de tamanho. Eles são amplamente usados para conectar computadores pessoais e estações de trabalho em escritórios e fábricas de empresas com o objetivo de compartilhar recursos e trocar informações. As LANs se distinguem de outros tipos de redes por três características: (1) seu tamanho, (2) sua tecnologia de transmissão e (3) sua topologia.

Ainda segundo [Tanenbaum, 1996], as LANs são restritas em tamanho, o que significa que o tempo de transmissão no pior caso é limitado e conhecido com antecedência. Conhecer esse limite torna possível usar certos tipos de projetos que de outra forma não seriam possíveis. Também simplifica o gerenciamento da rede.

As LANs tradicionais funcionam a velocidades de 10 Mbps, 100Mbps, 1Gbps e 10 Gbps, têm baixo atraso (dezenas de microssegundos) e cometem poucos erros.

Segundo [Tanenbaum, 1996] explica que uma rede metropolitana, ou MAN, é, na verdade, uma versão ampliada de uma LAN, pois basicamente os dois tipos de rede utilizam tecnologias semelhantes. Uma MAN pode abranger um grupo de escritórios vizinhos ou uma cidade inteira e pode ser privada ou pública. Esse tipo de rede é capaz de transportar dados, voz e vídeo, podendo inclusive ser associado à rede de televisão a cabo local.

Por sua vez, redes WAN abrange uma grande área geográfica, geralmente um país ou continente. Ele contém uma coleção de máquinas destinadas a executar programas do usuário (aplicações). Os hosts são conectados por uma sub-rede de comunicação. O trabalho da sub-rede é transportar mensagens de host para host. Na maioria das redes de longa distância, a sub-rede consiste em dois componentes distintos: linhas de transmissão e elementos de comutação. As linhas de transmissão movem bits entre máquinas. Os elementos de comutação são computadores

especializados usados para conectar duas ou mais linhas de transmissão. quando os dados chegam em uma linha de entrada, o elemento de comutação deve escolher uma linha de saída para encaminhá-los.

2.2 Topologias de Rede

Segundo [Soares et al., 1995], a topologia de uma rede refere-se à forma como os enlaces físicos e os nós de comutação estão organizados, determinando os caminhos físicos existentes e utilizáveis entre quaisquer pares de estações conectadas a essa rede. A topologia de uma rede descreve como o é o *layout* do meio através do qual há o tráfego de informações, e também como os dispositivos estão conectados a ele.

Ainda segundo [Soares et al., 1995], entre as principais topologias existentes, podem-se citar a topologia em barramento na qual todas as estações estão interligadas ao mesmo meio de transmissão. Cada estação conectada à barra pode escutar todas as informações transmitidas, similar às transmissões de radiofusão. Estrela que é caracterizada por um elemento central aonde cada estação da rede está interligada e através do qual todas as mensagens devem atravessar. Este elemento central age como centro de controle da rede, interligando todos as outras estações. A topologia em anel procura diminuir ao máximo o número de ligações no sistema. Esta topologia consiste em estações conectadas através de um caminho fechado.

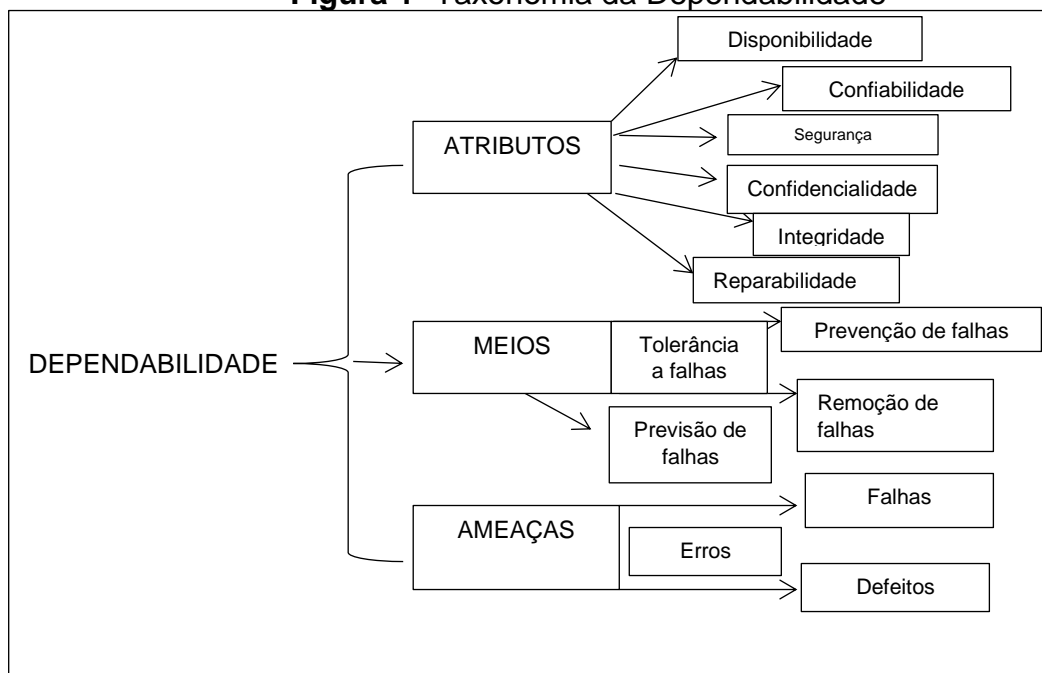
[SOARES et al, 1995] cita ainda a topologia parcialmente interligada aonde nem todas as ligações entre pares de estão presentes, mas caminhos alternativos existem e podem ser utilizados em caso de falhas ou congestionamento em determinadas rotas. No caso em que estações sem conexão física direta desejam se comunicar, elas deverão de alguma forma, encaminhar as suas mensagens para alguma outra estação que possa fazer a entrega da mensagem para a estação de destino.

2.3 Dependabilidade

Segundo [Guimarães, 2013], com relação às redes de computadores, 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. A possibilidade de evitar falhas que podem colocar os serviços em risco é um direcionamento que deve ser sempre seguido.

[AVIZIENIS et al., 2004] afirma que "a dependabilidade de um sistema de computadores deve ser entendida como a habilidade de se evitar falhas em serviços que são mais frequentes e mais severas do que seriam toleráveis". [AVIZIENIS, 2004] também define que "dependabilidade é uma propriedade dos sistemas computacionais que define a capacidade dos mesmos de prestar um serviço no qual se pode justificadamente confiar". O serviço prestado pelo sistema é o seu comportamento, como percebido pelos seus usuários. O usuário de um serviço pode ser outro sistema ou um ser humano, através de uma interface

Figura 1- Taxonomia da Dependabilidade



Fonte: [AVIZIENIS , 2004]

A função do sistema traduz e costuma dizer para o que ele foi projetado e deve ser descrito na sua especificação. A dependabilidade do sistema envolve um conjunto de conceitos que alguns autores costumam dividir em três grupos: atributos, meios pelos quais será alcançada e ameaças. Figura 1 abaixo ilustra uma taxonomia proposta por [AVIZIENIS et al., 2004].

Segundo [AVIZIENIS, 2004], com relação às ameaças pode-se definir que o correto serviço é entregue quando o serviço implementa a função especificada pelo sistema. Uma falha de serviço (failure) é um evento que ocorre quando o serviço é desviado do serviço correto.

Desde que um serviço é uma sequência estados externos do sistema, uma falha de serviço significa que pelo menos um (ou mais) estados externos do sistema desvia do serviço correto. O desvio é chamado de erro (error). A causa de um estado de erro (error) é denominada de falta (fault). [AVIZIENIS, 2004] define da seguinte maneira os meios para a obtenção de dependabilidade:

- Prevenção de Falhas: significa prevenir a ocorrência ou introdução de falhas.
- Tolerância a Falhas: significa evitar falhas dos serviços na presença de falhas.
- Remoção de falhas: significa reduzir o número e a gravidade das falhas.
- A previsão de falhas: significa estimar o presente número, a incidência futura e as consequências prováveis de falhas.

Ainda segundo [AVIZIENIS, 2004], a prevenção de falhas e a tolerância a falhas visam fornecer a capacidade de entregar um serviço que pode ser confiável, enquanto que remoção de falhas e previsão de falhas tem por objetivo alcançar confiança em suas capacidades, justificando que as especificações funcionais e de dependabilidade e segurança são adequadas e que é provável que o sistema as cumpra.

Segundo [AVIZIENIS et al., 2004], os atributos de dependabilidade englobam os conceitos de disponibilidade, confiabilidade, integridade, manutenibilidade e segurança". [ALBERTINI, 2008] proporciona as seguintes definições para estes atributos

- Disponibilidade: é definido como o tempo em um período em que o sistema encontrava-se funcionando normalmente. O fator importante é a fração de tempo na qual um sistema deverá ter condições de apresentar o serviço correto. Normalmente são usadas métricas de tempo médio para ocorrência de defeito e tempo médio para reparo para o levantamento da disponibilidade.
- Confiabilidade: pode ser traduzido como a probabilidade de o sistema não apresentar defeito durante o intervalo de tempo considerado.

- **Segurança:** é considerado sobre dois aspectos, contra catástrofes e convencional. No primeiro é a probabilidade do sistema não apresentar defeito que leve a consequências catastróficas contra os usuários ou meio ambiente em um intervalo de tempo. Na convencional, é a probabilidade de que não ocorra acesso ou mudança indevida no estado do sistema.
- **Manutenabilidade:** avalia o quanto um sistema pode ser restaurado, retornando ao estado correto, supondo que partiu do estado com defeito. É a capacidade de passar por reparos e modificações

Para os atributos da dependabilidade usam-se medidas probabilísticas para ponderá-los relativamente, visto que sua natureza é não determinística. Dependendo da aplicação os pesos considerados para cada atributo devem ser diferenciados [ALBERTINI, 2008].

Além disso, para o cálculo quantitativo 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*). 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 [Guimarães, 2013]. A métrica mais utilizada para o cálculo da confiabilidade é o *MTTF* e as métricas mais utilizadas para o cálculo da disponibilidade são *MTTF* e *MTTR*. A disponibilidade é calculada utilizando a Equação 1 abaixo [Sahner et al., 2012]:

$$A = \frac{MTTF}{MTTF + MTTR} \quad (1)$$

2.4 Mecanismos de Redundância

Redundância (*Redundancy*) é a propriedade de um dispositivo ou sistema ter mais do que um meio de executar sua função, sendo essa uma das técnicas utilizadas para aumentar o grau de tolerância a falhas [Weber, 2002]. Esta propriedade permite tolerar uma falha de um ou mais componentes, sem comprometer o funcionamento do sistema. A redundância do sistema não implica necessariamente haver componentes extras dentro do sistema (Redundância Física), mas implica que o mesmo fornece mais de uma forma de derivar e processar a informação desejada (Redundância Informacional) [PORTO, 2018].

Um sistema que não tem redundância é chamado de simplex. Quando se analisa essa configuração mais simples, que é o modo sem redundância (Simplex), a análise se limita em avaliar o funcionamento de cada componente, pois, se o sistema opera sem falhas, todos os componentes funcionam. Pode-se mensurar individualmente o grau de falha de cada componente e o impacto de cada componente no funcionamento do sistema [PORTO, 2018].

Ainda segundo [PORTO, 2018], componentes redundantes podem ser operados em duas maneiras diferentes no sistema: a redundância ativa (*Active redundancy*) e a redundância em espera (*standby redundancy*).

Redundância ativa se refere a uma configuração do sistema em que todos os componentes estão em funcionamento durante todo o período de operação. Neste caso, os conjuntos de componentes redundantes podem ser utilizados para verificar a consistência no seu funcionamento.

A redundância em espera é a que os componentes redundantes ficam inoperantes até ser necessário e somente após a necessidade (devido à falha) são colocados em serviço a partir de um processo de detecção e chaveamento.

Redundância em espera (*standby*) é bastante utilizada em projetos de sistemas computacionais [Kuo et al., 2003]. Neste tipo de redundância, apenas um único componente está ativo e operacional. Um ou mais componentes adicionais podem ser colocados no sistema, mas em condição de *espera*. Quando este componente principal falhar, um dos componentes em *espera* é levado ao estado ativo e operacional através do mecanismo de comutação [Kuo et al., 2003].

Abaixo, iremos detalhar os principais tipos de redundância em *espera*: *hot-standby*, *cold-standby* e *warm-standby*.

Segundo [Kuo et al., 2003], componentes em *hot-standby* são também denominados de componentes redundantes ativos. Um componente em *hot-standby* tem a mesma taxa de falha, λ , que o componente ativo. Por sua vez, em *cold-standby*, o componente em *espera* não poderá falhar enquanto estiverem nesta condição [Kuo et al., 2003]. O componente ativo possui uma taxa de falha constante igual a λ e o componente em espera possui uma taxa de falha igual a zero. Por fim, o mecanismo de *warm-standby* [Kuo et al., 2003] implica que o componente inativo tem uma taxa de falha que está entre zero e a taxa de falha do componente ativo, λ . Deste modo, o componente em espera poderá falhar mesmo enquanto estiver nesta condição.

Neste trabalho será utilizado o mecanismo de redundância em espera do tipo *hot-standby* e serão propostos modelos que utilizam este mecanismo de redundância.

2.5 Modelagem de sistema

Modelagem de sistemas nada mais é do que a atividade de construção de modelos que expliquem/ilustrem a forma de funcionamento de um sistema. Geralmente, esses modelos são criados graficamente, utilizando algum tipo de mecanismo de modelagem. Nesta seção iremos introduzir os métodos de modelagem utilizados para representação de aspectos da infraestrutura de redes de computadores através de modelos combinatoriais, tais como RBD [Sahner et al., 2012].

Dependendo da complexidade e tamanho e interação entre os componentes, os sistemas podem ser representados tanto por modelos baseados em espaço de estados ou por modelos combinatoriais [Guimarães, 2013]. RBD é levado em consideração quando não ocorrerem dependências dinâmicas entre os componentes do sistema [Sahner et al., 2012]. A abordagem adotada leva em consideração as vantagens e desvantagens do mecanismo de modelagem RBD para avaliar a dependabilidade de sistemas computacionais.

É importante enfatizar que os diferentes sistemas não adotam os mesmos mecanismos de modelagem, podendo possuir diferentes características. Na literatura, é observado que a distribuição exponencial é comumente utilizada para a avaliação da confiabilidade e da disponibilidade [Sahner, 2012]. Com relação aos modelos RBD, é assumido que o tempo para falha e o tempo para reparo dos componentes seguem também uma distribuição exponencial [Sahner et al., 2012].

2.5.1 – 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 [Guimarães, 2013]

2.5.1.1 – Reliability Block Diagram

RBD [Sahner et al., 2012] é considerado um modelo do tipo combinatorial, não baseado em espaço de estados. Ele representa a estrutura lógica de um sistema com relação à forma de como a confiabilidade de seus componentes afeta a confiabilidade do sistema. Ainda segundo [Sahner et al., 2012], 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 manutenibilidade.

Os componentes são representados por blocos e seus nomes podem ser fornecidos no próprio bloco. As estruturas comumente encontradas em topologias de redes convergentes são as do tipo série, paralelo e série/paralelo [Oggerino, 2001]. Figura 3 descreve três exemplos, aonde os blocos (C₁, C₂ e C₃) são estruturados em série (ver Figura 2(a)), em paralelo (ver Figura 2(b)) e em série-paralelo (ver Figura 2(c)).

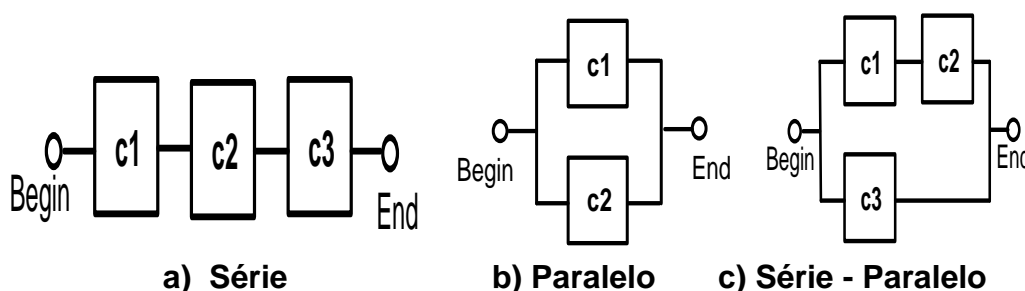


Figura 2. Estruturas Básicas

Todos estes construtores podem ser utilizados juntos em um único diagrama de blocos. Eles mapeiam a dependência operacional de um sistema em relação a seus componentes [Sahner et al., 2012]. Na estrutura em série, se um componente falhar, o sistema inteiro não estará mais operacional. Assumindo uma estrutura com

n componentes em série, a confiabilidade (disponibilidade) [Kuo et al., 2003] é obtida por:

$$P_{(s)}(t) = \prod_{i=1}^n P_i(t) \quad (1)$$

aonde $P_i(t)$ é a confiabilidade ou a disponibilidade do bloco i . Levando-se em consideração n componentes em uma estrutura em paralelo, a confiabilidade (disponibilidade) do sistema é [Kuo et al., 2003]:

$$P_{(p)}(t) = \prod_{i=1}^n (1 - P_i(t)) \quad (2)$$

aonde $P_i(t)$ é a confiabilidade ou a disponibilidade do bloco i . Com o objetivo de calcular a confiabilidade (disponibilidade) de uma estrutura série-paralelo, os resultados das equações relacionadas às estruturas em série devem ser combinados e colocados nas equações relacionados às estruturas em paralelo. Para outros exemplos e equações relacionadas, o leitor deve consultar [Kuo et al., 2003]. Configurações em série e em paralelo são casos especiais de configuração tipo k -of- n . Uma estrutura em série é do tipo n -of- n e uma estrutura em paralelo é do tipo 1 -of- n .

2.6 – Análise de Sensibilidade – Diferença Percentual

Para [Saltelli et al., 2004 apud MELO, 2017], o termo análise de sensibilidade é interpretado em diferentes comunidades técnicas e aplicado em diferentes configurações de problemas. Assim, até pouco tempo, a análise de sensibilidade foi concebida e muitas vezes definida como uma medida do efeito local de um determinado dado de entrada em relação ao dado de saída [Saltelli et al., 2004 apud MELO, 2017]. O autor complementa que a análise de sensibilidade tem por objetivo mapear os elos fracos dos sistemas computacionais e, a partir desse ponto, busca aplicar um conjunto de estratégias que visam melhorias para esses sistemas em diferentes contextos.

Um método para a determinação do parâmetro de análise de sensibilidade é para calcular a Diferença Percentual de saída quando varia um parâmetro de entrada a partir do seu valor mínimo para o seu valor máximo [HOFFMAN et al., 1983 apud MELO, 2017]. O índice de sensibilidade é calculado usando a Equação 3. Nesta equação $\max\{Y(\theta)\}$ e $\min\{Y(\theta)\}$ são os valores máximo e mínimo para a métrica considerada, calculados ao variar o parâmetro θ ao longo de uma gama de n possíveis valores de interesse [Matos et al., 2015].

$$S_{\theta}\{Y\} = \frac{\max\{Y(\theta)\} - \min\{Y(\theta)\}}{\max\{Y(\theta)\}} \quad (3)$$

onde,

$$\max\{Y(\theta)\} = \max\{Y(\theta_1), Y(\theta_2), \dots, Y(\theta_n)\}$$

e

$$\min\{Y(\theta)\} = \min\{Y(\theta_1), Y(\theta_2), \dots, Y(\theta_n)\}$$

A análise de sensibilidade utilizada neste trabalho utiliza o método de diferença percentual e será utilizada a ferramenta Mercury [Silva et al., 2015].

CAPÍTULO 3: TRABALHOS RELACIONADOS

Para compor a revisão de literatura do presente trabalho, foi realizada uma busca em trabalhos relacionados ao tema.

No trabalho realizado por [Medeiros, 2018] têm como objetivo a aplicação de técnicas de tolerância a falhas em sistemas embarcados baseados em microcontroladores com restrições de área, energia e custo, apresentando impacto da aplicação destas técnicas.

No trabalho realizado por [Carvalho e Guimarães, 2013], foi abordado o estudo de aspectos relativos à dependabilidade de redes corporativas, com vistas a que se garanta um melhor tempo de atividade e se alcance os objetivos organizacionais, nos quais foram utilizados os mecanismos de modelagem diagrama de blocos de confiabilidade [Sahner, 2012] (RBD) e árvore de falhas [Sahner, 2012] (FT) para avaliação analítica de cenários complexos.

[WEBER, 2002] abordou em seu trabalho assuntos acerca da confiabilidade e a disponibilidade de equipamentos e serviços de computação destacando que não são conceitos abstratos e absolutos, mas são atributos de um sistema que podem ser medidos quantitativamente. O autor afirma que várias técnicas de projeto podem ser usadas para aumentar o valor dessas medidas, que podem chegar próximas a cem por cento. Mesmo assim, sistemas totalmente infalíveis são impossíveis, pois falhas são inevitáveis.

Com relação a sistemas de *data centers*, o trabalho proposto em [Callou et al., 2013] avalia a métricas de dependabilidade, sustentabilidade e custos, associadas às infraestruturas de potência e de refrigeração, no contexto de projeto para estes sistemas. Com relação à dependabilidade, este estudo utiliza abordagem hierárquica considerando os mecanismos de modelagem *SPN* e *RBD*.

No trabalho mostrado em [Porto, 2018] destaca a importância e relevância em relação à temática de falhas para garantir a dependabilidade em sistemas. Mostra que muitos trabalhos vêm sendo estudados, mas ainda há muito por fazer. A conclusão que se obtém com a realização deste trabalho é que conceber com sucesso um sistema dependável é uma questão desafiadora, assunto de investigação em andamento na literatura.

Diferentemente dos trabalhos apresentados, o presente trabalho possui por objetivo proporcionar modelos analíticos, utilizando o mecanismo de modelagem

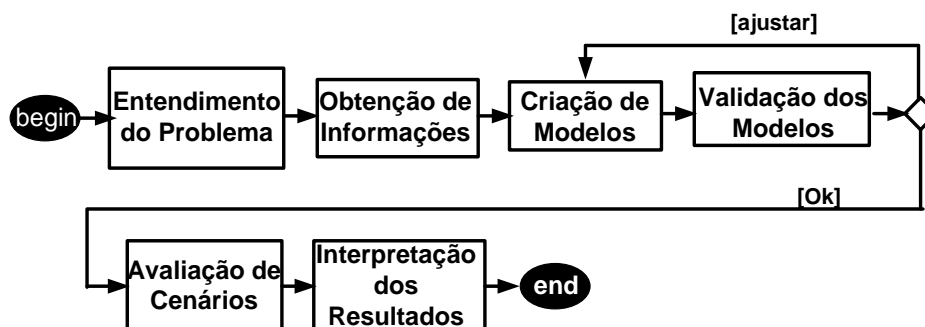
diagrama de blocos de confiabilidade, para analisar o nível de disponibilidade de redes de computadores considerando a implantação de redundância em componentes críticos para o funcionamento destas redes. Além disso, é utilizado o índice de análise de sensibilidade para determinar quais parâmetros, relacionados aos seus componentes, possuem maior impacto sobre sua disponibilidade a fim de direcionar as estratégias para o incremento desta métrica.

É preciso destacar aqui que outros trabalhos que compuseram o presente trabalho, mas que não foram destacados aqui, serviram de base para que o estudo de caso fosse possível.

CAPÍTULO 4: METOLOGIA

Nosso trabalho tem por objetivo o de realizar um estudo de dependabilidade de sistemas de redes de computadores. Nossa abordagem (ver Figura 3) consiste de seis atividades: entendimento do problema; obtenção de informações; criação de modelos; validação dos modelos; avaliação de cenários; interpretação dos resultados.

Figura 3: Metodologia Utilizada



A primeira atividade consiste no entendimento do problema assim como na identificação dos componentes. Para executar esta atividade, um conjunto de ações devem ser executadas. Dentre as ações podemos destacar: especificação da estrutura do sistema (componentes e interfaces); definição da estrutura e do ambiente (condições de contorno); definição do conjunto de métricas relacionadas à dependabilidade utilizadas para a análise do sistema.

A segunda atividade, Obtenção de Informações, trata a respeito da obtenção de informações para os modelos de dependabilidade. Estas informações são oriundas tanto dos fabricantes dos componentes a serem considerados no processo de modelagem, a exemplo do MTTF (sigla proveniente do inglês *Mean Time To Failure*), como das empresas responsáveis pelas políticas de reparo, tal como o MTTR (sigla proveniente do inglês *Mean Time To Repair*). Particularmente, com relação a componentes como enlaces, o MTTF pode ser fornecido pela empresa detentora deste recurso, através de uma base de informações.

A terceira atividade refere-se à criação de modelos para o sistema analisado. Com relação aos modelos de dependabilidade, as características do sistema, mecanismos de redundância, nível de interação entre os componentes, as políticas de reparo tratadas irão determinar qual mecanismo de modelagem utilizar. Esta atividade é executada pela composição de cada componente do sistema de acordo com regras específicas de cada um dos mecanismos de modelagem e pelo mapeamento das métricas desejadas através de expressões representadas pelos

elementos de cada modelo. A vantagem da utilização de modelos não baseados em espaço de estados é que eles são eficientes para a especificação e resolução de problemas. Contudo, a solução destes modelos assume a independência dos componentes. Por exemplo, em RBD os componentes devem ser completamente independentes uns dos outros em termos de comportamentos de falha e de reparo. A falha em um dos componentes não poderá afetar a operação de outro componente, e os componentes não podem compartilhar recursos de reparo. Modelos baseados em espaço de estados, tais como SPN, por sua vez, proporcionam a habilidade de modelar sistemas que violam a suposição feita pelos modelos não baseados em espaço de estados, RBD. 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.

Após a atividade de Validação dos modelos, a avaliação de diferentes cenários poderá ser executada. Finalmente, os resultados obtidos irão ser interpretados e explicados através da adoção de um vocabulário apropriado para o cliente.

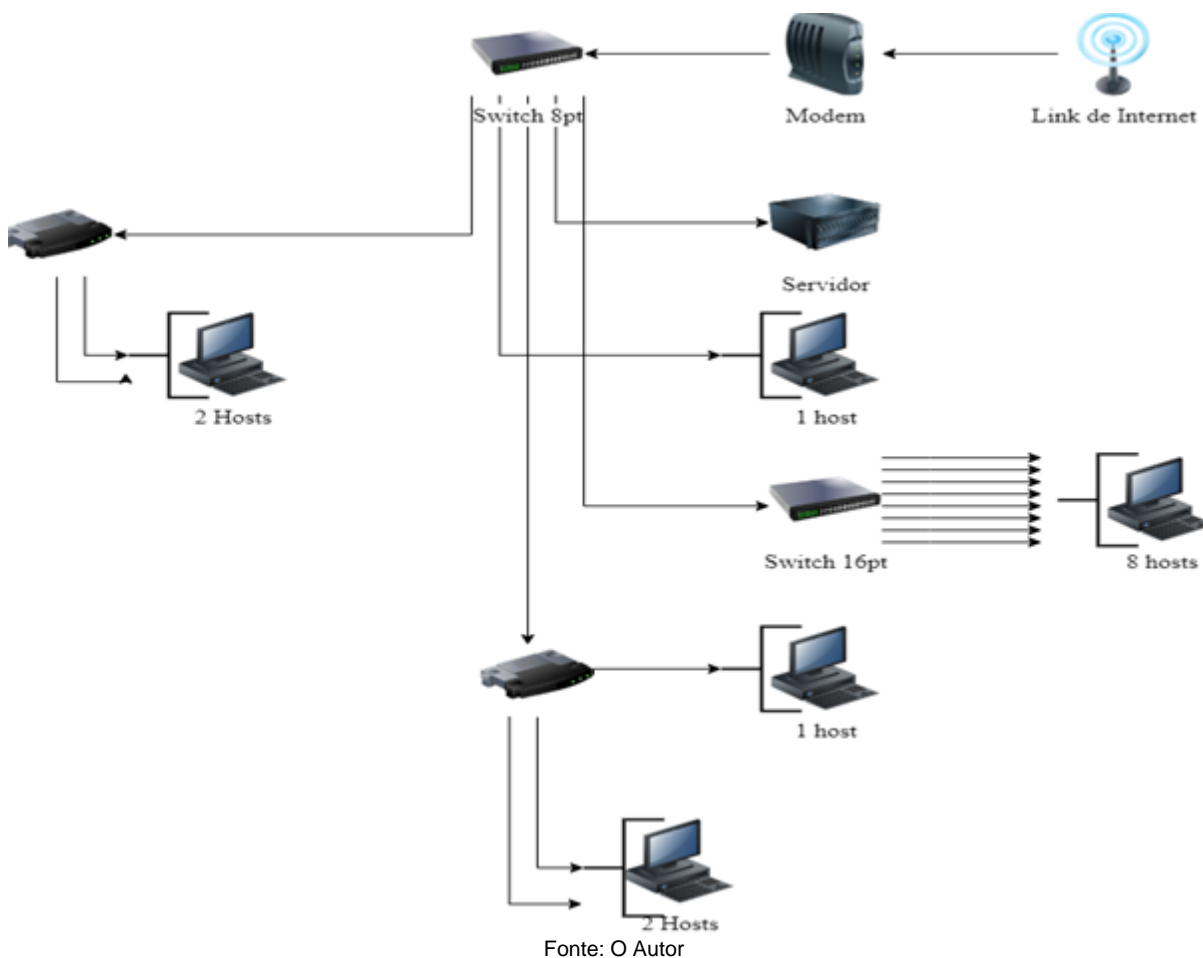
CAPÍTULO 5: ESTUDO DE CASO

O objetivo de nosso estudo de caso é o de analisar aspectos de dependabilidade de uma rede corporativa a fim de planejar ações com o objetivo incrementar a disponibilidade deste sistema.

A partir de uma arquitetura base, a qual será denominada de arquitetura A1, iremos propor mais duas arquitetura com a inserção de componentes redundantes em cada uma destas arquiteturas. A ideia é incrementar o nível de redundância em cada arquitetura, então analisar o índice de análise de sensibilidade em cada uma destas arquitetura e construir gráficos mostrando a variação da disponibilidade do sistema com os componentes que possuem os maiores valores deste índice.

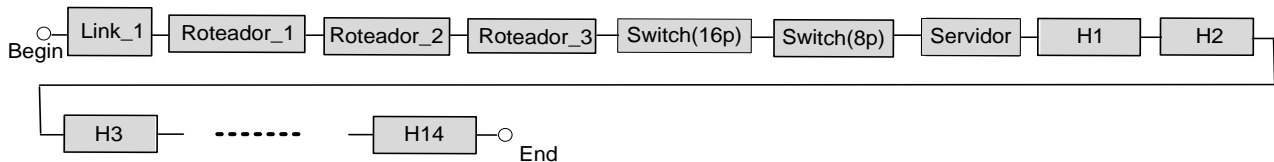
Inicialmente, Figura 4 mostra a arquitetura A1. A rede possui uma topologia em estrela aonde o componente principal é um switch de 8 portas (8p). Interligado a este switch temos três roteadores, um servidor, um switch de 16 portas (16p) e um servidor.

Figura 4 - Topologia da Rede Corporativa



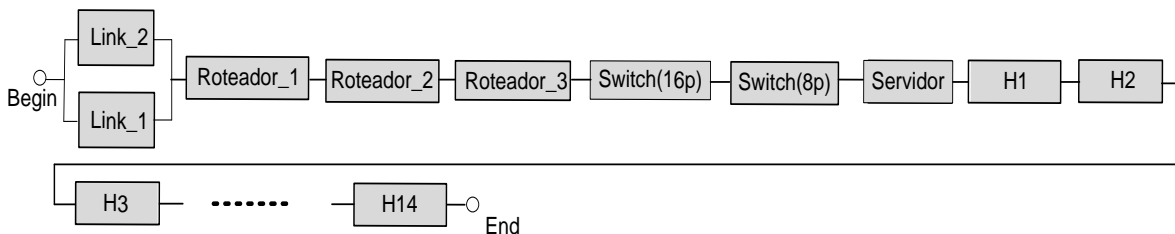
Para representar os aspectos de dependabilidade desta arquitetura foi criado o modelo de disponibilidade em RBD mostrado na Figura 5. Este modelo adota uma estrutura em série, sem nenhum grau de redundância, pois consideramos que caso ocorra uma falha em um dos componentes, toda a rede estará em falha.

Figura 5- Modelo de Dependabilidade RBD – Arquitetura A1



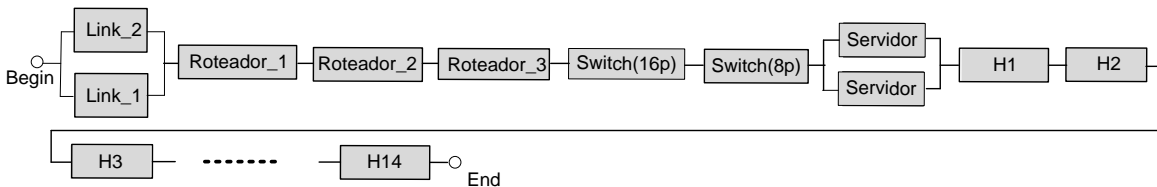
Fonte: O Autor

Figura 6 - Modelo de Dependabilidade RBD – Arquitetura A2



Fonte: O Autor

Figura 7 - Modelo de Dependabilidade RBD – Arquitetura A3



Fonte: O Autor

Por sua vez, Tabela 1 mostra os parâmetros MTTF e MTTR associados aos componentes desta rede que foram utilizados nos modelos RBD em cada uma das arquiteturas construídas.

Tabela 1 – Parâmetro de Entrada – Modelos RBD

Componente	MTTF (h)	MTTR (h)	Componente	MTTF (h)	MTTR (h)
Link_1	796,00	1,00	Servidor	1.417,29	0,99
Roteador _{1,2,3}	16.243,00	0,78	Host	2.654,96	0,35
Switch _{16p,8p}	13.043,48	0,35			

Fonte: O Autor

Além da Arquitetura A1, foram propostas as Arquiteturas A2 e A3 utilizando o mecanismo de redundância *hot-standby*. A Arquitetura A2 foi obtida através da inserção de um enlace redundante (Link_2) ao enlace principal (Link_1) de maneira que no caso de falha de um dos enlaces, a rede continuará funcionando. Por sua vez, a Arquitetura A3 foi obtida através da inserção tanto de um enlace redundante (Link_2) ao enlace principal (Link_1) quanto de um servidor redundante (Servidor_2) ao servidor principal (Servidor_1) de maneira que esta arquitetura proporciona redundância em dois dos principais componentes da rede (enlace e servidor). Os modelos RBD de dependabilidade das Arquiteturas A2 e A3 são mostrados nas Figuras 6 e 7 respectivamente. Estes modelos adotam uma estrutura em série-paralelo. Os componentes em paralelo, Link_1/Link_2 para a Arquitetura A2 junto com Link_1/Link_2 e Servidor_1/Servidor_2 para a Arquitetura A3 possuem um paralelismo que possibilita a continuidade de funcionamento da rede em cada uma das arquiteturas mesmo na ocorrência de falha em um dos componentes em paralelo. Os demais componentes das duas arquiteturas não possuem redundância e podem ocasionar falha da rede em caso de falha.

A partir dos modelos de dependabilidade construídos para as Arquiteturas A1, A2 e A3, serão calculados os índices de análise de sensibilidade em cada arquitetura. Tabelas 2, 3 e 4 mostram os valores calculados para este índice dos principais componentes arquitetura A1, A2 e A3 respectivamente. Para este cálculo, foi aplicado o método de diferença percentual [Hoffman et al., 1983] proporcionado na ferramenta Mercury [Silva et al., 2015]. Apêndice A, Apêndice B e Apêndice C detalham as configurações utilizadas para o cálculo da análise de sensibilidade nas Arquiteturas A1, A2 e A3 respectivamente através da ferramenta Mercury. Este método calcula a diferença percentual das saídas quando um parâmetro de entrada é variado a partir do seu valor mínimo para o seu valor máximo [Hoffman et al., 1983]. O índice de sensibilidade é calculado usando a Equação 1. Esta equação mostra a expressão para essa abordagem, onde $\max\{Y(\Theta)\}$ e $\min\{Y(\Theta)\}$ são os valores máximo e mínimo de saída calculados ao variar o parâmetro Θ ao longo de uma gama de n possíveis valores de interesse.

$$S_{\Theta} = \frac{\max\{(\Theta)\} - \min\{(\Theta)\}}{\max\{(\Theta)\}} \quad (1)$$

Vemos que na Arquitetura A1, os parâmetros MTTR_Link_1 e MTTF_Switch16p, 8p apresentam o maior e menor valor de análise de sensibilidade. Por questões de comparação, iremos construir um gráfico mostrando a variação da disponibilidade do sistema em função do parâmetro MTTR_Link_1 e em função do parâmetro MTTR_Roteador_1. O parâmetro MTTR_Roteador_1 foi utilizado por ter uma ordem de grandeza do valor de análise de sensibilidade próximo ao parâmetro MTTF_Switch16p,8p e a variação dos valores para este parâmetro (MTTR_Roteador_1) é próxima em relação à variação dos valores do parâmetro MTTR_Link_1.

Por sua vez, na Arquitetura A2, os parâmetros MTTR_Servidor, MTTF_Link_1 e MTTF_Link_2 apresentam o maior e os menores valores de análise de sensibilidade. Iremos construir um gráfico representado a variação da disponibilidade do sistema, de forma comparativa, em função dos parâmetros MTTR_Servidor e em função do parâmetro MTTF_Link_1. O parâmetro MTTR_Link_1, de forma semelhante à Arquitetura A1, foi utilizado por ter uma ordem de grandeza do valor de análise de sensibilidade próximo ao parâmetro MTTF_Link_1 e a variação dos valores para este parâmetro (MTTR_Link_1) é próxima em relação à variação dos valores do parâmetro MTTR_Servidor.

Por fim, na Arquitetura A3 os parâmetros MTTR_Host1..14, MTTF_Servidor1 e MTTF_Servidor2 apresentam o maior e os menores valores de análise de sensibilidade. O parâmetro MTTR_Host1..14 representa o parâmetros MTTR de cada um dos hosts da rede. Neste caso, iremos trabalhar com o parâmetro MTTR_Host_1. Será construído um gráfico representado a variação da disponibilidade do sistema, de forma comparativa, em função dos parâmetros MTTR_Host_1 e função do parâmetro MTTR_Servidor_1. O parâmetro MTTR_Servidor_1, de forma semelhante às Arquiteturas A1 e A2, foi utilizado por ter uma ordem de grandeza do valor de análise de sensibilidade próximo ao parâmetro MTTF_Servidor_1 e a variação dos valores para este parâmetro (MTTR_Servidor_1) é próxima em relação à variação dos valores do parâmetro MTTF_Servidor.

Tabela 2 – Análise de Sensibilidade Arquitetura 1

Componente	Análise_Sensib. (AS)	Componente	Análise_Sensib.(AS)
MTTF_Link_1	0,0008403	MTTR_Link_1	0,00375
MTTF_Roteador 1,2,3	0,0000 5046	MTTR_Roteador 1,2,3	0,00008003
MTTF_Switch _{16p,8p}	0,00003431	MTTR_Switch _{16p,8p}	0,0001149
MTTF_Servidor1	0,0004389	MTTR_Servidor	0.00212
MTTF_Host _{1,2,3...14}	0,00007009	MTTR_Host _{1,2,3...14}	0,0005643

Fonte: O Autor

Tabela 3 – Análise de Sensibilidade Arquitetura 2

Componente	Análise_Sensib. (AS)	Componente	Análise_Sensib.(AS)
MTTF_Link_1	0,000001053	MTTR_Link_1	0,000004699
MTTF_Link_2	0,000001053	MTTR_Link_2	0,000004699
MTTF_Roteador 1,2,3	0,00005046	MTTR_Roteador 1,2,3	0,00008002
MTTF_Switch _{16p,8p}	0,00003431	MTTR_Switch _{16p,8p}	0,0001149
MTTF_Servidor1	0,0004389	MTTR_Servidor	0.002118
MTTF_Host _{1,2,3...14}	0,00007008	MTTR_Host _{1,2,3...14}	0,0005643

Fonte: O Autor

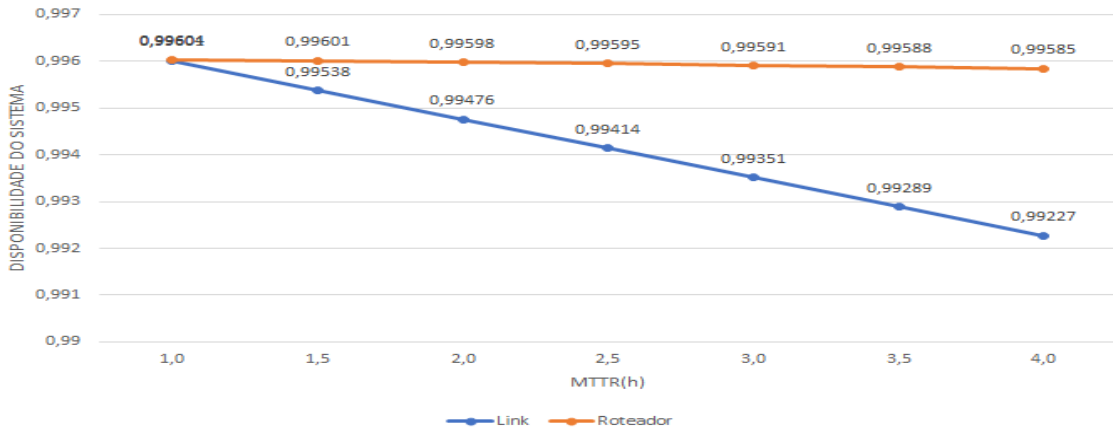
Tabela 4 – Análise de Sensibilidade Arquitetura 3

Componente	Análise_Sensib. (AS)	Componente	Análise_Sensib.(AS)
MTTF_Link_1	0,000001053	MTTR_Link_1	0,000004699
MTTF_Link_2	0,000001053	MTTR_Link_2	0,000004699
MTTF_Roteador 1,2,3	0,00005046	MTTR_Roteador 1,2,3	0,00008002
MTTF_Switch _{16p,8p}	0,00003431	MTTR_Switch _{16p,8p}	0,0001149
MTTF_Servidor1	0,0000003062	MTTR_Servidor_1	0,000001477
MTTF_Servidor2	0,0000003062	MTTR_Servidor_2	0,000001477
MTTF_Host _{1,2,3...14}	0,00007008	MTTR_Host _{1,2,3...14}	0,0005643

Fonte: O Autor

Figura 8 mostra a comparação do impacto que a variação dos parâmetros MTTR_Link_1 e MTTR_Roteador_1 possuem sobre a disponibilidade do sistema. Em virtude de possuir o maior valor de análise de sensibilidade (ver tabela 2) é visível o maior impacto do parâmetro MTTR_Link_1 em relação ao parâmetro MTTR_Roteador_1.

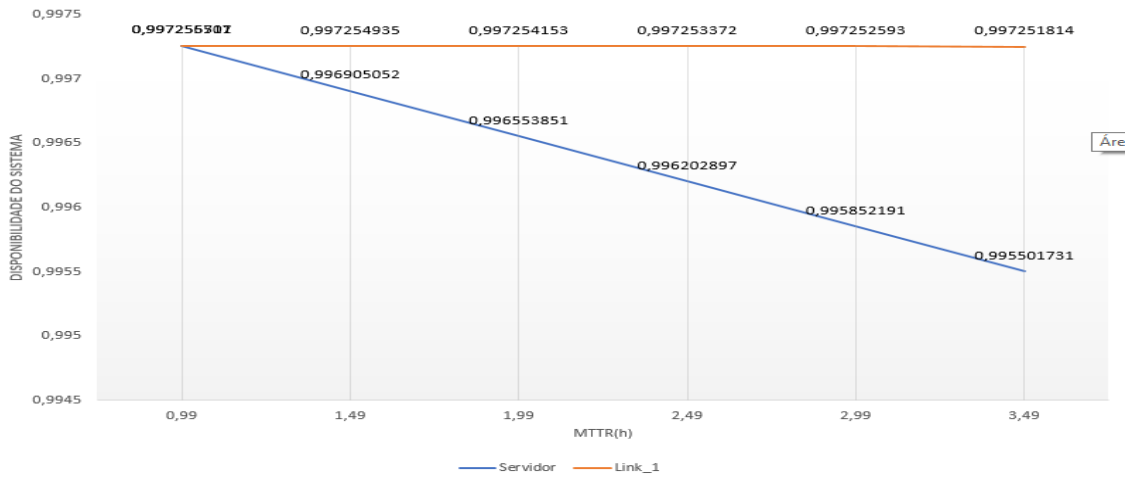
Figura 8: Variação MTTR X Disponibilidade Sistema – Arquitetura 1



Fonte: O Autor

Figura 9 mostra a comparação do impacto que a variação dos parâmetros MTTR_Servidor e MTTR_Link_1 possuem sobre a disponibilidade do sistema. Em virtude de possuir o maior valor de análise de sensibilidade (ver Tabela 3) é visível o maior impacto do parâmetro MTTR_Servidor em relação ao parâmetro MTTR_Link_1.

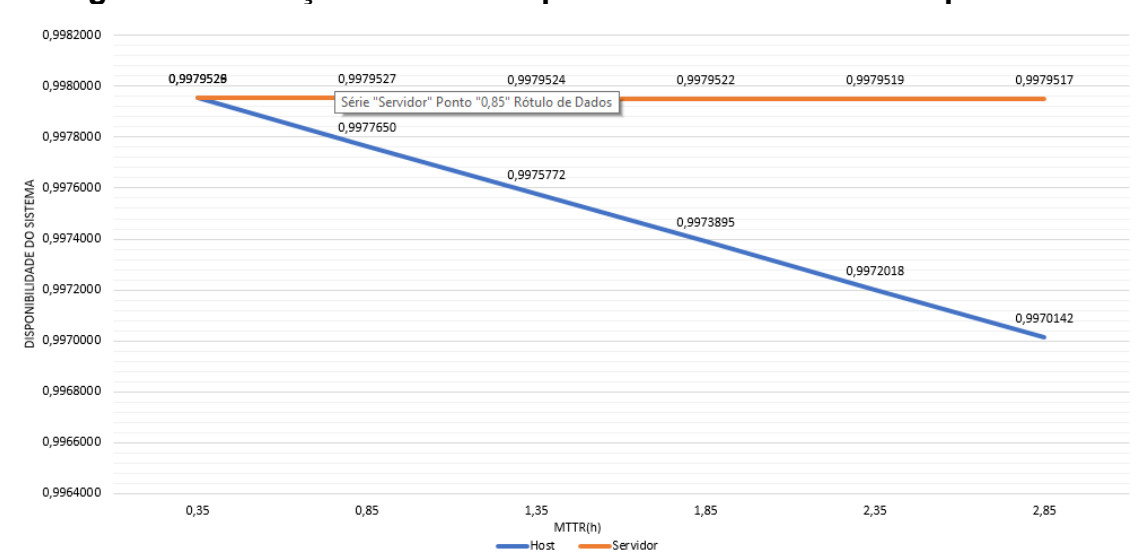
Figura 9: Variação MTTR X Disponibilidade Sistema – Arquitetura 2



Fonte: O Autor

Figura 10 mostra a comparação do impacto que a variação dos parâmetros MTTR_Host1...14 e MTTR_Servidor1,2 possuem sobre a disponibilidade do sistema. Em virtude de possuir o maior valor de análise de sensibilidade (ver Tabela 3) é visível o maior impacto do parâmetro MTTR_Host1 em relação ao parâmetro MTTR_Servidor_1.

Figura 10: Variação MTTR X Disponibilidade Sistema – Arquitetura 3



Fonte: O Autor

Por fim, a modelagem analítica, utilizando o mecanismo de modelagem RBD, junto com o índice de análise de sensibilidade proporcionou meios para uma avaliação dos parâmetros que possuem maior impacto sobre a disponibilidade do sistema em diferentes arquiteturas que utilizam o mecanismo de redundância *hot-standby* considerando diferentes componentes.

CONSIDERAÇÕES FINAIS

Nosso trabalho analisou aspectos de dependabilidade da infraestrutura de uma rede de computadores pertencente a uma pequena empresa através da utilização de modelagem analítica e do mecanismo de análise de sensibilidade. Além da arquitetura base, foram propostas diferentes arquiteturas que utilizam o mecanismo de redundância *hot standby* considerando os componentes críticos desta infraestrutura. Então, através da utilização dos mecanismos de modelagem analítica e de análise de sensibilidade, foram determinados os parâmetros que possuíam o maior e menor impacto sobre a disponibilidade do sistema e elaborados gráficos com esta variação.

Por sua vez, o mecanismo de análise de sensibilidade utilizou o método de diferença percentual para avaliar o impacto que diferentes parâmetros exercem sobre a disponibilidade do sistema. Algumas conclusões podem ser obtidas dos resultados deste estudo.

Primeiro, alguns componentes do sistema que possuem um maior índice de análise de sensibilidade impactam de forma mais significativa sobre a disponibilidade da infraestrutura da rede de computadores e como consequência, sobre a disponibilidade do sistema. Desta forma, estes componentes precisam de maior atenção para a alocação de políticas de manutenção sobre estes sistemas. Os custos decorrentes da adoção destes mecanismos podem ser compensados pela alta disponibilidade obtida por esta rede de computadores. Segundo, combinações de diferentes mecanismos de redundância podem ser propostos para estabelecer diferentes arquiteturas acarretando graus diferenciados de disponibilidade do sistema. Finalmente, para ilustrar a aplicabilidade dos resultados obtidos, um estudo foi realizado destacando os parâmetros que possuem maior e menor impactos sobre a disponibilidade do sistema. Os resultados mostraram que o parâmetro MTTR_Link_1 para a Arquitetura A1, MTTR_Servidor para a Arquitetura A2 e MTTR_Host1....14 para a Arquitetura A3 exercem as maiores influências sobre a disponibilidade da rede de computadores. Para trabalhos futuros, pretendemos estender a avaliação e propor soluções para as infraestruturas de comunicação e potência.

REFERÊNCIAS

- ALBERTINI, Pedro Henrique Buoro. **Estudo de Ferramentas de Software para melhoria de desempenho em ambiente de Cluster Computacional**. 2008. 74 f. Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação)- Universidade Federal de Santa Catarina. Florianópolis, 2008. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/184041/TCC%20-%20PedroH.pdf?sequence=-1&isAllowed=y>>. Acesso em: 20 maio. 2020.
- AVIZIENIS, A. et al. **Basic concepts and taxonomy of dependable and secure computing**. IEEE Transactions on Dependable and Secure Computing, IEEE, v.1, n. 1, p. 11–33, 2004
- CALLOU, G.; MACIEL, P.; TUTSCH, D.; ARAÚJO, C.; SOUZA, R. **Estimating sustainability impact of high dependable data centers: a comparative study between brazilian and us energy mixes**. Springer Computing Journal, 95:1137–1170 DOI:10.1007/s00607–013–0328–y, 2013.
- CARVALHO, Eduardo Rodrigues de; GUIMARÃES, Almir Pereira. **Dependabilidade em rede de computadores: uma análise baseada na importância para confiabilidade**. Revista Brasileira de Administração Científica. Vol.4. n. 2. p. 96-106, 2013. Disponível em: <<http://www.sustenere.co/index.php/rbadm/article/view/ESS2179-684X.2013.002.0007/300>>. Acesso em: 30 jul. 2020
- CAMBOIM, K.A. **Modelagem Hierárquica e Heterogênea para Infraestrutura de Redes Convergentes e Política de Manutenção para Garantia de Níveis de Serviços**. 2012. Dissertação de Mestrado (Pós-graduação em Ciência da Computação)- Centro de Informática da Universidade Federal de Pernambuco. Recife, 2012. Disponível em: <<http://www.modcs.org/>>. Acesso em: 11 Nov. 2020.
- COSTA, Hebert Luiz Amaral. **Alta Disponibilidade de balanceamento de Carga para Melhoria de Sistemas Computacionais Críticos Usando Software Livre: Um estudo de caso**. 2009. 138 f. Dissertação (Pós-Graduação em Ciência da Computação)- Universidade Federal de Viçosa. Minas Gerais, 2009. Disponível em: <http://www.dpi.ufv.br/arquivos/ppgcc/dissertacoes/2009-ms-Hebert_Luiz_Amaral_Costa.pdf>. Acesso em: 13 maio. 2020.
- EBELING, C. **An Introduction to Reliability and Maintainability Engineering**. Waveland Press. 1997.
- FERNANDES, Sérgio Murilo Maciel. **Avaliação de Dependabilidade de Sistemas com Mecanismos Tolerantes a Falha: Desenvolvimento de um Método Híbrido Baseado em EDSPN e Diagrama de Blocos**. 2007. 253 f. Tese de Doutorado (Doutorado em Ciências da Computação)- Universidade Federal de Pernambuco. Recife, 2007. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/2041/1/arquivo6424_1.pdf>. Acesso em: 07 maio. 2020.
- FOROUZAN, B.A.; MOSHARRAF, F. **Redes de Computadores: Uma Abordagem Top-Down**. AMGH Ltda. Porto Alegre, 2013.
- GOMES, Daniel Cardoso. **Proposta de otimização do tráfego de rede da Universidade Federal de Lavras utilizando a técnica de Spanning tree protocol**. 2005. 103f. Monografia

(Bacharel em Ciência da Computação.)- Universidade Federal de Lavras. Lavras, 2005. Disponível em: <http://repositorio.ufla.br/jspui/bitstream/1/8343/1/MONOGRAFIA_Proposta_de_otimiza%C3%A7%C3%A3o_do_tr%C3%A1fego_da_rede_da_universidade_federal_de_lavras_utilizando_a_t%C3%A9cnica_de_spanning_tree_protocol.pdf>. Acesso em: 22 jul. 2020

GUIMARÃES, A.P. **Modelos para Planejamento de Redes Convergentes Considerando a Integração de Aspectos de Infraestrutura e de Negócios**. Tese de Doutorado (Pós-graduação em Ciência da Computação)- Centro de Informática da Universidade Federal de Pernambuco. Recife, Dez. 2013.

HOFFMAN, F.; GARDNER, R. **Evaluation of Uncertainties in Environmental Radiological Assessment Models**. In: TILL, J.; MEYER, H. (Ed.). Radiological Assessments: a textbook on environmental dose assessment. Washington, DC: U.S. Nuclear Regulatory Commission, 1983. Report No. NUREG/CR-3332.

JÚNIOR, Antônio Carlos Venâncio; SILVA, Rodrigo Grumiche. **Computação distribuída de alto desempenho- um estudo de caso de acesso móvel**. 2004. 162 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação)- Universidade Federal de Santa Catarina. Florianópolis, 2004. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/183809/TCC-Antonio-Rodrigo.pdf?sequence=-1>>. Acesso em: 30 maio. 2020.

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

KUROSE, Keith W. Ross; JAMES F. **Redes de Computadores e a Internet: Uma Abordagem Top-down**. 5ª ed. Addison-Wesley, 2010.

LALA, P., "**Self-Checking and Fault Tolerant Digital Design**", Academic Press, 2001.

Matos Jr., R.; Araújo, J.; Oliveira, D.; Maciel, P.; Trivedi, K. "**Sensitivity analysis of a hierarchical model of mobile cloud computing**." Simulation Modelling Practice and Theory. v. 50, 151-164, 2015.

MACEDO, Ricardo Tombesi; FRANCISCATTO, Roberto; CUNHA, Guilherme Bernardino da; BERTOLINI, Cristiano. **Rede de Computadores**. 2018. 196f. Licenciatura em computação- Universidade Federal de Santa Maria. Rio Grande do Sul, 2018. Disponível em: <https://www.ufsm.br/orgaos-suplementares/nte/wp-content/uploads/sites/358/2019/08/MD_RedesdaComputadores.pdf>. Acesso em: 21 jul. 2020

MARÇAL, Joice Araújo; GOMES, Lidiege Terra Souza. Metodologia do Trabalho científico. **RE-Revista Eletrônica Acervo Saúde**, 2013. Vol.5, n.2, p. 474-489. Disponível em: <<https://revistas.ufpr.br/cogitare/article/viewFile/44358/26850>>. Acesso em: 30 maio. 2020.

MEDEIROS, Wanderson Ricardo de. **Tolerância a Falhas em Sistemas Embarcados Baseados em Microcontroladores**. 2018. 62 f. Monografia (Graduação em Ciência da Computação)- Centro de Ciências Exatas e da Terra da Universidade Federal do Rio Grande do Norte. Rio Grande do Norte, 2018. Disponível em: <

https://monografias.ufrn.br/jspui/bitstream/123456789/6746/1/SistemasEmbarcados_Medeiros_2018.pdf>. Acesso em: 03 mar. 2020.

MELO, Rosângela. **Análise de Sensibilidade aplicada à Identificação de Pontos que Requerem Melhoria na Disponibilidade em Infraestrutura de Cloud**. 2017. Tese de Doutorado (Pós-graduação em Ciência da Computação) - Centro de Informática da Universidade Federal de Pernambuco. Recife, 2015. Disponível em: < <http://www.modcs.org>>. Acesso em: 12 abr. 2020.

NASCIMENTO, Ramon Santos. **Avaliação de Dependabilidade e Análise de Sensibilidade de uma plataforma como serviço (PAAS)**. 2015. 98f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)- Universidade Federal Rural de Pernambuco. Garanhuns, 2015. Disponível em: < [https://pt.slideshare.net/RamonSantos28/tcc-avaliacao-de-dependabilidade-e-anlise-de-sensibilidade-de-uma-plataforma-como-servicio-paas](https://pt.slideshare.net/RamonSantos28/tcc-avaliacao-de-dependabilidade-e-analise-de-sensibilidade-de-uma-plataforma-como-servicio-paas)>. Acesso em: 18 abr. 2020.

OGGERINO, C. High Availability Network Fundamentals. Cisco Press. 2001.

PORTO, Roberta de Cassia Ferreira. **Dependabilidade em Sistemas Espaciais e análise de falhas reais em satélites**. 2018. 73 f. Monografia (Qualificação de Doutorado)- Instituto Nacional de Pesquisas Espaciais. São José dos Campos, 2018. Disponível em: <mtc-m21b.sid.inpe.br/col/sid.inpe.br/mtc-m21b/2018/02.08.18.35/doc/publicacao.pdf>. Acesso em: 15 abr. 2020.

RESENDE, Igor Henrique Correia. **Estudo para a Modelagem de um Sistema Moderno por meio da UML e extensões**. 2019. 53f. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação)- Universidade Federal de Uberlândia. Monte Carmelo, 2019. Disponível em: < <https://repositorio.ufu.br/bitstream/123456789/28179/4/EstudoModelagemSistema.pdf>>. Acesso em: 30 jul. 2020

SALTELLI, A. et al. **Sensitivity analysis in practice: a guide to assessing scientific models**. [S.l.]: John Wiley & Sons, 2004.

Sahner R, Trivedi K. S, Puliafito A. **Performance and reliability analysis of computer systems**. Kluwer Academic Publishers Group; 2002; ISBN0-7923- 9650-2.

SILVA, B.; MATOS, R.; CALLOU, G.; FIGUEIREDO, J.; OLIVEIRA, D.; FERREIRA, J.; DANTAS, J.; ALVES, V.; MACIEL, P. *Mercury: An integrated environment for performance and dependability evaluation of general system*. 45th Dependable Systems and Networks Conference (DSN), 2015.

SOARES, Guilherme Teixeira; PINHO, Pedro Henrique Vidal. **Sistema de Arquivos Distribuído Tolerante a Falhas Maliciosas: Uma Prova de Conceito**. 2016. 71f. Monografia (Licenciatura em Computação)-Universidade de Brasília. Brasília, 2016. Disponível em: < https://bdm.unb.br/bitstream/10483/14055/1/2016_GuilhermeTeixeiraSoares_PedroHenriqueVidalPinho.pdf>. Acesso em: 10 mar. 2020

SOARES, Luiz Fernando Gomes; LEMOS, Guido; Colcher, Sérgio. **Redes de Computadores: Das LANs, MANs e WANs às Redes ATM**. Editora Campus Ltda. Rio de Janeiro. 1995

SOUZA, Erica Teixeira Gomes de. **Modelagem de desempenho, dependabilidade e custo para o planejamento de infraestruturas e nuvens privadas**. 2015. 175 f. Tese de Doutorado (Pós-graduação em Ciência da Computação)- Centro de Informática da Universidade Federal de Pernambuco. Recife, 2015. Disponível em: < <http://www.modcs.org/wp-content/uploads/thesis/Tese-Erica.pdf>>. Acesso em: 12 abr. 2020.

TANENBAUM, Andrew S.. **Redes de Computadores**: tradução [ds 3. ed. original] Insight Serviços de Informática. Rio de Janeiro: Campus, 1997.

TORRES, Gabriel. **Redes de Computadores Curso Completo** – 2001, Axcel Books do Brasil Editora Ltda.

WEBER, Taisy Silva. **Um roteiro para exploração dos conceitos básicos de tolerância a falhas**. Especialização. 2002. Disponível em: < <http://www.inf.ufrgs.br/~taisy/disciplinas/textos/Dependabilidade.pdf>>. Acesso em: 05 abr. 2020.

APÊNDICE A

CONFIGURAÇÃO PARA O CÁLCULO DE ANÁLISE DE SENSIBILIDADE ARQUITETURA A1

t = 100;
tx = 1;
mttf_Link_1 = 796*tx;
mtrr_Link_1 = 1.0*tx;
mttf_Roteador_1 = 16243*tx;
mtrr_Roteador_1 = 0.78*tx;
mttf_Roteador_2 = 16243*tx;
mtrr_Roteador_2 = 0.78*tx;
mttf_Roteador_3 = 16243*tx;
mtrr_Roteador_3 = 0.78*tx;
mttf_Switch_16p = 13043*tx;
mtrr_Switch_16p = 0.35*tx;
mttf_Switch_8p = 13043*tx;
mtrr_Switch_8p = 0.35*tx;
mttf_Servidor = 1417.29*tx;
mtrr_Servidor = 0.99*tx;
mttf_H1 = 2654.96*tx;
mtrr_H1 = 0.3508*tx;
mttf_H2 = 2654.96*tx;
mtrr_H2 = 0.3508*tx;
mttf_H3 = 2654.96*tx;
mtrr_H3 = 0.3508*tx;
mttf_H4 = 2654.96*tx;
mtrr_H4 = 0.3508*tx;
mttf_H5 = 2654.96*tx;
mtrr_H5 = 0.3508*tx;
mttf_H6 = 2654.96*tx;
mtrr_H6 = 0.3508*tx;
mttf_H7 = 2654.96*tx;
mtrr_H7 = 0.3508*tx;
mttf_H8 = 2654.96*tx;
mtrr_H8 = 0.3508*tx;
mttf_H9 = 2654.96*tx;
mtrr_H9 = 0.3508*tx;
mttf_H10 = 2654.96*tx;
mtrr_H10 = 0.3508*tx;
mttf_H11 = 2654.96*tx;
mtrr_H11 = 0.3508*tx;
mttf_H12 = 2654.96*tx;
mtrr_H12 = 0.3508*tx;
mttf_H13 = 2654.96*tx;
mtrr_H13 = 0.3508*tx;
mttf_H14 = 2654.96*tx;
mtrr_H14 = 0.3508*tx;

RBD Model{

```

block Link_1( MTTF = mttf_Link_1, MTTR = mtrr_Link_1);
block Roteador_1( MTTF = mttf_Roteador_1, MTTR = mtrr_Roteador_1);
block Roteador_2( MTTF = mttf_Roteador_2, MTTR = mtrr_Roteador_2);
block Roteador_3( MTTF = mttf_Roteador_3, MTTR = mtrr_Roteador_3);
block Switch_16p( MTTF = mttf_Switch_16p, MTTR = mtrr_Switch_16p);
block Switch_8p( MTTF = mttf_Switch_8p, MTTR = mtrr_Switch_8p);
block Servidor( MTTF = mttf_Servidor, MTTR = mtrr_Servidor);
block H1( MTTF = mttf_H1, MTTR = mtrr_H1);
block H2( MTTF = mttf_H2, MTTR = mtrr_H2);
block H3( MTTF = mttf_H3, MTTR = mtrr_H3);
block H4( MTTF = mttf_H4, MTTR = mtrr_H4);
block H5( MTTF = mttf_H5, MTTR = mtrr_H5);
block H6( MTTF = mttf_H6, MTTR = mtrr_H6);
block H7( MTTF = mttf_H7, MTTR = mtrr_H7);
block H8( MTTF = mttf_H8, MTTR = mtrr_H8);
block H9( MTTF = mttf_H9, MTTR = mtrr_H9);
block H10( MTTF = mttf_H10, MTTR = mtrr_H10);
block H11( MTTF = mttf_H11, MTTR = mtrr_H11);
block H12( MTTF = mttf_H12, MTTR = mtrr_H12);
block H13( MTTF = mttf_H13, MTTR = mtrr_H13);
block H14( MTTF = mttf_H14, MTTR = mtrr_H14);
series s0(Link_1, Roteador_1, Roteador_2, Roteador_3, Switch_16p, Switch_8p, Servidor, H1,
H2, H3, H4, H5, H6, H7, H8, H9, H10, H11, H12, H13, H14 );

top s0;

metric av = availability;
metric rel = reliability( time = t );
metric mttf = mttf;
metric mtrr = mtrr;
}

main{

    av = solve(Model, av);
    rel = solve(Model, rel);
    mttf = solve(Model, mttf);
    mtrr = solve(Model, mtrr);

    println("Availability: " .. av );
    println("Reliability: " .. rel );
    println("Mean time to failure: " .. mttf );
    println("Mean time to repair: " .. mtrr );

    percentageDifference(
        model_ = "Model",
        metric_ = "av",
        samplingPoints = 7,
        parameters = (

```

```

mttf_Link_1 = [596, 1196],
mttr_Link_1 = [1, 4],
mttf_Roteador_1 = [11243, 41243],
mttr_Roteador_1 = [0.58, 1.88],
mttf_Roteador_2 = [11243, 41243],
mttr_Roteador_2 = [0.58, 1.88],
mttf_Roteador_3 = [11243, 41243],
mttr_Roteador_3 = [0.58, 1.88],
mttf_Switch_16p = [8043, 38043],
mttr_Switch_16p = [0.35, 1.85],
mttf_Switch_8p = [8043, 38043],
mttr_Switch_8p = [0.35, 1.85],
mttf_Servidor = [1417.29, 3817.29],
mttr_Servidor = [0.99, 4.00],
mttf_H1 = [2654.96, 5654.96],
mttr_H1 = [0.3508, 1.85],
mttf_H2 = [2654.96, 5654.96],
mttr_H2 = [0.3508, 1.85],
mttf_H3 = [2654.96, 5654.96],
mttr_H3 = [0.3508, 1.85],
mttf_H4 = [2654.96, 5654.96],
mttr_H4 = [0.3508, 1.85],
mttf_H5 = [2654.96, 5654.96],
mttr_H5 = [0.3508, 1.85],
mttf_H6 = [2654.96, 5654.96],
mttr_H6 = [0.3508, 1.85],
mttf_H7 = [2654.96, 5654.96],
mttr_H7 = [0.3508, 1.85],
mttf_H8 = [2654.96, 5654.96],
mttr_H8 = [0.3508, 1.85],
mttf_H9 = [2654.96, 5654.96],
mttr_H9 = [0.3508, 1.85],
mttf_H10 = [2654.96, 5654.96],
mttr_H10 = [0.3508, 1.85],
mttf_H11 = [2654.96, 5654.96],
mttr_H11 = [0.3508, 1.85],
mttf_H12 = [2654.96, 5654.96],
mttr_H12 = [0.3508, 1.85],
mttf_H13 = [2654.96, 5654.96],
mttr_H13 = [0.3508, 1.85],
mttf_H14 = [2654.96, 5654.96],
mttr_H14 = [0.3508, 1.85]
),
output = (
type = "swing",
yLabel = "Steady-state availability",
baselineValue = av
)
);
}

```

APÊNDICE B

CONFIGURAÇÃO PARA O CÁLCULO DE ANÁLISE DE SENSIBILIDADE ARQUITETURA A2

t = 100;
tx = 1;
mttf_Link_1 = 796*tx;
mtrr_Link_1 = 1.0*tx;
mttf_Link_2 = 796*tx;
mtrr_Link_2 = 1.0*tx;
mttf_Roteador_1 = 16243*tx;
mtrr_Roteador_1 = 0.78*tx;
mttf_Roteador_2 = 16243*tx;
mtrr_Roteador_2 = 0.78*tx;
mttf_Roteador_3 = 16243*tx;
mtrr_Roteador_3 = 0.78*tx;
mttf_Switch_16p = 13043*tx;
mtrr_Switch_16p = 0.35*tx;
mttf_Switch_8p = 13043*tx;
mtrr_Switch_8p = 0.35*tx;
mttf_Servidor = 1417.29*tx;
mtrr_Servidor = 0.99*tx;
mttf_H1 = 2654.96*tx;
mtrr_H1 = 0.3508*tx;
mttf_H2 = 2654.96*tx;
mtrr_H2 = 0.3508*tx;
mttf_H3 = 2654.96*tx;
mtrr_H3 = 0.3508*tx;
mttf_H4 = 2654.96*tx;
mtrr_H4 = 0.3508*tx;
mttf_H5 = 2654.96*tx;
mtrr_H5 = 0.3508*tx;
mttf_H6 = 2654.96*tx;
mtrr_H6 = 0.3508*tx;
mttf_H7 = 2654.96*tx;
mtrr_H7 = 0.3508*tx;
mttf_H8 = 2654.96*tx;
mtrr_H8 = 0.3508*tx;
mttf_H9 = 2654.96*tx;
mtrr_H9 = 0.3508*tx;
mttf_H10 = 2654.96*tx;
mtrr_H10 = 0.3508*tx;
mttf_H11 = 2654.96*tx;
mtrr_H11 = 0.3508*tx;
mttf_H12 = 2654.96*tx;
mtrr_H12 = 0.3508*tx;
mttf_H13 = 2654.96*tx;
mtrr_H13 = 0.3508*tx;
mttf_H14 = 2654.96*tx;
mtrr_H14 = 0.3508*tx;

```
RBD Model{
```

```
    block Link_1( MTTF = mttf_Link_1, MTTR = mtr_Link_1);
    block Link_2( MTTF = mttf_Link_2, MTTR = mtr_Link_2);
    block Roteador_1( MTTF = mttf_Roteador_1, MTTR = mtr_Roteador_1);
    block Roteador_2( MTTF = mttf_Roteador_2, MTTR = mtr_Roteador_2);
    block Roteador_3( MTTF = mttf_Roteador_3, MTTR = mtr_Roteador_3);
    block Switch_16p( MTTF = mttf_Switch_16p, MTTR = mtr_Switch_16p);
    block Switch_8p( MTTF = mttf_Switch_8p, MTTR = mtr_Switch_8p);
    block Servidor( MTTF = mttf_Servidor, MTTR = mtr_Servidor);
    block H1( MTTF = mttf_H1, MTTR = mtr_H1);
    block H2( MTTF = mttf_H2, MTTR = mtr_H2);
    block H3( MTTF = mttf_H3, MTTR = mtr_H3);
    block H4( MTTF = mttf_H4, MTTR = mtr_H4);
    block H5( MTTF = mttf_H5, MTTR = mtr_H5);
    block H6( MTTF = mttf_H6, MTTR = mtr_H6);
    block H7( MTTF = mttf_H7, MTTR = mtr_H7);
    block H8( MTTF = mttf_H8, MTTR = mtr_H8);
    block H9( MTTF = mttf_H9, MTTR = mtr_H9);
    block H10( MTTF = mttf_H10, MTTR = mtr_H10);
    block H11( MTTF = mttf_H11, MTTR = mtr_H11);
    block H12( MTTF = mttf_H12, MTTR = mtr_H12);
    block H13( MTTF = mttf_H13, MTTR = mtr_H13);
    block H14( MTTF = mttf_H14, MTTR = mtr_H14);
    parallel p1 (Link_1, Link_2);
    series s0(p1, Roteador_1, Roteador_2, Roteador_3, Switch_16p, Switch_8p, Servidor, H1, H2,
H3, H4, H5, H6, H7, H8, H9, H10, H11, H12, H13, H14 );

    top s0;

    metric av = availability;
    metric rel = reliability( time = t );
    metric mttf = mttf;
    metric mtr = mtr;
}
```

```
main{
```

```
    av = solve(Model, av);
    rel = solve(Model, rel);
    mttf = solve(Model, mttf);
    mtr = solve(Model, mtr);

    println("Availability: " .. av );
    println("Reliability: " .. rel );
    println("Mean time to failure: " .. mttf );
    println("Mean time to repair: " .. mtr );

    percentageDifference(
```

```

model_ = "Model",
metric_ = "av",
samplingPoints = 7,
parameters = (
    mttf_Link_1 = [596, 1196],
    mtrr_Link_1 = [1, 4],
    mttf_Link_2 = [596, 1196],
    mtrr_Link_2 = [1, 4],
    mttf_Roteador_1 = [11243, 41243],
    mtrr_Roteador_1 = [0.58, 1.88],
    mttf_Roteador_2 = [11243, 41243],
    mtrr_Roteador_2 = [0.58, 1.88],
    mttf_Roteador_3 = [11243, 41243],
    mtrr_Roteador_3 = [0.58, 1.88],
    mttf_Switch_16p = [8043, 38043],
    mtrr_Switch_16p = [0.35, 1.85],
    mttf_Switch_8p = [8043, 38043],
    mtrr_Switch_8p = [0.35, 1.85],
    mttf_Servidor = [1417.29, 3817.29],
    mtrr_Servidor = [0.99, 4.00],
    mttf_H1 = [2654.96, 5654.96],
    mtrr_H1 = [0.3508, 1.85],
    mttf_H2 = [2654.96, 5654.96],
    mtrr_H2 = [0.3508, 1.85],
    mttf_H3 = [2654.96, 5654.96],
    mtrr_H3 = [0.3508, 1.85],
    mttf_H4 = [2654.96, 5654.96],
    mtrr_H4 = [0.3508, 1.85],
    mttf_H5 = [2654.96, 5654.96],
    mtrr_H5 = [0.3508, 1.85],
    mttf_H6 = [2654.96, 5654.96],
    mtrr_H6 = [0.3508, 1.85],
    mttf_H7 = [2654.96, 5654.96],
    mtrr_H7 = [0.3508, 1.85],
    mttf_H8 = [2654.96, 5654.96],
    mtrr_H8 = [0.3508, 1.85],
    mttf_H9 = [2654.96, 5654.96],
    mtrr_H9 = [0.3508, 1.85],
    mttf_H10 = [2654.96, 5654.96],
    mtrr_H10 = [0.3508, 1.85],
    mttf_H11 = [2654.96, 5654.96],
    mtrr_H11 = [0.3508, 1.85],
    mttf_H12 = [2654.96, 5654.96],
    mtrr_H12 = [0.3508, 1.85],
    mttf_H13 = [2654.96, 5654.96],
    mtrr_H13 = [0.3508, 1.85],
    mttf_H14 = [2654.96, 5654.96],
    mtrr_H14 = [0.3508, 1.85]
),
output = (

```

```
    type = "swing",  
    ylabel = "Steady-state availability",  
    baselineValue = av  
  );  
}
```


APÊNDICE C

CONFIGURAÇÃO PARA O CÁLCULO DE ANÁLISE DE SENSIBILIDADE ARQUITETURA A3

t = 100;
tx = 1;
mttf_Link_1 = 796*tx;
mtrr_Link_1 = 1.0*tx;
mttf_Link_2 = 796*tx;
mtrr_Link_2 = 1.0*tx;
mttf_Roteador_1 = 16243*tx;
mtrr_Roteador_1 = 0.78*tx;
mttf_Roteador_2 = 16243*tx;
mtrr_Roteador_2 = 0.78*tx;
mttf_Roteador_3 = 16243*tx;
mtrr_Roteador_3 = 0.78*tx;
mttf_Switch_16p = 13043*tx;
mtrr_Switch_16p = 0.35*tx;
mttf_Switch_8p = 13043*tx;
mtrr_Switch_8p = 0.35*tx;
mttf_Servidor1 = 1417.29*tx;
mtrr_Servidor1 = 0.99*tx;
mttf_Servidor2 = 1417.29*tx;
mtrr_Servidor2 = 0.99*tx;
mttf_H1 = 2654.96*tx;
mtrr_H1 = 0.3508*tx;
mttf_H2 = 2654.96*tx;
mtrr_H2 = 0.3508*tx;
mttf_H3 = 2654.96*tx;
mtrr_H3 = 0.3508*tx;
mttf_H4 = 2654.96*tx;
mtrr_H4 = 0.3508*tx;
mttf_H5 = 2654.96*tx;
mtrr_H5 = 0.3508*tx;
mttf_H6 = 2654.96*tx;
mtrr_H6 = 0.3508*tx;
mttf_H7 = 2654.96*tx;
mtrr_H7 = 0.3508*tx;
mttf_H8 = 2654.96*tx;
mtrr_H8 = 0.3508*tx;
mttf_H9 = 2654.96*tx;
mtrr_H9 = 0.3508*tx;
mttf_H10 = 2654.96*tx;
mtrr_H10 = 0.3508*tx;
mttf_H11 = 2654.96*tx;
mtrr_H11 = 0.3508*tx;
mttf_H12 = 2654.96*tx;
mtrr_H12 = 0.3508*tx;
mttf_H13 = 2654.96*tx;
mtrr_H13 = 0.3508*tx;

```
mttf_H14 = 2654.96*tx;
mttr_H14 = 0.3508*tx;
```

```
RBD Model{
```

```
    block Link_1( MTTF = mttf_Link_1, MTTR = mttr_Link_1);
    block Link_2( MTTF = mttf_Link_2, MTTR = mttr_Link_2);
    block Roteador_1( MTTF = mttf_Roteador_1, MTTR = mttr_Roteador_1);
    block Roteador_2( MTTF = mttf_Roteador_2, MTTR = mttr_Roteador_2);
    block Roteador_3( MTTF = mttf_Roteador_3, MTTR = mttr_Roteador_3);
    block Switch_16p( MTTF = mttf_Switch_16p, MTTR = mttr_Switch_16p);
    block Switch_8p( MTTF = mttf_Switch_8p, MTTR = mttr_Switch_8p);
    block Servidor1( MTTF = mttf_Servidor1, MTTR = mttr_Servidor1);
    block Servidor2( MTTF = mttf_Servidor2, MTTR = mttr_Servidor2);
    block H1( MTTF = mttf_H1, MTTR = mttr_H1);
    block H2( MTTF = mttf_H2, MTTR = mttr_H2);
    block H3( MTTF = mttf_H3, MTTR = mttr_H3);
    block H4( MTTF = mttf_H4, MTTR = mttr_H4);
    block H5( MTTF = mttf_H5, MTTR = mttr_H5);
    block H6( MTTF = mttf_H6, MTTR = mttr_H6);
    block H7( MTTF = mttf_H7, MTTR = mttr_H7);
    block H8( MTTF = mttf_H8, MTTR = mttr_H8);
    block H9( MTTF = mttf_H9, MTTR = mttr_H9);
    block H10( MTTF = mttf_H10, MTTR = mttr_H10);
    block H11( MTTF = mttf_H11, MTTR = mttr_H11);
    block H12( MTTF = mttf_H12, MTTR = mttr_H12);
    block H13( MTTF = mttf_H13, MTTR = mttr_H13);
    block H14( MTTF = mttf_H14, MTTR = mttr_H14);
    parallel p1 (Link_1, Link_2);
    parallel p2 (Servidor1, Servidor2);
    series s0(p1, Roteador_1, Roteador_2, Roteador_3, Switch_16p, Switch_8p, p2, H1, H2, H3,
H4, H5, H6, H7, H8, H9, H10, H11, H12, H13, H14 );
```

```
top s0;
```

```
metric av = availability;
metric rel = reliability( time = t );
metric mttf = mttf;
metric mttr = mttr;
```

```
}
```

```
main{
```

```
    av = solve(Model, av);
    rel = solve(Model, rel);
    mttf = solve(Model, mttf);
    mttr = solve(Model, mttr);
```

```
    println("Availability: " .. av );
    println("Reliability: " .. rel );
```

```

println("Mean time to failure: " .. mttf );
println("Mean time to repair: " .. mtrr );

percentageDifference(
  model_ = "Model",
  metric_ = "av",
  samplingPoints = 7,
  parameters = (
    mttf_Link_1 = [596, 1196],
    mtrr_Link_1 = [1, 4],
    mttf_Link_2 = [596, 1196],
    mtrr_Link_2 = [1, 4],
    mttf_Roteador_1 = [11243, 41243],
    mtrr_Roteador_1 = [0.58, 1.88],
    mttf_Roteador_2 = [11243, 41243],
    mtrr_Roteador_2 = [0.58, 1.88],
    mttf_Roteador_3 = [11243, 41243],
    mtrr_Roteador_3 = [0.58, 1.88],
    mttf_Switch_16p = [8043, 38043],
    mtrr_Switch_16p = [0.35, 1.85],
    mttf_Switch_8p = [8043, 38043],
    mtrr_Switch_8p = [0.35, 1.85],
    mttf_Servidor1 = [1417.29, 3817.29],
    mtrr_Servidor1 = [0.99, 4.00],
    mttf_Servidor2 = [1417.29, 3817.29],
    mtrr_Servidor2 = [0.99, 4.00],
    mttf_H1 = [2654.96, 5654.96],
    mtrr_H1 = [0.3508, 1.85],
    mttf_H2 = [2654.96, 5654.96],
    mtrr_H2 = [0.3508, 1.85],
    mttf_H3 = [2654.96, 5654.96],
    mtrr_H3 = [0.3508, 1.85],
    mttf_H4 = [2654.96, 5654.96],
    mtrr_H4 = [0.3508, 1.85],
    mttf_H5 = [2654.96, 5654.96],
    mtrr_H5 = [0.3508, 1.85],
    mttf_H6 = [2654.96, 5654.96],
    mtrr_H6 = [0.3508, 1.85],
    mttf_H7 = [2654.96, 5654.96],
    mtrr_H7 = [0.3508, 1.85],
    mttf_H8 = [2654.96, 5654.96],
    mtrr_H8 = [0.3508, 1.85],
    mttf_H9 = [2654.96, 5654.96],
    mtrr_H9 = [0.3508, 1.85],
    mttf_H10 = [2654.96, 5654.96],
    mtrr_H10 = [0.3508, 1.85],
    mttf_H11 = [2654.96, 5654.96],
    mtrr_H11 = [0.3508, 1.85],
    mttf_H12 = [2654.96, 5654.96],
    mtrr_H12 = [0.3508, 1.85],
  )
)

```

```
mttf_H13 = [2654.96, 5654.96],
mttr_H13 = [0.3508, 1.85],
mttf_H14 = [2654.96, 5654.96],
mttr_H14 = [0.3508, 1.85]
),
output = (
type = "swing",
yLabel = "Steady-state availability",
baselineValue = av
)
);
}
```