

Jamil Kalil Naufal Júnior

**Algoritmo e Arquitetura
para a Localização de Falhas em
Sistemas Distribuídos**

*Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Mestre em Engenharia*

São Paulo

2000

Jamil Kalil Naufal Júnior

**Algoritmo e Arquitetura
para a Localização de Falhas em
Sistemas Distribuídos**

*Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Mestre em Engenharia*

*Área de concentração:
Sistemas Digitais*

*Orientador:
Prof. Dr. João Batista Camargo Júnior*

**São Paulo
2000**

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

NAUFAL JUNIOR, JAMIL KALIL
ALGORITMO E ARQUITETURA PARA A LOCALIZAÇÃO
DE FALHAS EM SISTEMAS DISTRIBUÍDOS / J. K. NAUFAL
JUNIOR -- versão corr. -- São Paulo, 2000.
235 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. FALHAS 2. REDE DE TELECOMUNICAÇÕES 3. GERÊNCIA DE REDES 4. ALGORITMOS 5. REDES DE COMPUTADORES I.
Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

Dedico este trabalho à minha mãe Esmeralda, meu pai Jamil,
minha esposa Rosimeire e minha filha Juliana.

AGRADECIMENTOS

Gostaria de manifestar meus agradecimentos a todos os que contribuíram de forma direta ou indireta para a realização deste trabalho:

- Ao prof. Dr. João Batista Camargo Júnior por ter me orientado desde o início deste trabalho de dissertação, incentivando e acreditando em meu potencial.
- A minha mãe e ao meu pai pelo apoio durante todo este período.
- A minha esposa pela compreensão e apoio nesta minha caminhada.
- Ao estagiário Claudio Macoto Hayashi, pela grandiosa ajuda nas simulações.

SUMÁRIO

Lista de figuras	
Lista de tabelas	
Lista de abreviaturas	
Lista de símbolos	
<i>Abstract</i>	
1. Introdução.....	18
1.1. Objetivos.....	18
1.2. Motivação.....	18
1.3. Apresentação da dissertação.....	20
2. Conceitos sobre Gerenciamento de Redes.....	22
2.1. Objetos e Base de Informação Gerencial – MIB.....	22
2.1.1. Objetos.....	22
2.1.2. Base de Informação Gerencial.....	23
2.2. Gerenciamento de Redes.....	24
2.2.1. Gerenciamento de Falhas.....	25
2.2.2. Gerenciamento de Configuração.....	25
2.2.3. Gerenciamento de Contabilização.....	26
2.2.4. Gerenciamento de Desempenho.....	26
2.2.5. Gerenciamento de Segurança.....	27
2.3. Arquiteturas de Gerenciamento.....	28
2.3.1. Arquitetura de Gerenciamento Centralizado.....	28
2.3.2. Arquitetura de Gerenciamento Distribuído.....	30
2.3.3. Arquitetura de Gerenciamento Hierárquico.....	32
3. Arquitetura de Gerenciamento Hierárquico Proposta.....	36
4. Modelamento para Sistemas Distribuídos e Conceituação Básica de falhas.....	39
4.1. Modelo de Sistema Distribuído.....	39
4.1.1. Modelo Físico de um Sistema Distribuído.....	40
4.1.2. Modelo Lógico de um Sistema Distribuído.....	41

4.2 Classificação das falhas.....	44
4.3. Confiabilidade.....	47
4.4. Disponibilidade.....	49
4.5. Cobertura de Falhas.....	50
5. Gerenciamento de Falhas.....	51
5.1. Tipos e Níveis de Severidade.....	51
5.2. Conceitos sobre Gerenciamento Hierárquico.....	53
5.3. Grafo de Dependência.....	55
5.4. Processo de Localização de Falhas em Sistemas Distribuídos.....	58
5.5. Modelos de Diagnóstico e Localização de Falhas.....	60
5.5.1. Modelo PMC.....	61
5.5.2. Algoritmos de Localização de Falhas.....	63
5.5.2.1. Algoritmo Pesquisa Exaustiva.....	64
5.5.2.2. Algoritmo Pesquisa Gulosa (<i>Greedy</i>)	65
5.5.2.3. Algoritmo <i>Divida e Conquiste</i>	65
6. Algoritmo <i>Divida e Conquista Proposto</i>	71
6.1 O Algoritmo Proposto.....	72
7. Algoritmos de Localização de Falhas aplicados às Arquiteturas de Gerenciamento.....	79
7.1. Testes na Arquitetura de Gerenciamento de Falhas Centralizado.....	79
7.2. Testes na Arquitetura de Gerenciamento de Falhas Distribuído.....	84
7.3. Testes na Arquitetura de Gerenciamento de Falhas Hierárquico Proposto..	87
7.3.1. Nível 1 de Gerenciamento.....	91
7.3.2. Nível 2 de Gerenciamento.....	95
7.3.3. Conclusão Preliminar.....	97
8. Verificação de Desempenho da Arquitetura e do Algoritmo Propostos.....	98
8.1. Ambiente de Simulação.....	98
8.2. Simulações Comparativas entre o Algoritmo <i>Divida e Conquiste</i> e o <i>Divida e Conquista Proposto</i>	99
8.2.1. Realização das Simulações.....	101
8.2.2. Análise das Simulações.....	103

8.3. Simulações Comparativas entre as Arquiteturas de Gerenciamento....	109
8.3.1. Realização das Simulações.....	111
8.3.1.1. Simulação para a Arquitetura Centralizada.....	113
8.3.1.2. Simulação para a Arquitetura Distribuída.....	117
8.3.1.3. Simulação para a Arquitetura Hierárquica Proposta.....	122
8.3.2. Resultados e análise das simulações.....	128
9. Considerações Finais.....	135
9.1. Resultados Alcançados.....	135
9.2. Futuras Linhas de Pesquisas.....	136
9.2.1. Técnicas para Coleta, Interpretação e Atribuição das diversas probabilidades utilizadas no algoritmo <i>Divida e</i> <i>Conquiste Proposto</i> , bem como no Gerenciamento Proativo da rede.....	136
9.2.2. Delegação de Processos de Localização de Falhas em Redes Distribuídas.....	137
9.2.3. Estudo de Técnicas de Busca para o Algoritmo <i>Divida e</i> <i>Conquiste Proposto</i> associado à Arquitetura Hierárquica Proposta.....	137
9.2.4. Análise do Algoritmo <i>Divida e Conquiste Proposto</i> e da Arquitetura Hierárquica Proposta em função da Variação do Grau do Arco de Dependência.....	138
9.2.5. Estudo de Técnicas para Automatizar o Processo de Geração de Grafos de Dependência.....	138
9.2.6. Análise de Propagação de Falhas por uma Rede de Comunicação e sua Interferência no algoritmo <i>Divida e Conquiste</i>	138
9.3. Aplicações Práticas relacionadas à Dissertação.....	139
9.3.1. Desenvolvimento de uma Interface Homem-Máquina.....	139
9.3.2. Implementação e avaliação do algoritmo <i>Divida e</i> <i>Conquiste Proposto</i> e da Arquitetura de Gerenciamento	

Hierárquico Proposto em um ambiente real.....	142
Anexo A.....	143
Anexo B.....	156
Anexo C.....	168
Anexo D.....	191
Anexo E.....	215
Referências Bibliográficas	231

Lista de figuras

- Figura 2.1: Arquitetura de Gerenciamento Centralizado
- Figura 2.2: Arquitetura Centralizada utilizando Objetos Gerenciados
- Figura 2.3: Arquitetura de Gerenciamento Distribuído
- Figura 2.4: Arquitetura Distribuída utilizando Objetos Gerenciados
- Figura 2.5: Arquitetura de Gerenciamento Hierárquico
- Figura 2.6: Arquitetura Hierárquica utilizando Objetos Gerenciados
- Figura 3.1: Arquitetura de Gerenciamento Hierárquico Proposta
- Figura 3.2: Relacionamento entre objetos na Arquitetura Hierárquica Proposta
- Figura 4.1: Modelos de Topologias de Rede
- Figura 4.2: Classificação de Falhas em Sistemas Distribuídos
- Figura 5.1: Definição de Domínios de Gerenciamento
- Figura 5.2: Definição de *Cluster* de Domínios de Alarmes
- Figura 5.3: Exemplo de um Modelo de Rede
- Figura 5.4: Grafo de Dependência para o Modelo de Rede
- Figura 5.5: Teste de um Sistema consistindo de 5 Unidades e utilizando o modelo PMC
- Figura 7.1: Topologia física da rede
- Figura 7.2: Visão da Arquitetura de Gerenciamento Centralizada de Rede
- Figura 7.3 : Grafo de dependência para o exemplo de Gerenciamento Centralizado
- Figura 7.4: Topologia Física da Rede
- Figura 7.5: Visão da Arquitetura de Gerenciamento Distribuído da Rede
- Figura 7.6: Arquitetura do Modelo Físico da Rede
- Figura 7.7: Visão da Arquitetura de Gerenciamento Hierárquico Proposto da Rede
- Figura 7.8: Arquitetura do modelo lógico da rede
- Figura 7.9: Modelo físico do domínio A
- Figura 7.10: Objetos do domínio A
- Figura 7.11: Modelo físico do domínio B
- Figura 7.12: Objetos do domínio B
- Figura 7.13: Modelo físico do domínio C

Figura 7.14: Objetos do domínio C

Figura 7.15: Modelamento dos domínios

Figura 7.16: Grafo de dependência para o exemplo proposto

Figura 8.1. Interfaces do ambiente de Simulação

Figura 8.2. Simulação do gráfico do objeto A_1 variando o seu fator de cobertura

Figura 8.3. Simulação do gráfico do objeto C_1 variando o seu fator de cobertura

Figura 8.4. Simulação do gráfico do objeto L_B variando o seu fator de cobertura

Figura 8.5 Expansão da Arquitetura Física

Figura 8.6: Expansão de uma rede utilizando Gerenciamento Centralizado

Figura 8.7: Expansão de uma rede utilizando Gerenciamento Centralizado

Figura 8.8: Expansão de uma rede utilizando Gerenciamento Centralizado

Figura 8.9: Expansão de uma rede utilizando Gerenciamento Centralizado

Figura 8.10: Expansão de uma rede utilizando Gerenciamento Distribuído

Figura 8.11: Expansão de uma rede utilizando Gerenciamento Distribuído

Figura 8.12: Expansão de uma rede utilizando Gerenciamento Distribuído

Figura 8.13: Expansão de uma rede utilizando Gerenciamento Distribuído

Figura 8.14: Expansão de uma rede utilizando Gerenciamento Hierárquico

Figura 8.15: Expansão de uma rede utilizando Gerenciamento Hierárquico

Figura 8.16: Expansão de uma rede utilizando gerenciamento

Figura 8.17: Expansão de uma rede utilizando Gerenciamento Hierárquico

Figura 8.18: Gráfico relacionando número de objetos gerenciados e tempo de localização de falhas utilizando o algoritmo *Divida e Conquiste Proposto*

Figura 8.19: Tempos de localização de falhas entre as Arquiteturas Centralizada e Distribuída e relação a Arquitetura Hierárquica.

Figura 8.20: Relação entre a taxa de crescimento dos objetos gerenciados e dos Tempos de localização de falhas para as Arquiteturas Centralizada, Distribuída e Hierárquica.

Figura 9.1: Tela do nível 1 de Gerenciamento

Figura 9.2: Tela de Gerenciamento de Falha com objetos em estado falho

Figura 9.3: Seqüência de telas do Ambiente de Gerenciamento

Lista de tabelas

Tabela 7.1: Relacionamento entre objetos de diferentes domínios de gerenciamento

Tabela 7.2: Relação de dependência entre objetos em diferentes níveis de gerenciamento.

Tabela 8.1: Número de objetos gerenciados em cada passo da simulação

Tabela 8.2 : Resultados obtidos para a simulação das arquiteturas

Tabela 8.3: Relação entre tempos de localização de falhas das Arquiteturas Centralizada e Distribuída em relação a Hierárquica.

Tabela 8.4: Relação entre crescimento dos objetos gerenciados e crescimento dos tempos de localização de falhas

Lista de Símbolos

A^t	Subconjunto t de alarmes do conjunto de alarmes A.
C_A	Fator de Cobertura do objeto A.
e_i	Objeto gerenciável i.
FCU	Fator de Cobertura da união entre conjuntos.
PU	Probabilidade de união das partições.
P_A	Probabilidade independente da partição A.
P_{AB}	Probabilidade da partição A falhar, dado que a partição B falhou.
p_i	Probabilidade de falha do objeto i.
P_{ij}	Probabilidade do objeto i falhar, dado que o objeto j falhou.
$Pr(S)$	Probabilidade do conjunto S.
PU	Probabilidade de união das partições.
$R(t)$	Confiabilidade.
S	Conjunto de elementos.
Sd^t	Filho direito do subconjunto t.
Se^t	Filho esquerdo do subconjunto t.
S_i	Subconjunto i do conjunto S.
S^t	Subconjunto t do conjunto S.
S_{sol}	Conjunto solução final.
w_{ij}	Peso de dependência entre os objetos i e j.
$x \in A$	Pertinência de um elemento x a um conjunto A .
∇j	Qualquer elemento j de um conjunto.
Σ	Somatória.
μ	Taxa de reparo de um componente/sistema.
λ	Taxa de falha de um componente/sistema.
\int	Integral de uma função.
∞	Infinito.
\prod	Produtória.
α	Disponibilidade Assintótica .

Lista de Abreviaturas

ATM	<i>Assincronous Tranfer Mode</i>
CMIP	<i>Common Management Protocol.</i>
FR	<i>Frame Relay</i>
LAN	<i>Local Area Network</i>
MIB	<i>Management Information Base.</i>
OSI	<i>Open Systems Interconnection.</i>
SNMP	<i>Simple Network Management Protocol.</i>
TCP/IP	<i>Transfer Control Protocol / Internet Protocol.</i>
MTTF	<i>Mean Time to Failure.</i>
MTTR	<i>Mean Time to Repair.</i>

Resumo

Devido à tendência mundial de crescimento sem precedentes na história das telecomunicações, verifica-se atualmente um aumento crescente no segmento das redes de comunicações com respeito ao seu tamanho e em seus correspondentes elementos, tornando o seu gerenciamento uma atividade árdua e complexa do ponto de vista de sua operação. Adicione-se a isto o fato de que o sucesso no empreendimento dos diferentes negócios atuais, no contexto da utilização das redes de comunicações, é dependente exclusivamente da qualidade do serviço e funcionamento dessas redes.

A ocorrência de falhas em qualquer sistema de comunicação é de certa forma inevitável e, portanto, mais críticos em sistemas de grande porte, seja em termos de quantidade e variedade de falhas. Dessa forma, é desejável que sejam desenvolvidas novas técnicas que permitam à rede de comunicação uma maior rapidez e eficiência na detecção e correção de operações sistêmicas anormais e, conseqüentemente, sejam consideradas como atributos prioritários em seu projeto. Em outras palavras, a implementação destas novas técnicas permitirá ao sistema a capacidade de detecção, isolamento e reconfiguração de um dado componente falho com referência aos requisitos de maior rapidez e eficiência, aumentando sobremaneira a disponibilidade da rede.

Neste trabalho de dissertação é proposto um algoritmo e uma arquitetura para o gerenciamento de falhas, além de verificar a sua aderência quanto ao requisito disponibilidade de rede.

Abstract

Due to the unprecedented world growth trend in the telecommunications history, it is currently realized the enormous increase in the communication networks segment regarding in the size and in its correspondent elements as well, becoming its own management an arduous and complex activity in the operation point of view consideration. Also, it must be taken into account that the real success concerning the currently and different business enterprise in the communications networks usage context are exclusively dependant on the quality of the service and the correct working of these networks.

The fault occurrence in any communication system is generally inevitable and therefore more critical when considering large systems configuration, in terms of number and variety of faults. In this way, it is desirable that new techniques be developed, allowing the communication networks the ability to handle faster and more efficiency the detection and correction in case of anormal systemic operations. Therefore, it must be considered as a priority attribute in the new networks design. In other words, the implementation of these new techniques will allow the systems to have the capacity to detect, isolate and reconfigure a given fault component, regarding the quickness and efficiency attribute, increasing the network availability.

It is proposed in this dissertation, an algorithm and an architecture for fault management and how they are adhered to the network availability.

1. Introdução

Neste capítulo são apresentados os objetivos e a motivação que levaram ao desenvolvimento deste trabalho de pesquisa.

1.1. Objetivos

Esta dissertação de mestrado teve dois objetivos básicos. O primeiro refere-se ao desenvolvimento de um algoritmo de localização de falhas, que procura selecionar, dentre os objetos gerenciados, aqueles com maior probabilidade de serem o objeto falho. Para tanto, são introduzidos os conceitos de fator de cobertura e seqüência de teste, que contribuem na precisão da seleção dos objetos a serem selecionados e, portanto, minimizam o número de testes a serem desenvolvidos. No escopo da dissertação os objetos gerenciados (conexões e nós de rede) são atômicos, e suas arquiteturas internas não são consideradas.

O segundo objetivo deste trabalho envolve a proposta de uma Arquitetura de Gerenciamento de Falhas Hierárquica, que procura proporcionar um menor tempo de processamento e, portanto, minimizar o tempo de execução do algoritmo de localização de falhas. No escopo desta dissertação está se considerando a rede como um sistema distribuído.

Desta forma, a dissertação procura também investigar a utilização do Algoritmo de Falhas Proposto na Arquitetura de Gerenciamento de Falhas Proposta, em especial aos aspectos relacionados com a disponibilidade dos serviços de rede aos usuários.

1.2. Motivação

Um sistema distribuído tem como base um ambiente de rede. Este ambiente cresce cada vez mais em número e diversidade de elementos, tornando seu gerenciamento uma

atividade extremamente complexa. Dentre estas atividades pode-se citar os testes, a identificação e isolamento de falhas, fundamentais na garantia da disponibilidade de um serviço [NIGATE-95].

Um estudo realizado pela Universidade de Austin, Texas (EUA), relata que o impacto de uma falha em uma rede reduz a receita e produz um aumento no custo da empresa que utiliza esta rede. De acordo com este estudo, o custo de paralisação da rede pode variar de 2% (correspondendo ao primeiro dia de paralisação) até 30% (correspondendo ao 30º dia ininterrupto de paralisação) da receita anual da empresa [CARVALHO-92].

Algumas ações normalmente podem ser realizadas na tentativa de amenizar o tempo de paralisação da rede e, desta forma, tentar garantir a continuidade do serviço disponível pela rede. Dentre estas ações pode-se citar:

- Utilizar um elemento redundante ou também chamado de elemento espelho. Este elemento espelho é idêntico ao elemento principal, entrando em atividade apenas na detecção de falha do elemento principal.
- Redirecionar o atendimento a outro elemento de rede que possua capacidade de realizar o serviço solicitado.
- Reconfigurar o elemento em caso de falha parcial, de forma que o serviço continue a ser disponível através da parte não afetada pela falha.

Entretanto, as alternativas anteriores nem sempre são possíveis de serem realizadas e, mesmo quando possíveis, podem apresentar um custo extremamente elevado, especialmente quando se mantêm elementos redundantes na rede. O fato torna-se ainda mais crítico quando o diagnóstico de uma falha é incorreto ou impreciso, podendo levar a uma manutenção ineficiente e, principalmente, à perda do serviço prestado, tornando o custo operacional da empresa extremamente alto.

Considerando a falha um fenômeno inevitável, e pelo exposto anteriormente, uma maneira eficiente de aumentar a disponibilidade da rede é diminuindo o tempo de paralisação da

rede, ou seja, aumentando a precisão e diminuindo o tempo de processamento do processo de localização de falhas.

Outro aspecto motivante é a utilização das técnicas discutidas neste trabalho com o intuito de antecipar, quando possível, a ocorrência de falhas na rede. Este tipo de abordagem pode ser implementado utilizando o algoritmo *Divida e Conquiste Proposta* e a Arquitetura de Gerenciamento Proposta na medição de fatores de desempenho da rede. Desta forma, pode-se identificar os objetos com maior probabilidade de virem a falhar, sendo então já substituídos. Esta atitude está relacionada com o gerenciamento proativo de gerenciamento de rede.

1.3. Apresentação da dissertação

A área de redes de comunicação torna-se a cada dia mais importante e fundamental, principalmente em um mundo globalizado, no qual a capacidade da comunicação, o acesso e a troca de informações tornam-se vitais. Desta forma, o gerenciamento de redes e a sua sub-área denominada gerenciamento de falhas são de fundamental importância para a garantia ininterrupta da comunicação. Esta problemática constitui-se num dos fatores que vem incentivando a pesquisa de novas técnicas e estruturas de redes mais eficientes visando permitir, aos usuários, uma maior disponibilidade dos serviços desejados.

A presente dissertação divide-se em 9 capítulos.

No capítulo 2 são apresentados os conceitos sobre gerenciamento de redes: as áreas de gerenciamento em redes, *Management Information Base* (MIB), e as Arquiteturas de Gerenciamento Centralizado, Distribuído e Hierárquico. Também apresentam-se as vantagens e desvantagens de cada uma destas arquiteturas de gerenciamento.

No capítulo 3 é apresentada a proposta de Arquitetura de Gerenciamento Hierárquico Proposta neste trabalho de pesquisa.

No capítulo 4 são apresentados outros conceitos também fundamentais para o desenvolvimento da dissertação: definição do modelo físico e lógico de sistema, bem como uma possível classificação de falhas. São definidos também os conceitos de confiabilidade e disponibilidade de sistema, além do termo cobertura, ainda muito pouco conhecido e utilizado em gerenciamento de falhas em redes.

O capítulo 5 apresenta os conceitos envolvidos como o Gerenciamento Hierárquico Proposto, o grafo de dependência e o processo de localização de falhas em sistemas distribuídos. São apresentados os algoritmos de localização de falha, *Pesquisa Exaustiva*, *Pesquisa Gulosa (Greedy)* e *Divida e Conquiste*. Este último constitui-se na base do algoritmo *Divida e Conquiste Proposto* neste trabalho de pesquisa.

Desta forma, no capítulo 6 é apresentado o algoritmo *Divida e Conquiste Proposto*.

O capítulo 7 apresenta exemplos ilustrativos, correspondentes à Arquitetura Centralizada, Distribuída e Hierárquica Proposta. No desenvolvimento dos exemplos são utilizados os algoritmos *Divida e Conquiste*, e *Divida e Conquiste Proposto* e, servem de subsídio para a adequada compreensão das comparações realizadas no capítulo 8.

No capítulo 8 são apresentadas algumas simulações com o objetivo de avaliar o comportamento da disponibilidade de uma rede hipotética em função do algoritmo de localização de falhas proposto, bem como da Arquitetura de Gerenciamento Proposta. Os parâmetros utilizados nestas simulações são o número de alarmes ativos, o fator de cobertura, além da probabilidade condicional e incondicional de falha dos objetos.

Concluindo, o capítulo 9 apresenta o resumo dos resultados alcançados durante este trabalho de mestrado, avaliando os objetivos inicialmente propostos, bem como destacando os aspectos que podem se constituir em futuras linhas de pesquisa.

2. Conceitos sobre Gerenciamento de Redes

Neste capítulo são apresentados alguns conceitos relacionados ao gerenciamento de redes, tais como, objetos gerenciados, Base de Informação Gerenciável (MIB – *Management Information Base*), áreas funcionais, além das arquiteturas utilizadas no gerenciamento em redes de computadores.

2.1. Objetos e Base de Informação gerencial - MIB

Neste item são apresentados os conceitos de objetos e de base de Informação Gerencial – MIB.

2.1.1. Objetos

Os conceitos ligados à orientação a objetos tornaram-se rapidamente populares na área de desenvolvimento de *software*, permitindo, de forma menos complexa, a troca, o re-uso, a atualização e interconexão das partes dos *softwares*. A motivação do uso da orientação a objetos é devido, também, a sua grande flexibilidade em representar uma grande variedade de fatos e conceitos, bastante adequada em redes de comunicação [STALLINGS-93].

Um objeto é definido, na área de desenvolvimento de *software*, como uma unidade de *software* distinta que contém uma coleção de procedimentos e dados relacionados. Geralmente, os procedimentos e os dados não são visíveis externamente ao objeto. Na prática, são desenvolvidas interfaces bem-definidas que permitem a *softwares* externos acessarem os seus dados e procedimentos. Os objetos interagem através de mensagens, característica importante para ambientes de redes distribuídas [VASSILA-95].

Do ponto de vista de gerenciamento, a rede de comunicação pode ser definida e representada através de objetos, sendo que cada objeto representa um elemento de

comunicação de rede com existência separada e distinta. Desta forma, um objeto poderá representar uma rede, um nó, um *switch*, uma camada da pilha de protocolo, um processo de *software*, um elemento físico de rede (por exemplo uma fibra óptica, uma conexão de comunicação) etc [KATZELA-93].

Assim, um objeto gerenciado pode ser definido como qualquer componente (*hardware* ou *software*) cujo comportamento possa ser controlado por um sistema de gerenciamento [SLOMAN-96]. Este objeto gerenciado, pode ser uma entidade de camada em uma arquitetura de protocolos de rede (por exemplo, arquitetura OSI, TCP/IP etc), uma conexão física ou um dispositivo de comunicação (por exemplo, um modem) [CARVALHO-93].

A abordagem de orientação a objetos é bastante adequada na representação da estrutura de dados das informações de gerenciamento em redes.

2.1.2. Base de Informação Gerencial - MIB

Todo sistema complexo de redes necessita armazenar as informações manipuladas em algum tipo de base de dados. A Base de Informação Gerencial (MIB) é uma base de dados distribuída que apresenta internamente objetos gerenciados, seus atributos, operações que eles executam e as notificações que eles fornecem [CARVALHO-92], sendo acessadas pelos agentes e gerentes de rede.

Um sistema de gerenciamento possui duas entidades básicas: agente e gerente. Um gerente é uma entidade centralizadora que obtém informações atualizadas sobre os objetos gerenciados. Para tanto, o gerente emite requisições de operações de gerenciamento aos agentes. Um agente localiza-se em diversos elementos de rede e, executa operações de gerenciamento sobre objetos gerenciados, podendo ainda enviar notificações assíncronas emitidas pelos objetos gerenciados ao gerente [CARVALHO-92] e [LEMOS-97].

Uma MIB pode também ser definida como um conjunto de objetos gerenciados que são acessíveis via um protocolo de gerenciamento de rede [LEINWAND-93] (por exemplo, o protocolo SNMP definido para a arquitetura TCP/IP ou o protocolo CMIP para a arquitetura OSI), na qual um objeto gerenciado é a visão abstrata de um recurso real e gerenciável de rede [CARVALHO-92].

2.2. Gerenciamento de Redes

Existem muitas definições para gerenciamento de redes.

De acordo com [CARVALHO-98] o gerenciamento é o processo de controlar uma rede de modo a maximizar a sua eficiência e produtividade.

O gerenciamento de rede é o processo de controlar uma complexa rede de dados, de tal forma a maximizar sua eficiência e produtividade de acordo com [LEINWAND-93].

Para [SIEGL-94] o objetivo da gerência de rede é garantir a operação regular de uma rede e, assim, permitir a máxima disponibilidade dos serviços a seus usuários.

De certa forma, essas definições vêm reforçar os objetivos deste trabalho de pesquisa, ou seja, melhorar a disponibilidade dos serviços de rede aos usuários, um fator importante de eficiência.

O gerenciamento de redes é muito abrangente, sendo definido pela OSI (*Open Systems Interconnection*) nas seguintes áreas funcionais: Gerenciamento de Falhas, Gerenciamento de Configuração, Gerenciamento de Contabilização, Gerenciamento de Desempenho e Gerenciamento de Segurança [CARVALHO-92, CARVALHO-98, LEINWAND-93, RAMAN-98 e STALLINGS-93].

2.2.1. Gerenciamento de Falhas

Trata-se de uma das áreas mais importantes de gerenciamento de redes, envolvendo o controle do estado operacional de cada componente, seja ele um modem ou um aplicativo sendo executado em um nó da rede. Normalmente, o Gerenciamento de Falhas envolve as seguintes atividades:

Detectar e reportar falhas ocorridas na rede através de um conjunto de alarmes associados aos vários objetos de rede gerenciados.

- Fornecer relatórios de eventos e erros ocorridos na rede para análises futuras.
- Realizar seqüências de teste em função de diagnósticos funcionais e de tempo de resposta.
- Isolar e corrigir falhas ocorridas na rede.
- Realizar, quando possível, a antecipação de falhas (gerenciamento proativo) através da monitoração de informações coletadas da rede.

O tema desta pesquisa enquadra-se dentro do conceito de Gerenciamento de Falhas.

2.2.2. Gerenciamento de Configuração

Envolve a visualização de informações relativas à topologia e ao fluxo de dados através da rede de comunicação, permitindo a intervenção no sistema para modificação de suas características através de aplicativos de gerenciamento. Normalmente, o Gerenciamento de Configuração envolve as seguintes atividades:

- Estabelecer parâmetros de operação com a finalidade de definir a configuração de gerenciamento da rede.
- Coletar informações sobre a configuração atual da rede, pois o ambiente de rede está continuamente modificando-se. É cada vez mais fundamental a habilidade para modificar a configuração da rede em tempo real.

- Modificar a configuração através da ativação, desativação e alteração dos parâmetros de configuração dos objetos da rede, por exemplo, quando necessita-se de maior largura de banda para um serviço fornecido a um usuário.
- Armazenar informações relativas à configuração da rede de forma a manter um inventário de todos os componentes gerenciados e emitir relatórios baseados nas mesmas.

2.2.3. Gerenciamento de Contabilização

Área de gerenciamento imprescindível para a identificação e registro de custos e volume de recursos utilizados pelos usuários da rede, dedicando-se a mensurar o conjunto de facilidades de rede utilizadas. Normalmente, o Gerenciamento de Contabilização envolve as seguintes atividades:

- Coletar dados sobre a utilização da rede (por exemplo, recursos de: comunicação, servidores, aplicativos e serviços), com a finalidade fornecer subsídios para uma melhor decisão de alocação de recursos, tarifação e custos da rede.
- Estabelecer escalas de tarifação associadas ao uso dos recursos de rede utilizados pelos vários usuários da rede.
- Determinar os custos envolvidos, realizando a combinação de custos quando múltiplos recursos são utilizados dentro de um dado objetivo de comunicação.
- Emitir relatórios sobre a utilização dos recursos de rede e custos relacionados, com a finalidade de permitir um melhor planejamento de futuras expansões da rede.

2.2.4. Gerenciamento de Desempenho

Esta área de gerenciamento é importante para garantir a qualidade de serviços especificada entre usuários da rede, ao menor custo possível, servindo assim para adequar os meios de comunicação às reais necessidades de seus usuários. Desta forma, é de fundamental importância no planejamento das capacidades da rede que está sendo gerenciada. Normalmente, o Gerenciamento de Desempenho engloba as seguintes atividades:

- Selecionar e utilizar indicadores apropriados para medir o desempenho da rede.
- Realizar atividades correlatas à medida de desempenho e sua análise:
 - Coletar informações estatísticas (por exemplo, tráfego de mensagens entre estações da rede, tipo e tamanho das mensagens, atraso de trânsito) com o objetivo de determinar o nível de serviço fornecidos aos usuários da rede.
 - Manter e examinar *logs* das informações coletadas sobre a rede.
 - Avaliar o comportamento dos recursos de rede gerenciados e a eficiência de sua utilização.
 - Determinar o desempenho da rede sob condições normais de funcionamento.
- Alterar o modo de operação do sistema com o objetivo de melhorar o desempenho da rede.

2.2.5. Gerenciamento de Segurança

Esta área utiliza o gerenciamento de redes para monitorar e controlar mecanismos de segurança que permitem desde o acesso aos sistemas computacionais, até o controle de informações sigilosas que trafegam nos circuitos de dados. Normalmente, o Gerenciamento de Segurança envolve as seguintes atividades:

- Gerenciar as facilidades, os serviços e os mecanismos de segurança, de modo a proteger os recursos computacionais e de rede contra ameaças ou violações por usuários não autorizados.
- Monitorar a utilização dos recursos de rede e as operações efetuadas por seus usuários.
- Criar, remover e controlar os serviços de segurança da rede.
- Manter e tratar *logs* de segurança sobre possíveis violações na rede.
- Emitir relatórios de eventos de segurança.

2.3. Arquiteturas de Gerenciamento

As arquiteturas utilizadas no gerenciamento em redes de computadores, pode ser dividida em: centralizada, distribuída e hierárquica, conforme é apresentado a seguir:

2.3.1. Arquitetura de Gerenciamento Centralizado

Um Sistema de Gerenciamento Centralizado possui um único gerente de rede que executa a maioria das operações e aplicações de gerenciamento, bem como, controla a atividade dos nós gerenciados através de seus agentes. Este gerente atua quando qualquer problema afeta a rede [NOGUEIRA-97].

As aplicações de Gerenciamento de Falhas e Configuração (presentes no gerente de rede) executam suas funções de forma coordenada. Assim, quando ocorre uma falha na rede, a aplicação de Gerenciamento de Falha detecta, identifica e informa qual é o objeto falho à aplicação de Gerenciamento de Configuração. A partir deste instante, a aplicação de Gerenciamento de Configuração reconfigura a rede de forma a isolar o elemento falho, e amenizar o impacto produzido pela falha nos serviços de rede disponibilizados.

A arquitetura de um Sistema de Gerenciamento Centralizado é apresentada na figura 2.1:

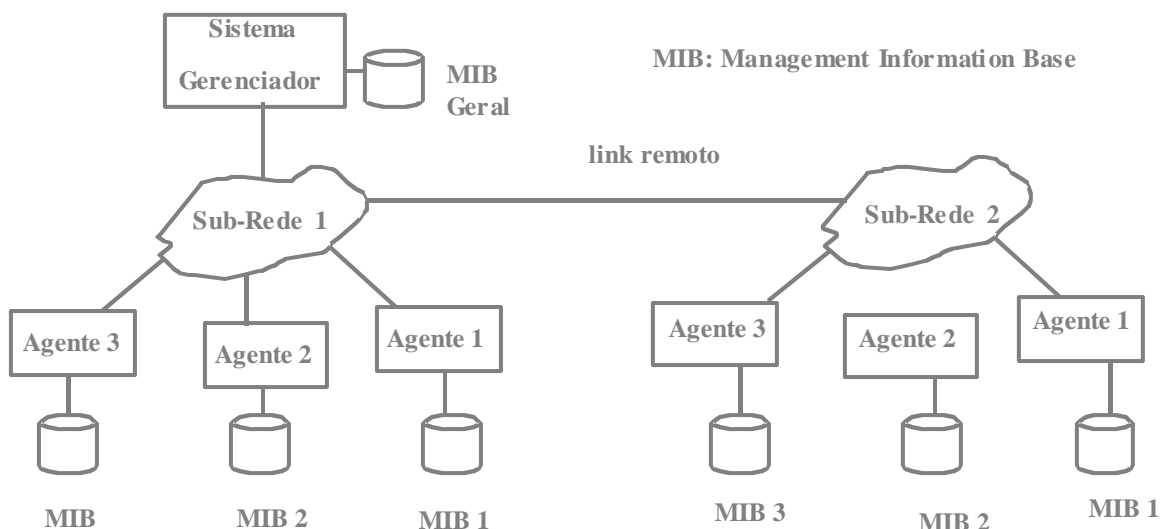


Figura 2.1: Arquitetura de Gerenciamento Centralizado

A figura 2.2 apresenta esta mesma arquitetura através do conceito de objetos [AIDAROUS-96]:

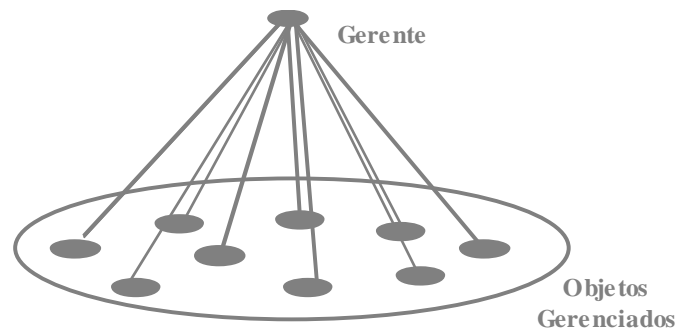


Figura 2.2: Arquitetura Centralizada utilizando Objetos Gerenciados

Um Sistema de Gerenciamento Centralizado apresenta as seguintes características:

Vantagem:

- Armazena todas as informações do sistema em uma base de dados centralizada (MIB Geral). Esta característica facilita a consulta, o controle e a segurança das informações de gerenciamento da rede [LEINWAND-93].

Desvantagens:

- Aumento não planejado da rede torna o seu gerenciamento muito complexo, na medida em que um número muito grande de informações de gerenciamento pode ser gerado. Este fato pode acarretar numa sobrecarga na rede (*overflow*), bem como ultrapassar os limites de tempo (*timeouts*) de algumas variáveis.
- A perda de comunicação com a sub-rede 2 (veja a figura 2.1) provoca a perda total do gerenciamento desta sub-rede.
- Quando o método de *polling* é utilizado para o monitoramento da rede, a largura de banda ocupada pelo gerenciamento é proporcional ao número de variáveis gerenciadas. Este fato torna o gerenciamento da rede remota um gargalo.

- No caso de futuras expansões apresenta problemas de flexibilidade, devido à necessidade de reconfigurar toda a rede, visando a compatibilidade com o Sistema de Gerenciamento Centralizado [LEINWAND-93].

2.3.2. Arquitetura de Gerenciamento Distribuído

Neste tipo de arquitetura o gerenciamento é realizado localmente em cada sub-rede, sendo compartilhado apenas algumas variáveis entre as sub-redes. Estas variáveis compartilhadas são necessárias apenas para fornecer uma visão parcial da sub-rede vizinha, e assim auxiliar na determinação do domínio do problema [BOULOUTAS-95 e SLOMAN-96].

No Gerenciamento Distribuído, igualmente ao Gerenciamento Centralizado, as aplicações de Gerenciamento de Falhas e Configuração (presentes nos gerentes da rede) executam suas funções de forma coordenada quando da ocorrência de uma falha na rede.

A arquitetura de um Sistema de Gerenciamento Distribuído pode ser apresentado como mostra a figura 2.3:

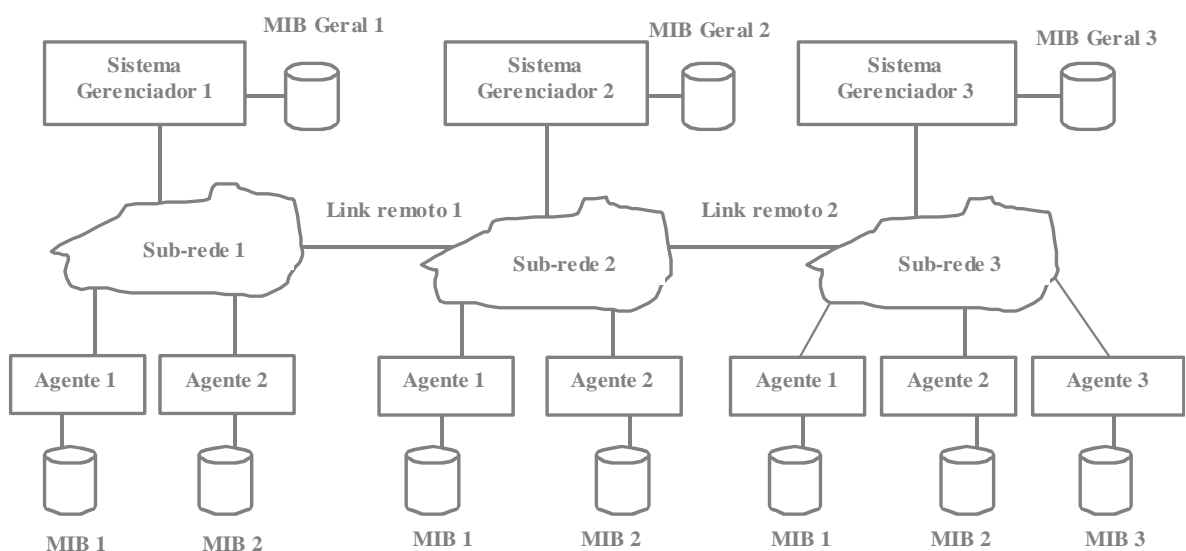


Figura 2.3: Arquitetura de Gerenciamento Distribuído

A figura 2.4 apresentada esta mesma arquitetura através do conceito de objetos:

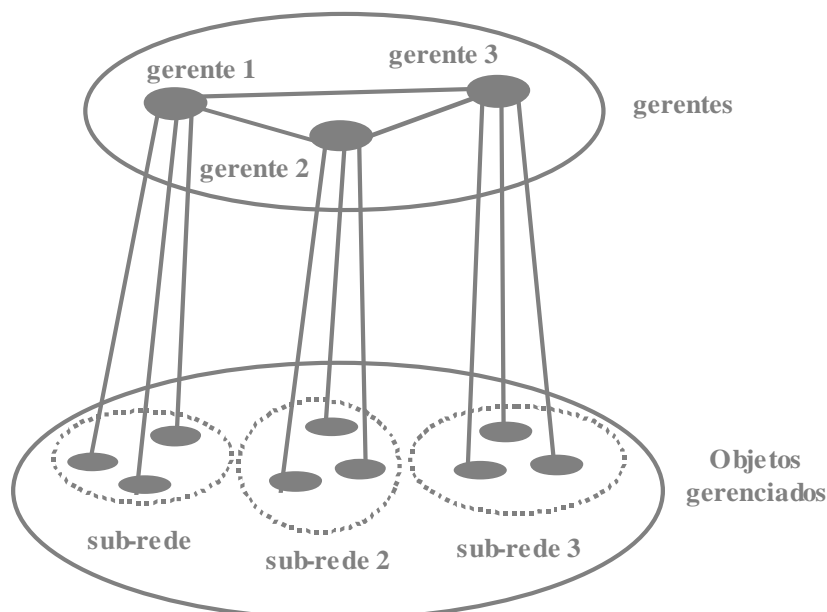


Figura 2.4: Arquitetura Distribuída utilizando Objetos Gerenciados

Um Sistema de Gerenciamento Distribuído apresenta as seguintes características:

Vantagens:

- Possui um menor número de variáveis de gerenciamento trocadas entre gerente e seus agentes, quando comparado ao Gerenciamento Centralizado. Este fato diminui a possibilidade de ocorrência de sobrecarga (*overflow*) e violação dos tempos máximos (*timeouts*).
- Apresenta maior flexibilidade em relação ao Gerenciamento Centralizado, devido à configuração da rede ser realizada localmente em cada sub-rede. As sub-redes vizinhas não necessitam ter visibilidade das variáveis locais gerenciadas em cada sub-rede de gerenciamento.

Desvantagens:

- Apresenta maior dificuldade de implementação, devido as interfaces entre os sistemas de gerenciamento apresentarem alta complexidade, em comparação ao Sistema de Gerenciamento Centralizado.
- Normalmente os algoritmos descentralizados são de alta complexidade. Este fato é explicado pelo volume de interações que os gerentes de um mesmo nível devem estabelecer, com objetivo de ter uma visão parcial dos domínios de seus vizinhos. Desta forma, necessita utilizar equipamentos (*hardware*) de gerenciamento com maior capacidade de processamento, quando comparado aos demais modelos de gerenciamento.
- As MIBs, agentes e sub-gerentes distribuídos fisicamente pela rede, também podem ser a fonte de falhas e, portanto, necessitam de mecanismos de proteção (por exemplo: utilizar mecanismos de redundância) de tal forma a garantir alta disponibilidade. O fato de adicionar elementos redundantes (MIBs, agentes e sub-gerentes) pela rede poderá elevar demasiadamente o custo da rede, principalmente quando comparado à Arquitetura de Gerenciamento Centralizada (possui apenas um agente de rede).

2.3.3. Arquitetura de Gerenciamento Hierárquico

Combina tanto as características do Sistema de Gerenciamento Centralizado, quanto do Sistema de Gerenciamento Distribuído. Utiliza o conceito de sub-redes de gerenciamento (igualmente ao Gerenciamento Distribuído), e permite que um nível de Gerenciamento Hierárquico coordene as atividades dos níveis hierárquicos imediatamente inferiores a ele (igualmente à relação entre agente e gerente realizado no Gerenciamento Centralizado).

Da mesma forma que nas Arquiteturas de Gerenciamento Centralizado e Distribuído, as aplicações de Gerenciamento de Falhas e Configuração (presentes nos sub-gerentes da rede) executam suas funções de forma coordenada quando da ocorrência de uma falha na rede.

Um Sistema de Gerenciamento Hierárquico é apresentado na figura 2.5 [SIEGL-95]:

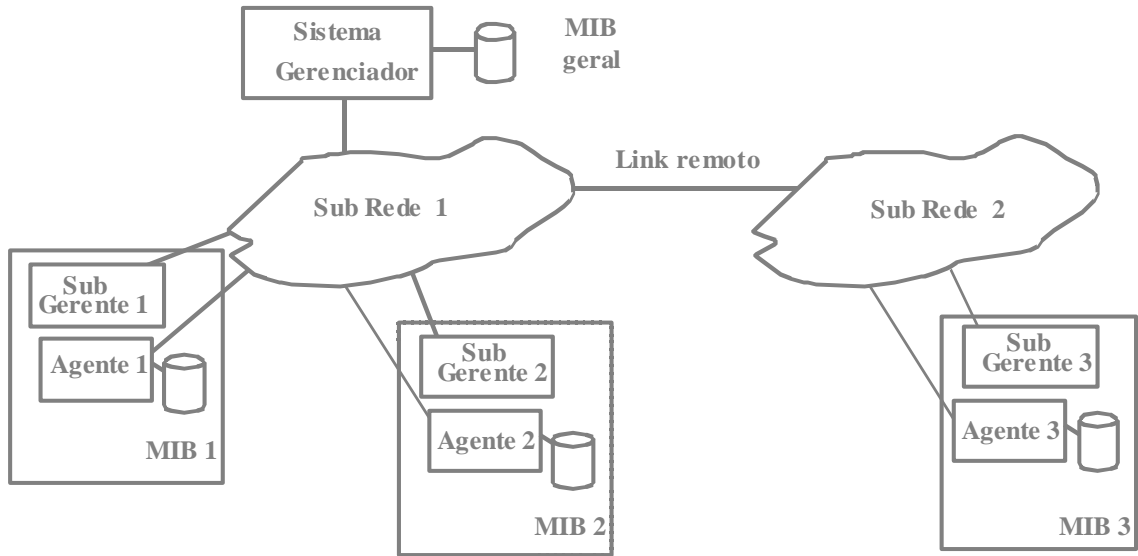


Figura 2.5: Arquitetura de Gerenciamento Hierárquico

Na figura 2.6 é apresentada a mesma arquitetura através do conceito de objetos:

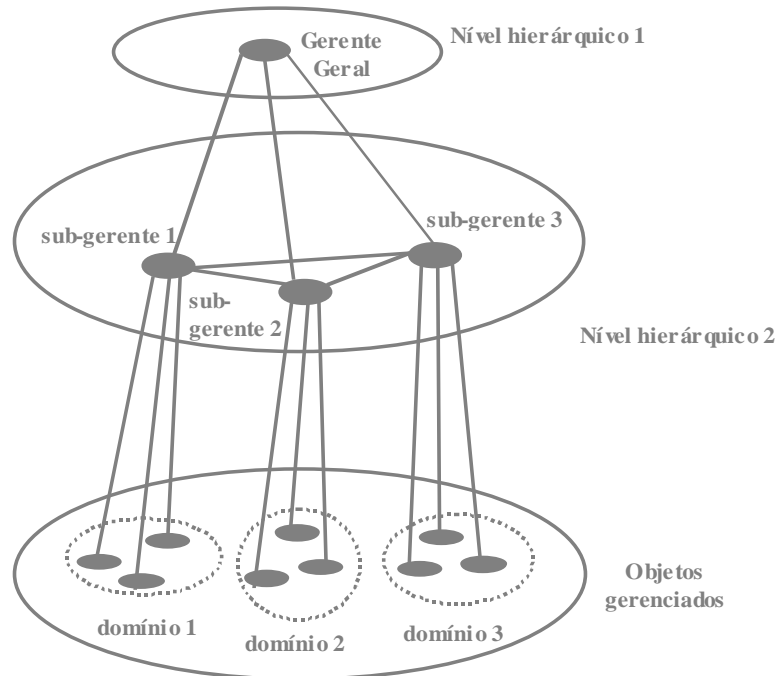


Figura 2.6: Arquitetura Hierárquica utilizando Objetos Gerenciados

Um Sistema de Gerenciamento Hierárquico apresenta as seguintes características:

Vantagens:

- Na Arquitetura de Gerenciamento Hierárquico, cada nível superior de Gerenciamento de Falhas apresenta um número decrescente de variáveis em relação ao nível inferior. Quanto maior o nível hierárquico, maior o conteúdo semântico das variáveis, quando comparada às demais arquiteturas de gerenciamento. Fica sob a responsabilidade dos sub-gerentes de níveis inferiores, a tarefa de gerenciar as variáveis de menor valor semântico.
- A possibilidade de delegar funções de gerenciamento aos sub-gerentes depende de cada sub-rede. Este fato permite um maior grau de independência dos sub-gerentes no Gerenciamento de Falhas da rede.
- Permite uma maior flexibilidade, no caso de expansão da rede, pois parte da configuração é realizada localmente em cada sub-gerente.
- Problemas localizados apenas em um domínio de gerenciamento (considerando-se apenas os casos em que não há propagação de falhas em mais de um domínio de gerenciamento) são resolvidos localmente pelo seu sub-gerente.
- Os níveis de gerenciamento hierarquicamente superiores atuam apenas em casos de falhas que ultrapassem mais de um domínio de gerenciamento. Esta característica facilita o monitoramento da rede nos níveis de gerenciamento superiores.
- O Sistema de Gerenciamento Hierárquico consome uma largura de banda menor quando comparado aos Sistemas de Gerenciamento Centralizado e Distribuído, pois apenas as variáveis de maior conteúdo semântico trafegam entre os domínios de gerenciamento.
- No caso de falha de comunicação entre o Gerente Geral e os sub-gerentes (por exemplo, pela perda de uma conexão remota), os sub-gerentes podem realizar momentaneamente (enquanto a conexão remota não é restabelecida) o gerenciamento local, independente dos demais níveis de gerenciamento.

- Os algoritmos de localização de falhas em cada domínio de gerenciamento podem ser mais simples, na medida que necessitam gerenciar um menor número de objetos gerenciáveis, principalmente quando comparado ao Gerenciamento Centralizado que utiliza um único algoritmo de localização de falhas para toda a rede.

Desvantagens:

- Quanto mais elevado o nível de gerenciamento, maior será o valor semântico das variáveis de gerenciamento. Este fato pode ser prejudicial em análises de falhas mais apuradas, pois pode-se estar realizando a análise em um determinado nível de gerenciamento e não ter acesso as variáveis locais (em MIBs locais) pertencentes a domínios de gerenciamento inferiores, que são importantes para uma maior precisão na localização do objeto falho.
- As MIBs, agentes e sub-gerentes distribuídos fisicamente pela rede, também podem ser a fonte de falhas e, portanto, necessitam de mecanismos de proteção (por exemplo: utilizar mecanismos de redundância) de tal forma a garantir alta disponibilidade.
- O fato de adicionar elementos redundantes (MIBs, agentes e sub-gerentes) pela rede poderá elevar demasiadamente o custo da rede, principalmente quando comparado à Arquitetura de Gerenciamento Centralizada (possui apenas um agente de rede).

3. Arquitetura de Gerenciamento Hierárquico Proposta

Este capítulo apresenta o Arquitetura de Gerenciamento Hierárquica Proposta que supõem-se apresentar um menor tempo para a localização de falhas, através do menor tempo de processamento do algoritmo *Divida e Conquiste*, quando comparado às Arquiteturas Centralizada e Distribuída.

A Arquitetura de Gerenciamento Hierárquico Proposta é apresentada na figura 3.1.

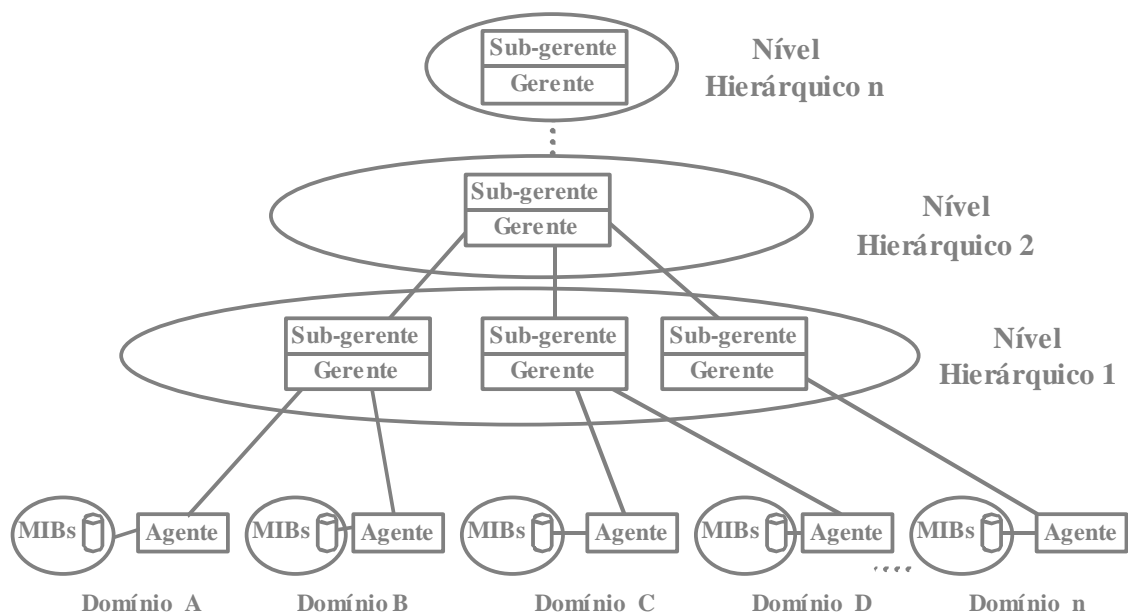


Figura 3.1: Arquitetura de Gerenciamento Hierárquico Proposta

Cada domínio de gerenciamento (sub-gerentes) se comunica apenas com o nível imediatamente superior. Nesta arquitetura não são trocadas informações entre os sub-gerentes do mesmo nível hierárquico.

Desta forma, cada domínio de gerenciamento inferior é mapeado em um único objeto no nível de gerenciamento imediatamente superior. Uma possível contribuição desta arquitetura poderá ser uma diminuição ainda maior na quantidade de variáveis que trafegam entre gerente e sub-

gerentes, bem como um aumento na disponibilidade total do sistema quando comparado com o Sistema de Gerenciamento Hierárquico apresentado no item 2.3.

Da mesma forma que nas Arquiteturas de Gerenciamento Centralizado, Distribuído e Hierárquico, as aplicações de Gerenciamento de Falhas e Configuração (presentes apenas nos sub-gerentes que comunicam com os agentes de rede) executam suas funções de forma coordenada quando da ocorrência de uma falha na rede.

Na figura 3.2 é apresentada a mesma arquitetura através do conceito por objetos:

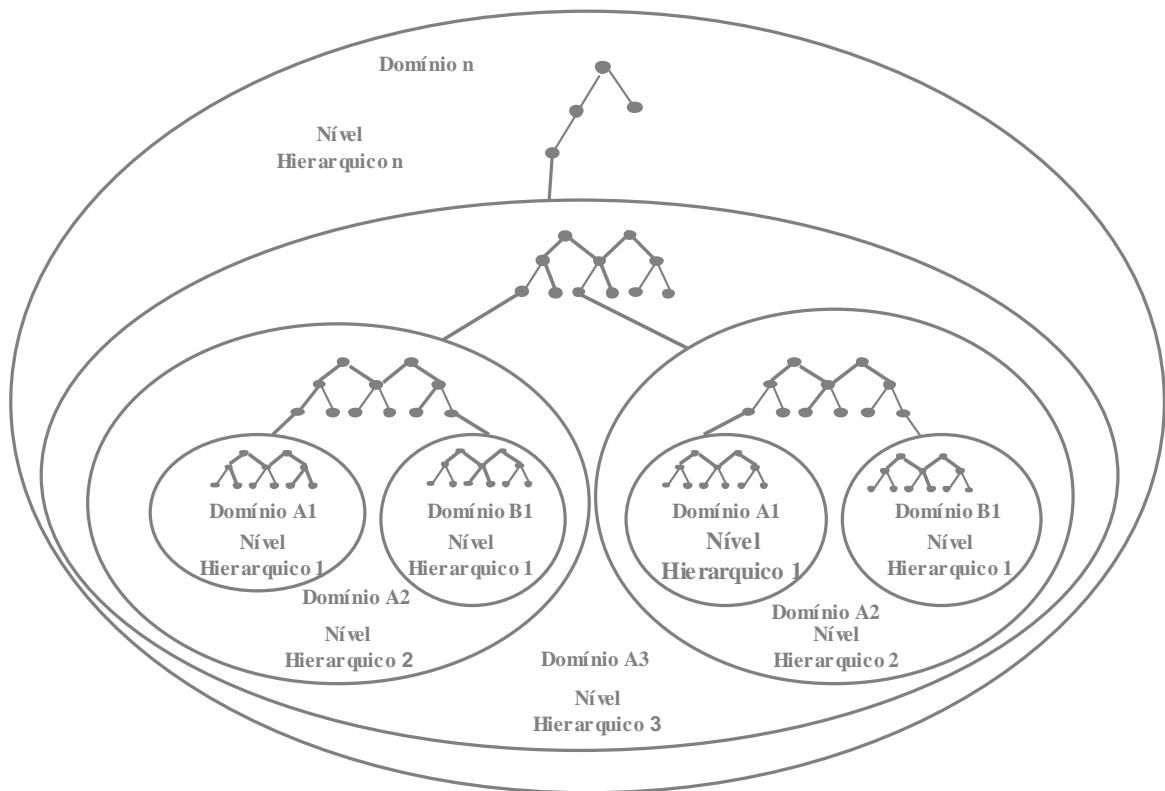


Figura 3.2: Relacionamento entre objetos na Arquitetura Hierárquica Proposta

Vantagens:

Apresenta as mesmas vantagens da Arquitetura de Gerenciamento Hierárquico citadas anteriormente, podendo adicionar:

- Quanto mais elevado o nível de gerenciamento, mais simples é o algoritmo de localização de falhas, pois cada domínio de gerenciamento é representado como um único objeto no nível de gerenciamento imediatamente superior.
- O processo de localização de falhas torna-se mais simples em comparação às demais arquiteturas de gerenciamento apresentadas, pois cada sub-gerente necessita apenas comunicar-se com o gerente de rede no nível de gerenciamento imediatamente superior, responsável pelo gerenciamento de seu domínio. Desta forma, não é necessário se comunicar com os sub-gerentes do mesmo nível hierárquico, como é realizado na Arquitetura de Gerenciamento Distribuída e Hierárquica.

Desvantagens:

Apresenta as mesmas desvantagens da Arquitetura de Gerenciamento Hierárquico citadas anteriormente, podendo adicionar-se:

- Como cada sub-gerente comunica-se apenas com um gerente de rede no nível de gerenciamento imediatamente superior, a perda da comunicação entre sub-gerente e gerente pode prejudicar o processo de localização de falha, já que os sub-gerentes pertencentes a um mesmo nível de gerenciamento não apresentam comunicação entre si.

4. Modelamento para Sistemas Distribuídos e Conceituação Básica de Falhas

Alguns conceitos fundamentais devem também ser apresentados devido a sua grande aplicabilidade dentro da área de Gerenciamento de Falhas.

Primeiro é abordado o conceito de Modelo de Sistema Distribuído. Em seguida é apresentada uma possível Classificação de Falhas seguida da definição dos conceitos de Confiabilidade, Disponibilidade e Cobertura de Falhas.

Na análise está sendo considerado o fenômeno da falha num âmbito sistêmico, no qual os componentes da rede são atômicos e, sua estrutura interna não é considerada. Desta forma, são analisados, neste capítulo, as falhas destes componentes independentemente de sua estrutura interna. Neste sentido, é apresentado a seguir a definição do conceito de Modelo de Sistema Distribuído a partir de seus componentes constituintes.

4.1. Modelo de Sistema Distribuído

Existem dois pontos de vista para um sistema distribuído: um definido pelos seus componentes físicos, e outro pelos seus componentes computacionais [JALOTE-96]. O primeiro é chamado de modelo físico do sistema, e o outro é denominado como modelo lógico. O ponto de vista computacional representa a visão do usuário em relação ao sistema. Nesta perspectiva são definidos os serviços, bem como a confiabilidade e a disponibilidade desejados. O modelo físico dá suporte ao modelo lógico.

O objetivo de um Sistema de Gerenciamento de Falhas é garantir as propriedades ou serviços do modelo lógico, mesmo na presença de falhas no modelo físico e, assim, permitir uma maior disponibilidade da rede.

4.1.1. Modelo Físico de um Sistema Distribuído

O modelo físico de um sistema distribuído consiste de computadores (frequentemente referidos como nós) que estão geograficamente em diferentes localidades, mas conectados através de uma rede de comunicação. A propriedade chave de um sistema distribuído é a separação geográfica e a natureza autônoma dos vários nós.

Em sistemas distribuídos, cada nó possui seu próprio relógio, sendo este utilizado no controle de execução de suas instruções. Portanto, um relógio em um nó não pode controlar diretamente as instruções executadas em um outro nó. Em contrapartida, um sistema paralelo pode existir um relógio global, que controla as instruções executadas em diferentes elementos de processamento distribuído pela rede.

Assim, pode-se definir um sistema distribuído formado por vários nós, onde cada nó é composto de um processador, algumas memórias voláteis que são inacessíveis por outros nós do sistema e, um relógio local, que controla a execução das instruções no processador. Em cada nó também existe uma interface de rede, através da qual o nó se conecta à rede de comunicação. Assim, dois nós em um sistema distribuído podem trocar mensagens pela rede de comunicação. Adicionalmente, assumi-se que cada nó apresente algum mecanismo de armazenamento não volátil, como por exemplo, um disco flexível [JALOTE-96]. Normalmente, cada nó pode utilizar-se de técnicas de tolerância à falha de forma a aumentar sua confiabilidade. Neste trabalho de pesquisa não é considerada a estrutura interna dos nós, sendo focado um sistema distribuído constituído pelos seus nós e pela rede de comunicação (rede).

A maneira pela qual os diferentes nós são conectados é chamada de topologia da rede. Várias topologias de rede são possíveis. As mais comuns são apresentadas na figura 4.1, caracterizando-se em redes ponto-a-ponto.

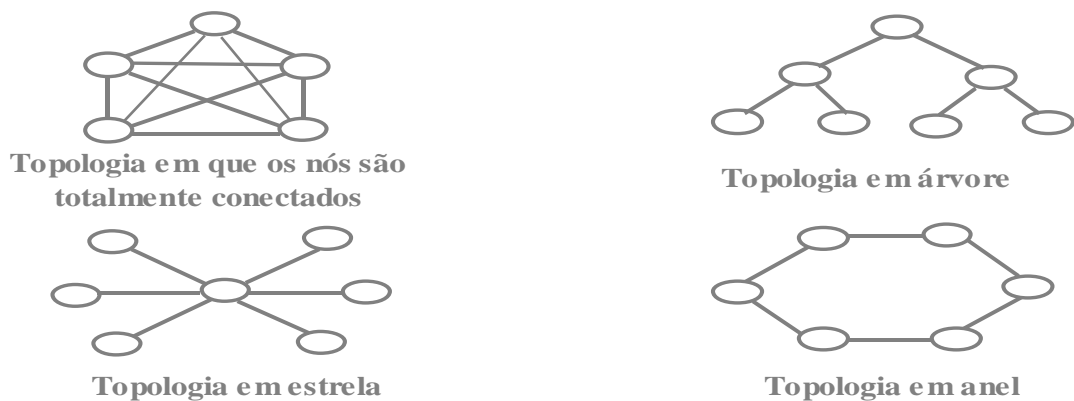


Figura 4.1: Modelos de Topologias de Rede

Enviar mensagens de um ponto a outro em uma rede requer um protocolo de comunicação. Os protocolos são necessários devido a natureza autônoma dos nós e a separação geográfica entre eles. Desde que os nós possam estar separados por uma grande distância, os dados enviados por uma conexão entre dois nós podem ser perdidos, devido a erros na transmissão e/ou a outros problemas. Desde que não compartilhem o mesmo relógio (*clock*), os nós podem trabalhar em diferentes velocidades. Muitos protocolos como TCP/IP, OSI, entre outros, foram desenvolvidos com o propósito de assegurar a confiabilidade da comunicação entre os nós em uma rede. Tipicamente, algumas partes do protocolo são executadas em *hardware*, pela interface de rede, sendo as demais partes realizadas por um sistema de *software*.

4.1.2. Modelo Lógico de um Sistema Distribuído

Um sistema distribuído pode ser observado a partir do ponto de vista de suas aplicações, sem se preocupar com seu sistema físico. Desta forma, pode-se também discutir o modelo de um sistema distribuído a partir do ponto de vista de seu processamento distribuído ou aplicação distribuída.

Para entender uma aplicação distribuída, deve-se entender o comportamento dos processos individualmente e, como estes interagem entre si. Um conjunto de processos concorrentes pode ser classificado em três categorias: independentes, competidores e cooperativos. Processos

concorrentes são ditos independentes, se os conjuntos de objetos acessados por eles são disjuntos. Visto que a atividade de cada processo pode ser completamente isolada de qualquer outro processo, este é conceitualmente igual ao caso de múltiplos processos fisicamente separados. Processos concorrentes são ditos competidores se eles compartilham recursos, mas não trocam informações entre si. Sistemas Operacionais Multiprocessamento normalmente suportam processos competidores. Os processos cooperativos trocam informações entre si, ou pelo uso compartilhado de dados, ou através da passagem de mensagens. Em um sistema distribuído é mais comum a utilização do método por passagem de mensagens.

Assume-se que os processos em análise são concorrentes, pois estes melhor representam um sistema distribuído.

No nível lógico, a estrutura da rede é tratada como totalmente conectada, se a rede física é conectada. Ou seja, se um nó é conectado a todos os outros nós, isto significa que uma mensagem pode ser enviada de um nó até qualquer outro. Isto é garantido pelos protocolos de comunicação que suportam a comunicação entre os diferentes nós da rede. Portanto, a rede, neste caso, pode ser tratada como totalmente conectada para o propósito de comunicação.

No nível lógico, considera-se um sistema distribuído como consistindo de um conjunto finito de canais que interligam os processos. Os canais representam a conexão lógica entre os processos. A estrutura da rede é tal que qualquer nó pode enviar mensagens para qualquer outro nó, em um nível lógico, desde que existam canais entre quaisquer dois processos. Pode-se dizer que existe um canal entre dois processos, se estes interagem entre si através de mensagens. Portanto, em um nível lógico, o sistema distribuído pode ser representado através de um grafo direcionado em que os vértices representam os processos do sistema.

A título de ilustração, considere dois processos representados pelos vértices P e Q e, que exista um arco direcionado interligando P a Q. Este arco representa o canal entre ambos, ou seja, representa que o processo P pode enviar mensagens para o processo Q.

Assume, nesta ilustração, que os canais são livres de erro e recebem as mensagens na ordem em que elas são enviadas. Isto é, P envia múltiplas mensagens até Q através do canal entre P e Q. As mensagens chegam até Q na mesma ordem em que elas foram enviadas por P. Entretanto, muitos protocolos de comunicação fornecem um serviço que satisfaz estas suposições. Desta forma, em um nível lógico, a comunicação é desenvolvida através dos serviços fornecidos pelos protocolos de comunicação.

Nota-se que a suposição de ordenação somente garante que a ordem das mensagens é preservada em um particular canal. Nada se pode falar sobre a ordem das mensagens que são recebidas por um processo a partir de diferentes canais que nele incidem (isto é, mensagens que são enviados por diferentes processos). Portanto, se diferentes processos enviam mensagens para um processo P, a ordem em que as mensagens são recebidas pelo processo P não podem ser especificadas. Os diferentes atrasos gerados em cada canal, durante a transmissão das mensagens, é a principal justificativa para o fato de que mesmo conhecendo-se o instante de envio das mensagens por diferentes processos para um único processo, ainda assim, não seja possível definir antecipadamente a ordem de recepção das mensagens.

No nível lógico, suposições são freqüentemente feitas sobre os limites de tempo no desempenho do sistema. A existência de limites de tempos em um sistema são normalmente necessários, por razões práticas. Um sistema é dito síncrono, quando trabalha corretamente e desenvolve sua função dentro de um limite de tempo finito e conhecido. Um sistema é dito assíncrono, caso contrário [JALOTE-96].

Um canal de comunicação síncrono é aquele em que o atraso máximo da mensagem é conhecido e limitado. A principal vantagem de um sistema síncrono é que a falha em um componente pode ser deduzida por uma falta de resposta dentro de um limite de tempo

predefinido. Portanto, se o sistema distribuído é síncrono, um esquema baseado em limite de tempo (*timeout*) pode ser utilizado para detectar uma falha em um nó, ou então, a perda da mensagem. Tal esquema não pode ser utilizado se o sistema distribuído é assíncrono, isto é, se o canal e o nó são assíncronos.

Note que as falhas no modelo físico causam falhas dos componentes do modelo lógico. Por exemplo, uma falha de um nó da rede física pode causar falhas de alguns nós (isto é, processos) da rede lógica. Similarmente, uma falha em uma linha de comunicação da rede física pode causar alguma falha no respectivo canal lógico.

O algoritmo de localização de falhas e a Arquitetura de Gerenciamento Hierárquico Proposto neste trabalho de mestrado serão exercidos sobre um grafo de dependência representado através do modelo lógico de um sistema distribuído síncrono.

4.2. Classificação das Falhas

Do ponto de vista de gerenciamento de rede, uma falha é uma condição anormal da rede que necessita ser corrigida através de um Sistema de Gerenciamento de Falhas.

Neste sentido dois conceitos são fundamentais: Fonte Primária de Falha e Fonte Secundária de Falha.

Fonte primária da falha no sistema: elemento que está no estado falha. Ele pode também ser chamado de causa raiz da falha.

Fonte secundária da falha no sistema: elemento que apesar de não estar com falha, devido a uma dependência de recursos de outros elementos da rede, leva o sistema a um estado falho.

Um sistema distribuído pode ser definido a partir do ponto de vista da rede física ou da rede lógica. Do ponto de vista do usuário, a rede lógica é a mais relevante.

As falhas ocorrem normalmente na rede física. O foco principal de um Sistema de Gerenciamento de Falhas em sistemas distribuídos são as falhas em seus componentes físicos, particularmente em componentes como nós e redes de comunicação.

Uma possível classificação das falhas em sistemas distribuídos é baseada no comportamento de seus componentes quando falham. Esta classificação de falhas é dividida em quatro categorias: desastre, omissão, temporização e Bizantina [JALOTE-96].

Falha por desastre:

É a falha que provoca a parada ou a perda do estado interno do componente. Com este tipo de falha, o componente nunca é submetido a qualquer tipo de transição.

Falha por omissão:

É uma falha que provoca um componente a não responder a algumas entradas.

Falha por temporização:

É uma falha que provoca um componente a responder em um tempo muito antecipado ou atrasado. Ela é também chamada algumas vezes como falha de desempenho.

Falha Bizantina:

É uma falha arbitrária que provoca um comportamento totalmente arbitrário do componente durante a falha.

Estas falhas formam uma hierarquia, no qual a falha por desastre é o tipo mais simples e restritivo (bem conhecido), enquanto que a Falha Bizantina é a mais complexa e menos restritiva. Esta categoria de falhas apresenta uma relação de inclusão conforme é apresentado na figura 4.3 [JALOTE-96].

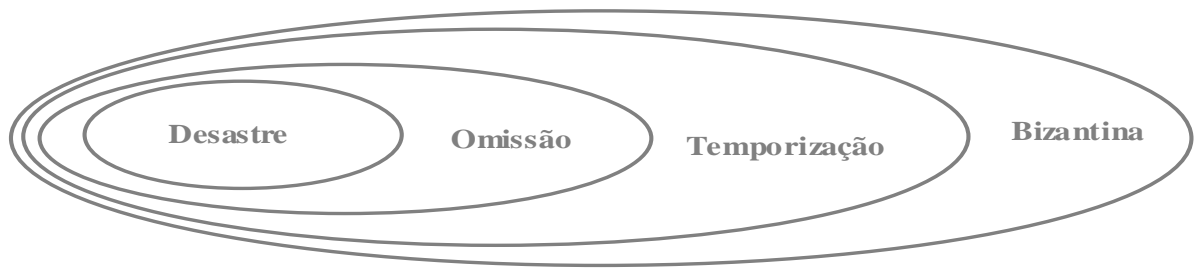


Figura 4.2: Classificação de Falhas em Sistemas Distribuídos

Existem outras classes que podem ser adicionadas, como por exemplo, a falha por computação incorreta. Evidentemente, esta classe é um subconjunto da falha Bizantina, mas ela é diferente das outras classes definidas na figura 4.2. Neste tipo de falha, um componente não apresenta qualquer falha por temporização. Simplesmente, o componente produz uma saída incorreta em resposta a uma dada entrada [JALOTE-96].

Para um processador assume-se que a falha típica pode ser desastre (quando o processador para a sua execução) ou Bizantina (quando nenhuma suposição pode ser feita sobre o comportamento da falha).

Em uma rede de comunicação considera-se todos os tipos de falhas definidos. Por exemplo, pode possuir uma falha por desastre (quando a rede não recebe ou envia qualquer mensagem), por computação incorreta (quando as mensagens são corrompidas), por omissão (quando mensagens são perdidas), por temporização (quando a rede entrega as mensagens após um tempo excessivamente longo) ou Bizantina (quando a rede apresenta um comportamento totalmente arbitrário).

Em um relógio, a falha normalmente considerada é por temporização, quando a sua execução é muito lenta ou rápida. No entanto, a falha Bizantina também pode ser considerada, no qual o comportamento do relógio (ou seja, o tempo que ele representa) é totalmente arbitrário. Se o tempo que o relógio registra é sempre o mesmo, pode ser considerada uma falha por omissão.

Um meio de armazenamento pode apresentar falha por desastre, na qual o armazenamento é totalmente inacessível para leitura ou escrita. Ele também pode apresentar falha por temporização (quando os dados chegam atrasados) ou falha por omissão (quando alguns dados são inacessíveis). Adicionalmente, pode também corromper os dados, considerando-se neste caso, uma falha por computação incorreta.

Componentes de *software* podem apresentar a maioria das falhas anteriormente apresentadas. Entretanto, a falha de maior interesse em componentes de *software* é a falha por computação incorreta, na qual, o *software* desenvolve um processamento incorreto. Neste caso, também pode ser chamada de falha de projeto de *software* [JALOTE-96].

Neste trabalho de pesquisa está se considerando as falhas de objetos, sem entrar no mérito de sua classificação.

4.3. Confiabilidade

A confiabilidade de um sistema/componente representa a probabilidade do sistema/componente funcionar de acordo com sua especificação num determinado período de tempo, dado que ele estava funcionando corretamente no instante inicial [CAMARGO-84, SIEGL-95 e HIGH AVAILABILITY: AN OVERVIEW-98].

$R(t)$ = probabilidade do sistema estar em operação correta em um período de tempo t .

Se a taxa de falhas (λ) de um sistema ou componente é constante no tempo, pode-se deduzir a função confiabilidade dada a seguir:

$$R(t) = e^{-\lambda t}$$

Em sistemas eletrônicos e programáveis a taxa de falhas dos componentes é considerada constante ao longo de sua vida útil [CAMARGO-94].

A partir desta definição, pode-se contabilizar o Tempo Médio para Falha, chamado de MTTF (*Mean Time to Failure*), ou também, expectativa de vida do componente/sistema

[SIEGL-95]. O MTTF é o tempo esperado que um componente/sistema irá operar antes que a primeira falha ocorra [CAMARGO-94], sendo uma medida comumente utilizada para especificar a confiabilidade de um sistema, dada por:

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

Considerando a função confiabilidade exponencial, apresentada anteriormente, têm-se:

$$\text{MTTF} = 1/\lambda$$

A expressão acima fornece o MTTF de um sistema simples.

Normalmente, um sistema pode ser considerado como uma combinação de múltiplos componentes independentes. Neste caso, o sistema falha se qualquer um dos componentes falhar. A confiabilidade total do sistema será dada por:

$$R(t) = \prod_{i=1}^n R_i(t), \text{ sendo } n \text{ o número de componentes.}$$

Considerando-se a taxa de falha constante de cada componente, tem-se que a taxa de falha do sistema é dada por:

$$\lambda = \sum_{i=1}^n \lambda_i .$$

Portanto, o MTTF do sistema série será dado por:

$$\text{MTTF}_{\text{série}} = 1/\sum_{i=1}^n \lambda_i$$

Assim, observa-se que o MTTF de um sistema série é menor do que o MTTF de qualquer um de seus componentes. Este fato já era esperado, pois a falha de qualquer componente do sistema em série causará a falha do sistema como um todo.

Um sistema também pode apresentar componentes redundantes, no qual os seus componentes são conectados em paralelo. Nesta organização, o sistema falha somente quando todos os componentes falharem.

Isto implica que a confiabilidade de um sistema paralelo é maior do que a confiabilidade de seus componentes. Assumindo-se uma distribuição exponencial para a confiabilidade dos componentes, o MTTF do sistema será dado por:

$$MTTF_{\text{paralelo}} = (1/\lambda) \sum_{i=1}^n (1/i) \approx \ln(n)/\lambda$$

Outro conceito importante é o MTTR (*Mean Time to Repair*) que é o tempo médio requerido para reparar um componente/sistema [CAMARGO-94], ou seja:

$$MTTR = (\sum_{i=1}^n t_i) / n$$

sendo, n o número de defeitos inseridos, e t_i é o tempo de reparo.

Normalmente o MTTR é especificado em termos de taxa de reparo (μ) que é o número médio de reparos que ocorrem num período de tempo, ou seja: $MTTR = 1/\mu$

4.4. Disponibilidade

A disponibilidade de um sistema/componente é definida como a probabilidade dele estar funcionando corretamente e, disponível para realizar suas funções em um determinado instante de tempo [CAMARGO-94 e HIGH AVAILABILITY: AN OVERVIEW-98].

Na ausência de reparo ou substituição, a disponibilidade é simplesmente igual à confiabilidade. De qualquer modo, com a utilização do processo de manutenção, a vida de um sistema/componente pode ser vista como uma seqüência de variáveis randômicas independentes, sendo que cada uma representa a vida de um sistema/componente até a ocorrência da próxima falha (e seu correspondente reparo).

O conceito de Disponibilidade Assintótica é mais utilizado, sendo calculado a partir do valor da Disponibilidade quando o limite do tempo tende a infinito. Considerando-se MTTF e o MTTR, tem-se que a Disponibilidade Assintótica (α) pode ser calculada como:

$$\alpha = \text{MTTF}/(\text{MTTF} + \text{MTTR})$$

4.5. Cobertura de Falhas

A cobertura de falhas num sistema distribuído pode ter um grande impacto na sua disponibilidade [CAMARGO-94].

Existem muitos tipos de cobertura, podendo classificá-las em cobertura de:

- Detecção da falha.
- Localização da falha.
- Contenção da falha.
- Recuperação da falha.

A cobertura da detecção/localização é a medida da habilidade de um determinado sistema em detectar/localizar um componente falho. Devido à existência de fontes primárias e secundárias de falhas, conforme exposto no item 4.2, o conceito de fator de cobertura torna-se fundamental para o aumento da disponibilidade final do sistema.

A cobertura de detecção/localização da falha num sistema distribuído influi diretamente na eficiência do algoritmo de detecção de falha. Este aspecto foi considerado no algoritmo *Divida e Conquiste Proposto*, apresentado no capítulo 6 desta dissertação de mestrado.

5. Gerenciamento de Falhas

O Gerenciamento de Falhas consiste no processo de prevenir e localizar problemas (por exemplo: serviço com nível inadequado) ou falhas (por exemplo: perda de serviço) com a máxima rapidez e precisão, tendo como objetivo principal garantir a maior disponibilidade dos serviços prestados em uma rede de comunicação. Este processo envolve os seguintes passos [SIEGL-94, KATZELA-93, NETWORK MANAGMENT FUNCTIONS-98, STALLINGS-93 e RAMAN-98]:

- Detectar a falha
- Determinar sua localização com máxima exatidão.
- Isolar a falha da parte restante da rede, de tal forma que a parte em bom funcionamento execute suas funções sem apresentar interferência da parte da rede com falha.
- Reconfigurar ou modificar a rede de tal forma a minimizar o impacto devido a falha, excluindo-se assim a interferência do componente falho no funcionamento da rede.
- Reparar ou substituir o componente falho, restabelecendo-se assim o serviço de rede, de forma automática ou manual.

5.1. Tipos de Alarmes e Níveis de Severidade

Alarme é um som ou mensagem utilizada para chamar a atenção do administrador da rede sobre uma ocorrência anormal [LEINWAND-93], ou então, são tipos específicos de notificações sobre falhas detectadas ou condições anormais que são emitidos pelos objetos gerenciados aos gerentes de rede [CARVALHO-92].

Da mesma maneira, são estabelecidos critérios para determinar o grau de mau funcionamento da rede em decorrência da falha, denominado de nível de severidade do alarme. Este nível de severidade é avaliado pelo nível de degradação da qualidade de serviço oferecido ao usuário de rede, ou pelo tipo do estado de um objeto gerenciado.

Normalmente os alarmes podem ser classificados de acordo com os seguintes tipos [STALLINGS-93 e CARVALHO-92]:

- Alarme de Comunicação: utilizado para informar quando um objeto gerenciado detecta um erro de comunicação. Normalmente está relacionado a procedimentos e processos utilizados na transferência de informações através da rede gerenciada.
- Alarme de Qualidade de Serviço: utilizado para informar quanto à degradação da qualidade de serviço especificada para a rede gerenciada.
- Alarme de Processamento: utilizado para informar a ocorrência de falhas nos equipamentos da rede.
- Alarme Ambiental: utilizado para informar a ocorrência de problemas relacionados às condições ambientais em que se encontra o equipamento gerenciado.

Pode-se classificar também os seguintes níveis de severidade de um alarme:

- Limpo: indica que um alarme ou conjunto de alarmes foi removido (*cleared*). Este conjunto de alarmes normalmente refere-se a alarmes do mesmo tipo, causa e problema associados a um objeto gerenciado.
- Indeterminado: quando não é possível estabelecer como as condições de funcionamento de um objeto gerenciado foram afetadas.
- Crítico: indica uma condição crítica de funcionamento, que exige uma ação corretiva imediata da rede. Por exemplo, quando um recurso definido por um objeto gerenciado não está em serviço (equipamento para de funcionar) e seu recurso é requerido.
- Maior: quando as condições de funcionamento da rede são afetadas, sendo necessária uma ação corretiva urgente. Por exemplo, quando uma severa degradação na capacidade de um objeto ocorre, e necessita-se restaurar toda a sua capacidade.
- Menor: indica a existência de falhas comprometendo o funcionamento adequado da rede, sendo necessário a utilização de ações corretivas para prevenir a ocorrência de falhas mais sérias. Neste caso, não é observado ainda a degradação de um objeto gerenciado.

Uma abordagem muito importante dentro da área de Gerenciamento de Falhas é o gerenciamento proativo, ou seja, as atividades de gerenciamento de redes devem atuar de

forma, não só reativa, mas preventiva, detectando as falhas antes que efeitos significativos sejam percebidos pelos usuários da rede. As atividades a seguir permitem a emissão de notificações de alarmes para o gerente de rede, que pode disparar ações corretivas prevenindo situações mais críticas. Estas atividades são:

- Coleta constante de informações do estado da rede, através da monitoração das taxas de: erro, perda e atraso de mensagens, entre outras.
- Gerar alarmes e tendências relacionadas com:
 - ◇ Taxas de erro, perda e atraso crescentes, excedendo limites predeterminados.
 - ◇ Evolução dos níveis de severidade dos alarmes.
- Banco de dados com histórico de equipamentos (tempo médio de funcionamento, tipo de falha mais freqüente etc).

Este trabalho de pesquisa está focado nos dois primeiros passos de Gerenciamento de Falhas (detecção e localização da falha com máxima exatidão), considerando-se que um objeto de rede pode estar no estado falho, ou então, em vias de entrar neste estado (gerenciamento proativo). Neste último caso, os alarmes estarão relacionados com prováveis objetos com maior probabilidade de virem a falhar.

5.2. Conceitos sobre Gerenciamento Hierárquico

O Gerenciamento Hierárquico divide a rede em domínios de gerenciamento, como é mostrado na figura 5.1:



Figura 5.1: Definição de Domínios de Gerenciamento

Cada um dos domínios de gerenciamento são definidos como [RECOMMENDATION M.3010-92]:

Logicamente autônomos, e

- Disjuntos

Da mesma forma pode-se caracterizar o domínio de um determinado alarme como mostra a figura 5.2:

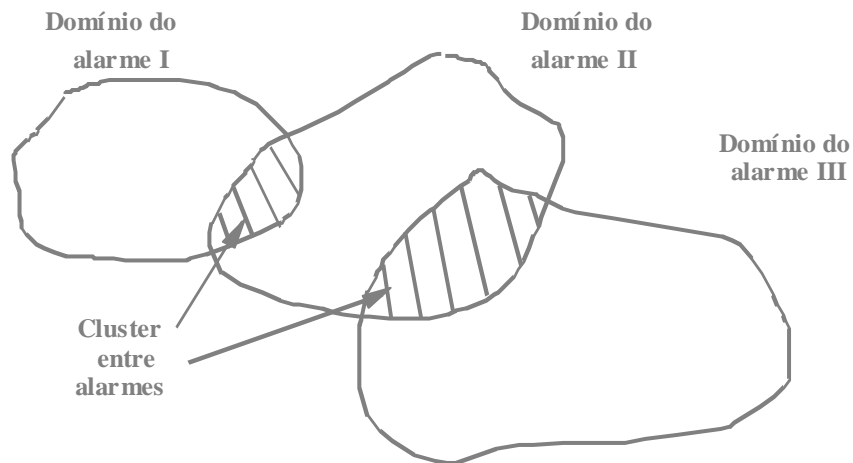


Figura 5.2: Definição de *Cluster* de Domínios de Alarmes

Um domínio de alarme caracteriza um alarme, e este depende de dois fatores:

- Semântica do alarme, e
- Topologia da Rede

O domínio de um alarme é definido como um conjunto de objetos que podem ter causado o alarme, isto é, podem ser a causa da falha (fonte primária ou causa raiz).

Para se encontrar o domínio de um alarme, deve-se associar alarmes a todos os objetos que estão relacionados ao mau funcionamento.

Um *cluster* entre alarmes é definido por [KATZELA-93] como uma região de interseção entre dois ou mais domínios de alarmes, em conjunto com os respectivos objetos contidos nesta região. Um único objeto falho normalmente provoca a geração de mais de um alarme e, desta forma pode gerar um ou mais *clusters*.

Existem duas situações para a identificação de um objeto falho pelo Sistema de Gerenciamento Hierárquico:

- Quando existe o *cluster* de alarmes, o Sistema de Gerenciamento de Falhas define qual o domínio de alarmes com maior probabilidade de apresentar um objeto falho (fonte primária de falha).
- Quando não existe o *cluster* de alarmes, o sub-gerente responsável pelo domínio de gerenciamento que sinalizou o alarme, identifica qual o objeto mais provável de ser a fonte primária de falha.

5.3. Grafos de Dependência

Normalmente, uma rede pode consistir de um sistema complexo composto por equipamentos de vários fabricantes e serviços diferenciados, nos quais a informação gerenciável pode estar lógica e fisicamente distribuída pela rede [KEHL-93].

Assim, pode-se modelar um sistema de comunicação em diferentes níveis de abstração, sendo que em cada um deles, a informação é apresentada com um grau diferente em significado.

Uma forma de modelar os níveis de abstração é através da utilização de grafos de dependência. Estes podem ser de extrema utilidade no processo de correlação de alarmes e identificação de falhas [JAKOBSON-93 e, MEIRA e NOGUEIRA-97].

A aplicabilidade de grafos de dependência é geral, sendo utilizado até na análise e testes de falhas em estruturas de *software* [TEWARY-98].

De uma forma geral, os esquemas de modelamento através de grafos de dependência e objetos apresentam as seguintes vantagens, de acordo com [KATZELA-93 e TEWARY-98]:

Generalidade:

Permite que diferentes tipos de sistemas de comunicação possam ser modelados facilmente.

Flexibilidade:

Permite modificar de forma fácil a representação de um sistema, pois a adição ou exclusão de um objeto de rede representa uma simples modificação no modelamento da estrutura da rede.

Simplicidade:

Permite modelar facilmente um sistema, e ao mesmo tempo representar a complexidade necessária de sistemas reais.

Similaridade:

Permite que sistemas similares tenham um modelo similar de representação.

Esta abordagem é mais ampla dentro da área de Gerenciamento de Falhas em Sistemas Distribuídos. Esta visão do problema difere daquela baseada em abordagens específicas, como por exemplo, através da utilização de uma base de conhecimento (*knowledge-base*) [LEMONS-97, SUTTER-96, CRONK-98, MANSFIELD-93 e WEINER-95].

Uma abordagem bastante interessante é utilizada em [MEIRA-97] e, procura classificar os diversos métodos e algoritmos de localização de falhas em: Correlação Baseada em Regras, Lógica Difusa, Redes Bayesianas, Raciocínio Baseado em Modelos, Quadro-Negro, Filtragem, “Event Forwarding Discriminator”- EDF, Raciocínio Baseado em Casos, Correlação por Codificação, Localização Explícita, Correlação por Votação, Correlação “Proativa”, Correlação Distribuída, Redes Neurais Artificiais e Diagnóstico por Comparação de Resultados de Teste. De acordo com esta classificação, este trabalho de dissertação pode ser enquadrado dentro da classe de Correlação Distribuída, apesar de também utilizar os conceitos de Redes Bayesianas, que também trata a falha como um fenômeno probabilístico.

A figura 5.3 ilustra um exemplo que utiliza o conceito de orientação a objetos e grafos de dependência utilizados no modelamento de uma rede:

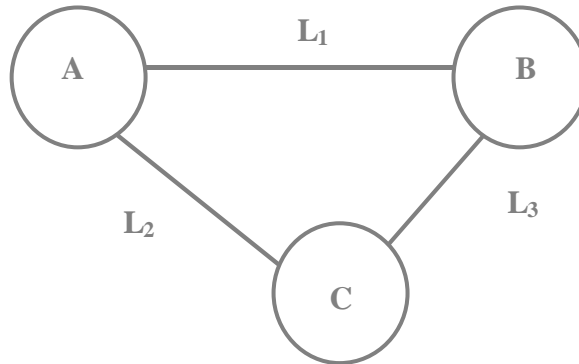


Figura 5.3: Exemplo de um Modelo de Rede

As localidades A, B e C representam três sub-redes em locais geograficamente distintos, e interconectados através das conexões L_1 , L_2 e L_3 [BOULOUTAS-91].

Assume-se que os domínios de gerenciamento sejam definidos pelos domínios geográficos de suas sub-redes.

Supor que existam os objetos A, B e C, que monitoram os estados (falho/não falho) das localidades A, B e C; e os objetos L_1 , L_2 e L_3 , que monitoram os estados (falho/não falho) das conexões L_1 , L_2 e L_3 .

Para simplificar a análise, assume-se que a localidade A necessite dos serviços fornecidos pelas localidades B e C e, por consequência, também dependa da disponibilidade das conexões L_1 e L_2 .

A localidade C depende dos serviços fornecidos pela localidade B e, por consequência, depende da disponibilidade da conexão L_3 e da disponibilidade da conexão L_2 , pois necessita enviar os serviços solicitados pela localidade A.

A localidade B necessita apenas da disponibilidade das conexões L_1 e L_3 para enviar os serviços solicitados pelas localidades A e C.

Assim, pode-se obter o seguinte grafo de dependência utilizando o conceito de objetos, conforme mostra a figura 5.4:

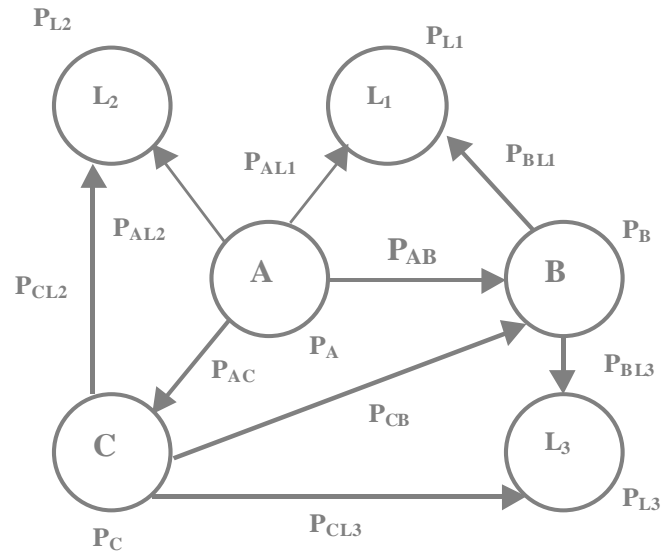


Figura 5.4: Grafo de Dependência para o Modelo de Rede

sendo que:

p_i = probabilidade incondicional do objeto i falhar independentemente de qualquer outro objeto.

p_{ij} = probabilidade condicional do objeto i falhar dado que o objeto j falhou.

5.4. Processo de Localização de Falhas em Sistemas Distribuídos

O processo de localização de falhas em sistemas distribuídos é composto basicamente por três etapas [BOULOUTAS-96]:

- Correlação de alarmes
- Identificação de falhas
- Estágio de teste

As duas primeiras etapas são melhor identificadas como processo de localização de falhas. Nelas são realizadas a correlação de alarmes e propostas as hipóteses de falhas, ou seja, sua identificação.

No estágio de teste cada hipótese de falha identificada é testada, com a finalidade de aumentar a precisão do processo de localização das falhas.

Assim, torna-se de grande interesse novas propostas de Gerenciamento de Falhas que procurem minimizar o tempo de localização de falhas

Pode-se definir o tempo de localização de falhas como:

$$\begin{array}{r} \text{Tempo para a} \\ \text{localização de} \\ \text{falhas} \end{array} = \begin{array}{r} \text{Tempo para propor} \\ \text{hipótese de falhas} \end{array} + \begin{array}{r} \text{Tempo para testar} \\ \text{as hipóteses} \\ \text{propostas} \end{array}$$

O tempo para propor hipóteses de falhas pode ser definido como:

$$\begin{array}{r} \text{Tempo para} \\ \text{propor hipóteses} \\ \text{de falhas} \end{array} = \begin{array}{r} \text{Tempo de complexidade do} \\ \text{algoritmo adotado para a} \\ \text{identificação de falhas} \end{array}$$

Portanto, desejam-se arquiteturas de gerenciamento e algoritmos de localização de falhas que procurem:

1. Minimizar o tempo de execução do algoritmo de identificação de falhas.
2. Maximizar a precisão do processo de hipóteses, ou seja, minimizar o número de testes a serem desenvolvidos.

Desta forma, é desejável que as arquiteturas de gerenciamento e os algoritmos de localização de falhas apresentem:

- Simplicidade;
- Baixa complexidade computacional;
- Resposta rápida em função da ocorrência de uma falha;
- Facilidade de implementação em um ambiente distribuído;
- Eficiência em manipular diferentes cenários de falhas;
- Robustez quanto a informações incompletas, redundantes e contraditórias.

5.5. Modelos de Diagnóstico e Localização de Falhas

Dado que um elemento falhou, o principal objetivo de um Sistema de Gerenciamento de Falhas é detectar qual é o elemento falho. Este fato pode implicar que outros componentes tenham que desenvolver outras atividades além das suas atividades normais, ou seja, a falha de um elemento tem que ser detectada e diagnosticada por outros elementos do sistema.

Algumas aplicações e métodos em sistemas distribuídos assumem que quando um nó do sistema falha, outros elementos do mesmo sistema devam encontrar a falha dentro de um tempo finito através de um algoritmo ou método particular e, como consequência, acionar uma ação específica para a correção da falha. O princípio para detecção de falhas pode ser satisfeito através da utilização de um método de força bruta que se baseia em tempo limite (*timeout*) em cada nó do sistema. Entretanto, em um grande sistema, esta abordagem pode implicar numa grande diminuição da disponibilidade.

O objetivo básico de um diagnóstico de falha é identificar todos os elementos no sistema com falha. Evidentemente, nem sempre é possível alcançar tal objetivo. Num caso extremo, quando todas as unidades em um sistema estão em falha, nenhuma unidade poderá desenvolver o diagnóstico e, portanto, alcançar o objetivo final de localizar a fonte primária de falha.

São apresentados, a seguir, alguns modelos básicos de diagnóstico de falha a nível de sistema.

O primeiro modelo foi introduzido em 1967 por *Preparata, Metze e Chien* [JALOTE-96], sendo conhecido como modelo *PMC*. Em seguida são apresentados alguns algoritmos de localização de falha, como *Pesquisa Exhaustiva*, *Pesquisa Gulosa* e *Divida e Conquiste*.

5.5.1. Modelo PMC

No modelo PMC, um sistema denominado como S é decomposto em n unidades, não necessariamente idênticas, denotadas pelo conjunto $U = \{ U_1, U_2, \dots, U_n \}$. Cada unidade é uma parte bem conhecida do sistema, não podendo ser decomposta novamente para o propósito de diagnóstico. Cada uma das unidades ou está trabalhando corretamente, ou está em falha. Cada unidade do sistema tem que ter capacidade suficiente para testar as outras unidades e, determinar se elas estão em falha, ou não.

O diagnóstico do sistema depende da habilidade de cada unidade testar o estado das outras unidades. Um teste no sistema envolve a aplicação controlada de alguns estímulos e a observação das correspondentes respostas. Os testes podem ser *softwares* de diagnósticos com entradas fixas e saídas bem conhecidas. Também é assumido que a unidade não falha sempre informa o estado das unidades que ele testa corretamente, enquanto as unidades falhas podem retornar resultados incorretos dos testes por eles conduzidos. A saída de um teste é simples, fornecendo os valores: 1 (falho) ou 0 (não falho). O teste é assumido ser completo, isto é, se o teste é aplicado por um nó não falho, então ele sempre retorna o estado do nó testado e, o resultado do teste reflete a condição da unidade testada.

O objetivo principal do diagnóstico é que algumas unidades do sistema testem outras unidades, de tal forma que, unindo os resultados dos testes, obtenha o estado completo do sistema.

No modelo PMC, a cada unidade pertencente a U é atribuído um subconjunto particular de U para ser testado (assume-se que nenhuma unidade testa si mesma). O conjunto completo de teste é chamado de conexão de designação, sendo representado por um grafo $G = (U, E)$. Neste grafo cada nó representa uma unidade e, cada arco representa uma conexão sendo testada. Um arco (u_i, u_j) pertence a G , se e somente se, o nó u_i testa o nó u_j .

Um resultado a_{ij} é associado a cada teste (u_i, u_j) , onde a_{ij} representa o resultado do teste. Se a unidade que realiza o teste u_i é não falha, então a_{ij} apresenta o valor 0 se u_j é não falha, e

1 se u_j é falha. Se u_i é falha, então o resultado do teste não é confiável e a_{ij} pode assumir qualquer valor, indiferente do estado de u_j . O conjunto de resultados do teste de um sistema S é chamado de síndrome de S .

Como exemplo, considere o modelo do sistema apresentado na figura 5.5. O sistema consiste de 5 unidades, e o teste é realizado como apresentado na figura:

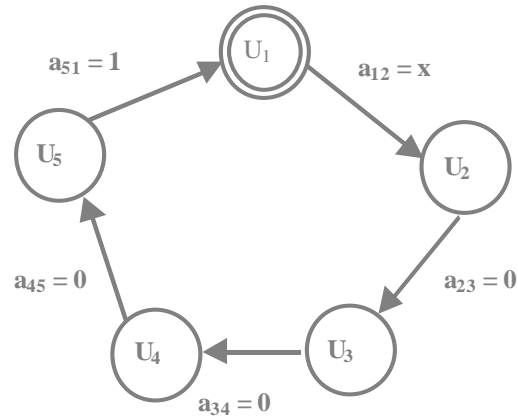


Figura 5.5: Teste de um sistema consistindo de 5 Unidades e utilizando o modelo PMC

A unidade U_1 testa a unidade U_2 , a unidade U_2 testa a unidade U_3 , e assim por diante.

A síndrome do sistema é um vetor de 5 bits representado da seguinte forma:

$$(a_{12}, a_{23}, a_{34}, a_{45}, a_{51})$$

Supondo que somente a unidade U_1 é falha, então a síndrome poderia ter o seguinte formato:

$$(x, 0, 0, 0, 1)$$

Portanto, U_5 identifica corretamente a unidade U_1 como falha, e as outras, exceto U_1 , obtêm o correto estado do seu teste. O valor de a_{12} pode ser tanto 1 ou 0, visto que U_1 é falha. Neste caso, o seu um valor é representado por um x .

Neste modelo, cada unidade conhece somente o resultado de seu teste. Para o diagnóstico, a unidade falha somente pode ser identificada a partir do resultado global do teste. No modelo PMC, a síndrome é analisada por um supervisor central, sendo este um processador ultra-confiável, que apresenta mecanismos de tolerância a falhas. Este supervisor diagnostica o estado do sistema, isto é, rotula cada unidade como falha ou não falha. Note que o problema do diagnóstico pelo supervisor torna-se muito difícil, a partir do momento em que o resultado de um nó possa não ser verdade, se ele foi declarado como falho por um outro nó.

O modelo PMC é a primeira proposição para diagnóstico de falha à nível de sistema. Desde então, um grande número de modelos, abordagens e resultados teóricos vem sendo apresentados. Com o crescimento dos sistemas computacionais, os resultados de diagnóstico à nível de sistema vem tornando-se cada vez mais necessários, principalmente quanto a mecanismos e Arquiteturas de Gerenciamento de Falhas que possam contribuir no acréscimo da disponibilidade dos serviços prestados por estes mesmos sistemas.

5.5.2. Algoritmos de Localização de Falhas

Os algoritmos de localização de falhas [SIEGL-94] utilizados em gerenciamento assumem que, em um dado momento, um número de alarmes tenham sido observados. O objetivo destes algoritmos é encontrar a melhor explicação dos alarmes recebidos.

Se dois ou mais alarmes compartilham uma interseção (isto é, pertencem ao mesmo *cluster* de alarmes), os alarmes podem ser examinados juntos, pois é muito provável que sejam causados pelo mesmo conjunto de falhas. Mas, se os alarmes recebidos criam mais de um *cluster*, então para cada *cluster* deve-se encontrar o conjunto de falhas que melhor os expliquem. A união de todos os conjuntos (um conjunto para cada *cluster*) fornecerá a melhor explicação dos alarmes recebidos.

Suposições:

1. Supor que o *cluster* é formado por um conjunto S associado a N objetos.

2. Para cada objeto $e_i \in S$, existem as probabilidades p_i e p_{ij} , $\forall j$, tal que $e_j \in S$, e p_i e p_{ij} são assumidos serem conhecidos.

Sendo, p_i a probabilidade de falha do objeto e_i independentemente de qualquer outro objeto do *cluster* e, $p_{ij} = P_r(e_i \text{ falhar/dado que } e_j \text{ falhou})$, a probabilidade condicional de e_i falhar como resultado da falha de outro objeto e_j contido no *cluster*.

Objetivo dos algoritmos:

O objetivo dos algoritmos de localização de falha é encontrar a melhor explicação dos alarmes recebidos. Para isso, eles pressupõem que a melhor explicação para um alarme recebido seja um subconjunto de S que tenha as seguintes propriedades:

- Os objetos no subconjunto explicam todos os alarmes do *cluster*, e
- Tem a máxima probabilidade de que pelo menos um dos objetos seja uma falha primária.

5.5.2.1. Algoritmo Pesquisa Exaustiva

Este algoritmo apresenta a abordagem direta para encontrar a melhor explicação dos alarmes recebidos.

O algoritmo de *Pesquisa Exaustiva* realiza os seguintes passos:

1. Forma todos os possíveis subconjuntos do conjunto S (conjunto formado por todos os objetos de monitoramento que recebem um alarme).
2. Seleciona o subconjunto S_i de S que explica todos os alarmes recebidos. Se existir mais de um subconjunto, então seleciona aquele que exibe a máxima probabilidade de que pelo menos um objeto seja uma falha primária.

Os dois passos anteriores garantem encontrar a melhor explicação para os alarmes recebidos. Se os alarmes recebidos associados a um *cluster* possuem N objetos, então serão formados 2^N subconjuntos diferentes para serem analisados. Assim, o tempo de execução deste algoritmo é exponencial em função do número de objetos gerenciados.

5.5.2.2. Algoritmo Pesquisa Gulosa (*Greedy*)

O número de subconjuntos torna-se muito grande no algoritmo *Pesquisa Exaustiva* sendo que, muitos deles, não são realmente explicação dos alarmes recebidos. Desta forma, no lugar de examinar todos os subconjuntos de S, é melhor examinar apenas alguns destes subconjuntos que ainda sejam capazes de apresentar a melhor explicação para os alarmes recebidos no *cluster*. Assim, este algoritmo *Greedy* propõem o desenvolvimento dos seguintes passos:

1. Iniciar com o conjunto S de objetos associados com o *cluster* de alarmes.
2. Encontrar todos as partições que consistem de dois subconjuntos disjuntos de igual tamanho.
3. Dos subconjuntos resultantes, selecionar o subconjunto que explica todos os alarmes recebidos e, tenha a máxima probabilidade de que pelo menos um dos objetos do subconjunto selecionado seja uma falha primária.
4. Aplicar os dois passos anteriores recursivamente para o subconjunto selecionado, enquanto os subconjuntos resultantes da partição corrente não sejam simples (ou seja, tenha mais de um elemento), ou não expliquem todos os alarmes recebidos.

5.5.2.3. Algoritmo *Divida e Conquiste*

A complexidade dos algoritmos anteriores é proporcional ao tamanho do S, conjunto dos objetos associados ao *cluster* de alarmes recebido. Quando o tamanho do conjunto S é pequeno os algoritmos *Pesquisa Exaustiva* e *Pesquisa Gulosa (Greedy)* podem ser preferentemente satisfatórios.

Assim, o *algoritmo Divida e Conquiste* procura solucionar o problema da complexidade do algoritmo para conjuntos S de grande tamanho, através do particionamento do problema em pequenos problemas e, encontrar as soluções para estes pequenos problemas. Em seguida combina as soluções em uma única solução do sistema.

O algoritmo inicia com um subconjunto S de objetos associados com o *cluster* de alarmes. Em seguida são realizados os seguintes passos:

1. Encontra a partição do subconjunto S que resulte em dois subconjuntos disjuntos, tal que em cada subconjunto os objetos exibam a máxima dependência mútua. Para os objetos e_i e e_j que pertençam ao mesmo subconjunto, e e_k que não pertença, o peso da dependência w_{ij} seja maior que os pesos da dependência w_{ik} , w_{ki} , w_{kj} e w_{jk} .
2. Seleciona um dos subconjuntos de S que:
 - Explique todos os alarmes recebidos e ,
 - Tenha a máxima probabilidade de que pelo menos um dos objetos do subconjunto seja uma falha primária.

Se tal subconjunto de S não existir, então encontrar o subconjunto de alarmes que cada um dos subconjuntos de S explique e , selecionar ambos.

A cada subconjunto de S selecionado, aplicar de forma recursiva os dois passos anteriores do algoritmo, enquanto o subconjunto resultante do particionamento não for simples, ou seja, possuir mais de um objeto. O principal objetivo do algoritmo é limitar o número de objetos do conjunto S .

A ação de cada estágio do algoritmo é eliminar os objetos do conjunto S que são menos prováveis de serem a causa da falha primária.

No caso de nenhum dos subconjuntos selecionados explicarem todos os alarmes recebidos, existe a indicação de que pode haver mais de um objeto com falha primária, sendo que estes objetos podem estar inclusos em mais de um subconjunto.

Para a implementação deste algoritmo são necessários:

Como Entrada:

- Um *cluster* de alarmes recebidos durante um intervalo de tempo T .

- Seja dado um conjunto S de N objetos associadas aos alarmes recebidos. A associação de alarmes pode ser fornecida, ou produzida por algum algoritmo de associação de alarmes. O critério para a geração dos alarmes adotado nesta dissertação, parte do princípio que o conjunto de objetos gerenciados, associados a um dado alarme, é formado pelo objeto que sinalizou a falha, mais aqueles dos quais ele depende diretamente (considere-se aqui um arco como distância de dependência). O critério para a criação dos alarmes pode ser diversificado, como por exemplo, associando um alarme a todos os objetos com distância de dependência menor ou igual a w a partir do objeto que emitiu a sinalização. O valor w refere-se a um parâmetro experimental que deve ser estimado.
- Cada objeto e_i pertencente a S deve ter as seguintes probabilidades conhecidas:
 - Probabilidade incondicional p_i em que o objeto e_i falhe independentemente do estado de qualquer outro objeto e_j pertencente a S .
 - Probabilidade condicional p_{ij} entre objetos pertencentes a S , ou seja, probabilidade na qual e_i apresenta uma falha secundária, dado que e_j falhou, sendo ambos objetos pertencentes ao mesmo conjunto S .

Como Saída:

Os menores subconjuntos de S que explicam o *cluster* de alarmes recebidos e , tem a maior probabilidade de que pelo menos um dos objetos do conjunto apresente uma falha primária.

A aplicação deste algoritmo é dividida em 2 fases:

Fase I (União):

Nesta primeira fase do algoritmo encontram-se as máximas partições mútuas (partições que apresentam as maiores dependências mútuas) dos objetos no conjunto S .

Passo 1:

- Cada objeto do conjunto S é tratado como uma partição. Portanto, no primeiro passo do algoritmo tem-se N partições, onde N é o número de objetos do conjunto S .

Passo 2:

- Examinam-se todas as partições e selecionam as duas partições que apresentam a máxima probabilidade condicional. Estas duas partições são incorporadas em uma única partição. Neste passo deve ser realizado a seguinte operação:

- Calcula a probabilidade da nova partição (partição união) da seguinte forma:

$$PU = P_A + P_{AB} \cdot P_B + P_B + P_{BA} \cdot P_A , \text{ no qual:}$$

P_U = probabilidade de união das partições.

P_A = probabilidade independente da partição A.

P_B = probabilidade independente da partição B.

P_{AB} = probabilidade dependente da partição A falhar, dado que a partição B falhou.

P_{BA} = probabilidade dependente da partição B falhar, dado que a partição A falhou.

Observação: este cálculo de probabilidade da nova partição não constava no algoritmo em [KATZELA-93], mas foi deduzido a partir de um exemplo no mesmo artigo.

- Calcula a probabilidade de pelo menos um objeto na nova partição ser uma falha primária.
- Calcula os arcos de dependência da nova partição para todas as demais partições do conjunto S.
- Para a nova partição gerada:
 - Se antes da união das partições selecionadas existia uma ou mais partições que apresentem arcos de dependência para ambos os objetos selecionados, ou seja, exista mais de um arco de dependência de alguma partição para a nova partição gerada, deve-se selecionar apenas o arco de dependência de maior probabilidade condicional.
 - Se antes da união das partições selecionadas existiam arcos de dependência destas partições para uma ou mais partições, ou seja, o nova partição apresenta mais de um arco de dependência para alguma outra partição, deve-

se selecionar apenas o arco de dependência de maior probabilidade condicional.

Observação: os dois critérios anteriores para a nova partição gerada, também não constavam no algoritmo em [KATZELA-93], mas foi deduzido a partir de um exemplo no mesmo artigo.

- Repete o passo 2 até o conjunto S apresentar uma única partição.

Fase II (Particionamento):

Divide a única partição obtida como resultado da Fase I (união) em 2 subconjuntos (filho esquerdo e direito), tal que os objetos em cada subconjunto exibam a máxima probabilidade incondicional.

Nesta fase o algoritmo encontra o menor subconjunto S_{sol} do conjunto S , tal que:

- Explique todos os alarmes recebidos, e
- Tenha a máxima probabilidade da falha primária estar nele contida.

Inicialização:

Inicialmente o conjunto solução S_{sol} é vazio. Aplicar o passo típico (apresentado a seguir) do algoritmo tendo como entrada (S,A) , no qual S é o conjunto de objetos gerenciados e, A é o conjunto de alarmes pertencentes ao *cluster* de alarmes.

Passo típico:

O passo típico do algoritmo apresenta como entrada o par (S^t, A^t) , no qual S^t é um subconjunto do conjunto S e, A^t é o subconjunto de alarmes do conjunto A que o conjunto S^t explica.

A partir dos filhos de S^t , Sd^t (filho direito) e Se^t (filho esquerdo) e, suas probabilidades $Pr(Sd^t)$ e $Pr(Se^t)$, realizar, conforme o caso, um dos três passos a seguir:

- Se ambos os filhos (esquerdo e direito) explicam todos os alarmes recebidos em A^t , então selecionar (incluir no conjunto S_{sol}) aquele que exiba a máxima probabilidade

incondicional de pelo menos um objeto do conjunto S ser uma fonte primária de falha. Repetir o passo típico para o novo subconjunto selecionado.

- Se um de seus filhos explica todos os alarmes recebidos e, exibe a maior probabilidade incondicional de pelo menos um objeto do subconjunto ser a fonte primária de falha, então selecionar este filho e repetir o passo típico para este subconjunto selecionado.
- Se somente um de seus filhos explica todos os alarmes, mas o outro filho tem a máxima probabilidade incondicional de pelo menos um objeto no subconjunto ser uma fonte primária de falha, ou nenhum de seus filhos explicam todos os alarmes, então:
 - Encontre o filho que exiba a máxima probabilidade incondicional, entre os dois filhos, de pelo menos um objeto do subconjunto ser a fonte primária de falha e,
 - Encontre o subconjunto A^t de alarmes a partir de A^t que este filho explica. Repita o passo típico para este filho, mas com o conjunto de alarmes de entrada A^t .
 - Repita o passo típico para o outro filho, mas com um conjunto de alarmes de entrada $A^t - A^t$.

O anexo A apresenta um exemplo de aplicação do algoritmo *Divida e Conquiste*.

6. Algoritmo *Divida e Conquista Proposto*

O algoritmo *Divida e Conquista Proposto* é baseado no algoritmo *Divida e Conquista* (apresentado no item 5.5.2.3.), pois supõem-se que este apresenta um melhor desempenho que os algoritmos *Pesquisa Exaustiva* e *Pesquisa Gulosa*.

Os itens listados a seguir foram inclusos no algoritmo *Divida e Conquista* visando melhorar a precisão do algoritmo na localização do objeto falho.

A partir do algoritmo *Divida e Conquista*, foram inclusos os seguintes pontos:

1. Nos algoritmos anteriores é assumido que existe um alarme associado a cada objeto pertencente ao sistema distribuído e, quando um objeto falha, o alarme correspondente a este objeto apresenta uma sinalização de notificação. No entanto, por trás desta suposição está a hipótese de cobertura perfeita dos alarmes. Tal característica não representa a realidade e, desta forma, será levado em consideração no algoritmo *Divida e Conquista Proposto* o Fator de Cobertura, como um parâmetro novo e essencial para a análise da localização de falhas.
2. Durante a simulação do algoritmo observou-se que quando existem vários objetos com o mesmo valor de probabilidade condicional, a escolha dos objetos na fase de união fica dependente apenas da implementação do algoritmo. O algoritmo torna-se mais preciso para fase de teste, quando os critérios a seguir foram inclusos no mesmo:
 - Selecionar as duas partições que sejam cobertas pelo maior número de alarmes ativos (recebidos pelo gerenciador da rede). Caso existam mais de duas partições que cumpram este critério, então
 - Selecionar as duas partições que apresentem a maior probabilidade incondicional.
3. Para a fase de teste foi incluso o conceito de seqüência de teste para os objetos finais selecionados. Através deste critério é testado primeiro o objeto com maior probabilidade de ser a fonte primária de falha, e assim, sucessivamente para os demais objetos selecionados para a fase de teste. Uma das justificativas para este critério

é que em determinadas circunstâncias todos os objetos selecionados não podem ser testados simultaneamente devido, por exemplo, ao consumo excessivo de recursos da rede e, portanto, sendo mais coerente testar primeiro os objetos com maior probabilidade de serem a fonte primária de falha.

Também está sendo levado em consideração para a aplicação deste algoritmo à Arquitetura Hierárquica Proposta no capítulo 3. Este fato é importante, pois permite mostrar como as duas propostas (Algoritmo de Localização de Falha e Arquitetura Hierárquica Proposta) interagem entre si e, como se viabiliza a execução e a análise da correlação de alarmes.

A arquitetura proposta apresenta um menor grau de complexidade em cada nível do Gerenciamento Hierárquico (este pode ser selecionado de forma conveniente) quando comparado às Arquiteturas Centralizada e Distribuída.

Assim, o tempo total do algoritmo de localização de falhas para a Arquitetura Hierárquica Proposta será a soma dos tempos de execução de cada nível do sistema hierárquico.

Tendo-se como parâmetro de comparação, por exemplo, o tempo de complexidade do algoritmo de localização de falhas (em função do número de objetos gerenciados, fator de cobertura de falhas) entre as diversas Arquiteturas, Centralizada, Distribuída e Hierárquica Proposta, pode-se avaliar em que circunstâncias uma determinada arquitetura apresenta um menor tempo de execução do algoritmo de localização de falhas, permitindo uma maior disponibilidade da rede. Outro aspecto importante é a determinação do número de níveis de gerenciamento intermediários (para o Gerenciamento Hierárquico de falhas Proposto) mais adequado às características de um determinado sistema distribuído.

6.1 O Algoritmo Proposto

Algumas partes do algoritmo *Divida e Conquiste* foram alteradas. Estas alterações são apresentadas em **negrito** no algoritmo a seguir.

O princípio deste algoritmo é semelhante ao anterior, ou seja, deseja-se particionar o problema em problemas menores e, desta forma, encontrar as soluções para estes problemas menores. Em seguida combina-se as soluções obtidas em uma única solução completa para o sistema.

O algoritmo inicia com um subconjunto S de objetos associados com o *cluster*, seguindo-se os passos:

1) Encontra a partição do subconjunto S que resulta em dois subconjuntos disjuntos, tal que em cada subconjunto os objetos exibam a máxima dependência mútua. Para os objetos e_i e e_j que pertençam ao mesmo subconjunto, e e_k que não pertença, o peso da dependência w_{ij} seja maior que os pesos da dependência w_{ik} , w_{ki} , w_{kj} e w_{jk} .

2) Seleciona um dos subconjuntos de S que:

- Explique todos os alarmes recebidos e,
- Tenha a máxima probabilidade de que pelo menos um dos objetos no subconjunto seja uma falha primária.

Se tal subconjunto de S não existir, então encontrar o subconjunto de alarmes que cada um dos subconjuntos de S explique e selecionar ambos.

A cada subconjunto de S selecionado, aplica de forma recursiva os dois passos anteriores do algoritmo, enquanto o subconjunto resultante do particionamento não for simples, ou seja, possuir mais de um objeto. O principal objetivo do algoritmo é limitar o número de objetos do conjunto S .

A ação de cada estágio do algoritmo é eliminar os objetos do conjunto S que são menos prováveis de serem a causa da falha primária.

No caso de nenhum dos subconjuntos selecionados explicarem todos os alarmes recebidos, existe a indicação de que pode haver mais de um objeto com falha primária, sendo que estes objetos podem ser inclusos em mais de um subconjunto.

Para a implementação deste algoritmo são necessários:

Como Entrada:

- Um *cluster* de alarmes recebidos durante um intervalo de tempo T.
- Seja dado um conjunto S de N objetos associadas aos alarmes recebidos. A associação de alarmes pode ser fornecida, ou produzida por algum algoritmo de associação de alarmes.
- Cada objeto e_i pertencente a S deverá ter as seguintes probabilidades conhecidas:
 - Probabilidade incondicional p_i de que o objeto e_i falhe independentemente do estado de qualquer outro objeto e_j pertencente a S.
 - Probabilidade condicional p_{ij} entre objetos pertencentes a S, ou seja, probabilidade na qual e_i apresenta uma falha secundária, dado que e_j falhou, sendo ambos objetos pertencentes ao mesmo conjunto S.
 - **O fator de cobertura de cada objeto de gerenciamento monitorado, que indica a probabilidade deste objeto sinalizar ou não uma falha (alarme), ou seja, representa a capacidade do sistema de gerenciamento em diagnosticar uma falha deste objeto e gerar uma sinalização.**

Como Saída:

O menores subconjuntos de S que explicam o *cluster* de alarmes recebidos e tem a maior probabilidade de que pelo menos um dos objetos no conjunto apresente uma falha primária.

A aplicação deste algoritmo é dividida em 2 fases:

Fase I (União):

Nesta primeira fase do algoritmo encontram-se as máximas partições mútuas (partições que apresentam as maiores dependências mútuas) dos objetos do conjunto S.

Passo 1:

- Cada objeto do conjunto S é tratado como uma partição. Portanto, no primeiro passo do algoritmo tem-se N partições, no qual N número de objetos no conjunto S.

Passo 2:

- Examinam-se todas as partições e:
 - 1) Selecionam as duas partições que apresentam a máxima probabilidade condicional. Caso existam mais de duas partições que cumpram este critério, então
 - 2) **Selecionam as duas partições que sejam cobertas pelo maior número de alarmes ativos (recebidos pelo gerenciador da rede). Caso existam mais de duas partições que cumpram este critério, então**
 - 3) **Selecionam as duas partições que apresentam a maior probabilidade incondicional. Se mesmo assim, existirem mais de dois objetos selecionados, então**
 - 4) **Selecionam duas partições aleatoriamente.**
- As duas partições selecionadas são incorporadas em uma única partição. Neste passo deve ser realizado as seguintes operações:

- Calcula a probabilidade da nova partição (partição união) da seguinte forma:

$$PU = P_A \cdot C_A + P_{AB} \cdot P_B \cdot C_A + P_B \cdot C_B + P_{BA} \cdot P_A \cdot C_B$$

no qual:

P_U = probabilidade de união entre partições.

P_A = probabilidade independente da partição A.

P_B = probabilidade independente da partição B.

C_A = **Fator de Cobertura do objeto A**

C_B = **Fator de Cobertura do objeto B**

P_{AB} = probabilidade dependente da partição A falhar, dado que a partição B falhou.

P_{BA} = probabilidade dependente da partição B falhar, dado que a partição A falhou.

- **Calcula o fator de cobertura da nova partição (partição união) de acordo com a seguinte formulação:**

como: $PU = P_A \cdot C_A + P_{AB} \cdot P_B \cdot C_A + P_B \cdot C_B + P_{BA} \cdot P_A \cdot C_B,$

considerando, $PU_A = P_A \cdot C_A + P_{AB} \cdot P_B \cdot C_A$ e

$$PU_B = P_B \cdot C_B + P_{BA} \cdot P_A \cdot C_B$$

onde,

PU_A = probabilidade de falha total do objeto A,

PU_B = probabilidade de falha total do objeto B.

Portanto, o fator de cobertura (FC) da nova partição pode ser calculado através da proporcionalidade das probabilidades:

$$FCU = (PU_A / PU) \cdot C_A + (PU_B / PU) \cdot C_B$$

- Calcula a probabilidade **de sinalização** de pelo menos um objeto na nova partição ser uma falha primária.
- Calcula os arcos de dependência da nova partição para todas as demais partições do conjunto S.
- Para a nova partição gerada:
 - Se antes da união das partições selecionadas existia uma ou mais partições que apresentem arcos de dependência para ambos os objetos selecionados, ou seja, exista mais de um arco de dependência de alguma partição para a nova partição gerada, deve-se selecionar apenas o arco de dependência de maior probabilidade condicional.
 - Se antes da união das partições selecionadas existiam arcos de dependência destas partições para uma ou mais partições, ou seja, o nova partição apresenta mais de um arco de dependência para alguma outra partição, deve-se selecionar apenas o arco de dependência de maior probabilidade condicional.
- Repita o passo 2 até o conjunto S apresentar uma única partição.

Fase II (Particionamento):

Divide a única partição obtida como resultado da Fase I (União) em 2 conjuntos (filho esquerdo e direito), tal que os objetos em cada subconjunto exibam a máxima probabilidade incondicional.

Nesta fase o algoritmo encontra o menor subconjunto S_{sol} do conjunto S tal que:

- Explique todos os alarmes recebidos, e
- Tenha a máxima probabilidade de **sinalização** da falha primária estar nele contida.

Inicialização:

Inicialmente o conjunto solução S_{sol} é vazio. Aplicar o passo típico (apresentado a seguir) do algoritmo, tendo como entrada o par (S,A) , onde S é o conjunto de todos os objetos gerenciados, e A é o conjunto de todos os alarmes pertencentes ao *cluster* de alarmes.

Passo típico:

O passo típico do algoritmo apresenta como entrada o par (S^t, A^t) , onde S^t é um subconjunto do conjunto S , e A^t é um subconjunto de alarmes do conjunto A que o conjunto A^t explica.

A partir dos filhos de S^t , Sd^t (filho direito) e Se^t (filho esquerdo), e suas probabilidades **de sinalização** $Pr(Sd^t)$ e $Pr(Se^t)$, realizar, conforme o caso, um dos três passos a seguir:

- Se ambos os filhos (esquerdo e direito) explicam todos os alarmes recebidos em A^t , então selecionar (incluir no conjunto S_{sol}) aquele que exiba a máxima probabilidade incondicional **de sinalização** de pelo menos um objeto do conjunto S ser uma fonte primária de falha. Repetir o passo típico para o novo subconjunto selecionado.
- Se um de seus filhos explica todos os alarmes recebidos, e exibe a maior probabilidade incondicional **de sinalização** de pelo menos um objeto no subconjunto ser a fonte primária de falha, então selecionar este filho e repetir o passo típico para este subconjunto selecionado.
- Se somente um de seus filhos explica todos os alarmes, mas o outro filho tem a máxima probabilidade incondicional **de sinalização** de pelo menos um objeto no conjunto ser uma fonte primária de falha, ou nenhum de seus filhos explicam todos os alarmes, então:
 - Encontre o filho que exiba a máxima probabilidade incondicional **de sinalização**, entre os dois filhos, de pelo menos um objeto do subconjunto ser a fonte primária de falha, e
 - Encontre o subconjuntos A^t de alarmes a partir de A^t que este filho explica. Repeta o passo típico para este filho, mas com o conjunto de alarmes de entrada A^t .
 - Repeta o passo típico para o outro filho, mas com um conjunto de alarmes de entrada $A^t - A^t$.

- Para cada objeto pertencente ao conjunto S_{sol} , calcular a sua probabilidades de falha total, conforme a seguir:

$$P_{U_A} = P_A \cdot C_A + P_{AB} \cdot P_B \cdot C_A + P_{AC} \cdot P_C \cdot C_A + \dots + P_{AX} \cdot P_X \cdot C_A$$

no qual:

P_{U_A} = probabilidade de falha total do objeto A.

P_A = probabilidade independente da partição A.

P_B = probabilidade independente da partição B.

P_x = probabilidade independente da partição X.

C_A = Fator de Cobertura do objeto A

P_{AB} = probabilidade dependente da partição A falhar, dado que a partição B falhou.

⋮

P_{AX} = probabilidade dependente da partição A falhar, dado que a partição X falhou.

Adotando-se a hipótese de que o objeto A depende dos objetos B a até X, considerando-se apenas um arco de dependência.

- A seqüência de teste dos objetos do conjunto S_{sol} é realizada na ordem decrescente da probabilidade de falha total de tais objetos.

7. Algoritmos de Localização de Falhas aplicados às Arquiteturas de Gerenciamento.

Neste capítulo é apresentada uma aplicação para cada uma das Arquiteturas de Gerenciamento de Falhas em redes (Centralizado, Distribuído e Hierárquico Proposto) dos algoritmos *Divida e Conquiste* e *Divida e Conquiste Proposto*.

Este capítulo tem como metas exemplificar a aplicação desses algoritmos nas diversas arquiteturas, apresentar algumas conclusões preliminares sobre suas vantagens e desvantagens, além de fornecer subsídios para a adequada compreensão das comparações realizadas no capítulo 8.

7.1. Testes na Arquitetura de Gerenciamento de Falhas Centralizado

O Gerenciamento Centralizado baseia-se no princípio da existência de um único gerente responsável pelo gerenciamento de toda a rede. Este gerente possui o algoritmo de localização de falha que determina os objetos gerenciados com maior probabilidade de serem a fonte primária de falha.

A figura 7.1 apresenta uma topologia física de uma rede hipotética, na qual é aplicado o Gerenciamento de Falhas Centralizado.

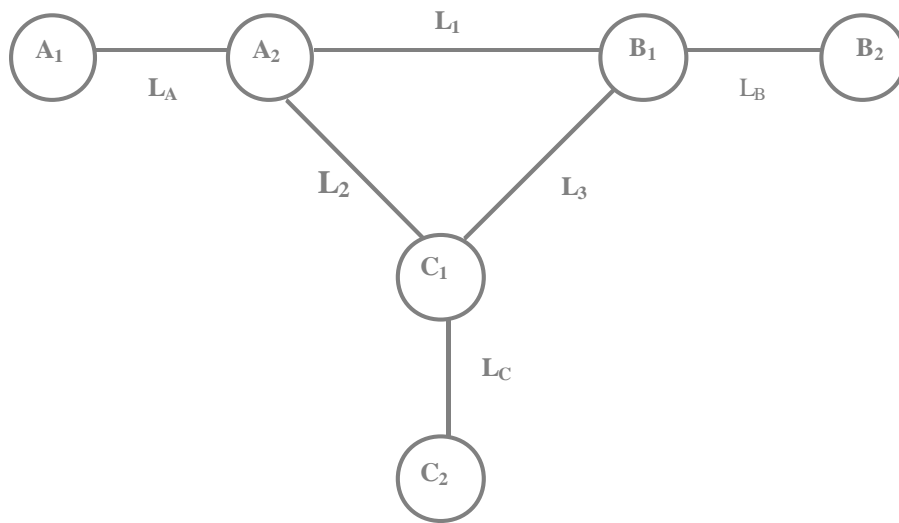


Figura 7.1: Topologia física da rede

Supor que existam os seguintes objetos realizando o monitoramento da rede:

A_1 : objeto que monitora a atividade do nó A_1 .

A_2 : objeto que monitora a atividade do nó A_2 .

L_A : objeto que monitora a atividade do nó L_A .

B_1 : objeto que monitora a atividade do nó B_1 .

B_2 : objeto que monitora a atividade do nó B_2 .

L_B : objeto que monitora a atividade do nó L_B .

C_1 : objeto que monitora a atividade do nó C_1 .

C_2 : objeto que monitora a atividade do nó C_2 .

L_C : objeto que monitora a atividade do nó L_C .

Supor que existam as seguintes dependências de recursos entre os objetos:

Nó A_1 necessita dos recursos do nó A_2 , via conexão L_A .

Nó A_2 necessita dos recursos do nó B_1 , via conexão L_1 .

Nó A_2 necessita dos recursos do nó C_1 , via conexão L_2 .

Nó B_1 necessita dos recursos do nó B_2 , via conexão L_B .

Nó C_1 necessita dos recursos do nó B_1 , via conexão L_3 .

Nó C_1 necessita dos recursos do nó C_2 , via conexão L_C .

Pode-se apresentar a arquitetura de gerenciamento em função do seu nível de gerenciamento conforme apresentado na figura 7.2:

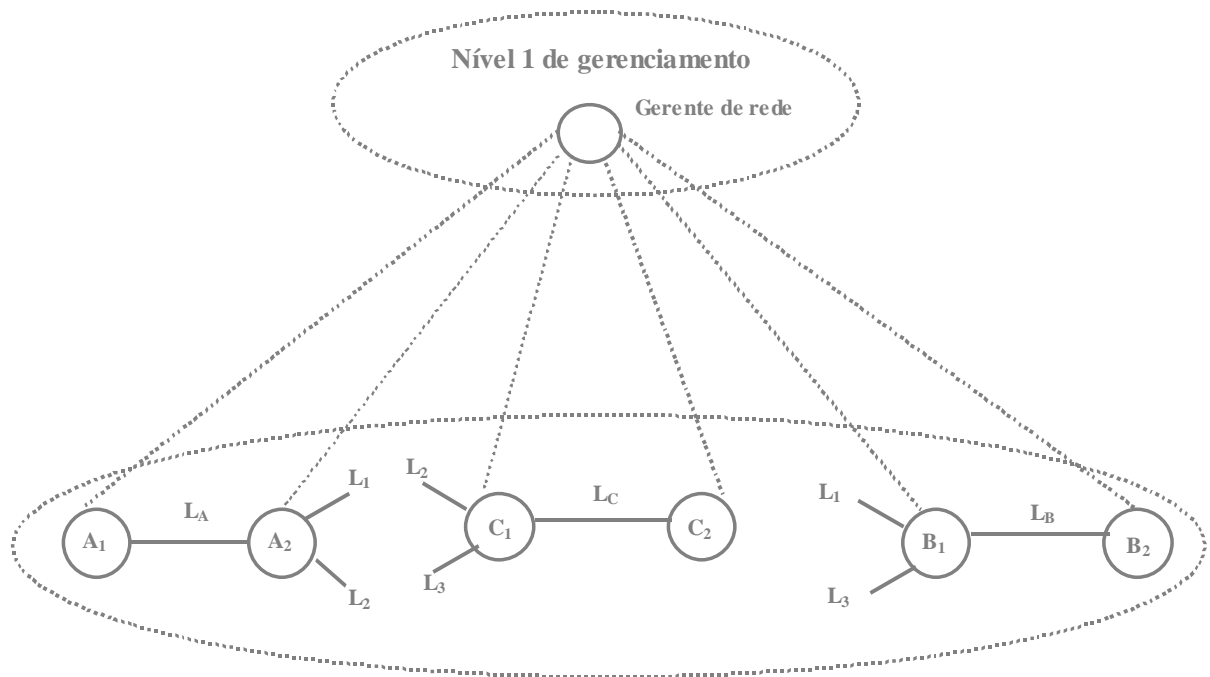
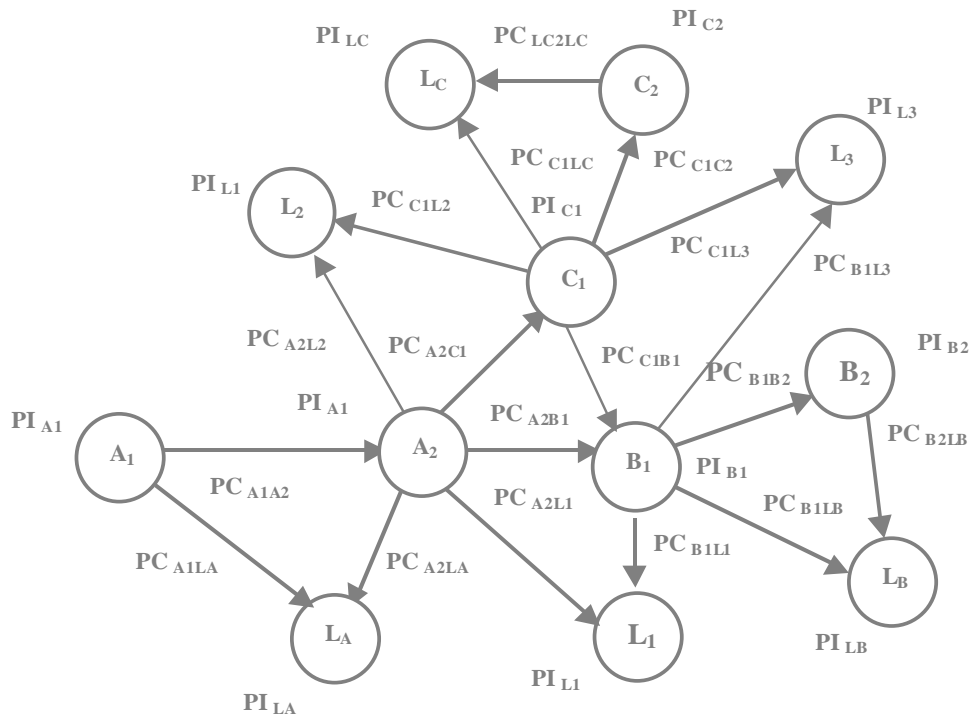


Figura 7.2: Visão da Arquitetura de Gerenciamento Centralizada de Rede

Conforme a figura 7.2, o gerente de rede recebe mensagens de gerenciamento (notificação de falhas ocorridas na rede) dos objetos gerenciados.

Pode-se obter o seguinte grafo de dependência para essa Arquitetura de Gerenciamento Centralizado:



PC = Probabilidade condicional de falha PI = Probabilidade incondicional de falha

Figura 7.3 : Grafo de dependência para o exemplo de Gerenciamento Centralizado

A aplicação do algoritmo *Divida e Conquiste* nessa mesma Arquitetura de Gerenciamento de Falhas centralizada, está apresentada no Anexo B desta dissertação.

O conjunto solução final obtido é $S_{sol} = \{ A_2, C_1, C_2 \}$.

A fonte primária de falha será encontrada após realizar a fase de teste nos objetos selecionados pelo conjunto S_{sol} .

A aplicação do algoritmo *Divida e Conquiste Proposto* nessa Arquitetura de Gerenciamento de Falhas Centralizada, está apresentada no Anexo B desta dissertação.

O conjunto solução final obtido é composto pela união das soluções parciais $S_{Sol\ 1.1.1} = \{ A_2 \}$, $S_{Sol\ 1.1.2} = \{ C_1 \}$ e $S_{Sol\ 1.2} = \{ C_2 \}$, ou seja, a solução final é:

$$S_{sol} = \{ A_2, C_1, C_2 \}$$

Determinação da seqüência de teste através do Algoritmo *Divida e Conquiste Proposto*

A fase de testes deverá ser realizados nos objetos A_2 , C_1 e C_2 .

Neste caso, a seqüência de teste foi baseada nas probabilidades de falha dos objetos (probabilidade condicional adicionada a probabilidade incondicional para cada objeto selecionado).

A seqüência de testes deverá ser realizada na ordem decrescente das probabilidades de falhas, pois os objetos com maior probabilidade apresentam maior chance de serem a fonte primária de falha. Assim, poderá ser realizado as seguintes seqüências de testes para este exemplo:

$$C_2, A_2 \text{ e } C_1$$

Esta característica significa uma melhora em relação ao algoritmo *Divida e Conquiste*, que não determina a forma de teste dos objetos selecionados.

Assim sendo, o algoritmo *Divida e Conquiste Proposto* pode proporcionar um aumento na disponibilidade da rede, pois apresenta economia de tempo na fase de teste.

Conclusão Preliminar:

A complexidade do grafo gerado para o Gerenciamento Centralizado é proporcional ao tamanho da rede. Este fato pode tornar a sua implementação extremamente complexa, bem como a execução do algoritmo *Divida e Conquiste* pode consumir um tempo excessivo de processamento, principalmente quando se leva em consideração redes de grande dimensões. Outro fato importante é o tráfego gerado pelo gerenciamento da rede, sendo este proporcional ao número de mensagens trocadas (ou seja, é função do número de objetos gerenciados), bem como da velocidade de conexão entre objetos gerenciados. Este fato pode tornar o gerenciamento da rede crítico, bem como afetar a disponibilidade da rede.

Na simulação do algoritmo *Divida e Conquiste Proposto*, verifica-se que será consumido uma menor quantidade de recursos de rede durante a fase de teste, quando comparado ao algoritmo *Divida e Conquiste*. O algoritmo *Divida e Conquiste* não possui nenhum critério de seqüência de teste para os objetos selecionados, afetando a disponibilidade da rede.

7.2. Testes na Arquitetura de Gerenciamento de Falhas Distribuído

Considere a mesma topologia física do Gerenciamento Centralizado apresentada no item anterior, com as mesmas probabilidades condicionais, incondicionais e fator de cobertura. Na figura 7.4 é apresentada a topologia física da rede.

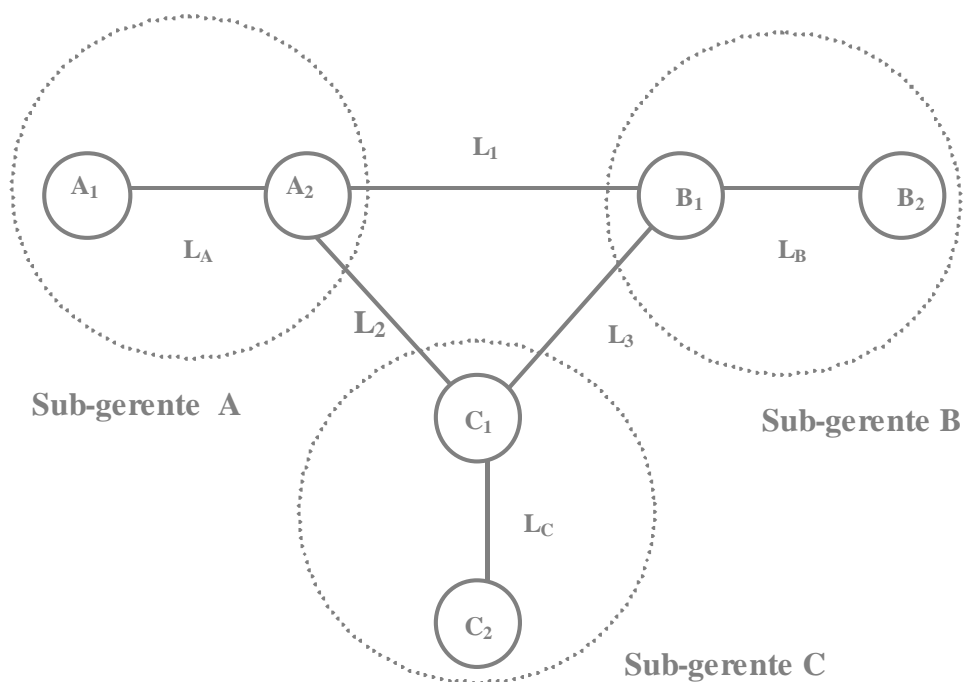


Figura 7.4: Topologia Física da Rede

Para manter a compatibilidade com o exemplo apresentado no item anterior, são considerados os mesmos objetos realizando o monitoramento da rede com as mesmas dependências de recursos anteriormente consideradas. Para este caso de Arquitetura de Gerenciamento de Falhas Distribuído, considere que existam os seguintes sub-gerentes:

Sub-gerente A responsável pelo Gerenciamento de Falhas dos nós A₁, A₂, L₁, L₂ e L_A.

Sub-gerente B responsável pelo Gerenciamento de Falhas dos nós B_1 , B_2 , L_1 , L_3 e L_B .
 Sub-gerente C é responsável pelo Gerenciamento de Falhas dos nós C_1 , C_2 , L_2 , L_3 e L_C .
 Os sub-gerentes apresentam, entre si, um protocolo de comunicação de informações referentes a objetos falhos existentes na rede. Pode-se também apresentar a hierarquia de gerenciamento pelos seus níveis de gerenciamento conforme mostrado na figura 7.5.

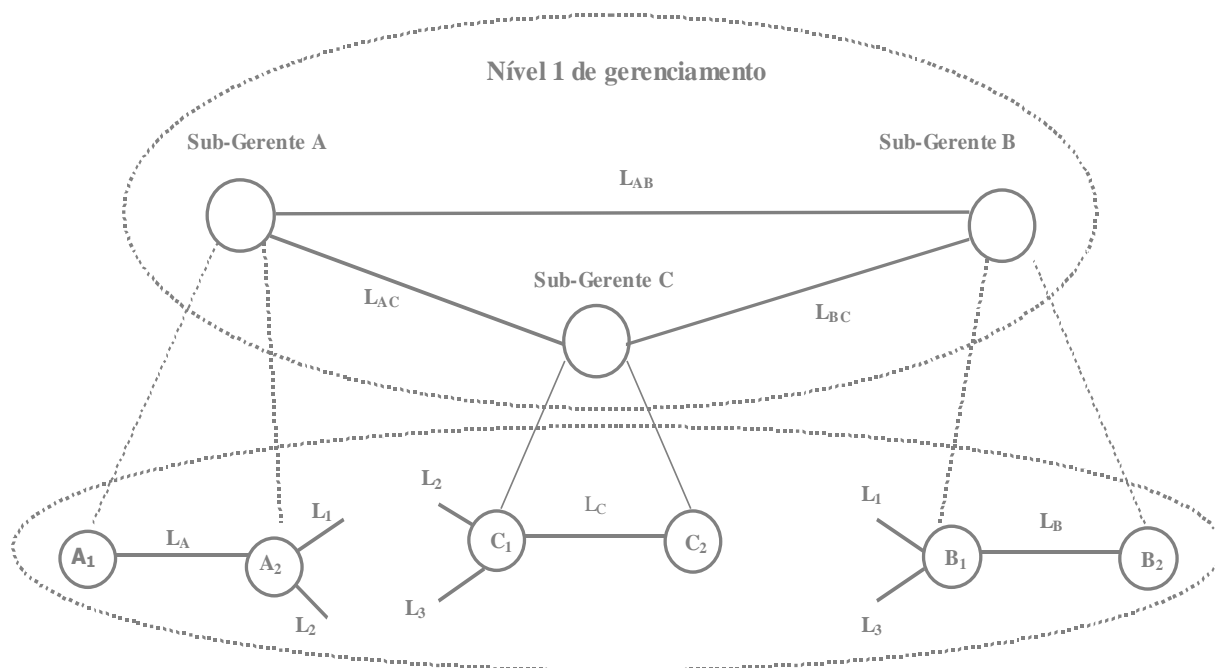


Figura 7.5: Visão da Arquitetura de Gerenciamento Distribuído da Rede

A troca de mensagens de gerenciamento entre os sub-gerentes pode ser realizada pelas conexões L_{AB} , L_{AC} e L_{BC} . Estas são consideradas de alta disponibilidade e utilizadas apenas para troca de mensagens de gerenciamento de falhas.

A aplicação do algoritmo *Divida e Conquiste* na Arquitetura de Gerenciamento de Falhas Distribuído é apresentada no Anexo C desta dissertação.

O conjunto solução final obtido é $S_{sol} = \{ C_2 \}$ para o Sub-gerente C.

A aplicação do algoritmo *Divida e Conquiste Proposto* na Arquitetura de Gerenciamento de Falhas Distribuído, é apresentada no Anexo C deste trabalho.

O conjunto solução final obtido é composto pelos conjuntos soluções parciais $S_{sol\ 1} = \{ C_2 \}$ e $S_{sol\ 2} = \{ C_1, L_2 \}$. Portanto, o conjunto solução $S_{sol} = \{ C_1, C_2, L_2 \}$ é a solução final para o Sub-gerente C.

Determinação da seqüência de teste através do Algoritmo *Divida e Conquiste Proposto*

A fase de testes deverá ser realizada em função do valor decrescente da probabilidade de falha. Calcula-se a seqüência de teste obtendo-se a seguinte ordem: C_2, C_1, L_2 . Pode-se observar que através da aplicação do algoritmo *Divida e Conquiste Proposto* obtiveram-se outras opções de teste de prováveis objetos falhos além de C_2 , permitindo, de certa forma, o aumento da disponibilidade da rede.

Conclusão Preliminar:

Nesta arquitetura não existe um nível superior de decisão (gerente central) aos Sub-gerentes A, B e C. Este fato acarreta uma maior complexidade da comunicação entre os sub-gerentes, principalmente quando do aumento dos sub-gerentes da rede e das mensagens trocadas entre eles. Este fato dificulta na determinação dos objetos de rede com maior probabilidade de serem uma falha primária, bem como na determinação da seqüência de testes a serem realizados sobre os objetos.

Outro fator a ser considerado nesta arquitetura, é que em alguns casos os sub-gerentes que recebem alarmes de falhas podem realizar a fase de testes independentemente da comunicação com os sub-gerentes vizinhos, mesmo antes dos sub-gerentes trocarem informações de gerenciamento. Este fato pode acarretar a realização de testes desnecessários, principalmente quando a fase de localização da falha em cada domínio engloba muitos objetos para a fase de teste. Estes aspectos considerados podem diminuir significativamente a disponibilidade do sistema.

Com relação aos algoritmos pode-se dizer que, para os domínios A e B (Sub-gerentes A e B) não apresentaram alteração dos objetos selecionados para teste. No entanto, para o domínio C (Sub-gerente C), o algoritmo *Divida e Conquiste Proposto* (com fator de cobertura) selecionou um maior número de objetos a serem testados, quando comparado ao algoritmo *Divida e Conquiste* (sem o fator de cobertura). Este fato ocorreu devido a influência do fator de cobertura, apresentando mais opções de teste de prováveis objetos falhos, aumentando, assim, a disponibilidade da rede.

7.3. Testes na Arquitetura de Gerenciamento de Falhas Hierárquica Proposta

Considere a mesma topologia física do gerenciamento centralizado e distribuído apresentada nos itens anteriores, contendo as mesmas probabilidades condicionais, incondicionais e fator de cobertura. As aplicações dos capítulos 7.1 e 7.2 representam o nível 1 de gerenciamento dentro do Gerenciamento Hierárquico (primeiro nível de gerenciamento). Considere que seja adicionado mais um nível de gerenciamento (nível 2 de gerenciamento) conforme descrito a seguir.

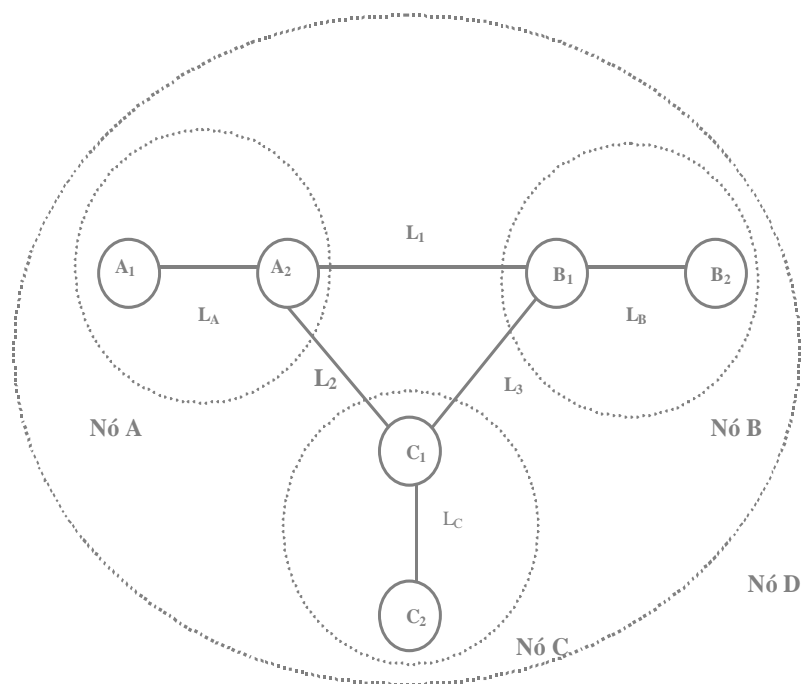


Figura 7.6: Arquitetura do Modelo Físico da Rede

A figura 7.6 apresenta a topologia física da aplicação. Os nós A, B e C (Sub-gerentes A, B e C) do nível de gerenciamento 1 estão conectados via as conexões L_1 , L_2 , L_3 . O nó D (gerente de rede) pertence ao nível de gerenciamento 2.

O nó A (Sub-domínio A) apresenta como elementos internos os nós A_1 e A_2 que estão interligados via a conexão L_A . O nó A_2 conecta-se aos nós B e C (do nível 1 de gerenciamento).

O nó B (Sub-domínio B) apresenta como elementos internos os nós B_1 e B_2 que estão interligados via a conexão L_B . O nó B_1 conecta-se aos nós A e C (do nível 1 de gerenciamento).

O nó C (Sub-domínio C) apresenta como elementos internos os nós C_1 e C_2 que estão interligados via a conexão L_C . O nó C_1 conecta-se aos nós B e C (do nível 1 de gerenciamento).

O nó D representa toda a rede gerenciada, ou seja, é o gerente da rede. Esta representação é apropriada nos casos de expansão da rede, em que este nó é interligado aos demais nós da rede.

Para manter a compatibilidade com os exemplos anteriores, são considerados, no nível hierárquico 1, os mesmos objetos realizando o monitoramento da rede com as mesmas dependências de recursos anteriormente consideradas.

Desta forma, tem-se as seguintes dependências no nível hierárquico 2:

Nó A necessita dos recursos do nó B, via conexão L_1 .

Nó A necessita dos recursos do nó C, via conexão L_2 .

Nó C necessita dos recursos do nó B, via conexão L_3 .

O objeto D que monitora a atividade do nó D.

O objeto D representa toda a rede e sinaliza seu alarme quando ocorre uma falha em qualquer objeto monitorado na rede (domínio formado por toda a rede).

Pode-se também apresentar a hierarquia de gerenciamento pelos seus vários níveis de gerenciamento conforme mostrado na figura 7.7 .

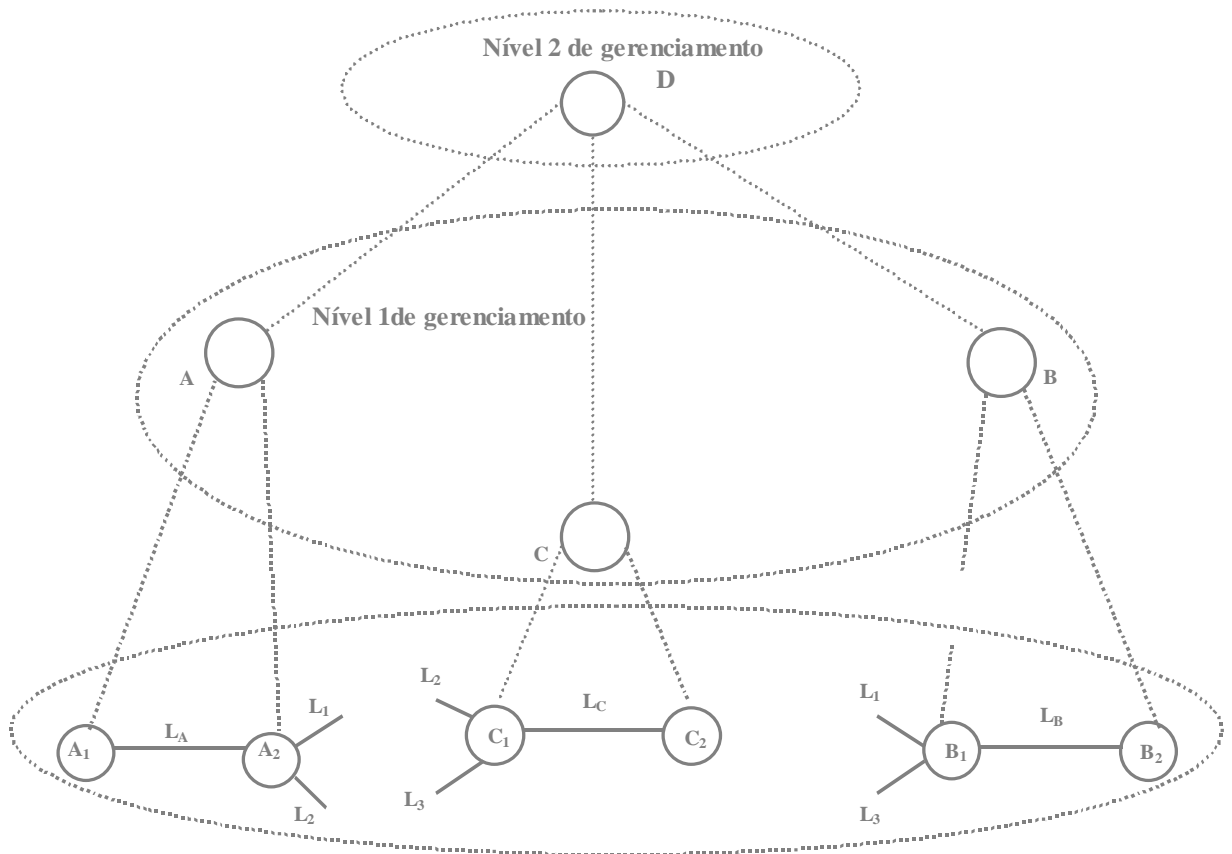


Figura 7.7: Visão da Arquitetura de Gerenciamento Hierárquico Proposto da Rede

Uma terceira forma de apresentar a arquitetura é através da topologia lógica. Nela pode-se observar o relacionamento dos objetos gerenciados nos diversos níveis da hierarquia e suas interconexões com os seus sub-gerentes, e destes com o gerente da rede. A figura 7.8 apresenta o modelo lógico para o presente exemplo.

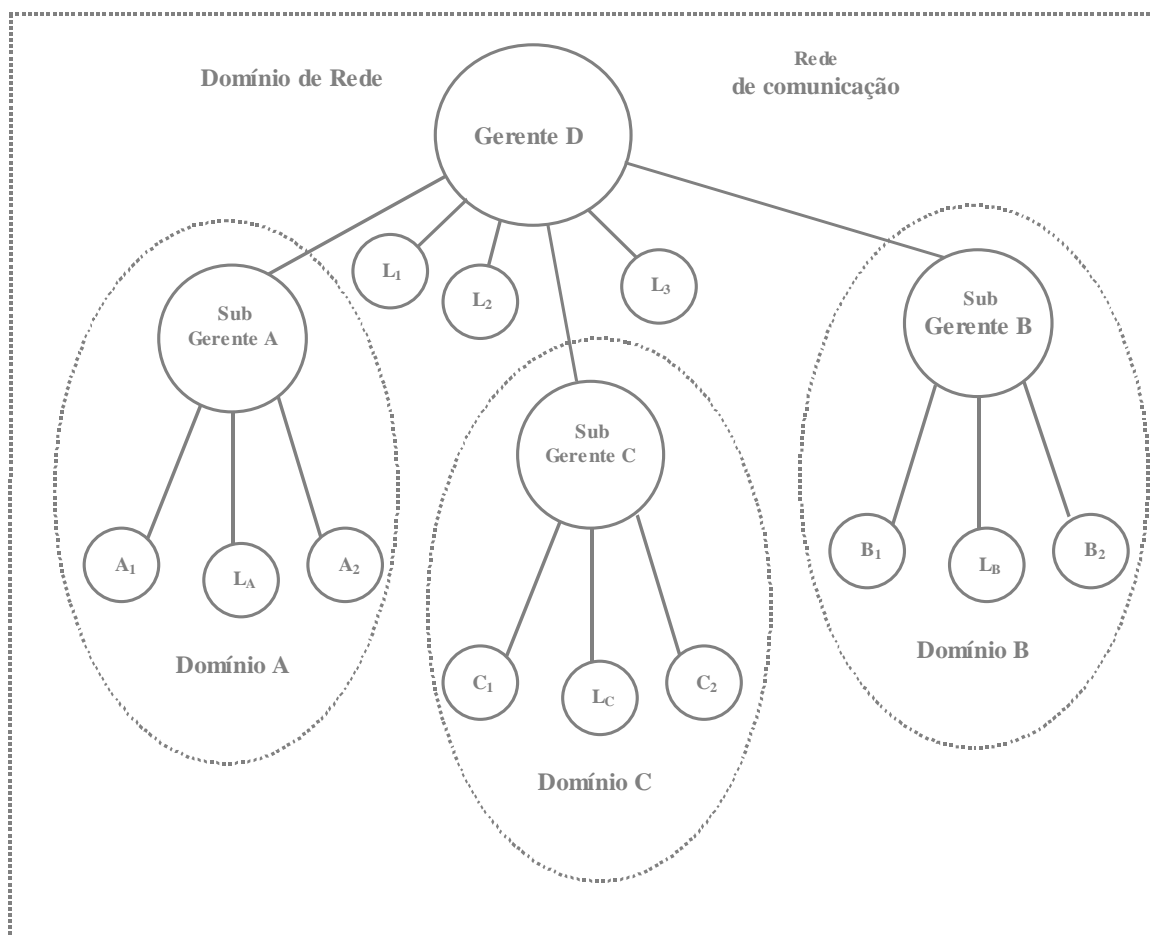


Figura 7.8: Arquitetura do modelo lógico da rede

O gerente da rede (representado pelo objeto D) é responsável pelo Gerenciamento de Falhas de toda a rede.

O domínio A é gerenciado pelo Sub-gerente A, da mesma forma ocorre o domínio B com o Sub-gerente B e o domínio C com o Sub-gerente C. Se uma falha não ultrapassa o domínio de seu sub-gerente, então não há necessidade de atuação do gerente da rede, pois o respectivo sub-gerente apresenta capacidade suficiente para localizar o objeto ou objetos com maiores probabilidades de serem uma falha primária.

Dada a visão hierárquica da rede, pode-se descrever cada nível hierárquico de forma mais detalhada com relação a composição de seus grafos de dependência e alarmes gerados.

7.3.1. Nível 1 de Gerenciamento

O Nível 1 de Gerenciamento é composto de por 3 localidades: A, B e C.

Localidade A

Considera-se que a localidade A composta pelos nós A_1 e A_2 interligados pela conexão L_A , conforme apresentado na figura 7.9:



Figura 7.9: Modelo físico do domínio A

Premissas:

Assume-se a seguinte dependência:

- Nó A_1 necessita dos recursos do nó A_2 .

Supor que existam os seguintes objetos realizando o monitoramento da rede:

A_1 : objeto que monitora a atividade do nó A_1 .

A_2 : objeto que monitora a atividade do nó A_2 .

L_A : objeto que monitora a atividade do nó L_A .

Desta forma pode-se obter o seguinte grafo de dependência:

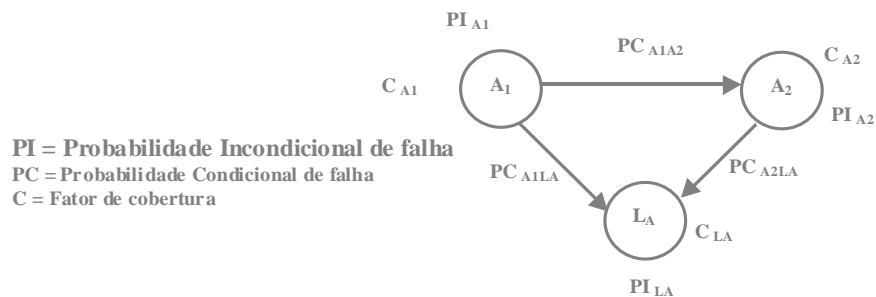


Figura 7.10: Objetos do domínio A

O objeto A_1 depende da atividade do objeto A_2 (dos recursos desta localidade) e do objeto L_A (conexão necessária para o envio de solicitação de serviços).

O objeto A_2 depende da atividade do objeto L_A (conexão necessária para o envio do serviço solicitado pelo objeto A_1). O objeto L_A não depende dos objetos A_1 e A_2 .

Localidade B

Considera-se a localidade B composta pelos nós B_1 e B_2 interligados pela conexão L_B , conforme apresentado na figura 7.11.

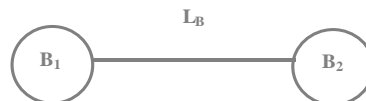


Figura 7.11: Modelo físico do domínio B

Premissas:

Assume-se a seguinte dependência:

- Nó B_1 necessita dos recursos do nó B_2 .

Supor que existam os seguintes objetos realizando o monitoramento da rede:

B_1 : objeto que monitora a atividade do nó B_1 .

B_2 : objeto que monitora a atividade do nó B_2 .

L_B : objeto que monitora a atividade do nó L_B .

Desta forma pode-se obter o seguinte grafo de dependência:

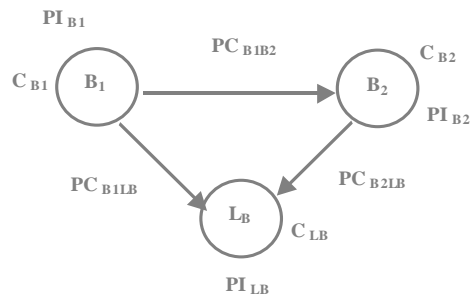


Figura 7.12: Objetos do domínio B

O objeto B_1 depende da atividade do objeto B_2 (dos recursos desta localidade) e do objeto L_B (conexão necessária para o envio de solicitação de serviços). O objeto B_2 depende da atividade do objeto L_B (conexão necessária para o envio do serviço solicitado pelo objeto B_1). O objeto L_B não depende dos objetos B_1 e B_2 .

Localidade C

Considera-se a localidade C composta pelos nós C_1 e C_2 interligados pela conexão L_C , conforme é apresentado na figura 7.13.

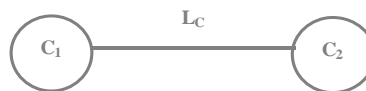


Figura 7.13: Modelo físico do domínio C

Premissas:

Assume-se a seguinte dependência:

- Nó C_1 necessita dos recursos do nó C_2 .

Supor que existam os seguintes objetos realizando o monitoramento da rede:

C_1 : objeto que monitora a atividade do nó C_1 .

C_2 : objeto que monitora a atividade do nó C_2 .

L_B : objeto que monitora a atividade do nó L_B .

Desta forma pode-se obter o seguinte grafo de dependência:

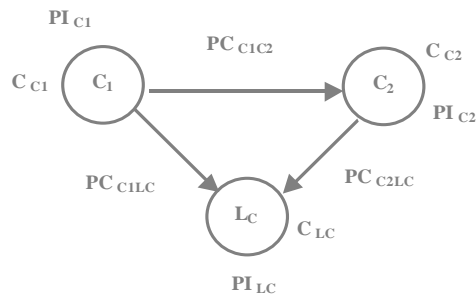


Figura 7.14: Objetos do domínio C

O objeto C_1 depende da atividade do objeto C_2 (dos recursos desta localidade) e do objeto L_C (conexão necessária para o envio de solicitação de serviços). O objeto C_2 depende da atividade do objeto L_C (conexão necessária para o envio do serviço solicitado pelo objeto C_1). O objeto L_C não depende dos objetos C_1 e C_2 .

Dependências entre objetos de diferentes domínios:

Supor que os objetos de diferentes domínios do nível 1 de gerenciamento apresentem uma interdependência implícita de fornecimento e/ou consumo de recursos, conforme mostrados na tabela 7.1. Esta interdependência também é refletida nos demais níveis de gerenciamento.

Objeto	Objeto de que depende
A_2	C_1
A_2	B_1
C_1	B_1

Tabela 7.1: Relacionamento entre objetos de diferentes domínios de gerenciamento

A dependência entre os objetos gerenciado no nível 1 de gerenciamento permite a seguinte dependência entre os objetos do nível 2 na Tabela 7.2.

Dependência no nível 1 de gerenciamento	Implica na dependência no nível 2 de gerenciamento
$A_2 \rightarrow C_1$	$A \rightarrow C$
$A_2 \rightarrow B_1$	$A \rightarrow B$
$C_1 \rightarrow B_1$	$C \rightarrow B$

Tabela 7.2: Relação de dependência entre objetos em diferentes níveis de gerenciamento.

7.3.2. Nível 2 de Gerenciamento

O Nível 2 de Gerenciamento é composto pelas localidades A, B e C e pelas conexões L_1 , L_2 e L_3 que as interligam, conforme representado na figura 7.15.

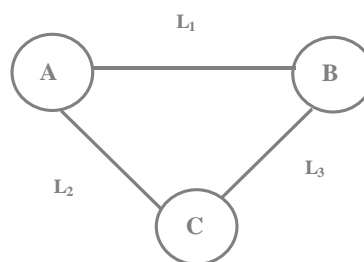


Figura 7.15: Modelamento dos domínios

Premissas:

Assumem-se as seguintes dependências:

- Nó A necessita dos recursos do nó B.
- Nó A necessita dos recursos do nó C.
- Nó C necessita dos recursos do nó B.
- Nó B não necessita dos recursos do nó A.
- Nó B não necessita dos recursos do nó C.

Supor que existam os seguintes objetos realizando o monitoramento da rede:

A : objeto que monitora a atividade do nó A.

B : objeto que monitora a atividade do nó B.

C : objeto que monitora a atividade do nó C.

L1 : objeto que monitora a atividade da conexão L1.

L2 : objeto que monitora a atividade da conexão L2.

L3 : objeto que monitora a atividade da conexão L3.

Desta forma pode-se obter o seguinte grafo de dependência:

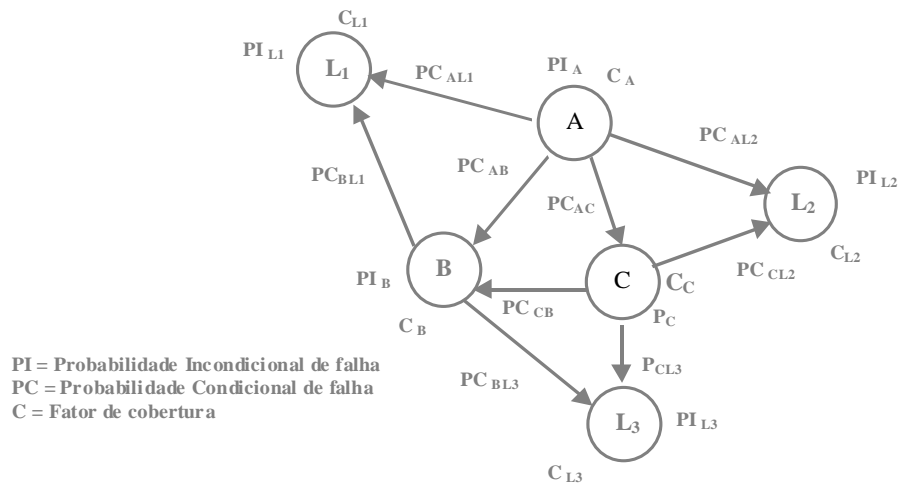


Figura 7.16: Grafo de dependência para o exemplo proposto

O objeto A depende da atividade dos objetos B e C (dos recursos destas localidades) e dos objetos L₁ e L₂ (conexões necessárias para o envio de solicitação de serviços). O objeto C depende da atividade dos objetos B (dos recursos destas localidades) , e do objeto L₂ (conexão necessária para o envio de solicitação de serviços). O objeto B depende da atividade dos objetos L₂ e L₃ (conexões necessárias para o envio do serviço solicitado pelos objetos A e C). Os objetos L₁, L₂ e L₃ não dependem de nenhum outro objeto.

Ocorrendo qualquer falha nos objetos A, B e C (ou seja, nos domínios A,B e C) é enviada uma notificação de falha ao objeto D (gerente de rede).

A aplicação do algoritmo *Divida e Conquiste* na Arquitetura de Gerenciamento de Falhas Hierárquico Proposta é apresentada no Anexo D desta dissertação.

Desta aplicação resultou que o Sub-gerente C deverá submeter o objeto C_2 a fase de teste.

A aplicação do algoritmo *Divida e Conquiste Proposto* na Arquitetura de Gerenciamento de Falhas Hierárquica Proposta está apresentada no Anexo D deste trabalho.

Desta aplicação resultou que o Sub-gerente C deverá submeter também o objeto C_2 a fase de teste.

7.3.3. Conclusão Preliminar:

O Sistema de Gerenciamento Hierárquico proposto apresenta um grafo de dependência com menor complexidade, quando comparado ao Gerenciamento Centralizado, pois a rede é subdividida em níveis e domínios de gerenciamento. Um domínio em um nível de gerenciamento pode ser dividido em sub-domínios. Da mesma forma, podem ser criados níveis de gerenciamento adicionais, à medida que a complexidade dos grafos de dependência afetem a disponibilidade do sistema. O número de objetos a serem submetidos a testes podem obedecer a determinado critério de seqüência de teste. Se em um determinado nível 1 de gerenciamento, um objeto é definido como o mais provável de ser uma falha primária, todos os objetos do nível 2 de gerenciamento pertencentes ao mesmo domínio deste objeto do nível 1 deverão ser testados. Somente após estes testes, o segundo objeto mais provável de ser uma falha primária no nível 1 de gerenciamento deverá ser testado. Este raciocínio pode ser realizado de forma análoga em qualquer dos níveis de gerenciamento.

O fator de cobertura permite modelar de forma mais adequada redes de comunicação reais. Para as aplicações realizadas neste capítulo, não se pode verificar a sensibilidade deste parâmetro na seleção ou na seqüência de teste dos objetos, pois o fator de cobertura foi considerado constante para todos os objetos durante as simulações. O capítulo 8 apresenta simulações em que é analisada a influência do fator de cobertura.

8. Verificação de Desempenho da Arquitetura e do Algoritmo Propostos

Neste capítulo são realizadas simulações com o objetivo de analisar o comportamento da disponibilidade da rede hipotética, baseada na arquitetura física apresentada no capítulo 7.

Para atingir esse objetivo foram realizadas as seguintes etapas.

- 1) Comparar o algoritmo *Divida e Conquiste Proposto* com o algoritmo *Divida e Conquiste*, através da análise dos objetos selecionados para a fase de teste durante as simulações nas diversas arquiteturas propostas: Centralizado, Distribuída e Hierárquica Proposta.
- 2) Comparar o tempo de execução do algoritmo *Divida e Conquiste Proposto*, quando aplicado as Arquiteturas Centralizada, Distribuída e Hierárquica Proposta, bem como, avaliar os objetos selecionados para a fase de teste em cada uma das arquiteturas analisadas.

8.1. Ambiente de simulação

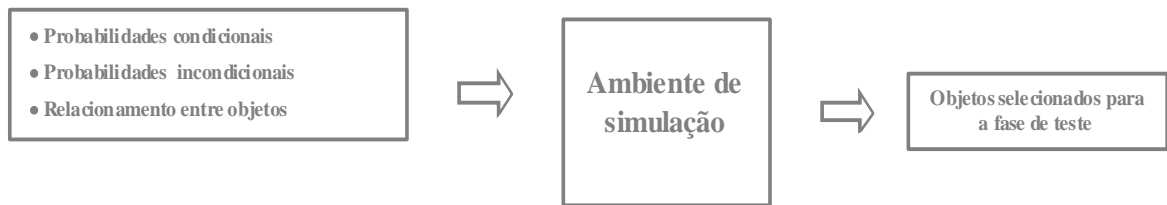
Os algoritmos *Divida e Conquiste* e *Divida e Conquiste Proposto* foram implementados e simulados no ambiente de programação C++ Builder 3.0 da Borland versão Standart e, utilizando um microcomputador PC Pentium-MMX 200MHz, 32 MB – SDRAM, HD 2.1GB e Sistema Operacional Windows 95.

Como entrada do ambiente de simulação existe um arquivo contendo as probabilidades condicionais, incondicionais, os fatores de cobertura (este dado foi utilizado apenas no algoritmo *Divida e Conquiste Proposto*) e uma descrição de dependência entre os objetos gerenciados.

Como saída do ambiente de simulação foi obtido um arquivo contendo os objetos selecionados para a fase de teste e a seqüência que estes objetos devem ser testados (válido somente para o algoritmo *Divida e Conquiste Proposto*).

A figura 8.1 apresenta um esquema geral da interface do ambiente de simulação.

Simulação do algoritmo *Divida e Conquiste*



Simulação do algoritmo *Divida e Conquiste Proposto*

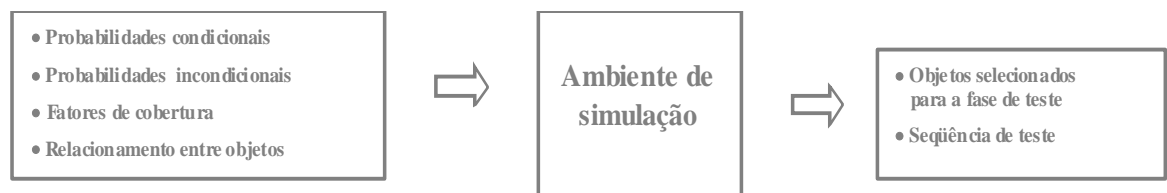


Figura 8.1: Interfaces do ambiente de Simulação

Os itens seguintes apresentam as etapas de simulação para cada um dos algoritmos analisados.

8.2. Simulações Comparativas entre o Algoritmo *Divida e Conquiste* e o *Divida e Conquiste Proposto*

O objetivo destas simulações é analisar o comportamento do *Algoritmo Divida e Conquiste Proposto* em relação ao algoritmo *Divida e Conquiste*, independente das Arquiteturas Centralizada, Distribuída e Hierárquica Proposta. Para tanto, avalia-se o conjunto de objetos selecionados para a fase de teste quando utilizado ambos os algoritmos, bem como, a seqüência de objetos para a fase de teste (apenas para o algoritmo *Divida e Conquiste Proposto*) quando é observado um certo número de alarmes ativos.

Para a realização destas simulações comparativas foi adotada seguinte suposição:

Os alarmes ativos gerados não se alteram durante a simulação do algoritmo de localização de falhas. Esta simplificação é necessária para a realização da simulação, nem sempre sendo válida em aplicações reais.

Parâmetros em Análise

Foram utilizados os seguintes parâmetros para a análise dos algoritmos:

- Sensibilidade à variação da probabilidade condicional.
- Sensibilidade à variação da probabilidade incondicional.
- Sensibilidade à variação do fator de cobertura.

Assim, utilizou-se os seguintes critérios para os testes:

Objetos selecionados

Considerando a arquitetura apresentada no capítulo 7, selecionam-se os objetos A_1 , C_1 e L_B , pois estes objetos apresentam as seguintes particularidades:

Objeto A_1 : não apresenta nenhum objeto dependente dele.

Objeto C_1 : um dos objetos que apresenta o maior número dependências na rede.

Objeto L_B : objeto que não depende de nenhum outro objeto, contendo apenas objetos dependentes dele.

Estes três tipos de objetos cobrem a grande maioria de dependências existentes em uma rede. As simulações foram realizadas em função dos parâmetros e objetos selecionados anteriormente.

Alarmes ativos

Os alarmes ativos foram selecionados em duas fases:

- Alarmes individuais: supondo os alarmes ativos individualmente e, verificando quais os objetos, e em que seqüência (apenas para o algoritmo *Divida e Conquiste Proposto*) foram selecionados para a fase de teste.
- Seqüência de Alarmes ativos: partindo de apenas um alarme ativo e, adicionando, em cada simulação, um novo alarme ao conjunto de alarmes ativos simultaneamente. Para cada caso, foram verificados quais objetos e, em que seqüência foram selecionados para a fase de teste.

8.2.1. Realização das Simulações

As simulações, conforme descrito a seguir, foram realizadas para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto* sempre considerando os critérios apresentados no item anterior.

- 1) Simulações considerando todos os objetos com iguais probabilidades condicional, incondicional e fator de cobertura (este parâmetro válido apenas para o algoritmo proposto).
- 2) Simulações considerando o objeto A_1 com probabilidade incondicional (p_i) maior que os outros objetos, sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.
- 3) Simulações considerando o objeto C_1 com probabilidade incondicional (p_i) maior que os outros objetos, sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.
- 4) Simulações considerando o objeto L_B com probabilidade incondicional (p_i) maior que os outros objetos, sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.

- 5) Simulações considerando o objeto A_1 com uma de suas probabilidades condicionais (p_{ij}) maior que os outros objetos (escolhido a maior probabilidade condicional entre os objetos A_1 e A_2), sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.
- 6) Simulações considerando o objeto C_1 com uma de suas probabilidades condicionais (p_{ij}) maior que os outros objetos (escolhido a maior probabilidade condicional entre os objetos C_1 e C_2), sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.
- 7) Simulações considerando o objeto L_B com uma de suas probabilidades condicionais (p_{ij}) maior que os outros objetos (escolhido a maior probabilidade condicional entre os objetos B_1 e L_B), sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.
- 8) Simulações variando o fator de cobertura do objeto A_1 (de 0% a 100%), supondo os alarmes ativos de E_1 a E_6 , sendo os demais objetos apresentando probabilidade condicional, incondicional e fator de cobertura iguais aos apresentados no capítulo 7.
- 9) Simulações variando o fator de cobertura do objeto C_1 (de 0% a 100%), supondo os alarmes ativos de E_1 a E_5 , sendo os demais objetos apresentando probabilidade condicional, incondicional e fator de cobertura iguais aos apresentados no capítulo 7.
- 10) Simulações variando o fator de cobertura do objeto L_B (de 0% a 100%), supondo os alarmes ativos de E_1 a E_8 , sendo os demais objetos apresentando probabilidade condicional, incondicional e fator de cobertura iguais aos apresentados no capítulo 7.

Para analisar a influência do fator de cobertura nas simulações 8, 9 e 10, utilizou-se os seguintes critérios de avaliação: o conjunto de alarmes ativos escolhido deve abranger vários objetos para a fase de teste e, o objeto com fator de cobertura igual a 100% fosse o primeiro na seqüência de teste.

Assim, selecionou-se como alarmes ativos de E_1 a E_6 para a simulação 8, de E_1 a E_5 para a simulação 9 e, de E_1 a E_8 para a simulação 10.

8.2.2. Análise das Simulações

Neste item são apresentadas as análises das diversas simulações apontadas anteriormente. As tabelas contendo os resultados destas simulações encontram-se no Anexo E desta dissertação.

- 1) Simulações com todos os objetos com iguais probabilidades condicional, incondicional e fator de cobertura.

Verificou-se, para os dois algoritmos em análise, que praticamente os mesmos objetos foram selecionados para a fase de teste. Fato importante a ser observado é que na seqüência de teste (para o algoritmo *Divida e Conquiste Proposto*), para a maioria dos alarmes ativos simultaneamente, os primeiros objetos a serem testados sempre pertenciam a um grande número de *clusters* de alarmes. Este resultado é bastante importante, pois objetos pertencentes a um grande número de *clusters* de alarmes apresentam a maior probabilidade de serem a fonte primária de falha.

- 2) Simulação com o objeto A_1 com probabilidade incondicional maior que os outros objetos, sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura iguais.

Durante a simulação do algoritmo *Divida e Conquiste* observou-se que o objeto A_1 não era selecionado para fase de teste. Este fato motivou a criação de dois critérios adicionais para a fase de união do algoritmo *Divida e Conquiste Proposto*. Os critérios foram: selecionar os objetos que cobrem o maior número de alarmes ativos simultaneamente e, selecionar os objetos com maior probabilidade incondicional apresentados no capítulo 6.

Observou-se que era mais coerente que os objetos com maior probabilidade incondicional fossem os primeiros a serem selecionados para fase de teste, quando os demais objetos tinham a probabilidade incondicional, condicional e fator de cobertura iguais. Este critério era fundamental quando o alarme ativo correspondesse ao objeto selecionado (por exemplo, ao alarme ativo E_1 corresponde ao objeto A_1), conforme pode ver observado na tabela 7.2 do capítulo 7.

Na simulação do algoritmo *Divida e Conquiste Proposto* verificou-se que o objeto A_1 foi selecionado em várias simulações, principalmente quando os alarmes E_1 e E_2 estavam ativos. Observou-se que o objeto A_1 foi selecionado como o primeiro objeto a ser testado nas fases de teste nas quais ele foi selecionado.

- 3) Simulação com o objeto C_1 com probabilidade incondicional (p_i) maior que os outros objetos, sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.

Observou-se que o objeto C_1 foi selecionado na maioria das simulações realizadas para o algoritmo *Divida e Conquiste Proposto*, quando comparado ao algoritmo *Divida e Conquiste*, que pouco selecionou o objeto C_1 para a fase de teste. Este fato ocorreu principalmente nos casos em que mais de um alarme foi considerado como ativo.

Observou-se também que o objeto C_1 foi selecionado como o primeiro objeto a ser testado nas fases de teste em que ele foi selecionado.

- 4) Simulação com o objeto L_B com probabilidade incondicional maior que os outros objetos, sendo os demais objetos com iguais probabilidade condicional, incondicional e fator de cobertura.

Observou-se que o objeto L_B foi selecionado em um número maior de vezes nas simulações realizadas para o algoritmo *Divida e Conquiste Proposto*, quando comparado ao algoritmo *Divida e Conquiste*. Quando existia mais de um alarme ativo, ambos os algoritmos selecionaram o objeto L_B para a fase de teste. O objeto L_B foi selecionado como o primeiro objeto a ser testado nas fases de teste em todos os casos em que houve mais de um alarme ativo.

- 5) Simulação considerando o objeto A_1 com uma de suas probabilidades condicionais maior que os outros objetos (escolhido a maior probabilidade condicional entre os objetos A_1 e A_2), sendo os demais objetos com iguais probabilidades condicional, incondicional e fator de cobertura.

- 6) O objeto A_2 foi selecionado para a fase de teste praticamente nas mesmas simulações para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto*. Entretanto, observou-se que quando foram considerados vários alarmes ativos simultaneamente, o algoritmo *Divida e Conquiste* apresentou um maior número de objetos selecionados para a fase de teste. Este fato é importante, pois um menor número de objetos a serem testados com a maior probabilidade de serem a fonte primária de falha é um dos objetivos a ser alcançado pelo gerenciamento de falhas.

Observou-se que o objeto A_2 foi selecionado para a fase de teste na maioria das simulações realizadas. Este resultado é bastante coerente, pois a dependência entre os objetos A_1 e A_2 apresenta maior valor no grafo de dependência.

- 7) Simulação considerando o objeto C_1 com uma de suas probabilidades condicionais maior que os outros objetos (escolhida a maior probabilidade condicional entre os objetos C_1 e C_2), sendo os demais objetos com iguais probabilidades condicional, incondicional e fator de cobertura.

O objeto C_2 foi selecionado para a fase de teste praticamente em todas as simulações para o algoritmo *Divida e Conquiste Proposto*, enquanto que para o algoritmo *Divida e Conquiste*, apareceu um número menor de vezes. Este resultado é bastante importante, pois o objeto C_2 apresenta a maior probabilidade de ser a fonte primária de falha.

O algoritmo *Divida e Conquiste Proposto* apresentou um maior número de objetos selecionados para a fase de teste quando comparado ao algoritmo *Divida e Conquiste*, principalmente quando vários alarmes foram considerados como ativos. Uma possível justificativa, seria devido ao fato da adição de dois novos critérios na fase de união do algoritmo *Divida e Conquiste Proposto* (selecionar as duas partições que sejam cobertas pelo maior número de alarmes ativos e, selecionar as duas partições que apresentem a maior probabilidade incondicional, apresentados no capítulo 6).

Observa-se que quando foi considerado alarmes isolados, o objeto C_2 praticamente não foi selecionado no algoritmo *Divida e Conquiste*, em oposição, no algoritmo *Divida e Conquiste Proposto*. Neste último o objeto C_2 foi selecionado na maioria dos alarmes

ativos isolados. Este fato também pode ser justificado pela adição dos dois novos critérios na fase de união do algoritmo *Divida e Conquiste Proposto*. Tal característica representa um aspecto positivo do algoritmo *Divida e Conquiste*.

- 8) Simulação considerando o objeto L_B com umas de suas probabilidades condicionais maior que os demais objetos (escolhido a maior probabilidade condicional entre os objetos B_1 e L_B), sendo os demais objetos com iguais probabilidades condicional, incondicional e fator de cobertura.

O objeto L_B apareceu praticamente em todas as simulações para o algoritmo *Divida e Conquiste* e *Divida Conquiste Proposto*, tanto para os alarmes isolados, quanto para vários alarmes ativos simultaneamente. Este fato justifica-se pois o objeto L_B não apresenta dependência de nenhum outro objeto da rede sendo, portanto, coerente a sua presença em quase todas as seqüências de testes a serem utilizadas.

- 9) Simulação variando o fator de cobertura do objeto A_1 (de 0% a 100%), supondo os alarmes ativos de E_1 a E_6 , sendo os demais objetos apresentando probabilidades condicional, incondicional e fator de cobertura iguais aos apresentados no capítulo 7.

A figura 8.2 apresenta o resultado obtido para esta simulação, observa-se que enquanto o fator de cobertura foi igual ou superior a 30%, a lista de objetos para a fase de teste não se alterou, sendo o objeto A_1 o primeiro a ser testado. Verificou-se que quando o fator de cobertura foi igual a 20%, o objeto A_1 foi selecionado na quarta posição de teste e, quando o fator de cobertura passou a ser 10%, o objeto A_1 foi selecionado na quinta posição de teste. É importante observar que quando o fator de cobertura foi considerado igual a 0%, o objeto A_1 foi retirado da lista de teste. Uma possível justificativa para tal evento possa ser devido ao baixo número de dependências envolvendo o objeto A_1 .

Pelo resultado obtido, percebe-se a grande influência do fator de cobertura na posição dos objetos selecionados para a fase de teste. O fator de cobertura auxilia na representação de situações reais, pois é extremamente difícil ter-se um objeto gerenciado (equipamento ou *software*) com fator cobertura igual a 100%.

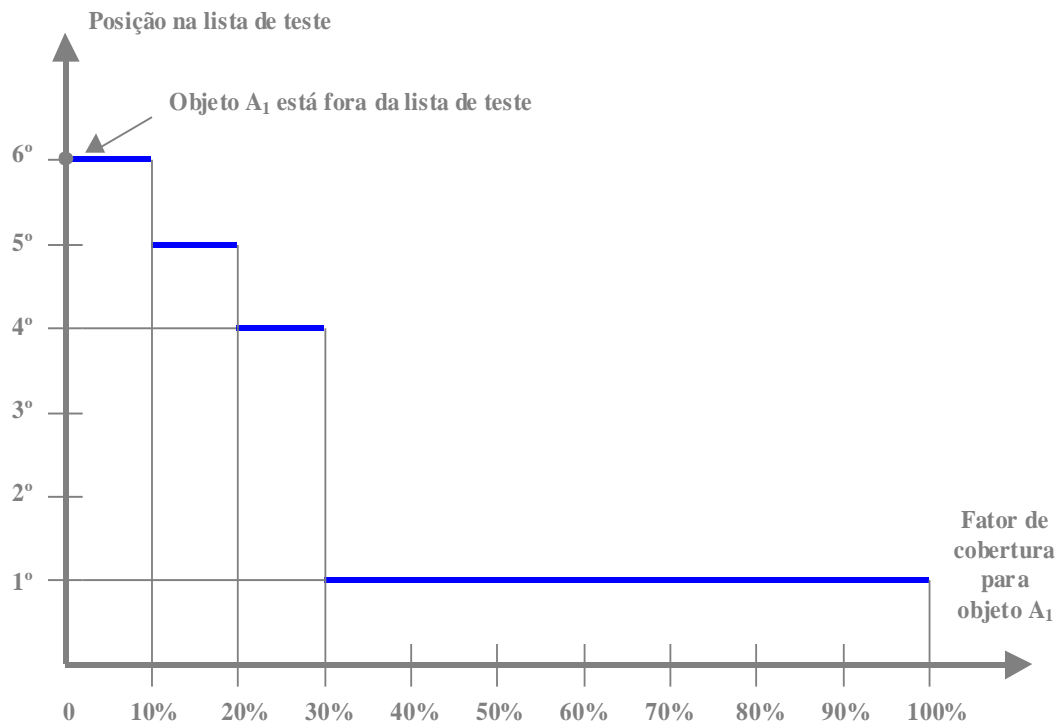


Figura 8.2: Simulação do gráfico do objeto A₁ variando o seu fator de cobertura

- 10) Simulação variando o fator de cobertura do objeto C₁ (de 0% a 100%), supondo os alarmes ativos de E₁ a E₆, sendo os demais objetos apresentados probabilidades condicional, incondicional e fator de cobertura iguais ao apresentados no capítulo 7.

A figura 8.3 apresenta o resultado obtido para esta simulação. O objeto C₁ apareceu em todas as simulações, mesmo quando o seu fator de cobertura foi igual a zero. Uma possível justificativa para tal evento possa ser devido ao grande número de dependências envolvendo o objeto C₁.

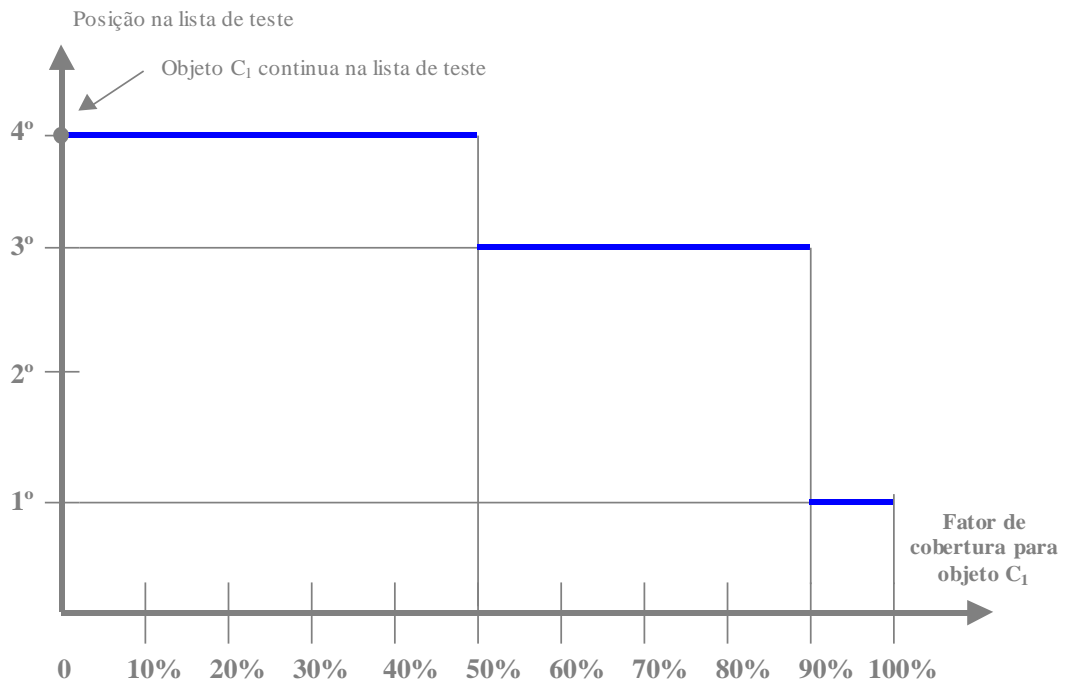


Figura 8.3: Simulação do gráfico do objeto C₁ variando o seu fator de cobertura

- 11) Simulação variando o fator de cobertura do objeto L_B (de 0% a 100%), supondo os alarmes ativos de E₁ a E₈, sendo os demais objetos apresentando probabilidades condicional, incondicional e fator de cobertura iguais aos apresentados no capítulo 7. A figura 8.4 apresenta o resultado obtido para esta simulação. Observa-se que o objeto L_B permaneceu na segunda posição enquanto o fator de cobertura esteve entre 90% a 100%. O objeto L_B foi retirado da lista de teste, quando o fator de cobertura foi igual ou inferior a 50%, mesmo o objeto L_B não dependendo de qualquer outro objeto. Uma possível justificativa para tal evento possa ser devido ao fato de que o fator de cobertura do objeto L_B tem grande influência nas probabilidades condicionais e incondicionais na fase de união. Assim os objetos gerados na fase de união em que o objeto L_B pertenceu, apresentaram valores pequenos o suficiente para que na fase de partição, estes objetos união (contém o objeto L_B) não fossem selecionados.

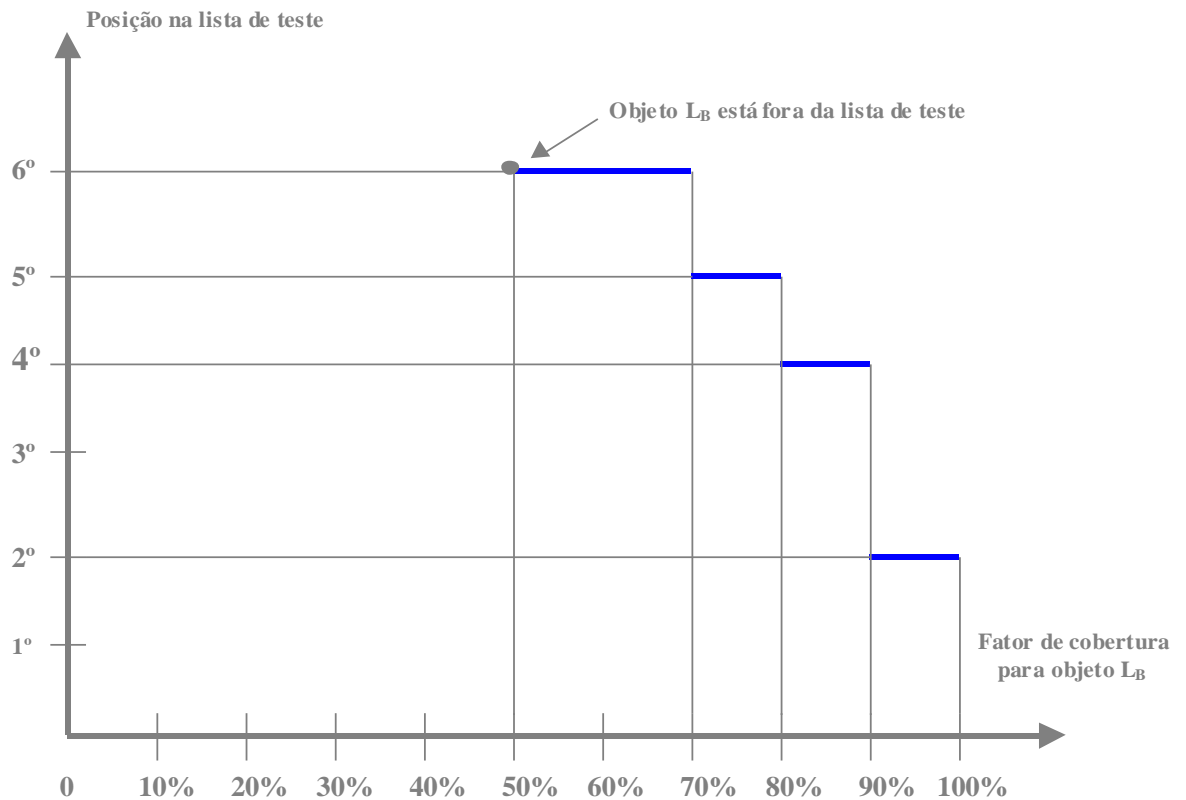


Figura 8.4: Simulação do gráfico do objeto L_B variando o seu fator de cobertura

8.3. Simulações Comparativas entre as Arquiteturas de Gerenciamento

O objetivo destas simulações é verificar, de modo comparativo, o tempo total de execução do algoritmo de localização de falhas *Divida e Conquista Proposto* sobre as Arquiteturas Centralizada, Distribuída e Hierárquica Proposta, bem como verificar, de forma comparativa, quais objetos foram selecionados para a fase de teste em cada uma das arquiteturas analisadas.

Para tanto, foram utilizados os seguintes parâmetros para a análise das arquiteturas:

- Tempo total da localização dos objetos mais prováveis de serem fonte primária de falha.
- Sequência de objetos a serem testados.

Para a realização das simulações foram adotadas as seguintes suposições:

- As redes simuladas apresentam a mesma arquitetura física para as Arquiteturas Centralizada, Distribuída e Hierárquica.
- Os grafos de dependência para todas as arquiteturas simuladas apresentam os mesmos valores de probabilidades condicionais, incondicionais e fatores de cobertura.
- Para ter-se como comparar os tempos de localização de falhas para as Arquiteturas de Gerenciamento Centralizado, Distribuído e Hierárquico Proposto, estas arquiteturas apresentam a mesma forma de expansão de sua arquitetura física, ou seja, foram inseridos os mesmos objetos gerenciáveis e, nas mesmas conexões físicas para todas as arquiteturas de gerenciamento simuladas. Da mesma forma, foi suposto que os objetos falhos em cada passo de simulação eram idênticos para todas as arquiteturas simuladas.
- Os alarmes ativos gerados não se alteram durante a simulação do algoritmo de localização de falhas.
- O tempo de propagação das informações entre objetos gerenciados e sub-gerentes (válido apenas para as Arquiteturas Distribuída e Hierárquica), entre sub-gerentes e, entre estes e o Gerente Geral da rede (válido apenas para a Arquitetura Distribuída), não foi considerado durante a simulação das arquiteturas. Este critério foi adotado, pois o tempo de propagação foi considerado constante e igual para todas as arquiteturas simuladas e, portanto, não influenciando nas simulações realizadas.
- Os sub-gerentes (válido para as Arquiteturas Distribuída e Hierárquica) e, o Gerente Geral (válido apenas para a Arquitetura Distribuída) da rede estão fisicamente separados e, portanto, neste caso, os algoritmos de localização de falha em cada sub-gerente e Gerente Geral da rede podem ser executados paralelamente, ou seja, a rede pode ser considerada um sistema distribuído e concorrente.
- O tempo de localização de falhas foi baseado no número de operações realizadas pelo algoritmo *Divida e Conquiste Proposto*, pois este mesmo algoritmo foi utilizado em todas as arquiteturas de gerenciamento. Adotou-se este método como mais adequado para se medir o tempo de execução do algoritmo, independentemente da plataforma de *hardware* do sistema de gerenciamento.

- As fontes primárias de falha em cada passo das simulações das Arquiteturas Centralizada, Distribuída e Hierárquica foram sempre adotadas nos mesmos pontos da rede.

8.3.1. Realização das Simulações

Para a realização das simulações das Arquiteturas Centralizada, Distribuída e Hierárquica foi utilizado a arquitetura básica apresentada no capítulo 7. A expansão da rede foi realizada utilizando o nó A como elemento de simetria, adicionando um ou mais vezes este elemento a cada passo de simulação, conforme apresentado na figura 8.5.

Foram assumidas as seguintes dependências entre os nós:

- Primeiro passo de simulação: o nó A depende do nó B.
- Segundo passo de simulação: além das dependências do primeiro passo, o nó B depende do nó C e, o nó C depende do nó A.
- Terceiro passo de simulação: além das dependências do segundo passo, o nó B depende do nó D, o nó D depende do nó E, o nó E depende do nó F e, o nó F depende do nó D.
- Quarto passo de simulação: além das dependências do terceiro passo, o nó F depende do nó H, o nó G depende do nó H, o nó H depende do nó I, o nó I depende do nó G e, o nó G depende do nó C.

Foi assumido também que as conexões entre os objetos apresentam probabilidade condicional e incondicional igual a 0,02 e fator de cobertura igual a 0,8.

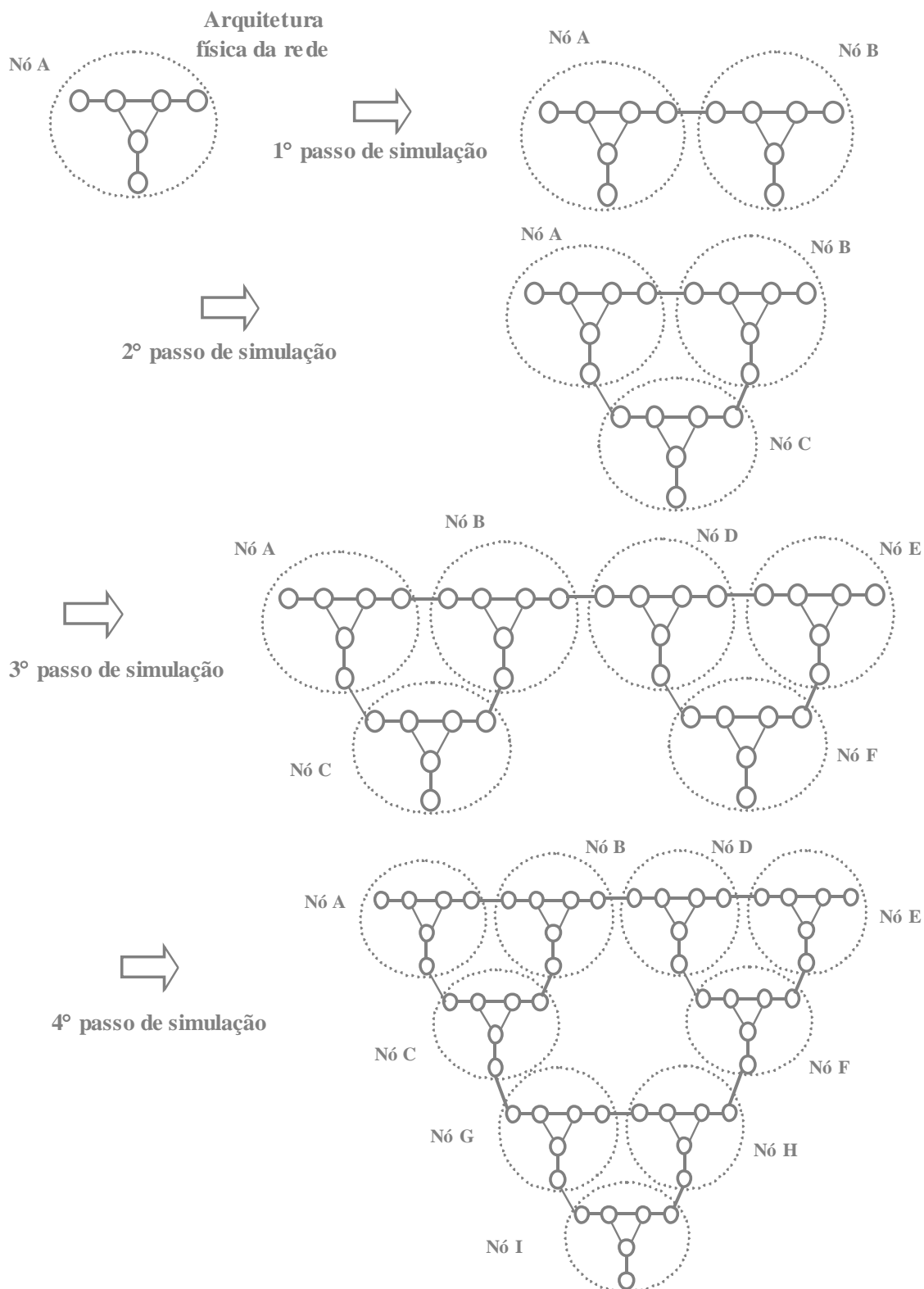


Figura 8.5 Expansão da Arquitetura Física

8.3.1.1. Simulação para a Arquitetura Centralizada

As figuras de 8.6 a 8.9 apresentam os passos realizados para a simulação da Arquitetura Centralizada, nas quais são apresentadas os grafos de dependência em cada passo da simulação. O grafo inicialmente utilizado é idêntico ao apresentado no capítulo 7, figura 7.3. Foi considerado como fonte primária de falha o objeto LD_1 , que representa o link entre domínios A e B, pois uma falha neste objeto pode provocar falhas em objetos internos a cada domínio.

O tempo de execução do algoritmo *Divida e Conquiste Proposto* baseou-se na contabilização do número de operações realizadas pelo único gerente de rede que possui o algoritmo de localização de falhas.

Os objetos que representam as conexões entre os nós apresentado na figura 8.5 e, as respectivas dependências geradas entre estes objetos está apreseto na figura 8.6 em cor diferenciada.

No primeiro passo da simulação centralizada a seguir, foi considerado como falho o objeto que representam o link LD_1 entre os domínios A e B.

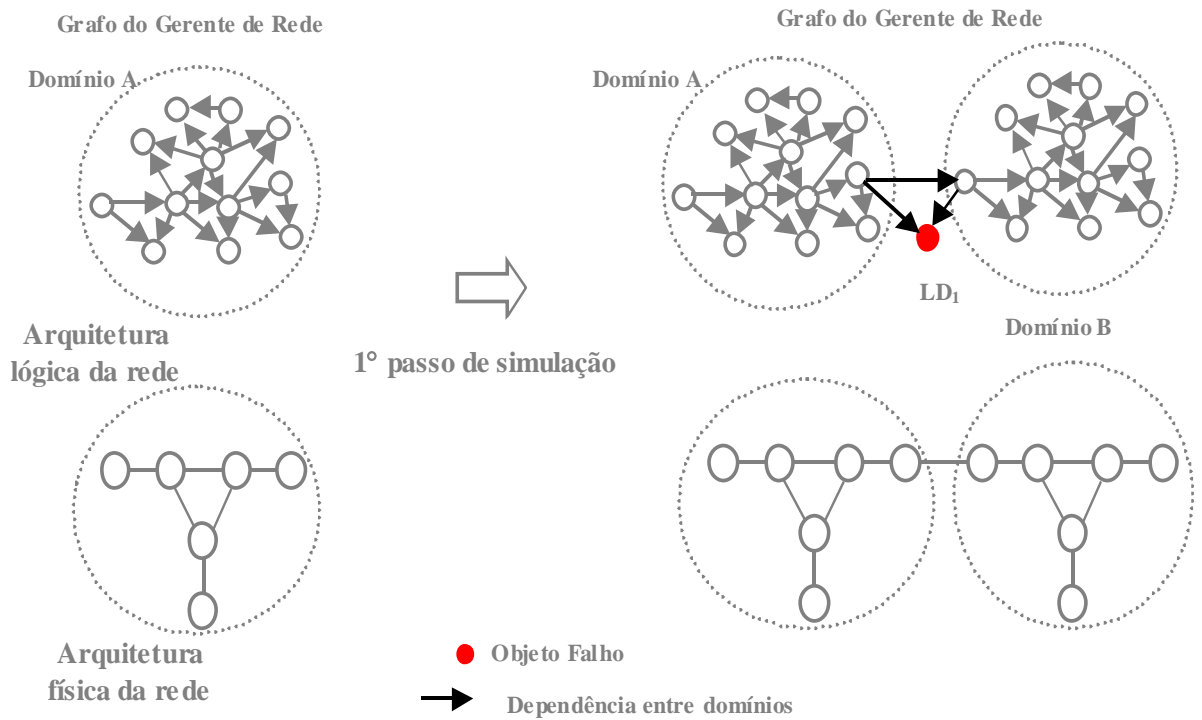


Figura 8.6: Expansão de uma rede utilizando Gerenciamento Centralizado - 1º Passo

No segundo passo da simulação centralizada, foram considerados como falhos (falha múltipla) os objetos que representam os links LD₁, LD₂ e LD₃ entre os domínios A e B, B e C e, entre C e A, conforme apresentado na figura 8.7.

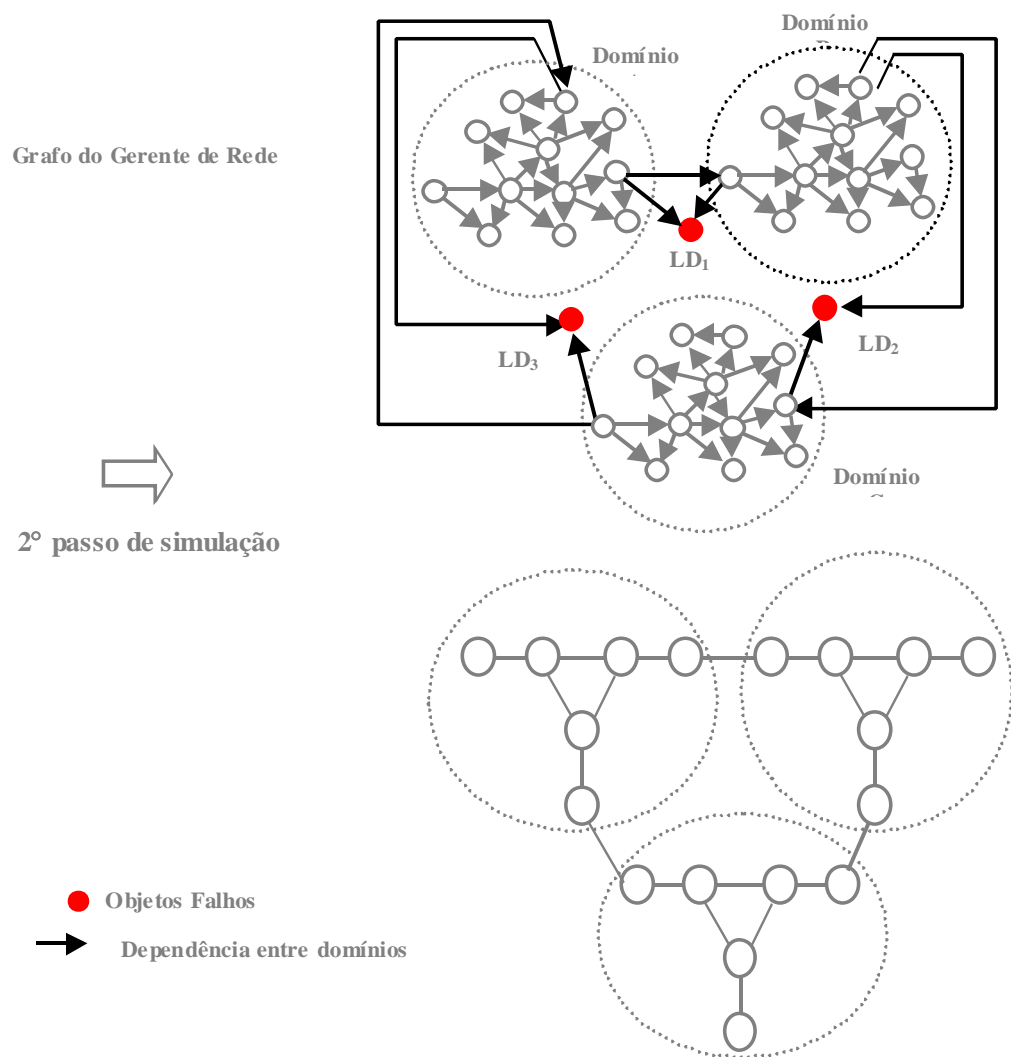


Figura 8.7: Expansão de uma rede utilizando Gerenciamento Centralizado - 2º Passo

No terceiro passo da simulação centralizada a seguir, foi considerado como falho o objeto que representa o link LD₄ entre os domínios B e D, conforme apresentado na figura 8.8.

Grafo do Gerente de Rede

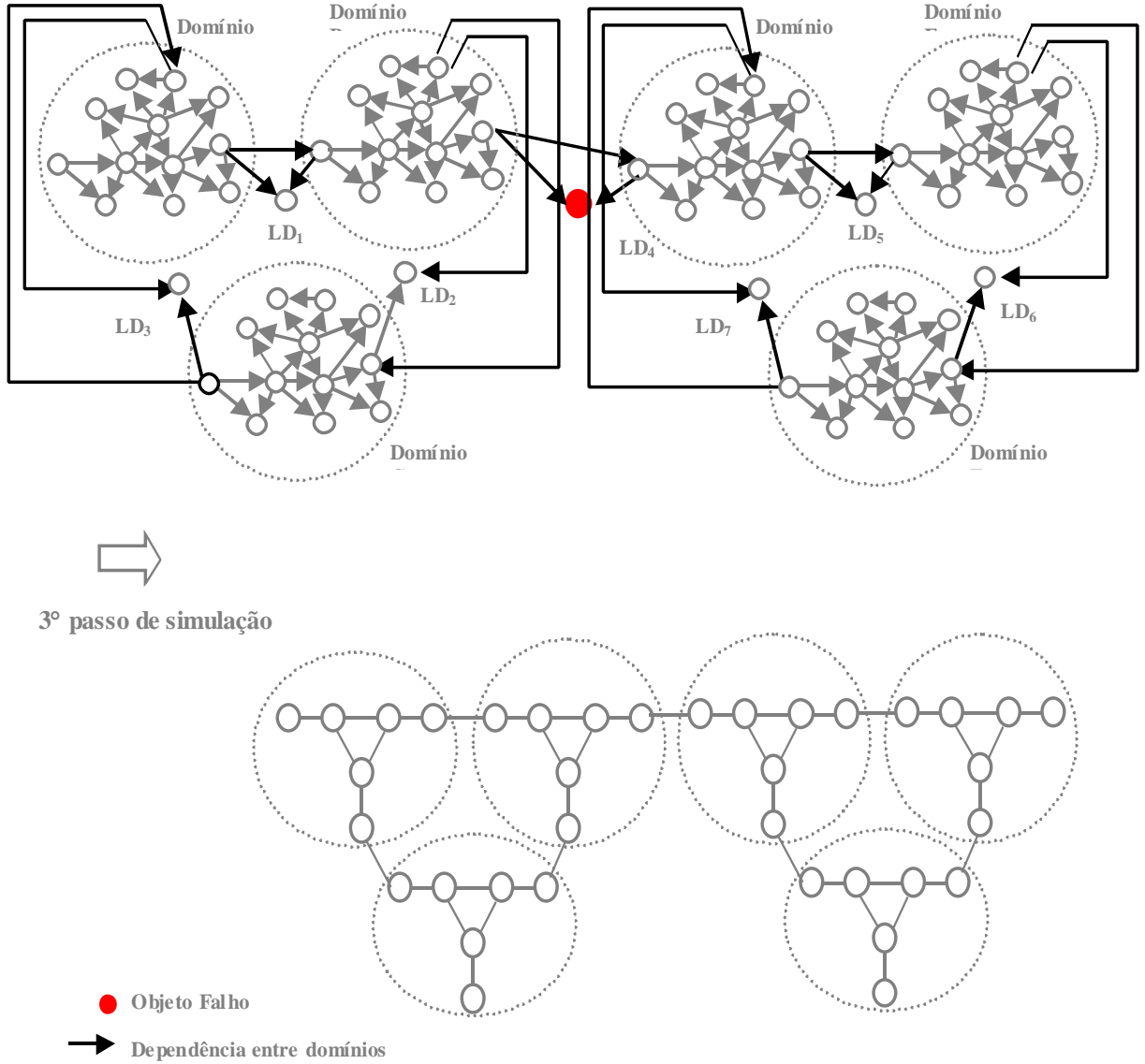


Figura 8.8: Expansão de uma rede utilizando Gerenciamento Centralizado - 3º Passo

No quarto passo da simulação centralizada a seguir, foram considerado como falhos os objetos que representam os links LD₄, LD₈ e LD₁₂ entre os domínios B e D, F e H e, entre C e G, conforme apresentado na figura 8.9.

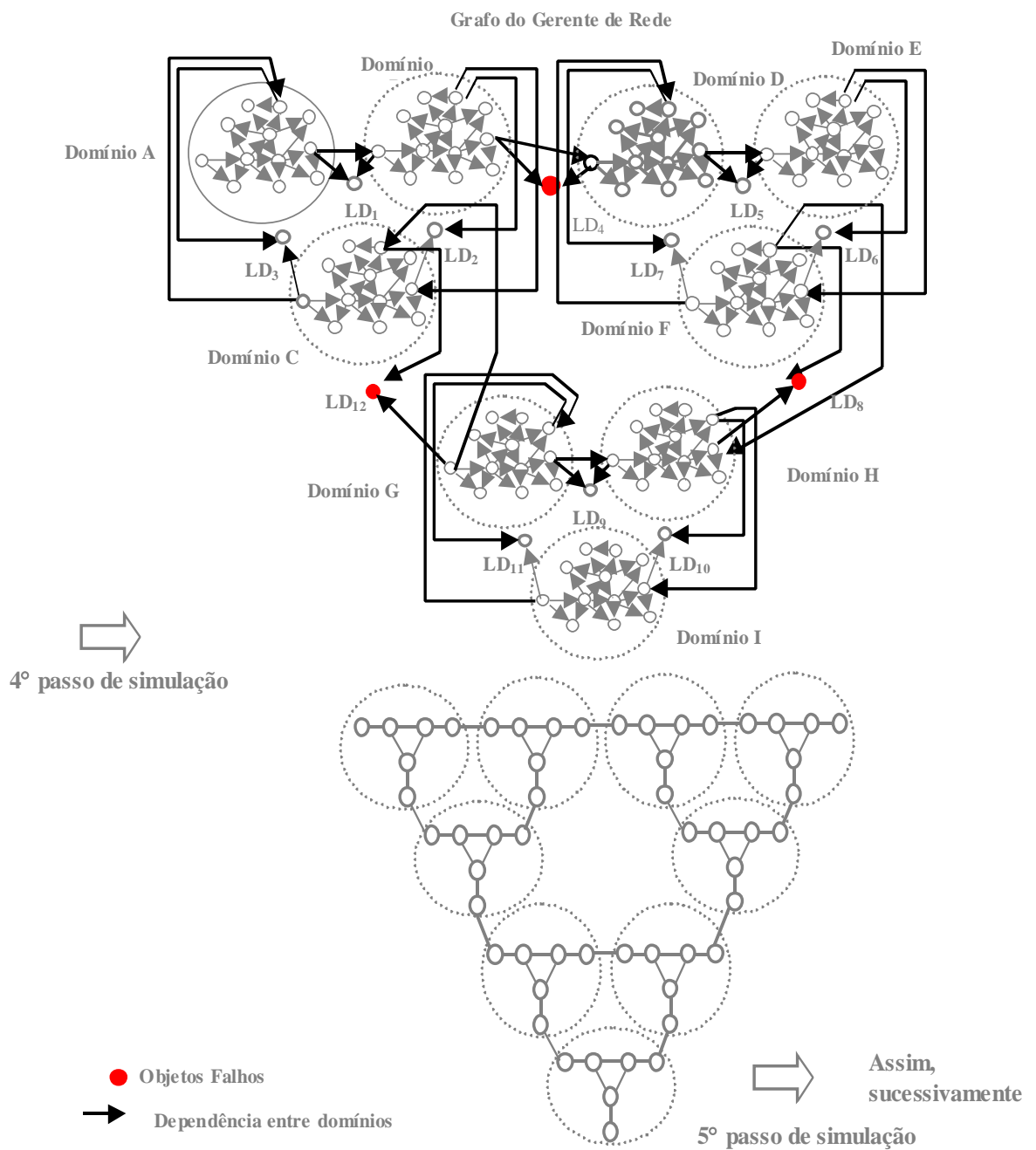


Figura 8.9: Expansão de uma rede utilizando Gerenciamento Centralizado - 4º Passo

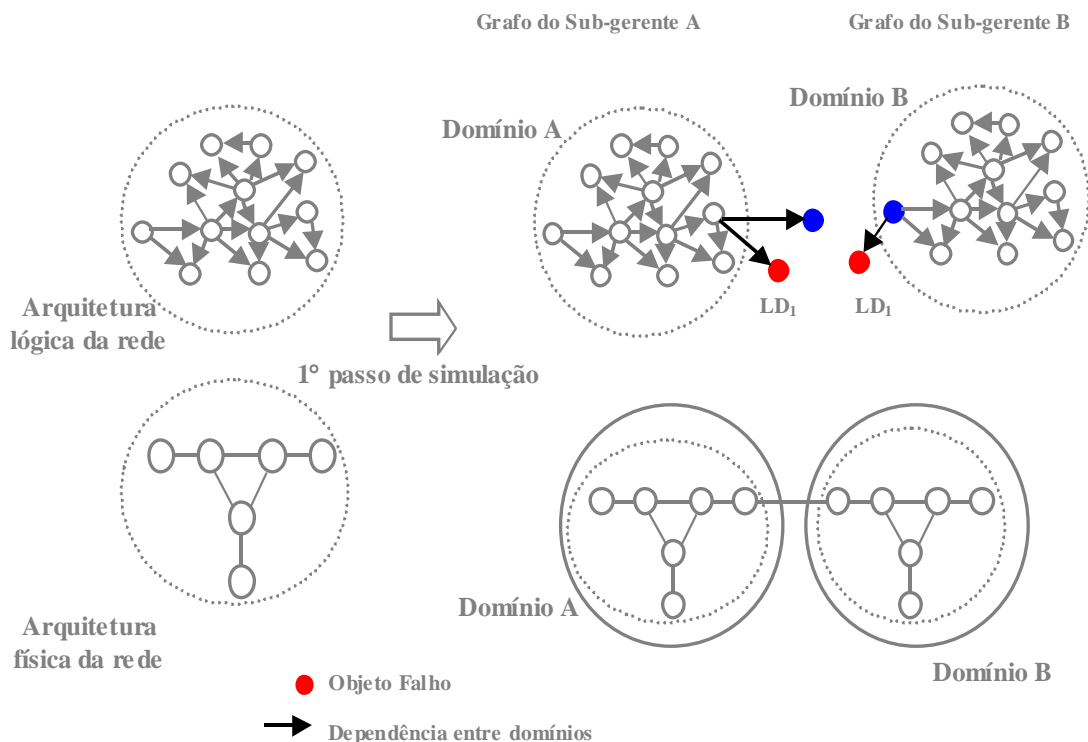
8.3.1.2. Simulação para a Arquitetura Distribuída

As figuras de 8.10 a 8.13 apresentam os passos realizados para a simulação distribuída. É apresentado também o grafo de dependência para cada sub-gerente de rede, gerado em cada

passo de simulação. O grafo inicialmente utilizado pelo Sub-gerente A é idêntico ao apresentado no capítulo 7, figura 7.3 . O grafo de dependência em cada domínio de gerenciamento possui objetos que representam as conexões entre os nós da figura 8.5. Os objetos idênticos, mas pertencentes a domínios diferentes, possuem a mesma cor de preenchimento, conforme apresentado na figura 8.10.

O tempo de execução do algoritmo *Divida e Conquiste* baseou-se na contabilização do número de operações realizadas pelos vários sub-gerentes. A partir deste fato, considerou-se o tempo de execução para a Arquitetura Distribuída, àquela em que o sub-gerente apresentou o maior número de operações realizadas.

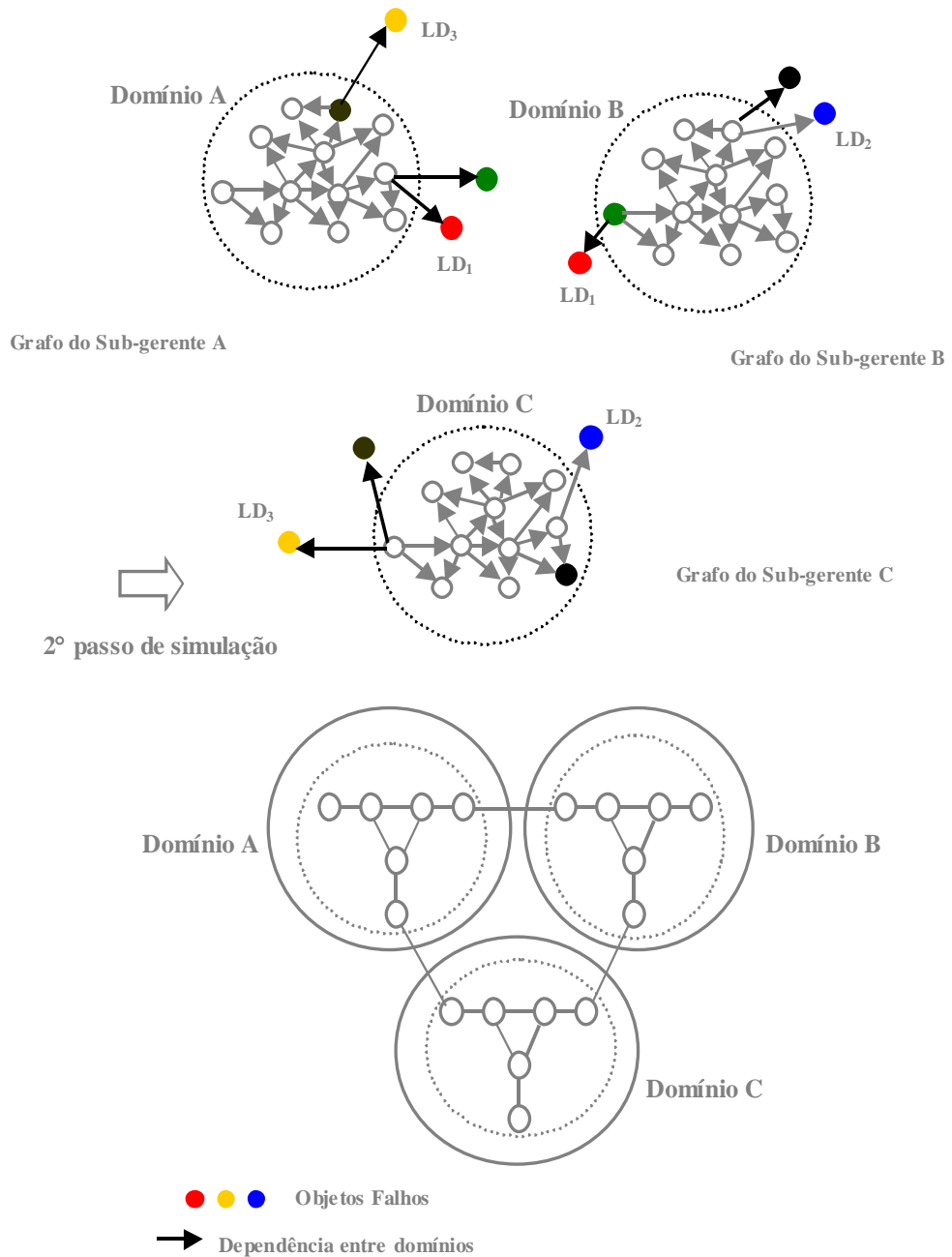
Igualmente ao primeiro passo da simulação centralizada, foi considerado como fonte primária de falha, no primeiro passo da simulação distribuída, o objeto LD₁, que representa o link entre domínios A e B. Uma falha neste objeto pode provocar falhas em objetos internos a cada domínio. O primeiro passo da simulação está apresentada na figura 8.10.



Objetos com a mesma cor de preenchimento, mas em domínios diferentes, representam o mesmo objeto gerenciado

Figura 8.10: Expansão de uma rede utilizando Gerenciamento Distribuído - 1º Passo

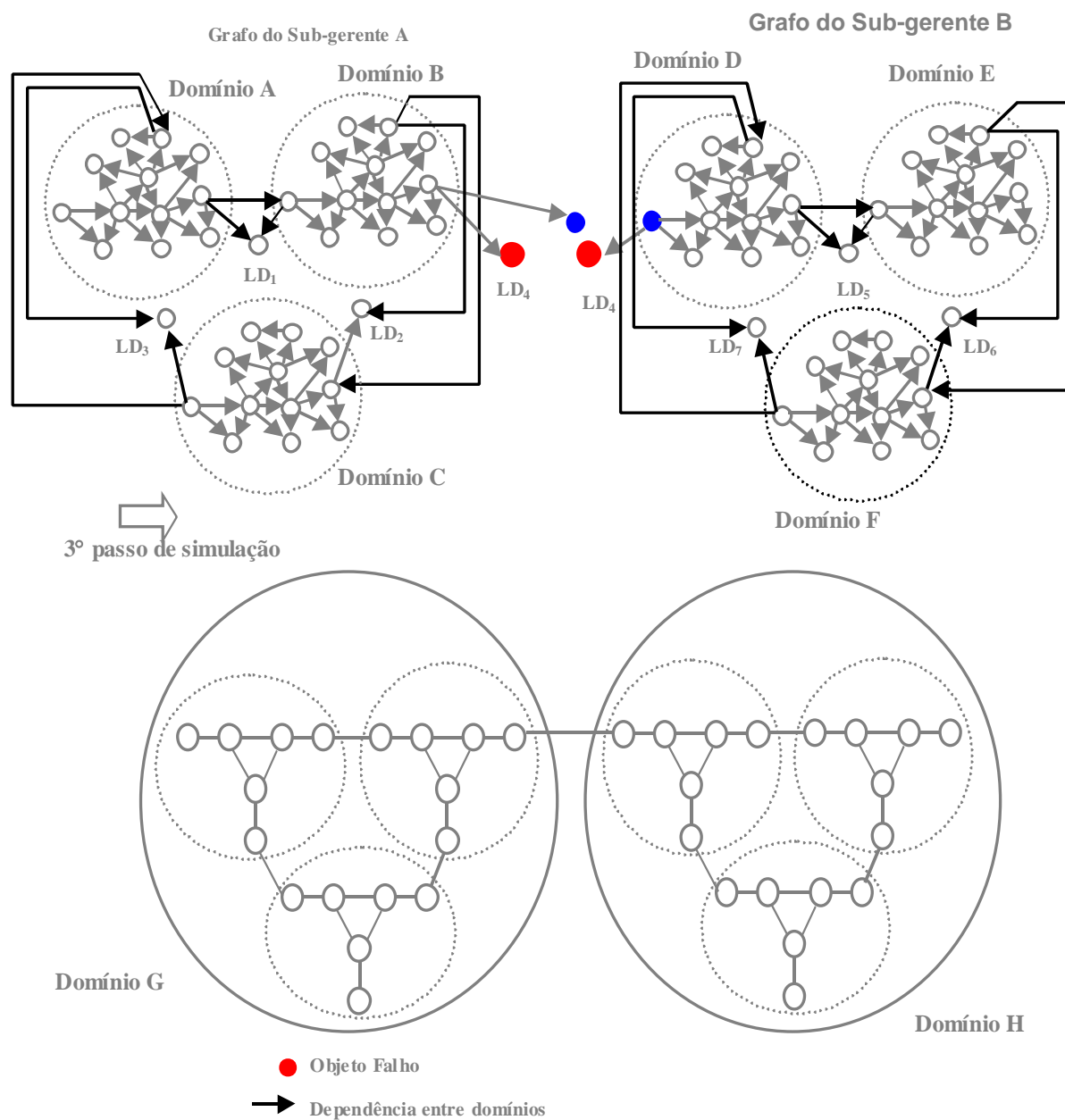
Igualmente ao segundo passo da simulação centralizada, foram considerados como falhos, no segundo passo da simulação distribuída, os objetos que representam os links LD₁, LD₂ e LD₃ entre os domínios A e B, B e C e, entre C e A, conforme figura 8.11.



Objetos com a mesma cor de preenchimento, mas em domínios diferentes, representam o mesmo objeto gerenciado

Figura 8.11: Expansão de uma rede utilizando Gerenciamento Distribuído - 2º Passo

Igualmente ao terceiro passo simulação centralizada, foi considerado como falho no terceiro passo da simulação distribuída, o objeto que representa o link LD₄ entre os domínios B e D, conforme apresentado na figura 8.12.



Objetos com a mesma cor em de preenchimento, mas em domínios diferentes, representam o mesmo objeto gerenciado

Figura 8.12: Expansão de uma rede utilizando Gerenciamento Distribuído - 3º Passo

Igualmente ao quarto passo da simulação centralizada, foram considerados como falhos no quarto passo da simulação distribuída, os objetos que representam os links LD₄, LD₈ e LD₁₂ entre os domínios B e D, F e H e, entre G e C, conforme apresentado na figura 8.13.

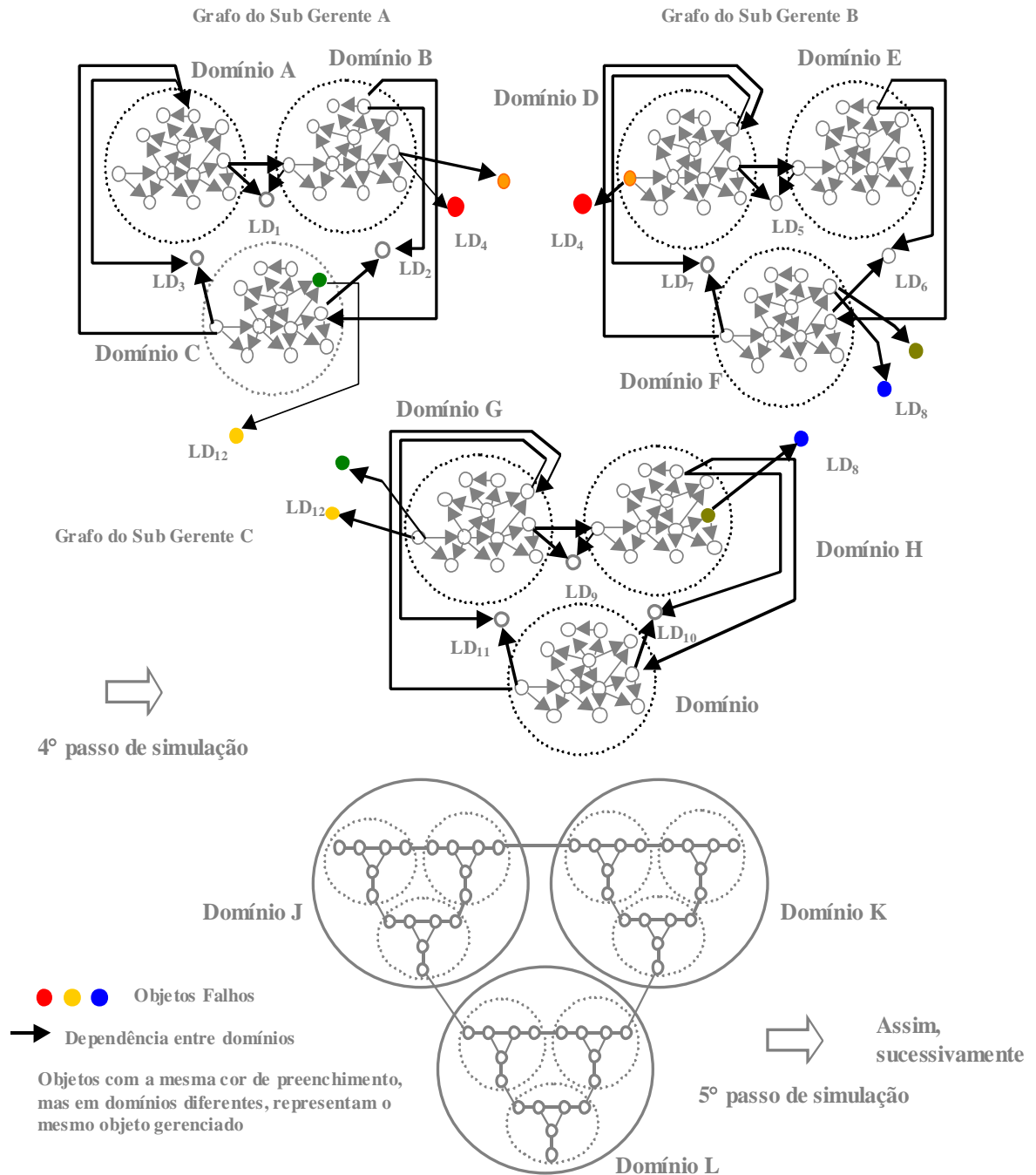


Figura 8.13: Expansão de uma rede utilizando Gerenciamento Distribuído - 4º Passo

8.3.1.3. Simulação para a Arquitetura Hierárquica Proposta

As figuras de 8.14 a 8.17 apresentam os passos realizados para a simulação hierárquica. É também apresentado o grafo de dependência para os sub-gerentes em cada nível de gerenciamento de rede gerado em cada passo de simulação. O grafo inicialmente utilizado pelo Sub-Gerente A é idêntico ao apresentado no capítulo 7, figura 7.3

Da mesma maneira que na simulação distribuída, o grafo de dependência em cada nível e em cada domínio de gerenciamento, apresenta os objetos que representam as conexões entre os nós da figura 8.5, bem como os objetos pertencentes a outros domínios. Cada domínio representado no nível de gerenciamento 1 foi representado pelo grafo de dependência obtido no capítulo 7. Quando dois domínios foram interligados, também foi criado o nível de gerenciamento 2, superior ao nível de gerenciamento 1. Da mesma forma, quando domínios do nível de gerenciamento 2 foram interligados, foi criado o nível de gerenciamento 3. A figura 8.14 apresenta em maiores detalhes o primeiro passo da simulação hierárquica.

O tempo de execução do algoritmo *Divida e Conquiste Proposto* foi contabilizado da seguinte forma:

- 1) Um objeto gerenciado em um determinado nível de gerenciamento representa um domínio, no nível de gerenciamento imediatamente inferior. Assim, a probabilidade condicional, incondicional e o fator de cobertura de cada objeto gerenciado em um determinado nível de gerenciamento, é calculado a partir de seu correspondente domínio no nível de gerenciamento imediatamente inferior.
- 2) Dada o exposto em 1, em cada nível de gerenciamento, considerou-se o maior número de operações na fase de união do algoritmo *Divida e Conquiste Proposto*, entre diversos domínios deste nível. Este aspecto foi adotado pois supõem-se que todos os domínios, em um mesmo nível de gerenciamento, executam seus algoritmos *Divida e Conquiste Proposto* de forma concorrente.

- 3) A fase de particionamento foi realizada apenas para os domínios de gerenciamento que foram selecionados para a fase de teste no nível de gerenciamento imediatamente superior. Este domínios representam a maior probabilidade de possuírem a fonte primária de falha.
- 4) Dada o exposto em 3, em cada nível de gerenciamento, considerou-se o maior o número de operações na fase de particionamento do algoritmo *Divida e Conquiste Proposto*, entre os diversos domínios deste nível. Este aspecto foi adotado pois supõem que todos os domínios, em um mesmo nível de gerenciamento, executam seus algoritmos *Divida e Conquiste Proposto* de forma concorrente.
- 5) O tempo total de execução do algoritmo *Divida e Conquiste Proposto* para a Arquitetura Hierárquica Proposta, foi obtido através da soma das operações em todos os níveis de gerenciamento obtidos através dos itens 2 e 4 anteriores.

Igualmente ao primeiro passo das simulações anteriores, foi considerado como fonte primária de falha no primeiro passo da simulação distribuída, o objeto LD_1 que representa o link entre os domínios A e B, conforme apresentado na figura 8.14.

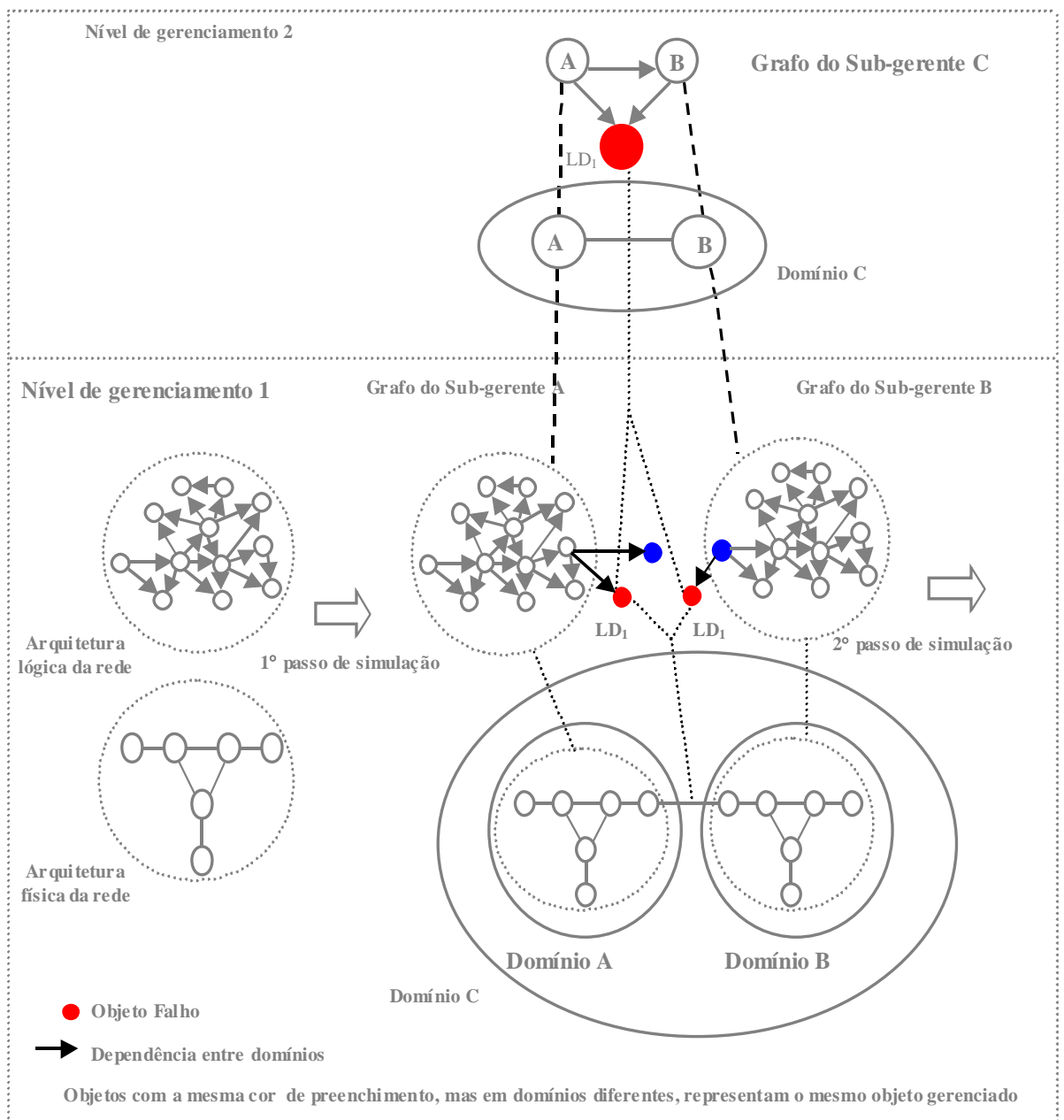


Figura 8.14: Expansão de uma rede utilizando Gerenciamento Hierárquico - 1º Passo

Igualmente ao segundo passo das simulações anteriores, foram considerados como fontes primárias de falhas (falhas múltiplas) no segundo passo da simulação distribuída, os objetos LD₁, LD₂ e LD₃ que representam os links entre domínios A e B, B e C, e entre C e A, conforme apresentado na figura 8.15.

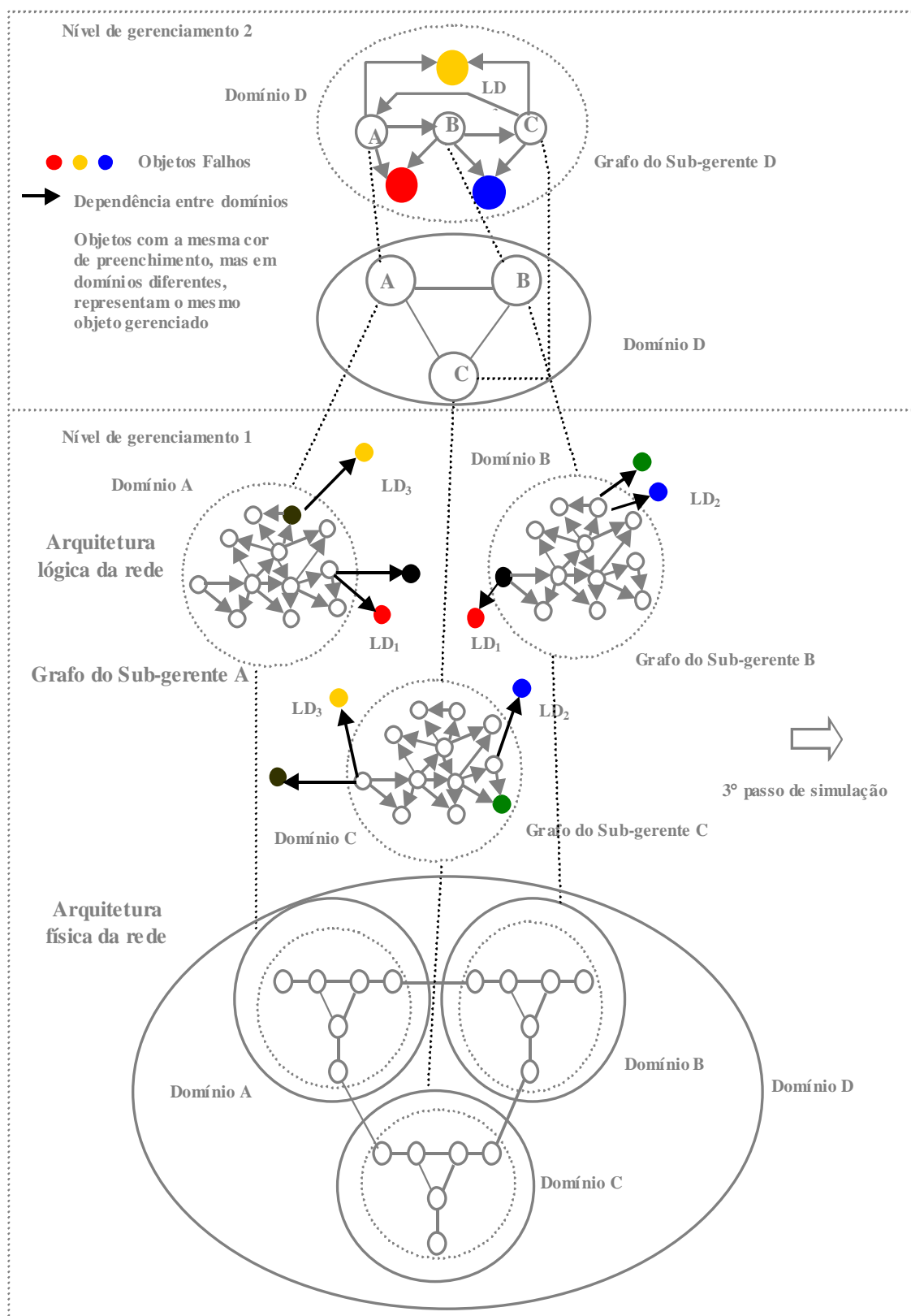


Figura 8.15: Expansão de uma rede utilizando Gerenciamento Hierárquico - 2º Passo

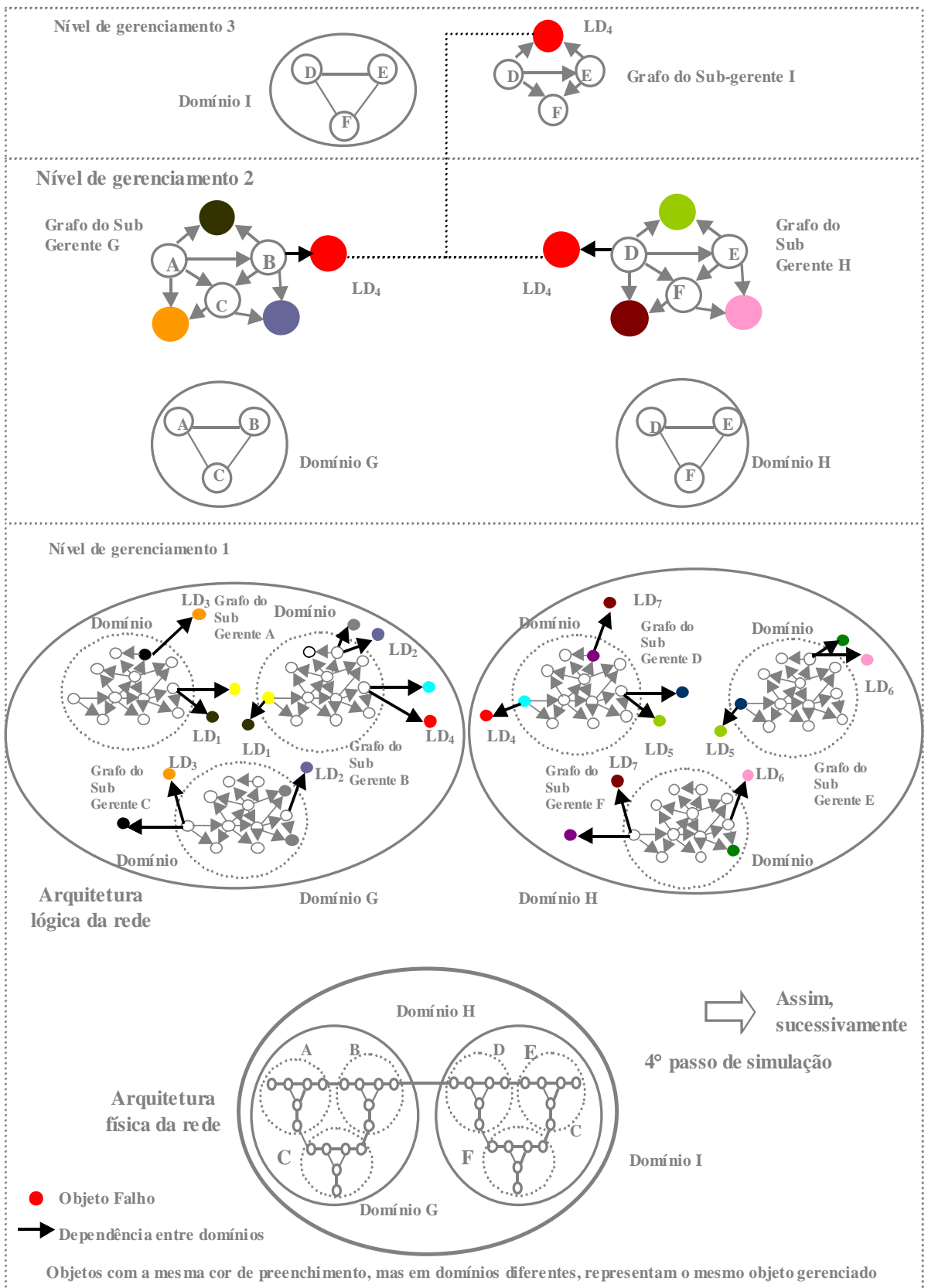


Figura 8.16: Expansão de uma rede utilizando Gerenciamento Hierárquico - 3º Passo

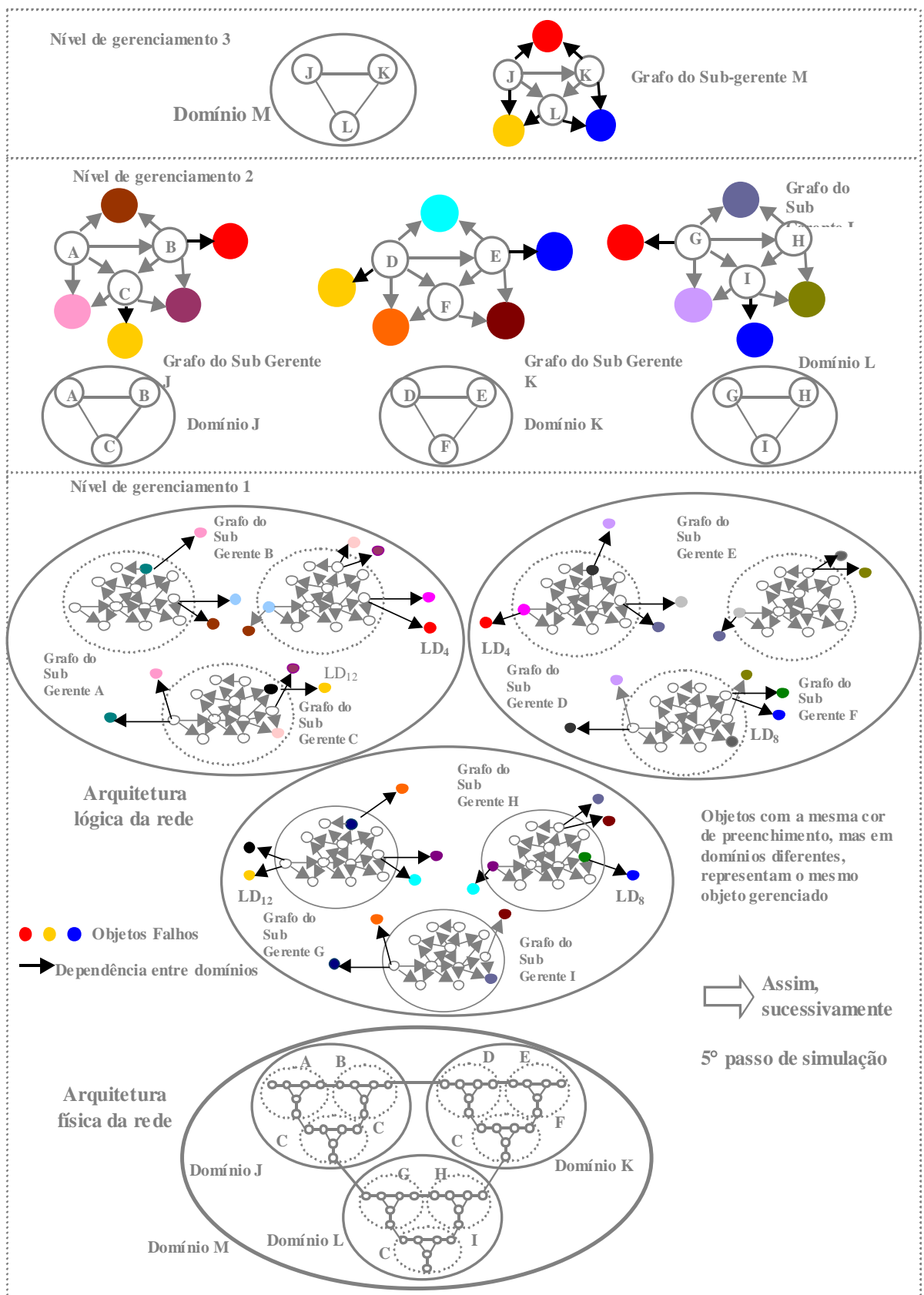


Figura 8.17: Expansão de uma rede utilizando Gerenciamento Hierárquico - 4º Passo

No terceiro passo das simulações foi considerado como fonte primária de falha no segundo passo da simulação distribuída, o objeto LD₄, que representa o link entre domínios B e D, conforme apresentado na figura 8.16.

No quarto passo das simulações foram considerado como fontes primárias de falhas no terceiro passo da simulação distribuída, os objetos LD₄, LD₈ e LD₁₂ que representam os links entre domínios B e D, F e H e, entre G e C, conforme apresentado na figura 8.17.

8.3.2. Resultados e Análise das Simulações

Em cada passo das simulações o número total de objetos gerenciados está apresentado na tabela a seguir:

Passo \ Arquitetura	Centralizado	Distribuído	Hierárquico Proposto
1 ^o	25	28	30
2 ^o	39	45	48
3 ^o	79	81	87
4 ^o	120	126	159

Tabela 8.1: Número de objetos gerenciados em cada passo da simulação

O número total de objetos gerenciados em cada arquitetura de gerenciamento foi obtido através das figuras 8.14 a 8.17.

A tabela a 8.2 apresenta o número de operações do algoritmo *Divida e Conquiste Proposto*, representando o tempo de localização de falhas, e os objetos selecionados para a fase de teste para as Arquiteturas Centralizada, Distribuída e Hierárquica Proposta.

	Centralizado		Distribuído		Hierárquico Proposto	
Passo	Número de Operações	Objetos selecionados	Número de Operações	Objetos selecionados	Número de Operações	Objetos selecionados
1°	753422	LD₁	753422	1° - LD₁ 2°-A1 do domínio B 3°-L2 do domínio A	753422	1° - LD₁ 2° - B2 do domínio A 3° - L2 do domínio A
2°	5389656	1° - A2 do domínio A 2° - LD₃ 3° - LD₂ 4° - LD₁ 5° - B2 do domínio A	92322	1° - LD₂ 2° - LD₃ 3° - A2 do domínio B 4° - LD₁ 5° - B2 do domínio B	34557	1° - LD₁ 2° - LD₂ 3° - LD₃ 4° - A2 do Domínio B 5° - C1 do domínio B
3°	143.135.442	1° - B2 do domínio B 2° - L2 do domínio B 3° - LD₄	8.767.562	1° - B2 do domínio B 2° - A2 do domínio B 3° - LD₄	38.499	1° - LD₄ 2° - B2 do domínio B 3° - L2 do domínio B
4°	15.432.962.5 14	1° - LD₄ 2° B2 do domínio B 3° - LD₈ 4° - A1 do domínio G 5° - LD₁₂	8.774.077	1° - LD₈ 2° - LD₄ 3° - B2 do domínio B 4° - LD₁₂ 5° - A1 do domínio G	39.176	1° - LD₄ 2° - A1 do domínio D 3° - LD₁₂ 4° - A1 do domínio G 5° - LD₈

Tabela 8.2 : Resultados obtidos para a simulação das arquiteturas

Os objetos em negrito representam os objetos considerados como fonte primária de falha durante as simulações.

Observa-se pela tabela acima, que os objetos considerados como falhos em cada passo das simulações, apresentaram para a Arquitetura Hierárquica melhores posições para a fase de teste, quando comparado com as Arquiteturas Centralizada e Distribuída. Outro fato importante verificado é que o tempo de localização de falhas, expresso pelo número de operações, foi igual a 200 vezes menor para a Arquitetura Hierárquica em relação a Distribuída e, 390000 vezes menor que a Centralizada. Tal constatação indica que a Arquitetura Hierárquica Proposta apresenta uma grande melhora no desempenho da localização de falhas, quando é considerado como parâmetro o tempo requerido para localizar objetos falhos.

O número de objetos selecionados para a fase de teste não aumentou significativamente em relação ao número de objetos gerenciados. Este fato é importante, pois um menor número de objetos selecionados para a fase de teste poderá contribuir para o aumento da disponibilidade da rede.

A partir da tabela obtida anteriormente pode-se apresentar, de forma ilustrativa, o número de objetos gerenciados em função tempo de localização de falhas, utilizando o algoritmo *Divida e Conquiste Proposto*, aplicado sobre as Arquiteturas Centralizada, Distribuída e Hierárquica Proposta. Este aspecto é apresentado na figura 8.18.

Através do gráfico apresentado na figura 8.18, verifica-se que as Arquiteturas Centralizada e Distribuído apresentam um tempo de localização de falhas muito superior a Arquitetura Hierárquica, principalmente se for observado que o eixo do tempo de localização de falhas é logarítmico.

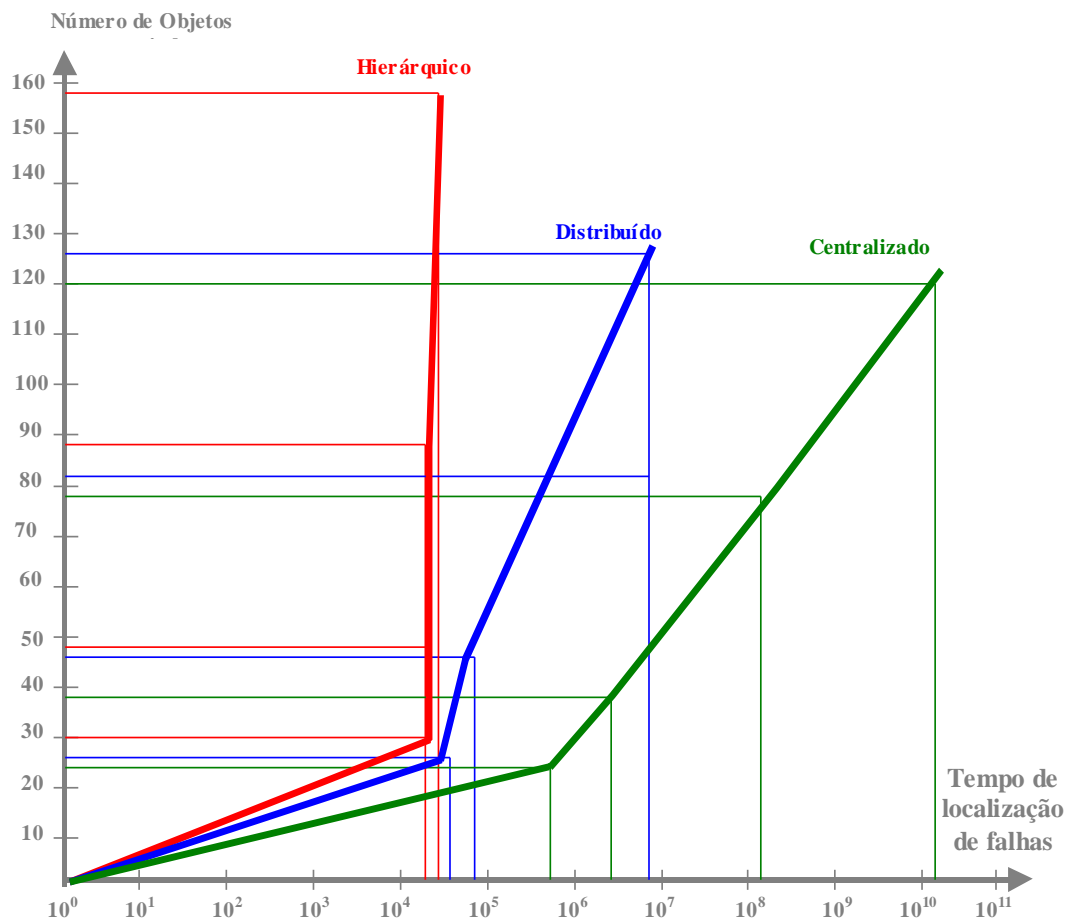


Figura 8.18: Gráfico relacionando número de objetos gerenciados e tempo de localização de falhas utilizando o algoritmo *Divida e Conquista Proposto*

Passo \ II	Centralizado/Hierárquico	Distribuído/Hierárquico
1°	23	1,7
2°	155	2,6
3°	3718	228
4°	393939	224

Tabela 8.3: Relação entre tempos de localização de falhas das Arquiteturas Centralizada e Distribuída em relação a Hierárquica Proposta.

A Tabela 8.3 apresenta a relação entre os tempos de localização de falhas das Arquiteturas Centralizada e Distribuída em relação à Hierárquica. Através desta tabela, pode-se

verificar que a relação dos tempos de localização de falhas entre as Arquiteturas Centralizada e Distribuída em relação à Arquitetura Hierárquica Proposta é bastante significativa. A figura 8.19 apresenta de forma ilustrativa a relação entre os tempos de localização de falhas entre as Arquiteturas Centralizada e Distribuída em relação a Arquitetura Hierárquica, considerando o número de objetos gerenciados aos valores obtidos para a Arquitetura Centralizada, e apresentada na Tabela 8.1.

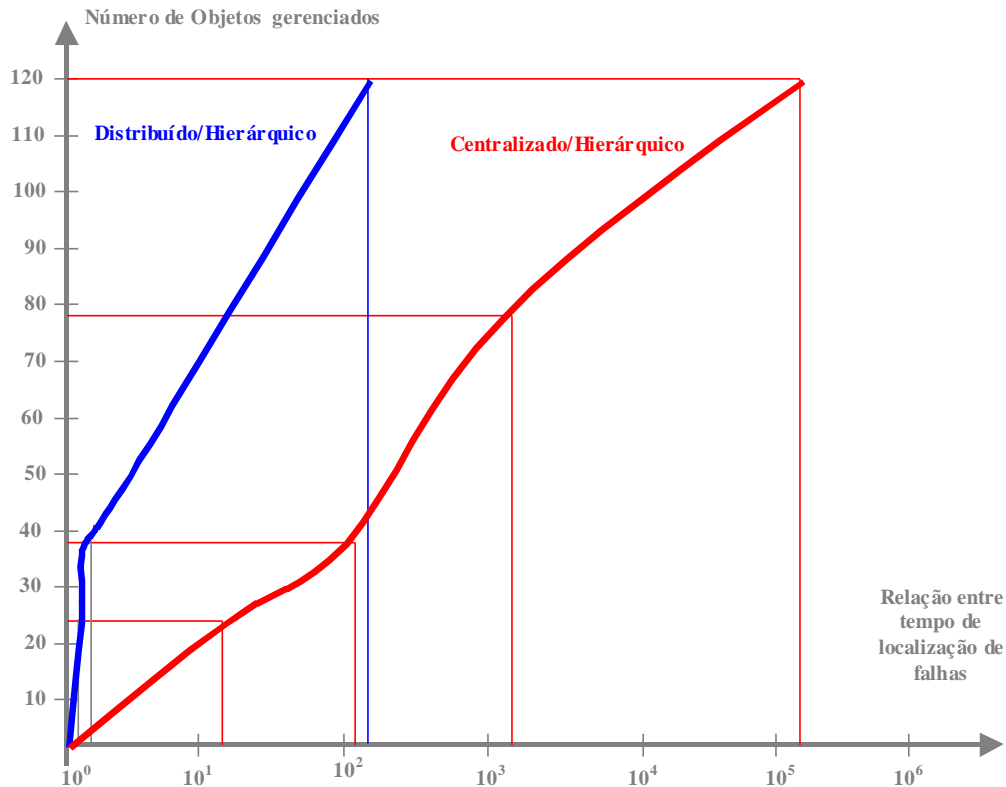


Figura 8.19: Tempos de localização de falhas entre as Arquiteturas Centralizada e Distribuída e relação a Arquitetura Hierárquica Proposta.

Pode-se verificar pela figura 8.19 que a relação entre tempos de localização de falhas entre as Arquiteturas Centralizada e Distribuída, em relação à Arquitetura Hierárquica Proposta é crescente e mais significativa quanto maior o número de objetos gerenciados, principalmente se for verificado que o eixo de tempo de localização de falhas esta na escala logarítmica. Nota-se também que para uma quantidade de objetos gerenciados (menor do que 40) é baixa a relação entre os tempos obtidos para a Arquitetura Distribuída em relação à Hierárquica Proposta.

A tabela 8.4 apresenta o aumento dos tempos de localização de falhas em função do crescimento do número de objetos gerenciados em cada passo da simulação.

I \ II	Centralizado	Distribuído	Hierárquico Proposta
1,6	7,15	1,66	1,09
3,0	190	157	1,22
5,0	20484	158	1,24

Tabela 8.4: Relação entre crescimento do número de objetos gerenciados e aumento dos tempos de localização de falhas

I – Taxa de crescimento médio do número de objetos gerenciados, obtido em cada passo de simulação, a partir das Arquiteturas Centralizada, Distribuída e Hierárquica Proposta. O número médio de objetos gerenciados entre as três arquiteturas para os passos de 1 a 4 são: 27, 44, 82 e 135, obtidos a partir da tabela 8.1. Assim, definiu-se que a taxa de crescimento média dos objetos gerenciados corresponde à relação entre o número médio de objetos gerenciados nos passos 2, 3 e 4 em relação ao passo 1.

II – Taxa de crescimento dos Tempos de localização de falhas para as Arquiteturas Centralizada, Distribuída e Hierárquica Proposta. A Taxa de crescimento dos tempos de localização de falhas foi definida como a relação entre o número de operações nos passos 2, 3 e 4 e o número de operações no passo 1, a partir da tabela 8.2.

Pela tabela 8.4 pode-se observar que a Arquitetura Distribuída e, principalmente a Centralizada, apresentaram um aumento significativo na Taxa de crescimento do tempo de localização de falhas, quando comparado com a Arquitetura Hierárquica Proposta. A figura 8.20 mostra de forma gráfica que o desempenho da Arquitetura Hierárquica Proposta é significativamente melhor que as Arquiteturas Centralizada e Distribuída.

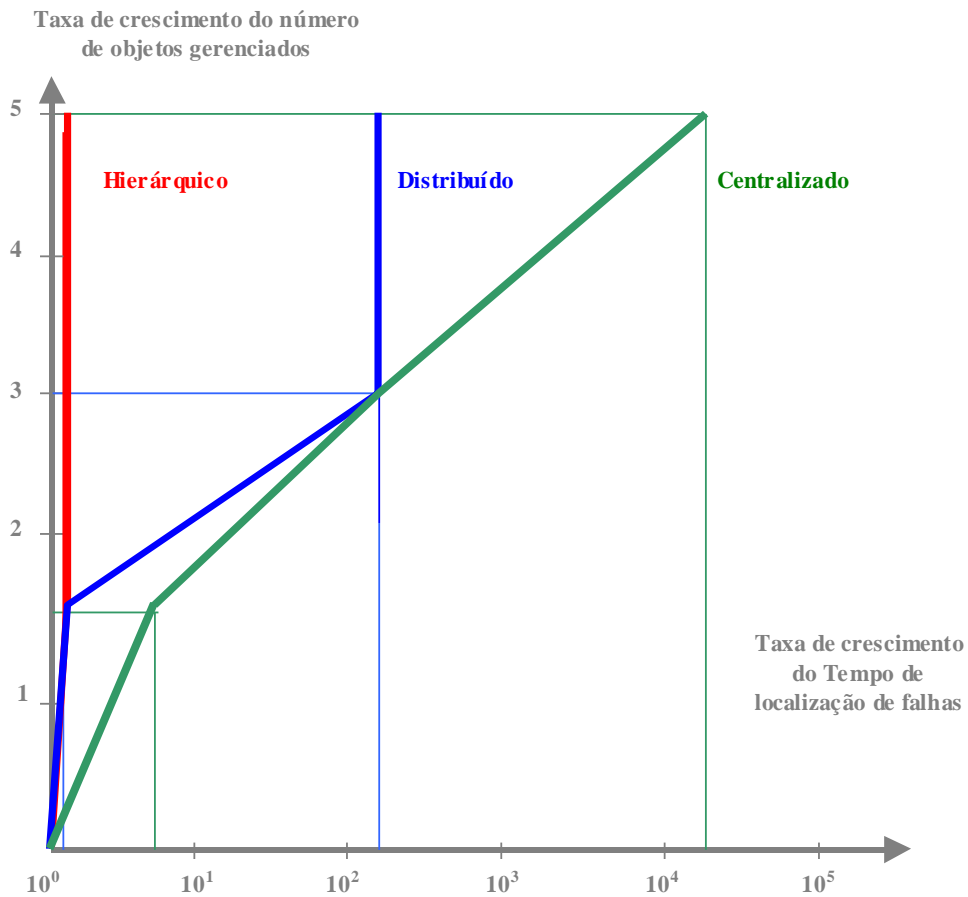


Figura 8.20: Relação entre a taxa de crescimento dos objetos gerenciados e dos Tempos de localização de falhas para as Arquiteturas Centralizada, Distribuída e Hierárquica.

Na Arquitetura Hierárquica Proposta observa-se que para a taxa de crescimento médio do número de objetos gerenciados igual a 5, o tempo de localização de falhas foi apenas 1,24, conforme pode ser observado na tabela 8.3. Este fato torna-se ainda mais significativo, quando observa-se que para a mesma taxa de crescimento médio do número de objetos gerenciados, na Arquitetura Centralizada, o tempo de localização de falhas foi igual a 20484 e, para a Arquitetura Distribuída foi igual a 158.

9. Considerações Finais

Este capítulo apresenta um resumo dos resultados alcançados durante este trabalho de pesquisa, tendo-se como base os objetivos inicialmente propostos. São apresentadas também alguns aspectos que podem-se constituir em futuras linhas de pesquisa.

9.1. Resultados Alcançados

Esta dissertação propôs investigar o efeito na disponibilidade dos serviços de rede decorrente da associação de um Algoritmo de Localização de Falhas com uma Arquitetura de Gerenciamento.

Em função de tais objetivos, pode-se apresentar as seguintes conclusões:

1) Quanto ao algoritmo *Divida e Conquiste Proposto*

- O fator de cobertura pode ser um parâmetro importante para a representação de sistemas reais e , portanto, pode proporcionar uma melhora significativa no processo de localização de falhas em redes de comunicação. Durante a simulação do algoritmo *Divida e Conquiste Proposto* pode-se observar uma grande sensibilidade do algoritmo em relação a este parâmetro.
- Através da simulação dos algoritmos considerados, pode-se observar que a introdução de dois novos critérios para a fase de união (selecionar as duas partições que sejam cobertas pelo maior número de alarmes ativos e, selecionar as duas partições que apresentem a maior probabilidade incondicional) contribuíram para que o algoritmo tornasse menos dependente de sua implementação. Tal aspecto é observado quando alguns objetos gerenciados apresentaram valores iguais para as probabilidades condicional, incondicional e fator de cobertura.
- A inclusão do conceito de seqüência de teste para os objetos selecionados para a fase de teste é um critério que aumenta a precisão dos objetos a serem testados. Conforme foi verificado durante a simulação do algoritmo *Divida e Conquiste Proposto*,

normalmente os primeiros objetos da seqüência de teste correspondiam aos objetos com maior probabilidade de serem a fonte primária de falha.

2) Quanto a Arquitetura de Gerenciamento Hierárquica Proposta:

- A Arquitetura de Gerenciamento Hierárquica Proposta apresentou significativa melhora no tempo de localização de falhas, quando comparada à Arquitetura Distribuída e, principalmente, à Arquitetura Centralizada. Esta característica foi verificada através das simulações dos algoritmos sobre diversas arquiteturas. Esta melhora torna-se mais significativa quando formam-se pequenos domínios de gerenciamento, que podem ser executados em paralelo, beneficiando-se do processamento distribuído.
- Durante a simulação das arquiteturas verificou-se que os objetos selecionados, bem como a sua seqüência de teste foram bastante semelhantes para as Arquiteturas Centralizada, Distribuída e Hierárquica Proposta. Tal fato, mostra que as arquiteturas analisadas não interferiram significativamente na seleção dos objetos para a fase de teste.

9.2. Futuras Linhas de Pesquisas

A seguir são apresentadas algumas linhas de pesquisas e aplicações que podem ser desenvolvidas, em futuros trabalhos de pesquisa.

9.2.1. Técnicas para Coleta, Interpretação e Atribuição das diversas probabilidades utilizadas no algoritmo *Divida e Conquiste Proposto*, bem como no Gerenciamento Proativo da rede

O elemento principal do algoritmo *Divida e Conquiste Proposto* é a precisão das probabilidades condicionais, incondicionais e fatores de cobertura dos objetos gerenciados. Portanto, é fundamental a utilização de técnicas que propiciem a correta coleta, interpretação e atribuição destas diversas probabilidades utilizadas no algoritmo *Divida e Conquiste Proposto*. Em tal análise deverá ser analisado como os parâmetros, tais como o MTTF, MTTR entre outros, devem ser contabilizados no cálculo das probabilidades

condicionais, incondicionais e no fator de cobertura para cada objeto de rede. Será também importante que tais técnicas contemplem a identificação de diversos níveis de alarmes de tal forma a propiciar um gerenciamento proativo da rede, antecipando prováveis objetos da rede que possam vir a ser falhos.

Neste aspecto podem ser estudadas técnicas baseadas em inteligência artificial como forma de automatizar a coleta, seleção e atribuição das diversas probabilidades utilizadas pelo algoritmo *Divida e Conquiste Proposto*, bem como forma de atribuir os diversos níveis de alarmes propiciando o gerenciamento proativo da rede. Os processos de coleta, seleção e atribuição das diversas probabilidades, bem como a seleção do nível de alarme a ser gerado, pode ser bastante complexo e, muitas vezes, baseadas no conhecimento de especialistas.

9.2.2. Delegação de Processos de Localização de Falhas em Redes Distribuídas

Estudo de técnicas de como realizar a fase de testes pelos sub-gerentes e identificados, com maior precisão, quais objetos são a fonte primária de falha. Estas técnicas podem ser realizadas através de funções de gerenciamento automatizadas, nas quais os sub-gerentes podem, além de detectar e selecionar os objetos falhos, também realizar seqüências de testes em tais objetos, isolar o objeto falho identificado como fonte primária de falha e, redirecionar automaticamente o serviço, antes executado pelo objeto falho, para outro objeto da rede. Algumas destas técnicas podem ser encontradas em [GOLDSZMIDT-98] e [VASSILA-95].

9.2.3. Estudo de Técnicas de Busca para o Algoritmo *Divida e Conquiste Proposto* associado à Arquitetura Hierárquica Proposta

Estudo de métodos de busca alternativos para a seleção dos objetos para a fase de teste. Neste aspecto pode ser avaliada a eficiência da localização do objeto falho tendo como método de busca a técnica de busca em profundidade (*depth-first search*), ou seja, aplica-se o algoritmo *Divida e Conquiste Proposto* apenas para o nível de gerenciamento

imediatamente inferior ao objeto selecionado da seqüência de teste e, assim sucessivamente até encontrar os objetos do nível de gerenciamento mais inferior da rede.

O método utilizado neste trabalho de pesquisa corresponde à técnica de busca em largura (*breadth-first search*), ou seja, aplica-se o algoritmo *Divida e Conquiste Proposto* a todos os domínios do nível de gerenciamento inferior ao nível de gerenciamento em análise.

9.2.4. Análise do Algoritmo *Divida e Conquiste Proposto* e da Arquitetura Hierárquica Proposta em função da Variação do Grau do Arco de Dependência

Pode ser estudo qual o comportamento do algoritmo *Divida e Conquiste Proposto* e da Arquitetura Hierárquica Proposta, quando é variado o grau do arco de dependência do grafo de dependência do algoritmo *Divida e Conquiste Proposto*. Neste sentido pretende-se estudar se a variação do grau do arco de dependência altera a seqüência de teste dos objetos selecionados, bem como, o tempo de localização dos objetos falhos para a Arquitetura Hierárquica Proposta.

9.2.5. Estudo de Técnicas para Automatizar o Processo de Geração de Grafos de Dependência

Um dos processos mais trabalhosos, encontrados neste trabalho de pesquisa, refere-se à geração do grafo de dependência. Este processo torna-se mais trabalhoso em função do crescimento e, da interdependência entre os objetos da rede. Desta forma, é extremamente útil o estudo de técnicas que permitem a geração automática de grafos de dependência, sem a intervenção humana utilizando, por exemplo, técnicas de inteligência artificial.

9.2.6. Análise da Propagação de Falhas por uma Rede de Comunicação e sua Interferência no algoritmo *Divida e Conquiste Proposto*

Durante as simulações dos algoritmos *Divida e Conquiste* e *Divida e Conquiste Proposto*, foi suposto que o conjunto de alarmes ativos recebidos pelo gerente e sub-gerentes da rede

não se altera. Tal suposição pode nem sempre ocorrer em uma rede de comunicação real, pois um alarme pode ser ativado em qualquer momento, inclusive durante a execução do algoritmo *Divida e Conquiste Proposto*. Por exemplo, se após o recebimento de um conjunto de alarmes ativos, este conjunto de alarmes se alterar pela inclusão de um novo alarme ativo, ou excluir um alarme por não estar mais ativo, deverá ou não recommençar a execução do algoritmo *Divida e Conquiste Proposto* para este novo conjunto de alarmes ativos. Neste caso, poderá ser analisado qual a influência na disponibilidade final de rede em se recommençar a execução do algoritmo *Divida e Conquiste Proposto*, ou então, como considerar esta nova informação na simulação em curso.

9.3. Aplicações Práticas relacionadas à Dissertação

A seguir são apresentadas algumas aplicações práticas que poderão ser desenvolvidas em futuros trabalhos de pesquisa.

9.3.1. Desenvolvimento de uma Interface Homem-Máquina

Poderá ser desenvolvido uma interface homem-máquina que permita administrar a rede de maneira fácil, fornecendo uma visão uniforme e de baixa complexidade do sistema de gerenciamento [MEIRA-96].

O último nível do Gerenciamento Hierárquico poderá ser apresentado como na figura 9.1, na qual o estado da rede é mostrado de forma gráfica. Este nível tem a função de determinar em qual domínio existe a maior probabilidade de possuir a fonte primária de falha. Como exemplo, cada objeto poderia apresentar os seguintes estados: normal, em falha, normal e testado, em falha e testado.

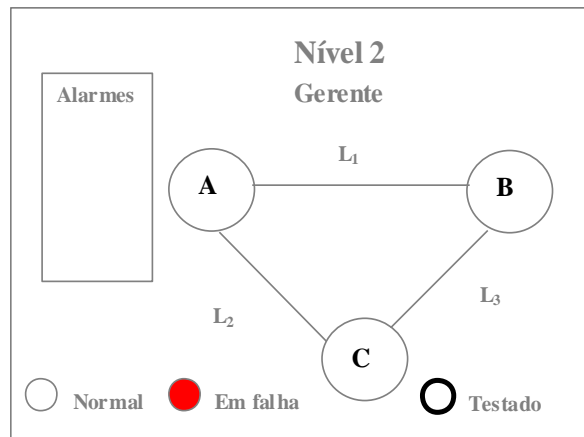


Figura 9.1: Tela do Nível 1 de Gerenciamento

Na ocorrência de uma falha o painel do administrador poderá ser apresentado como a figura 9.2, onde a sub-tela apresenta os objetos que são cobertos pelos alarmes ativos. É suposto a sinalização de falhas nos objetos B e C e, dos alarmes E_1 , E_2 e E_3 .

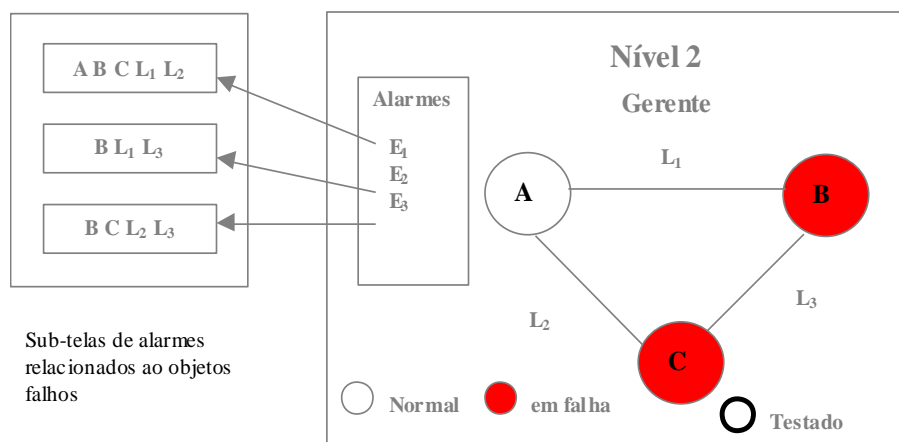


Figura 9.2: Tela de Gerenciamento de Falha com objetos em estado falho

Os alarmes apresentados pela tela de gerenciamento do administrador de rede poderão basear-se em códigos de cores para a captura de eventos ou para indicar a evolução do estado da rede, no caso de gerenciamento proativo [FAULT MANAGMENT-98]. A figura 9.3 ilustra a composição das telas da interface homem-máquina para cada um dos níveis do Gerenciamento Hierárquico. Considera-se que os objetos A, B e C representem os Sub-

gerentes A, B e C e, que os sub-gerentes B e C possuam objetos gerenciados (B_1, B_2, C_1, C_2).

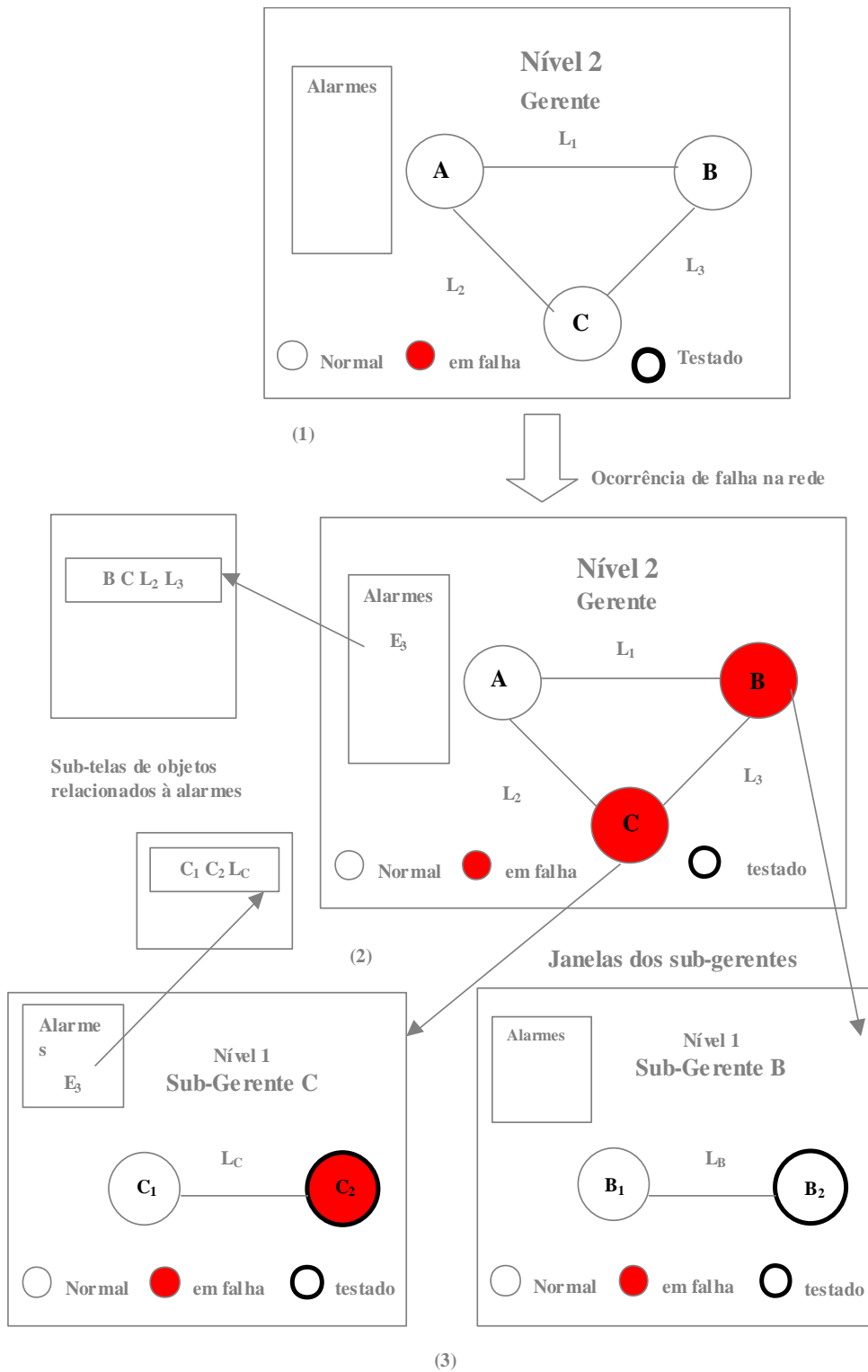


Figura 9.3: Seqüência de Telas do Ambiente de Gerenciamento

A primeira tela (1) corresponde ao nível 2 de gerenciamento. Na tela (2) é apresentada a fase de detecção e localização da falha na rede. Neste caso, o sistema de gerenciamento detectou uma falha, o Gerente de rede recebeu as mensagens de notificação de falha, executou o seu algoritmo *Divida e Conquiste Proposto* e, determinou que as localidades B e C correspondem as mais prováveis de possuírem a fonte primária de falha. Em seguida, o Gerente enviou uma mensagem de teste para o Sub-Gerente B e outra mensagem de teste para o Sub-Gerente C. Ambos os sub-gerentes executaram seus algoritmos *Divida e Conquiste Proposto* e localizam os objetos mais prováveis de serem a fonte primária de falha em seus respectivos domínios (objetos B₂ e C₂). O Sub-gerente B executou a fase de teste no objeto B₂, mas não encontrou nenhuma falha neste objeto. O Sub-gerente C executou a fase de teste no objeto C₂, identificando-o como o objeto falho na rede.

O desenvolvimento da Interface Homem Máquina poderá permitir de forma automática e gráfica a inserção de novos objetos gerenciados. O desenvolvimento desta característica deverá ser desenvolvida em conjunto com o item 9.2.5. apresentado anteriormente, ou seja, a Geração Automática do Grafo de Dependência.

9.3.2. Implementação e Avaliação do algoritmo *Divida e Conquiste Proposto* e da Arquitetura de Gerenciamento Hierárquico Proposto em Ambiente Real

Implementação e Avaliação do comportamento do algoritmo *Divida Conquiste Proposto* e da Arquitetura de Gerenciamento Hierárquico Proposta em um ambiente de gerenciamento real composto por equipamentos de rede de diversos fabricantes. Os ambientes reais podem contemplar as diversas tecnologias de redes, por exemplo, *Ethernet*, *Token Ring*, *FDDI*, *ATM (Assincronos Transfer Mode)*, *X.25*, *Frame Relay* etc.

ANEXO A

Exemplo de Aplicação do Algoritmo *Divida e Conquiste*

Para melhor entendimento do algoritmo, supor uma rede composta pelos nós A, B e C interligados pelas conexões L_1 , L_2 e L_3 , conforme apresentado na figura A.1.

Para simplificar a análise, supor que os domínios também sejam definidos pelas localidades A, B e C.

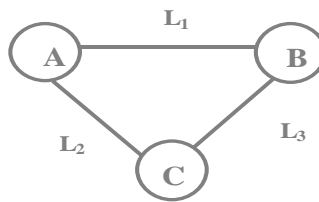


Figura A.1: Modelamento dos domínios

Premissas:

Assuma as seguintes dependências:

- Nó A necessita dos recursos do nó B.
- Nó A necessita dos recursos do nó C.
- Nó C necessita dos recursos do nó B.
- Nó B não necessita dos recursos do nó A.
- Nó B não necessita dos recursos do nó C.

Protocolo de comunicação:

Quando um elemento em um nó necessita de um recurso pertencente a outro nó, ele realiza os seguintes passos:

- Emite uma mensagem de solicitação de serviço.
- O nó que possui o serviço, retorna uma mensagem com o serviço solicitado.

A figura A.2 ilustrar melhor o processo:

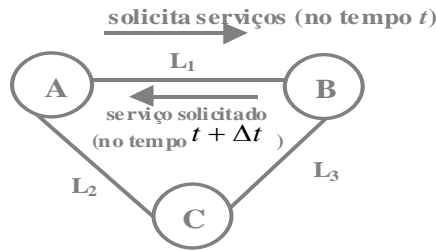


Figura A.2: Solicitação de serviços em localidades diferentes

Supor que existam os seguintes objetos realizando o monitoramento da rede:

A : objeto que monitora a atividade do nó A.

B : objeto que monitora a atividade do nó B.

C : objeto que monitora a atividade do nó C.

L_1 : objeto que monitora a atividade da conexão L_1 .

L_2 : objeto que monitora a atividade da conexão L_2 .

L_3 : objeto que monitora a atividade da conexão L_3 .

No processo de monitoramento alguns dados são obtidos a partir dos objetos gerenciados. Estes dados podem normalmente ser coletados em intervalos regulares de tempo, conhecidos como intervalos de amostragem [PERFORMANCE MANAGEMENT-98].

Desta forma pode-se obter o seguinte grafo de dependência para o exemplo anteriormente citado e apresentado na figura A.3:

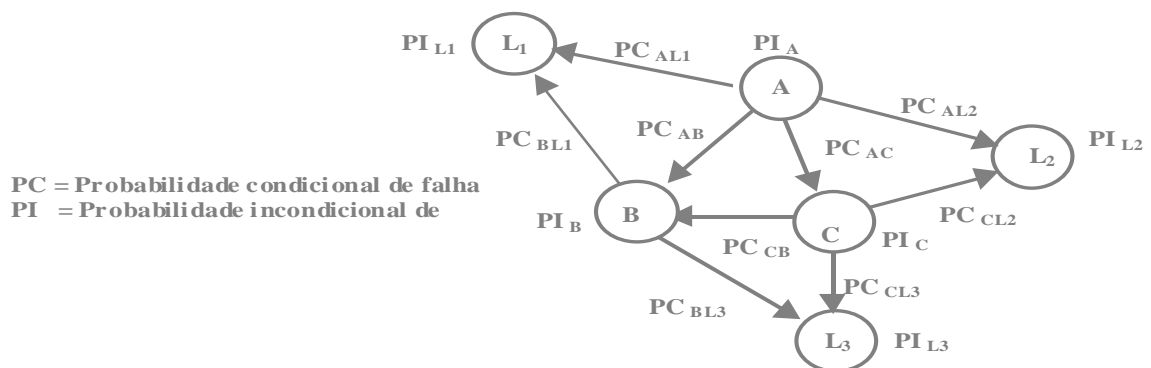


Figura A.3: Grafo de Dependência para o exemplo

Descrição:

O objeto A depende da atividade dos objetos B e C (dos recursos disponíveis nestas localidades) e dos objetos L₁ e L₂ (conexões necessárias para o envio da solicitação de serviços pela localidade A). O objeto C depende da atividade do objeto B (dos recursos disponíveis por esta localidade), do objeto L₂ (conexão necessária para o envio dos serviços solicitados pela localidade A), e do objeto L₃ (conexão necessária para o envio de solicitação de serviços para a localidade B). O objeto B depende da atividade dos objetos L₂ e L₃ (conexões necessárias para o envio do serviço solicitado pelas localidades A e C). Os objetos L₁, L₂ e L₃ não dependem de nenhum outro objeto.

Supor que os objetos gerenciados apresentem as seguintes probabilidades incondicional de falha (PI) e condicional de falha (PC), conforme mostrado na tabela A.1:

$PI_A = 0,03$	$PI_{L_3} = 0,02$	$PC_{AB} = 0,04$
$PI_B = 0,02$	$PC_{AL_2} = 0,04$	$PC_{AL_1} = 0,02$
$PI_C = 0,04$	$PC_{BL_3} = 0,02$	$PC_{CL_2} = 0,06$
$PI_{L_1} = 0,01$	$PC_{AC} = 0,06$	$PC_{CL_3} = 0,03$
$PI_{L_2} = 0,02$	$PC_{CB} = 0,01$	$PC_{BL_1} = 0,02$

Tabela A.1: Probabilidades de Falhas para os Objetos Gerenciados

Adicionando as probabilidades tem-se o grafo da figura A.4:

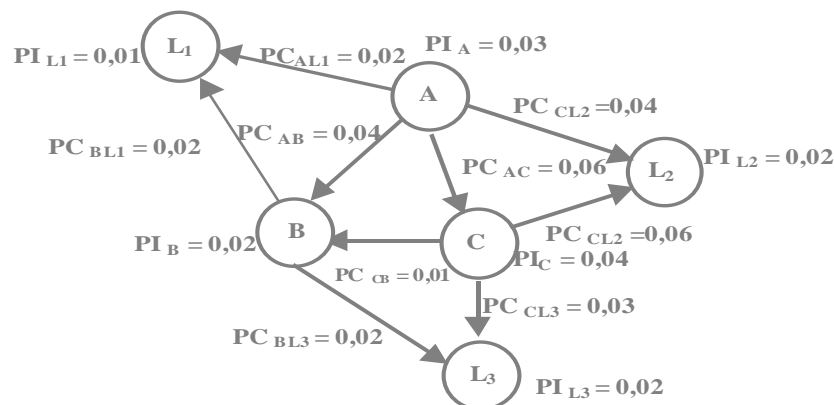


Figura A.4: Grafo de Dependência adicionado as Probabilidades de Falhas

Geração dos alarmes:

Supor como critério para a geração dos alarmes, que o conjunto de objetos gerenciados associado a um dado alarme é formado pelo objeto que sinalizou a falha, adicionando-se aqueles que ele depende diretamente (considera-se aqui um arco como distância de dependência direta).

Assim, tem-se para o exemplo em questão, a tabela A.2 relacionando: objeto sinalizado (ou seja, que emitiu uma mensagem de falha), objetos que o objeto sinalizado depende diretamente, e o respectivo alarme gerado:

Objeto Sinalizado	Objetos que o Objeto Sinalizado Depende	Alarme Gerado
A	B, C, L ₁ e L ₂	E ₁
B	L ₁ e L ₃	E ₂
C	B, L ₂ e L ₃	E ₃
L ₁	-	E ₄
L ₂	-	E ₅
L ₃	-	E ₆

Tabela A.2: Relacionamento entre Objetos Sinalizados e Alarmes Gerados

Assim, os alarmes formados e seus objetos gerenciados são:

$E_1 = \{A, B, C, L_1, L_2\}$; $E_2 = \{B, L_1, L_3\}$; $E_3 = \{B, C, L_2, L_3\}$; $E_4 = \{L_1\}$; $E_5 = \{L_2\}$; $E_6 = \{L_3\}$.

Outra forma de representar os alarmes é através da relação Objeto-Alarmes Relacionados ou Cobertos, conforme apresentado na tabela A.3.

Objeto	Alarmes Cobertos
A	E ₁
B	E ₁ , E ₂ , E ₃
C	E ₁ , E ₃
L ₁	E ₁ , E ₂ , E ₄
L ₂	E ₁ , E ₃ , E ₅
L ₃	E ₂ , E ₃ , E ₆

Tabela A.3: Relacionamento entre os Alarmes Cobertos e Objetos Gerenciados

Supor que os alarmes gerados anteriormente estejam associados aos tipos básicos (Comunicação, Qualidade de Serviço, Processamento e Ambiental) e associados a níveis de severidade (Cleared, Indeterminado, Crítico, Maior e Menor). Ou seja, supõem-se que os objetos gerenciados possam coletar informações sobre o estado da rede, através da monitoração de taxas de erro, de perda e atraso de mensagens etc. Assim, supõem-se que os objetos gerenciados possam estar no estado falho, ou então, em via de entrar neste estado (estado este observado pelo Gerenciamento proativo).

Simulação de uma falha simples:

Supor uma falha no nó C, e como consequência desta falha, os objetos A e C tenham sinalizado a falha, ou seja, tenha sinalizado apenas os alarmes E₁ e E₃ (veja a tabela A.2.) . A seguir é apresentado a simulação do algoritmo *Divida e Conquiste*.

Simulação do algoritmo *Divida e Conquiste*:

Fase de União

A fase de união utiliza um critério básico em cada passo de sua execução com a finalidade de encontrar o objeto com maior probabilidade de ser uma falha primária, este critério é:

- Unir os dois objetos com maior probabilidade condicional.

Passo 1:

Dado o grafo fornecido inicialmente:

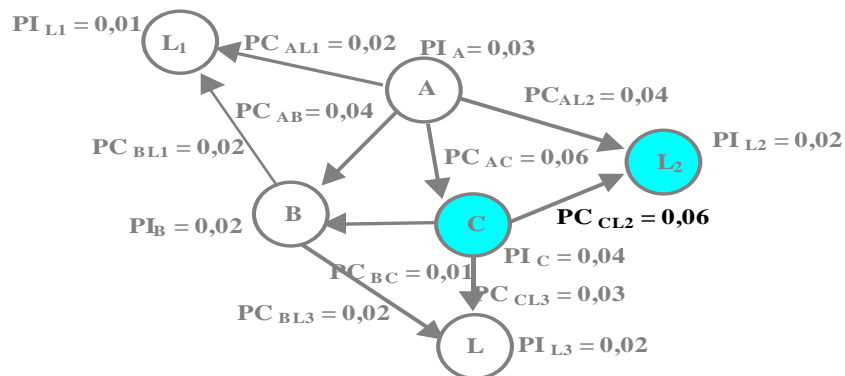


Figura A.5: União dos Objetos C e L₂

Aplicando-se o critério para a fase de união, tem-se:

- Unir os objetos com maior probabilidade condicional

Neste caso, existem duas probabilidades condicionais com valor igual a 0,06, entre os objetos A e L₂ e entre os objetos C e L₂. Seleciona-se arbitrariamente o arco de dependência entre os objetos C e L₂, pois o algoritmo *Divida e Conquiste* [KATZELA-93] não apresenta nenhuma referência no caso de existirem duas dependências com o mesmo valor máximo de probabilidade condicional. A probabilidade do novo objeto CL₂ pode ser calculado como:

$$P_{U_{CL_2}} = \frac{0,04}{\text{objeto C}} + \frac{0,06 \times 0,02}{\text{objeto L}_2} + 0,02 = 0,0612 = P_{I_{CL_2}}$$

Pela figura A.5, observa-se que existem dois arcos de probabilidade condicional a partir do objeto A para o novo objeto CL₂ com valores de 0,06 (do objeto A para o objeto C) e 0,04 (do objeto A para o objeto L₂). Assim, deverá ser selecionado apenas a maior probabilidade condicional, ou seja, o valor 0,06.

Este critério, ou seja, de selecionar a maior probabilidade condicional no caso de existirem dois arcos de dependência a partir de um mesmo objeto para o novo objeto, não é apresentado de forma explícita no algoritmo *Divida e Conquiste*, mas foi observado através de um exemplo analisado em [KATZELA-93].

Passo 2:

Portanto, obtém-se a figura a seguir a partir do grafo do passo anterior:

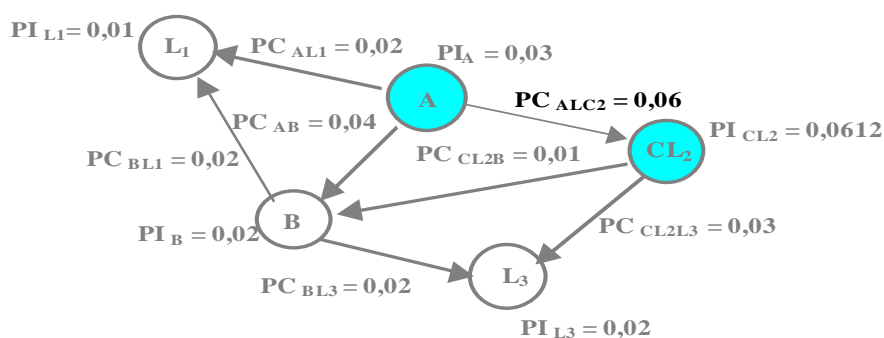


Figura A.6: União dos Objetos A e CL₂

Aplicando-se novamente o critério para a fase de união :

- Unir os objetos com maior probabilidade condicional

Verifica-se que a maior dependência neste caso ocorre entre os objetos A e CL_2 com o valor 0,06. A probabilidade do novo objeto ACL_2 pode ser calculado como:

$$PU_{ACL_2} = \frac{0,03}{\text{objeto A}} + \frac{0,06 \times 0,0612}{\text{objeto } CL_2} + 0,0612 = 0,0949 = PI_{ACL_2}$$

Pela figura A.6 verifica-se que existem dois arcos de probabilidade condicional do novo objeto ACL_2 para o objeto B, ou seja, dos objetos A e CL_2 para o objeto B, sendo respectivamente os valores 0,04 (do objeto A para o objeto B) e 0,01 (do objeto CL_2 para o objeto B). Seleciona-se a maior probabilidade condicional, ou seja, o valor 0,04.

Passo 3:

Assim, obtém-se a figura a seguir a partir do grafo do passo anterior:

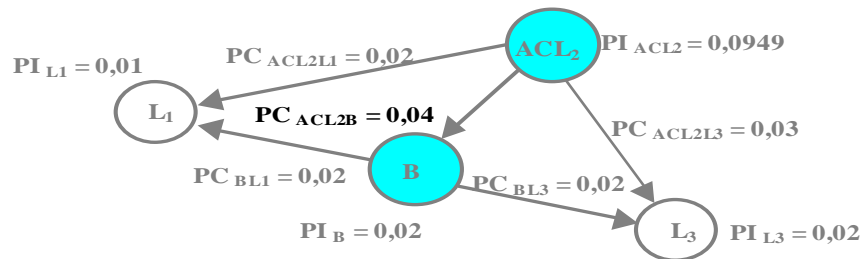


Figura A.7: União dos Objetos B e ACL_2

Aplicando-se o critério para a fase de união :

- Unir os objetos com maior probabilidade condicional

Verifica-se que a maior dependência neste caso ocorre entre os objetos B e ACL_2 com o valor 0,04. A probabilidade do novo objeto $ABCL_2$ pode ser calculado como:

$$PU_{ABCL_2} = \frac{0,0949}{\text{objeto } ACL_2} + \frac{0,04 \times 0,02}{\text{objeto B}} + 0,02 = 0,1157 = PI_{ABCL_2}$$

Pela figura A.7, verifica-se que existem dois arcos de dependência do novo objeto $ABCL_2$ para o objeto L_1 , ou seja, dos objetos ACL_2 e B para o objeto L_1 ambos com o mesmo valor 0,02. Neste caso, como ambos os arcos de dependência condicional tem o mesmo valor, será adotado o valor 0,02 para o arco de dependência entre os o novo objeto $ABCL_2$ e o objeto L_1 . Da mesma forma, observa-se que existe também dois arcos de dependência do novo objeto $ABCL_2$ para o objeto L_3 , ou seja, dos objetos ACL_2 e B para o objeto L_3 , com valores respectivamente 0,03 e 0,02. Selecionando-se o de maior valor, ou seja, o valor 0,03.

Passo 4:

Assim, obtém-se a seguinte figura a partir do grafo do passo anterior:

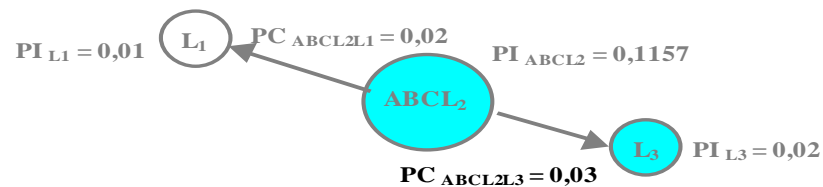


Figura A.8: União dos Objetos $ABCL_2$ e L_3

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Verifica-se o maior arco de dependência neste caso ocorre entre os objetos $ABCL_2$ e L_3 com o valor 0,03. A probabilidade do novo objeto $ABCL_2L_3$ pode ser calculado como:

$$PU_{ABCL_2L_3} = \frac{0,1157 + 0,03 \times 0,02}{\text{objeto } ABCL_2} + \frac{0,02}{\text{objeto } L_3} = 0,1363 = PI_{ABCL_2L_3}$$

Passo 5:

Assim, a partir do grafo do passo anterior, obtém-se a seguinte figura:

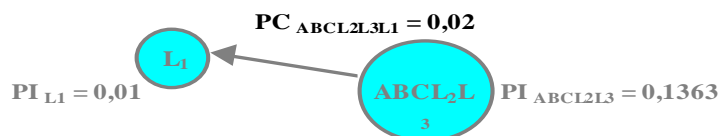


Figura A.9: União dos Objetos $ABCL_2L_3$ e L_1

Aplicando-se o critério para a fase de união terá:

- Unir os objetos com maior probabilidade condicional

Como neste caso apenas existe um arco de dependência, este é selecionado. A probabilidade do novo objeto $ABCL_1L_2L_3$ pode ser calculado como:

$$PU_{ABCL_1L_2L_3} = \underbrace{0,1363}_{\text{objeto } ABCL_2L_3} + \underbrace{0,02 \times 0,01}_{\text{objeto } L_1} + 0,01 = 0,1465 = PI_{ABCL_1L_2L_3}$$



Figura A.10: Objeto final da Fase de União

Fase de particionamento

Partindo-se do objeto final obtido na fase de União, realiza-se agora o processo inverso da fase de união, ou seja, o objeto da fase final de união é particionado sempre em dois filhos (direito e esquerdo), estes filhos são gerados em uma seqüência inversa ao realizado na fase de união, devendo-se selecionar em cada passo desta fase do algoritmo, o filho que cumpre os seguintes dois critérios:

- 1) Apresente a maior probabilidade incondicional
- 2) Seja coberto por todos os alarmes ativos

Se ambos os critérios em um determinado passo não for cumprido, seleciona-se ambos os filhos e aplica-se os dois critérios anteriormente citados, mas a partir deste passo o número de alarmes ativos se reduz ao número de alarmes ativos correspondente a este passo. Por exemplo, se inicialmente existir 3 alarmes ativos, e num determinado passo ambos os critérios não são cumpridos, e se um filho é coberto apenas por 2 alarmes, a partir deste passo o número de alarmes ativos para este filho passará ser igual a 2 e não mais 3. A seguir é aplicado a fase de particionamento para o exemplo.

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios definidos para a fase de particionamento.

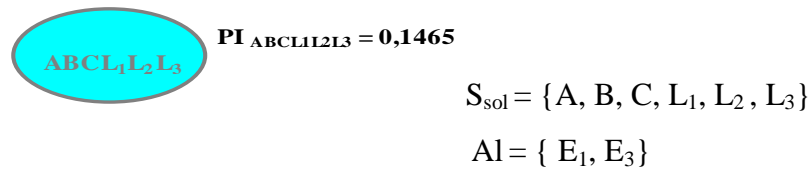


Figura A.11: Objeto obtido na Fase de União

O conjunto S_{sol} apresentado na figura A.11 possui como elementos iniciais os objetos básicos (A, B, C, L_1 , L_2 , L_3) que estão inclusos no objeto final da fase de união. O conjunto S_{sol} ao final da fase de particionamento apresentará o objeto ou objetos com maior probabilidade de serem uma fonte primária de falha. O conjunto A1 apresentado na figura A.11 representa o conjunto de alarmes ativos (ativados pelos objetos no estado falho) nesta simulação.

Passo 1:

Particionar o objeto em:

- Filho direito: L_1
- Filho esquerdo: $ABCL_2L_3$ (obtido a partir do passo 5 da fase de união).

Assim, obtém-se o seguinte grafo:

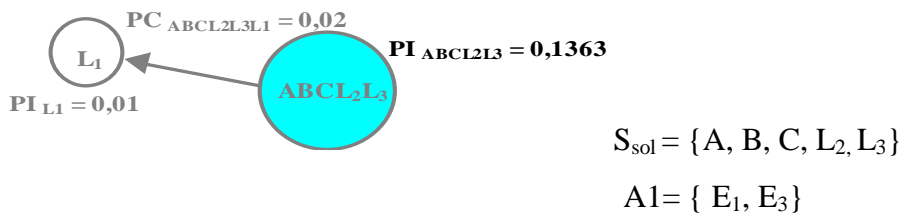


Figura A.12: Particionando os Objetos em $ABCL_2L_3$ e L_1

Selecionar o filho que cumpra os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional com valor 0,1363.

- 2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelos alarmes ativos E_1 e E_3 . O filho esquerdo é coberto pelo alarme E_1 . O filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, sendo portanto o selecionado.

Passo 2:

Particionar o filho selecionado no passo anterior em:

- Filho direito: L_3
- Filho esquerdo: $ABCL_2$

Assim, obtém-se o seguinte grafo:

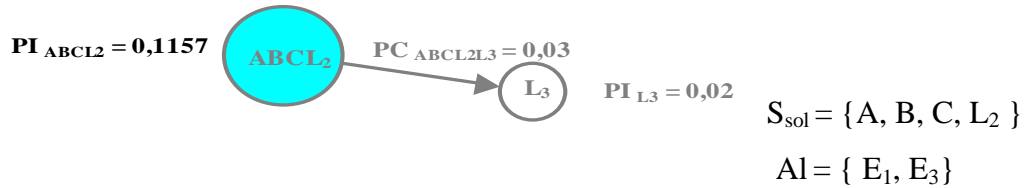


Figura A.13: Seleção entre os Objetos $ABCL_2$ e L_3

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional com valor 0,1157.

- 2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelo alarme ativo E_3 . O filho esquerdo é coberto pelos alarmes ativos E_1 e E_3 . Como filho esquerdo é coberto por todos os alarmes e tem a maior probabilidade incondicional, portanto ele é o selecionado.

Passo 3:

Particionar o filho selecionado no passo anterior em:

- Filho direito: ACL_2
- Filho esquerdo: B

Assim, obtém-se o seguinte grafo:

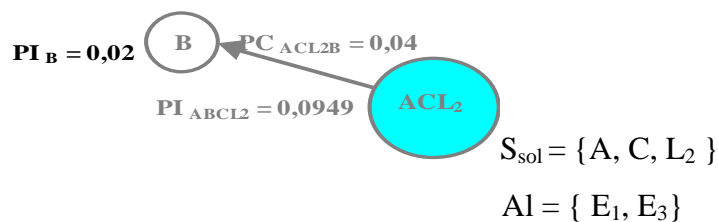


Figura A.14: Seleção entre os Objetos ACL_2 e B

Selecionar o filho que cumpre os seguintes critérios:

1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional com valor 0,0949.

2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelos alarmes ativos E_1 e E_3 . O filho esquerdo é coberto pelos alarmes ativos E_1 e E_3 . Como filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, portanto ele é o selecionado.

Passo 4:

Particionar o filho direito selecionado no passo anterior em:

- Filho direito: CL_2
- Filho esquerdo: A

Assim, obtém-se o seguinte grafo:

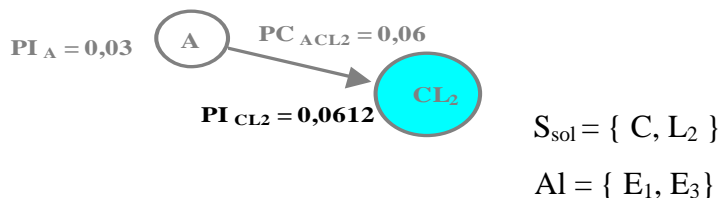


Figura A.15: Seleção entre os Objetos CL_2 e A

Selecionar o filho que cumpre os seguintes critérios:

1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelos alarmes ativos E_1 e E_3 . O filho esquerdo é coberto pelo alarme ativo E_1 . O filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, sendo o selecionado neste passo.

Passo 5:

Particionar o filho selecionado no passo anterior em:

- Filho direito: L_2
- Filho esquerdo: C

Assim, obtém-se o seguinte grafo:

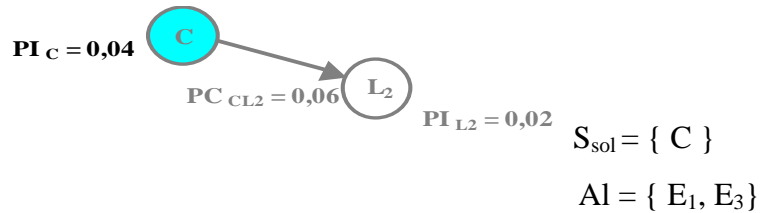


Figura A.16: Seleção entre os Objetos C e L_2

Selecionar o filho que cumpre os seguintes critérios:

1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelos alarmes ativos E_1 e E_3 . O filho esquerdo é coberto pelos alarmes ativos E_1 e E_3 . Como o filho esquerdo cumpre os dois critérios, portanto é o selecionado. O algoritmo termina a sua execução, pois o conjunto solução apresenta um único objeto selecionado.

Solução final: $S_{sol} = \{ C \}$

O conjunto solução final (S_{sol}) apresenta um único objeto a ser testado. A fase de teste deve ser realizada para todos os objetos selecionados no conjunto Solução final.

Como o objeto C é a fonte primária de falha (suposição feita no início do exemplo), a falha será detectada no teste deste objeto.

ANEXO B

Arquitetura de Gerenciamento de Falhas Centralizado

1. Aplicação utilizando o algoritmo *Divida e Conquiste*

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC):

$PI_{A1} = 0,02$	$PI_{L2} = 0,02$	$PC_{C1B1} = 0,01$
$PI_{A2} = 0,03$	$PI_{L3} = 0,02$	$PC_{C1LC} = 0,02$
$PI_{B1} = 0,02$	$PC_{A1A2} = 0,01$	$PC_{C1L2} = 0,06$
$PI_{B2} = 0,03$	$PC_{A1LA} = 0,02$	$PC_{C1L3} = 0,03$
$PI_{C1} = 0,02$	$PC_{A2LA} = 0,03$	$PC_{B1B2} = 0,01$
$PI_{C2} = 0,03$	$PC_{A2L2} = 0,04$	$PC_{B1LB} = 0,02$
$PI_{LA} = 0,02$	$PC_{A2L1} = 0,02$	$PC_{B1L1} = 0,01$
$PI_{LB} = 0,02$	$PC_{A2B1} = 0,04$	$PC_{B1L3} = 0,02$
$PI_{LC} = 0,02$	$PC_{A2C1} = 0,06$	$PC_{B2LB} = 0,03$
$PI_{L1} = 0,01$	$PC_{C1C2} = 0,01$	$PC_{C2LC} = 0,03$

Tabela B.1.1: Probabilidades de falhas dos objetos gerenciados para arquitetura centralizada

Adicionando as probabilidades ao grafo tem-se:

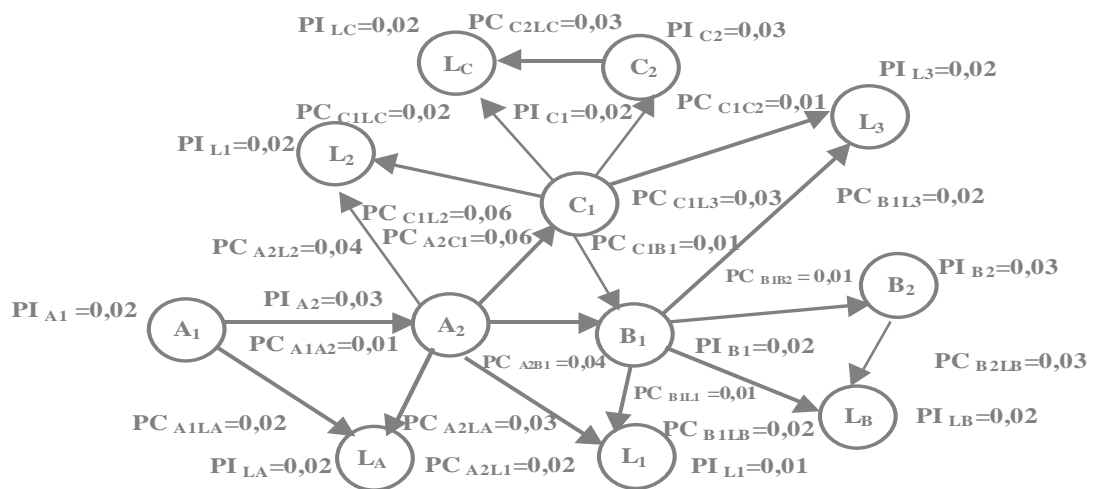


Figura B.1.1: Grafo de dependência adicionado as probabilidades de falhas

Geração dos alarmes:

Supor como critério para geração dos alarmes que o conjunto de objetos associado a um dado alarme é formado pelo objeto que sinalizou a falha, adicionando-se todos os objetos que ele depende diretamente (considerar um arco como distância de dependência). Este critério também é utilizado nas demais aplicações para Gerenciamento Distribuído e Hierárquico Proposto.

Assim, tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e o respectivo alarme gerado.

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
A ₁	A ₂ , L _A	E ₁
A ₂	B ₁ , C ₁ , L _A , L ₁ , L ₂	E ₂
B ₁	B ₂ , L _B , L ₁ , L ₃	E ₃
B ₂	L _B	E ₄
C ₁	B ₁ , C ₂ , L _C , L ₂ , L ₃	E ₅
C ₂	L _C	E ₆
L _A	-	E ₇
L _B	-	E ₈
L _C	-	E ₉
L ₁	-	E ₁₀
L ₂	-	E ₁₁
L ₃	-	E ₁₂

Tabela B.1.2: Relacionamento entre objetos sinalizados e alarmes gerados.

Assim o alarmes são formados como a seguir:

$$E_1 = \{ A_1, A_2, L_A \}; E_2 = \{ A_2, B_1, C_1, L_A, L_1, L_2 \}; E_3 = \{ B_1, B_2, L_B, L_1, L_3 \}; E_4 = \{ B_2, L_B \}$$
$$E_5 = \{ B_1, C_1, C_2, L_C, L_2, L_3 \}; E_6 = \{ C_2, L_C \}; E_7 = \{ L_A \}; E_8 = \{ L_B \}; E_9 = \{ L_C \}; E_{10} = \{ L_1 \}$$
$$E_{11} = \{ L_2 \}; E_{12} = \{ L_3 \}$$

O critério para a criação de alarmes pode ser diversificado, ou seja, poder-se associar cada objeto sinalizado com outros objetos, segundo algum outro critério, por exemplo, associando um alarme a todos os objetos com distância menor ou igual a w do objeto que emitiu a sinalização, sendo w um parâmetro experimental estimado. Outra forma bastante

interessantes e que é muito utilizada, relacionam objetos gerenciados e alarmes emitidos apresentados na tabela B.1.3:

Objeto	Alarmes emitidos
A ₁	E ₁
A ₂	E ₁ , E ₂
B ₁	E ₂ , E ₃ , E ₅
B ₂	E ₃ , E ₄
C ₁	E ₂ , E ₅
C ₂	E ₅ , E ₆
L _A	E ₁ , E ₂ , E ₇
L _B	E ₃ , E ₄ , E ₈
L _C	E ₅ , E ₆ , E ₉
L ₁	E ₂ , E ₃ , E ₁₀
L ₂	E ₂ , E ₅ , E ₁₁
L ₃	E ₃ , E ₅ , E ₁₂

Tabela B.1.3: Relacionamento entre objetos gerenciados e alarmes gerados.

Supor para todo o capítulo que os alarmes gerados anteriormente estejam associados a tipos básicos (Comunicação, Qualidade de Serviço, Processamento e Ambiental), e associados a níveis de severidade (Cleared, Indeterminado, Crítico, Maior e Menor). Ou seja, supõem-se que os objetos gerenciados possam coletar informações sobre o estado da rede, através da monitoração de taxas de erro, de perda e atraso de mensagens etc. Assim, supõem-se que os objetos gerenciados possam estar no estado falho, ou então, em via de entrar neste estado (estado este observado pelo Gerenciamento proativo).

Simulação de falha simples:

Supor que apenas os objetos A₁ e C₂ emitiam uma mensagem notificação de falha ao gerenciador de falhas, ou seja, apenas as falhas nos objetos A₁ e C₂ são cobertas pelos alarmes ativos. Assim, os objetos A₁ e C₂ emitem uma mensagem de notificação de falha que aciona os alarmes associados a estes objetos. Os alarmes ativos para estes objetos são: E₁, E₅ e E₆ (veja tabela B.1.3). A seguir é apresentado a simulação do algoritmo *Divida e Conquiste*.

Fase de União

Utiliza em cada passo de sua execução o seguinte critério, com a finalidade de encontrar o objeto com maior probabilidade de ser uma falha primária:

- Unir os dois objetos com maior dependência condicional.

Passo 1:

Dado o grafo fornecido inicial:

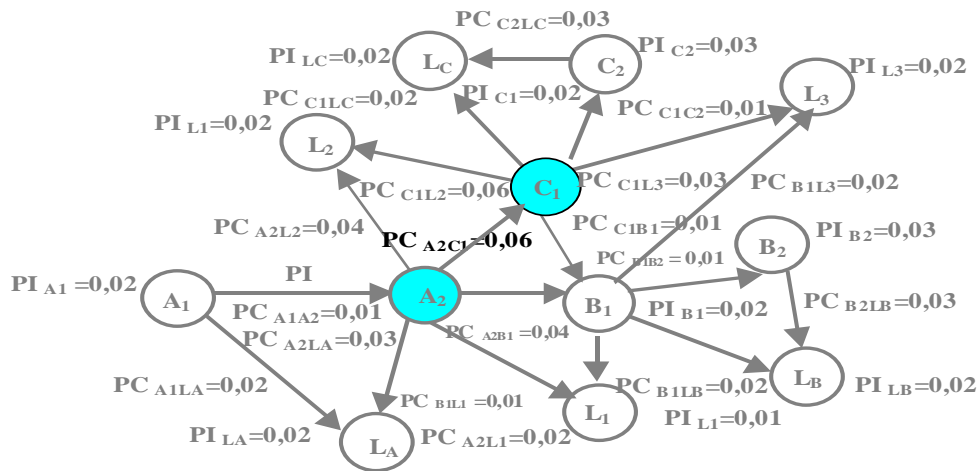


Figura B.1.2: União dos objetos C₁ e L₂

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Neste caso, existem duas probabilidades condicionais com o mesmo valor de dependência igual a 0,06 (entre os objetos C₁ e L₂, e entre os objetos A₂ e C₁). Seleciona-se arbitrariamente a dependência entre os objetos A₂ e C₁. A probabilidade do novo objeto A₂C₁ pode ser calculado como:

$$PU_{A_2C_1} = \frac{0,03}{\text{objeto } A_2} + \frac{0,06 \times 0,02}{\text{objeto } C_1} + 0,02 = 0,0512 = PI_{A_2C_1}$$

Pela figura B.1.2, observa-se que existem dois arcos de probabilidade condicional a partir do novo objeto A₂C₁ para o objeto L₂ com os valores de 0,06 (do objeto C₁ para o objeto L₂) e 0,04 (do objeto A₂ para o objeto L₂). Assim, deverá ser selecionado apenas a maior dependência a partir do novo objeto A₂C₁ para o objeto L₂, ou seja, a dependência de valor 0,06. Da mesma forma, observa-se que existem dois arcos de probabilidade condicional a

partir do novo objeto A_2C_1 para o objeto B_1 com os valores de 0,04 (do objeto A_2 para o objeto B_1) e 0,01 (do objeto C_1 para o objeto B_1). Assim, deverá ser selecionado apenas a maior dependência a partir do novo objeto A_2C_1 para o objeto B_1 , ou seja, a dependência de valor 0,04. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.

$$A_1A_2B_1B_2C_1C_2L_AL_BL_CL_L1L_2L_3 \quad \text{PI}_{A_1B_1B_2C_1C_2L_AL_BL_CL_L1L_2L_3} = 0,2727$$

Figura B.1.3: Objeto final da fase de união

Fase de particionamento

Partindo-se do objeto final obtido na fase de União, realiza-se agora o processo inverso desta fase. O objeto da fase final de união é particionado sempre em dois filhos (direito e esquerdo), estes filhos são gerados em uma seqüência inversa ao realizado na fase de união, e seleciona-se em cada passo desta fase do algoritmo, o filho que cumpre os seguintes dois critérios:

- 1) Apresenta a maior probabilidade incondicional.
- 2) Coberto por todos os alarmes ativos.

Se ambos os critérios em um determinado passo não for cumprido, seleciona-se ambos os filhos e aplica-se os dois critérios anteriormente citados, mas a partir deste passo o número de alarmes ativos se reduz ao número de alarmes válidos a partir do passo corrente. A seguir é aplicado a fase de particionamento para o exemplo. Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

$$A_1A_2B_1B_2C_1C_2L_AL_BL_CL_L1L_2L_3 \quad \text{PI}_{A_1B_1B_2C_1C_2L_AL_BL_CL_L1L_2L_3} = 0,2727$$

$$S_{\text{sol}} = \{ A_1, A_2, B_1, B_2, C_1, C_2, L_A, L_B, L_C, L_1, L_2, L_3 \}$$

$$A = \{ E_1, E_5, E_6 \}$$

Figura B.1.4: Objeto obtido na fase de união

O conjunto S_{sol} apresentado na figura B.1.4 possui como elementos iniciais os objetos ($A_1, A_2, B_1, B_2, C_1, C_2, L_A, L_B, L_C, L_1, L_2, L_3$) que estão inclusos no objeto final da fase de união. O conjunto S_{sol} ao final da fase de particionamento apresentará o objeto ou objetos com

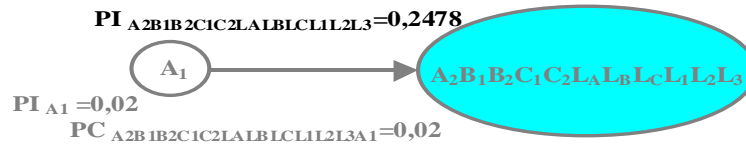
maior probabilidade de serem uma falha primária. A figura B.1.4 apresenta também o conjunto de alarmes e objetos cobertos.

Passo 1:

Particionar o objeto em:

- Filho direito: A_1
- Filho esquerdo: $A_2B_1B_2C_1C_2L_AL_BL_CL_1L_2L_3$

Os filhos formados nesta fase são determinados de acordo com a ordem inversa da fase de união. Assim, obtém-se o seguinte grafo:



$$S_{sol} = \{ A_2, B_1, B_2, C_1, C_2, L_A, L_B, L_C, L_1, L_2, L_3 \}$$

$$A = \{ E_1, E_5, E_6 \}$$

Figura B.1.5: Particionando os objetos em $A_2B_1B_2C_1C_2L_AL_BL_CL_1L_2L_3$ e A_1

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho direito é coberto por todos os alarmes ativos E_1, E_5 e E_6 . O filho esquerdo é coberto pelo alarme ativo E_1 . Como o filho direito é coberto por todos os alarmes ativos, e tem a maior probabilidade incondicional, então ele é selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se o conjunto solução final descrito a seguir.

Solução final:

O conjunto solução final será $S_{sol} = \{ A_2, C_1, C_2 \}$.

2. Aplicação utilizando a algoritmo *Divida e Conquista Proposto*

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC) idênticos ao apresentado na tabela B.1.1 e, seja adicionado o fator de cobertura (C) para cada objeto conforme a seguir:

$C_{A1} = 0,7$	$C_{B2} = 0,6$	$C_{LC} = 0,8$
$C_{LA} = 0,8$	$C_{LB} = 0,8$	$C_{L1} = 0,9$
$C_{A2} = 0,6$	$C_{C1} = 0,7$	$C_{L2} = 0,8$
$C_{B1} = 0,7$	$C_{C2} = 0,7$	$C_{L3} = 0,9$

Tabela B.2.1: Probabilidades de falhas dos objetos adicionando o fator de cobertura

Adicionando as probabilidades ao grafo tem-se:

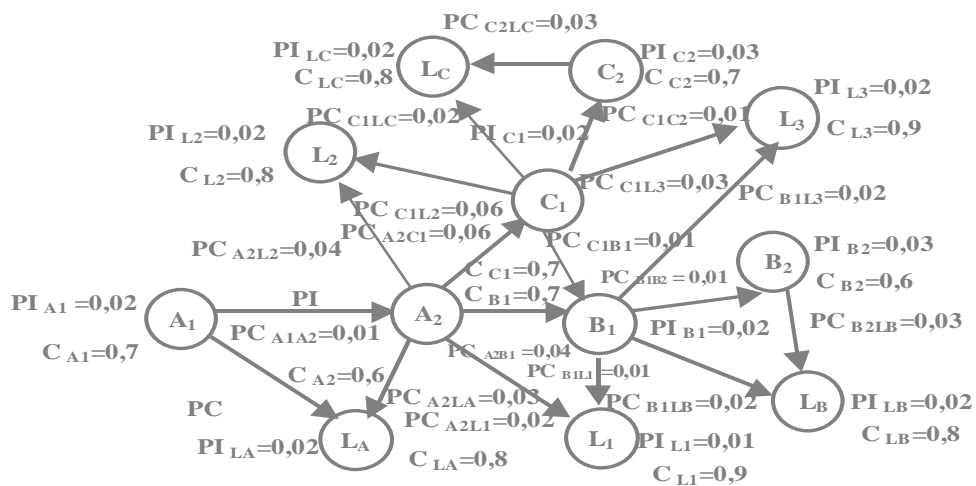


Figura B.2.1: Grafo de dependência adicionado as probabilidades de falhas

Supor igualmente ao item 1 anterior, que os objetos A_1 e C_2 emitam uma mensagem notificação de falha ao gerenciador de falhas e, estejam ativos os alarmes E_1 , E_5 e E_6 .

Fase de União

A fase de união utiliza quatro critérios (sendo novos os critérios 2 e 3) em cada passo de sua execução com a finalidade de encontrar o objeto com maior probabilidade de ser uma falha primária, estes critérios são:

- 1) Unir os dois objetos com maior dependência condicional, caso existam mais de dois objetos que cumpram este critério, deve-se,
- 2) Selecionar a maior dependência que inclua o maior número de alarmes ativos, caso existam mais de dois objetos que cumpram este critério, deve-se,
- 3) Unir os dois objetos com maior dependência incondicional, caso existam mais de dois objetos que cumpram este critério, deve-se,
- 4) Unir dois objetos arbitrariamente.

Os alarmes ativos para esta simulação são E_1 , E_5 e E_6 .

Passo 1:

Dado o grafo fornecido inicialmente:

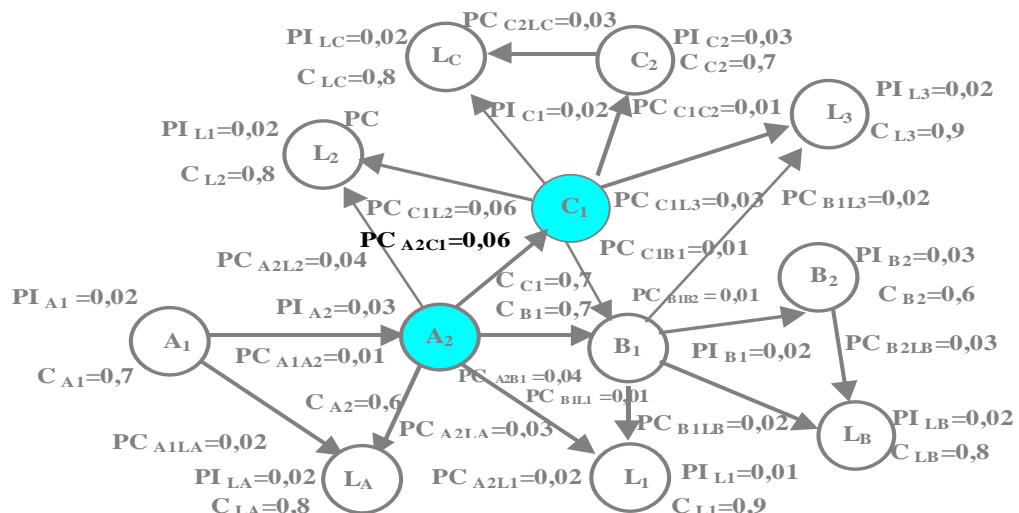


Figura B.2.2: União dos objetos C_1 e L_2

Aplicando-se o critério para a fase de união para o algoritmo *Divida e Conquiste Proposto*:

- 1) Unir os objetos com maior probabilidade condicional

Neste caso, existem duas probabilidades condicionais com o mesmo valor de dependência igual a 0,06 (entre os objetos C_1 e L_2 , e entre os objetos A_2 e C_1). Assim deve-se aplicar o segundo critério a seguir.

- 2) Coberto pelo maior número de alarmes ativos:

Os objetos C_1 e L_2 são cobertos pelos seguintes alarmes ativos: O objeto C_1 é coberto pelo alarme E_5 . O objeto L_2 é coberto pelo alarme E_5 . Ou seja, a dependência entre os objetos C_1

e L_2 é coberta pelo alarme E_5 . Os objetos A_2 e C_1 são cobertos pelo seguinte alarme: O objeto A_2 é coberto pelo alarme ativo E_1 . O objeto C_1 é coberto pelo alarme ativo E_5 . Ou seja, a dependência entre os objetos A_2 e C_1 é coberta pelos alarmes ativos E_1 e E_5 . As coberturas dos alarmes são facilmente verificadas pela tabela B.1.3. Verifica-se que dependência A_2C_1 é a selecionada, por cobrir o maior número de alarmes ativos.

3) Unir os objetos com maior probabilidade incondicional

Não é necessário verificar este terceiro critério, pois apenas foi selecionado uma única dependência no critério anterior.

4) Unir dois objetos arbitrariamente.

Também não é necessário verificar este critério neste passo do algoritmo. A probabilidade do novo objeto A_2C_1 pode ser calculado como:

$$PU_{A_2C_1} = \frac{0,03 \times 0,6}{\text{objeto } A_2} + \frac{0,06 \times 0,02 \times 0,6 + 0,02 \times 0,7}{\text{objeto } C_1} = 0,0327 = PI_{A_2C_1}$$

$$PU_{A_2} = 0,03 \times 0,6 + 0,06 \times 0,02 \times 0,6 = 0,0187$$

$$PU_{C_1} = 0,02 \times 0,7 = 0,014$$

$$FCU_{A_2C_1} = (PU_{A_2} / PU_{A_2C_1}) \times CU_{A_2} + (PU_{C_1} / PU_{A_2C_1}) \times CU_{C_1}$$

$$FCU_{A_2C_1} = (0,0187 / 0,0327) \times 0,6 + (0,014 / 0,0327) \times 0,7 = 0,6428$$

Pela figura B.2.2, observa-se que existe dois arcos de probabilidade condicional a partir do novo objeto A_2C_1 para o objeto L_2 com os valores de 0,06 (do objeto A_2 para o objeto L_2) e 0,04 (do objeto C_1 para o objeto L_2). Assim, deverá ser selecionado apenas a maior dependência a partir do novo objeto A_2C_1 para o objeto L_2 , ou seja, a dependência de valor 0,06. Da mesma forma, observa-se que existem dois arcos de probabilidade condicional a partir do novo objeto A_2C_1 para o objeto B_1 com os valores de 0,04 (do objeto A_2 para o objeto B_1) e 0,01 (do objeto C_1 para o objeto B_1). Assim, deverá ser selecionado apenas a maior dependência a partir do novo objeto A_2C_1 para o objeto B_1 , ou seja, a dependência de valor 0,04. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de particionamento a seguir. Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.



Figura B.2.3: Objeto final da fase de união

Ou seja, a rede apresenta uma probabilidade de falha incondicional de 0,06314, e com um fator de cobertura de 0,7580 (75,80 %) das falhas surgidas na rede. A obtenção da probabilidade de falha total da rede em análise não foi observada em [KATZELA-93], além de apresentar um novo parâmetro que é o número de falhas detectadas em relação ao número total de falhas que corridas na rede.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.



$$S_{sol} = \{ A_1, A_2, B_1, B_2, C_1, C_2, L_A, L_B, L_C, L_1, L_2, L_3 \}$$

$$A = \{ E_1, E_5, E_6 \}$$

Figura B.2.4: Objeto obtido na fase de união

O conjunto S_{sol} apresentado na figura B.2.4 representa o conjunto solução inicial para a fase de particionamento. O conjunto S_{sol} ao final da fase de particionamento apresentará o objeto ou objetos com maior probabilidade de serem uma falha primária. A figura B.2.4 apresenta também o conjunto de alarmes que o conjunto S_{sol} é coberto.

Passo 1:

Particionar o objeto em:

- Filho direito: A_1
- Filho esquerdo: $A_2B_1B_2C_1C_2L_AL_B L_C L_1L_2L_3$

Assim, obtém-se o seguinte grafo:

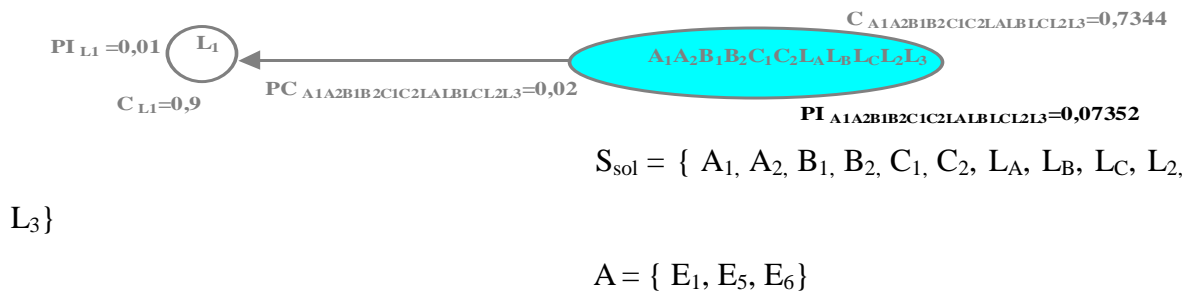


Figura B.2.5: Particionando os objetos em $A_2B_1B_2C_1C_2L_AL_BL_CL_1L_2L_3$ e L_1

Selecionar o filho que cumpre os seguintes critérios:

1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

2) Coberto por todos os alarmes ativos:

O filho direito é coberto por todos os alarmes ativos E_1, E_5 e E_6 . O filho esquerdo não é coberto por nenhum alarme ativo. Como o filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, ele é selecionado. Assim, sucessivamente deve é aplicado aos demais passos até obter-se o conjunto solução final descrito a seguir.

Solução final:

O conjunto solução final será composto pela união das soluções parciais $S_{sol\ 1.1.1} = \{ A_2 \}$, $S_{sol\ 1.1.2} = \{ C_1 \}$ e $S_{sol\ 1.2} = \{ C_2 \}$, ou seja, a solução final será: $S_{sol} = \{ A_2, C_1, C_2 \}$

Determinação da seqüência de teste:

A fase de testes deverá ser realizados nos objetos A_2, C_1 e C_2 .

Neste caso, a seqüência de teste poderá ser baseada nas probabilidades de falha dos objetos (probabilidade condicional adicionado a probabilidade incondicional para cada objeto selecionado), ou seja:

$$PU_{A_2} = 0,03 \times 0,6 + 0,03 \times 0,02 \times 0,6 + 0,02 \times 0,01 \times 0,6 + 0,04 \times 0,02 \times 0,6 + 0,06 \times 0,02 \times 0,6 + 0,04 \times 0,02 \times 0,6 = 0,0202$$

$$\frac{PI_{A1}}{PC_{A2LA}} \quad \frac{PC_{A2L1}}{PC_{A2B1}} \quad \frac{PC_{A2C1}}{PC_{A2L2}}$$

$$PU_{C1} = 0,02 \times 0,7 + 0,01 \times 0,02 \times 0,7 + 0,03 \times 0,02 \times 0,7 + 0,01 \times 0,03 \times 0,7 + 0,02 \times 0,02 \times 0,7 + 0,06 \times 0,02 \times 0,7 = 0,0159$$

$$\frac{PI_{C1}}{PC_{C1B1}} \quad \frac{PC_{C1L3}}{PC_{C1C2}} \quad \frac{PC_{C1LC}}{PC_{C1L1}}$$

$$PU_{C2} = 0,03 \times 0,7 + 0,03 \times 0,02 \times 0,7 = 0,0214$$

$$\frac{PI_{C2}}{PC_{LC}}$$

A seqüência de testes deverá ser realizada na ordem decrescente das probabilidades de falhas, pois os objetos com maior probabilidade tem maior chance de serem a fonte primária de falha. Assim, poderá ser realizado as seguintes seqüências de testes para este exemplo: C_2 , A_2 e C_1 . Como o primeiro objeto a ser testado é o objeto C_2 , a fonte primária de falha será encontrada logo no primeiro nó a ser testado, isto significa uma melhora em relação ao algoritmo *Divida e Conquiste*, pois este não determina a forma de teste dos objetos selecionados.

ANEXO C

Arquitetura de Gerenciamento de Falhas Distribuído

1. Aplicação utilizando o algoritmo *Divida e Conquiste*

Para a topologia de rede física apresentada na figura 7.1 (capítulo 7), pode-se dividir o Gerenciamento de Falhas em três domínios (Gerente A, Gerente B e Gerente C). Assim, será analisado cada domínio individualmente, quando uma falha ocorre na rede.

Simulação de falha:

Supor igualmente a Arquitetura de Gerenciamento Centralizada, que os objetos A_1 e C_2 emitiam uma mensagem notificação de falha.

1.1. Sub-gerente A

Pode-se obter o seguinte grafo de dependência para a arquitetura de Gerenciamento Distribuído referente ao domínio de gerenciamento do Sub-gerente A:

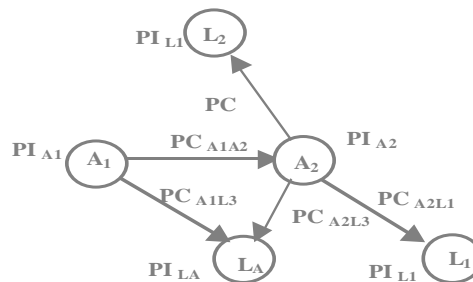


Figura C.1.1: Grafo de dependência para Gerenciamento de Falhas do Sub-gerente A.

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC) para este Sub-domínio A, estas probabilidades são compatíveis com as probabilidades da tabela B.1.1:

$PI_{A1} = 0,02$	$PI_{L2} = 0,02$	$PC_{A2LA} = 0,03$
$PI_{A2} = 0,03$	$PC_{A1A2} = 0,01$	$PC_{A2L2} = 0,04$
$PI_{LA} = 0,02$	$PC_{A1LA} = 0,02$	$PC_{A2L1} = 0,02$
$PI_{L1} = 0,01$		

Tabela C.1.1: Probabilidades de falhas dos objetos gerenciados pelo Sub-gerente A.

Adicionando as probabilidades ao grafo tem-se:

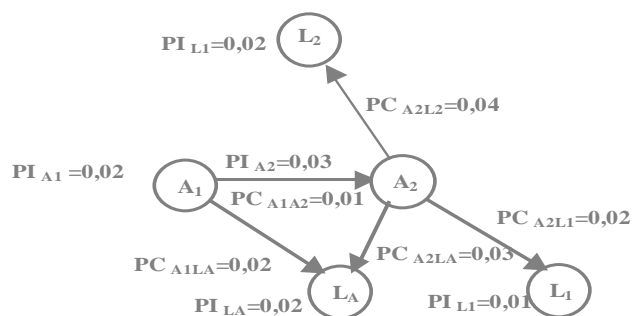


Figura C.1.2: Grafo de dependência adicionado as probabilidades de falhas

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e o respectivo alarme gerado.

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
A_1	A_2, L_A	E_1
A_2	L_A, L_1, L_2	E_2
L_A	-	E_3
L_1	-	E_4
L_2	-	E_5

Tabela C.1.2: Relacionamento objetos sinalizados e alarmes gerados para o Sub-gerente A.

Assim os alarmes são formados como a seguir:

$$E_1 = \{ A_1, A_2, L_A \}; E_2 = \{ A_2, L_A, L_1, L_2 \}; E_3 = \{ L_A \}; E_4 = \{ L_1 \}; E_5 = \{ L_2 \}.$$

Relação de alarmes cobertos para cada objeto:

Objeto	Alarmes cobertos pelo objeto
A ₁	E ₁
A ₂	E ₁ , E ₂
L _A	E ₃
L ₁	E ₄
L ₂	E ₅

Tabela C.1.3: Relacionamento objetos gerenciados e alarmes gerados para Sub-gerente A .

Supor também para todo este capítulo que os alarmes gerados estejam associados a tipos (Comunicação, Qualidade de Serviço, Processamento e Ambiental), e associados a níveis de severidade (Cleared, Indeterminado, Crítico, Maior e Menor).

Simulação de falha:

Como foi suposto inicialmente, o objeto A₁ envia uma mensagem de falha ao Sub-gerente A, esta falha provoca a sinalização do alarme E₁. Este Sub-gerente executa o algoritmo de localização de falha utilizando o seu grafo de dependência como mostrado a seguir.

Fase de União

Passo 1:

Dado o grafo fornecido inicialmente para o Sub-gerente A:

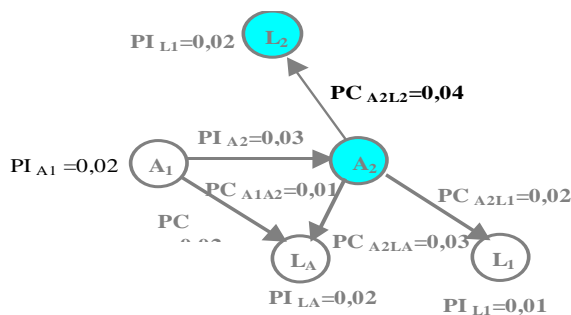


Figura C.1.3: União dos objetos A₂ e L₂

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Neste caso, existem apenas uma dependência com o valor de 0,04 (entre os objetos C₂ e L₂). Cálculo da probabilidade incondicional da nova partição:

$$PU_{A_2L_2} = \frac{0,03}{\text{objeto } A_2} + \frac{0,04 \times 0,02}{\text{objeto } L_2} + \frac{0,02}{\text{objeto } L_2} = 0,0508 = PI_{A_2L_2}$$

Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.



Figura C.1.4: Objeto final da fase de união

Fase de particionamento

Dado o objeto final da fase de união apresentado, pode-se obter o objeto inicial para a fase de particionamento, conforme a seguir:

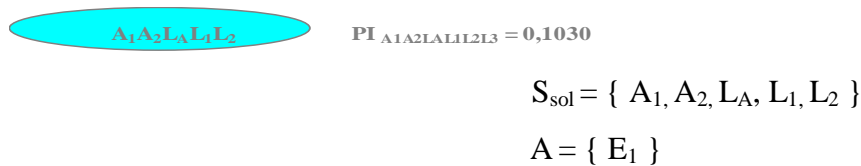


Figura C.1.5: Objeto inicial para a fase de particionamento

Fase de particionamento

Passo 1:

Particionar o objeto em:

- Filho direito: $A_1A_2L_AL_2$
- Filho esquerdo: L_1

Assim, obtém-se o seguinte grafo:

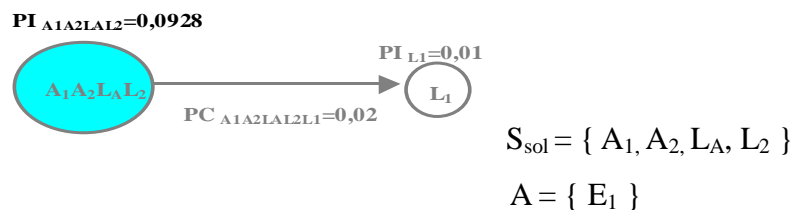


Figura C.1.6: Particionando os objetos em $A_1A_2L_AL_2$ e L_1

Selecionar o filho que cumpre os seguintes critérios:

3) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

4) Coberto por todos os alarmes ativos:

O filho esquerdo é coberto pelo alarme ativo E_1 . O filho direito não é coberto por nenhum alarme ativo. Portanto, o filho esquerdo é selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se o conjunto solução final descrito a seguir.

Solução final:

O conjunto solução final para o Sub-gerente A será: $S_{sol} = \{ A_2 \}$.

1.2. Sub-gerente B

Pode-se obter o seguinte grafo de dependência para a arquitetura de Gerenciamento Distribuído para o domínio do Sub-gerente B:

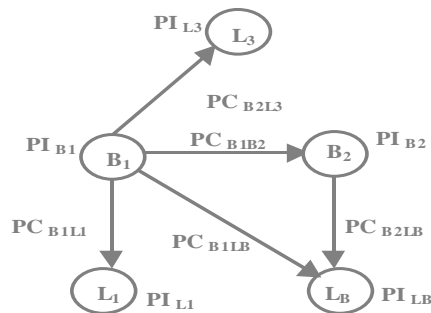


Figura C.1.7: Grafo de dependência para Gerenciamento de Falhas do Sub-gerente B.

Supor que os objetos de gerenciamento apresentam as seguintes probabilidades de falha incondicional (PI) e condicional (PC) para este Sub-domínio B, estas probabilidades são compatíveis com as probabilidades da tabela B.1.1:

$PI_{B1} = 0,02$	$PI_{L3} = 0,02$	$PC_{B1L1} = 0,02$
$PI_{B2} = 0,03$	$PC_{B1B2} = 0,01$	$PC_{B1L3} = 0,02$
$PI_{LB} = 0,02$	$PC_{B1LB} = 0,02$	$PC_{B2LB} = 0,03$
$PI_{L1} = 0,01$		

Tabela C.1.4: Probabilidades de falhas dos objetos gerenciados para o Sub-gerente B.

Adicionando as probabilidades ao grafo tem-se:

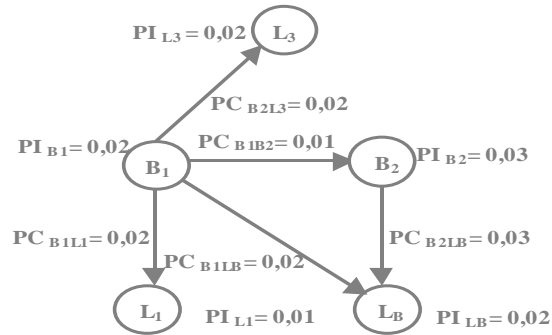


Figura C.1.8: Grafo de dependência adicionado-se suas probabilidades de falhas

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e o respectivo alarme gerado:

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
B ₁	B ₂ , L _B , L ₁ , L ₃	E ₁
B ₂	L _B	E ₂
L _B	-	E ₃
L ₁	-	E ₄
L ₃	-	E ₅

Tabela C.1.5: Relacionamento entre objetos sinalizados e alarmes gerados

Assim os alarmes são formados como a seguir:

$$E_1 = \{ B_1, B_2, L_B, L_1, L_3 \}; E_2 = \{ B_2, L_B \}; E_3 = \{ L_B \}; E_4 = \{ L_1 \}; E_5 = \{ L_3 \}.$$

Relação de objetos e os alarmes que os cobrem:

Objeto	Alarmes cobertos pelo objeto
B ₁	E ₁
B ₂	E ₁ , E ₂
L _B	E ₁ , E ₂ , E ₃
L ₁	E ₁ , E ₄
L ₃	E ₁ , E ₅

Tabela C.1.6: Objetos gerenciados e objetos cobertos pelos alarmes

Simulação de falha:

Como foi suposto inicialmente, o objeto não B_1 envia nenhuma mensagem de falha ao Sub-Gerente B e, portanto, não há nenhuma sinalização de alarme ativo. Este sub-gerente irá executar o algoritmo de localização de falha utilizando o seu grafo de dependência como mostrado a seguir.

Fase de União

Passo 1:

Dado o grafo fornecido inicial para o Sub-gerente A:

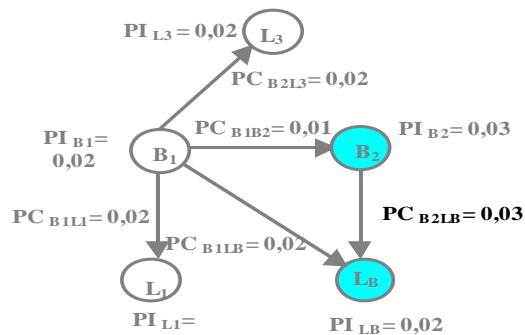


Figura C.1.9: União dos objetos B_2 e L_2

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Neste caso, a dependência B_2 e L_2 é a maior dependência e apresenta o valor de 0,03.

Cálculo da probabilidade incondicional da nova partição:

$$PU_{B_2L_2} = \frac{0,03 + 0,03 \times 0,02 + 0,02}{0,02} = 0,0506 = PI_{B_2L_2}$$

objeto B_2 objeto L_2

Pela figura C.1.9, observa-se que existe dois arcos de probabilidade condicional a partir do objeto B_1 para o novo objeto B_2L_2 com os valores de 0,01 (do objeto B_1 para o objeto B_2) e 0,02 (do objeto B_1 para o objeto L_2). Assim, deverá ser selecionado apenas a maior dependência a partir do objeto B_1 para o novo objeto B_2L_2 , ou seja, a dependência de valor 0,02. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.

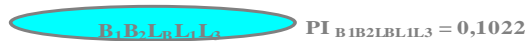


Figura C.1.10: Objeto final da fase de união

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

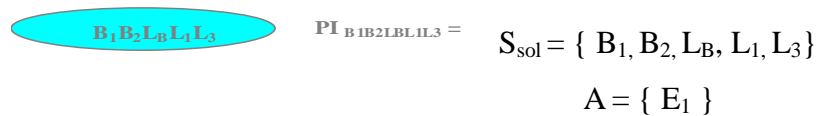


Figura C.1.11: Objeto obtido na fase de união

Fase de particionamento

Passo 1:

Particionar o objeto em:

- Filho direito: $B_1B_2L_B L_1$
- Filho esquerdo: L_3

Assim, obtém-se o seguinte grafo:

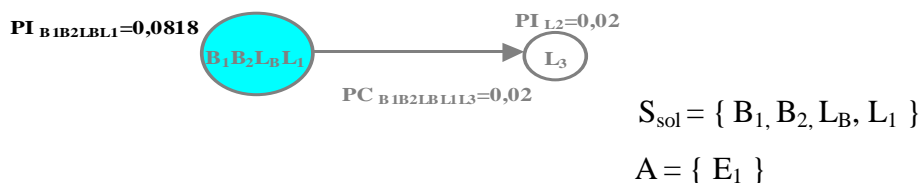


Figura C.1.12: Particionando os objetos em $B_1B_2L_B L_1$ e L_3

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

Não existe nenhum alarme ativo, portanto, este critério não é considerado nos demais passos.

Portanto, o filho esquerdo é o selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de particionamento descrita a seguir.

Solução final:

O conjunto solução final será $S_{Sol} = \{ B_2 \}$, sendo esta a solução final para o Sub-gerente B

1.3. Sub-gerente C

Pode-se obter o seguinte grafo de dependência para a arquitetura de Gerenciamento Distribuído para o domínio do Sub-gerente C:

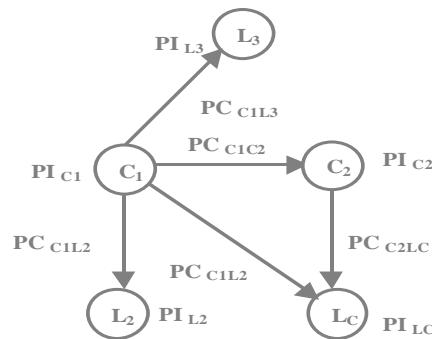


Figura C.1.13: Grafo de dependência para Gerenciamento de Falhas do Sub-gerente C.

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC) para este Sub-domínio C, estas probabilidades são compatíveis com as probabilidades da tabela B1.1:

$PI_{C1} = 0,02$	$PI_{L3} = 0,02$	$PC_{C1L2} = 0,06$
$PI_{C2} = 0,03$	$PC_{C1C2} = 0,01$	$PC_{C1L3} = 0,02$
$PI_{LC} = 0,02$	$PC_{C1LC} = 0,02$	$PC_{C2LC} = 0,03$
$PI_{L2} = 0,02$		

Tabela C.1.7: Probabilidades de falhas dos objetos gerenciados para o Sub-gerente C.

Adicionando as probabilidades ao grafo tem-se:

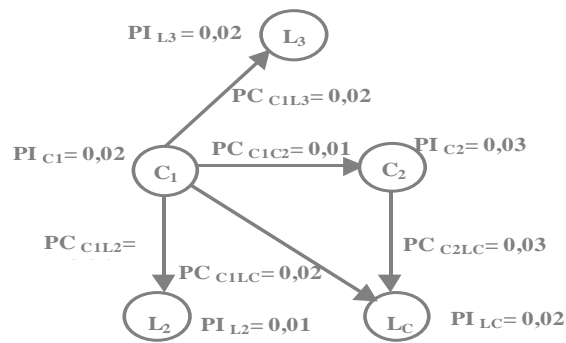


Figura C.1.14: Grafo de dependência adicionado as probabilidades de falhas

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e o respectivo alarme gerado:

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
C_1	C_2, L_C, L_2, L_3	E_1
C_2	L_C	E_2
L_C	-	E_3
L_2	-	E_4
L_3	-	E_5

Tabela C.1.8 : Objetos sinalizados e alarmes gerados para o Sub-gerente C.

Assim o alarmes são formados como a seguir:

$$E_1 = \{ C_1, C_2, L_C, L_2, L_3 \}; E_2 = \{ C_2, L_C \}; E_3 = \{ L_C \}; E_4 = \{ L_2 \}; E_5 = \{ L_3 \}.$$

Relação de objetos cobertos por alarmes:

Objeto	Alarmes cobertos pelo objeto
C_1	E_1
C_2	E_1, E_2
L_C	E_1, E_2, E_3
L_2	E_1, E_4
L_3	E_1, E_5

Tabela C.1.9: Objetos gerenciados e objetos cobertos por alarmes para o Sub-gerente C.

Simulação de falha:

Como foi suposto inicialmente, o objeto C_2 envia uma mensagem de falha ao Sub-gerente C, esta falha provoca a sinalização do alarme E_1 e E_2 . Este sub-gerente irá executar o

algoritmo de localização de falha utilizando o seu grafo de dependência como mostrado a seguir:

Fase de União

Passo 1:

Dado o grafo fornecido inicialmente para o Sub-gerente C:

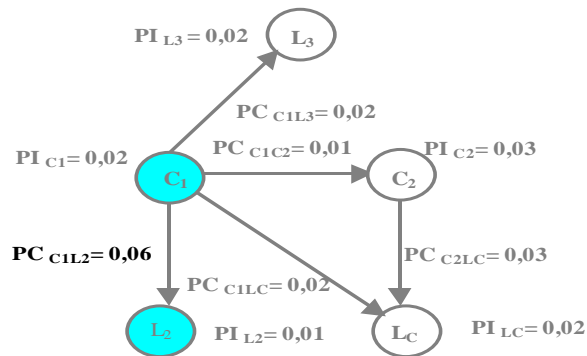


Figura C.1.15: União dos objetos C_1 e L_2

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Neste caso, a dependência C_1 e L_2 tem a maior dependência com valor de 0,06.

Cálculo da probabilidade incondicional da nova partição:

$$PU_{C_1L_2} = \frac{0,02}{\text{objeto } C_1} + \frac{0,06 \times 0,02}{\text{objeto } L_2} + \frac{0,02}{\text{objeto } L_C} = 0,0412 = PI_{C_1L_2}$$

objeto C_1 objeto L_2

Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.

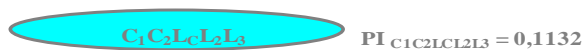


Figura C.1.16: Objeto final da fase de união

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

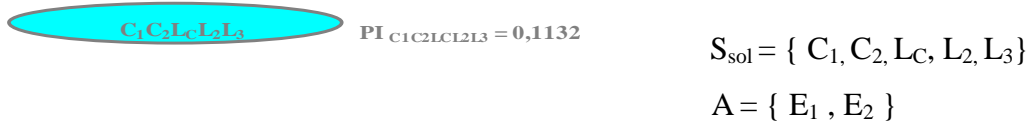


Figura C.1.17: Objeto obtido na fase de união

Fase de particionamento

Passo 1:

Particionar o objeto em:

- Filho direito: $B_1B_2L_B L_1$
- Filho esquerdo: L_3

Assim, obtém-se o seguinte grafo:

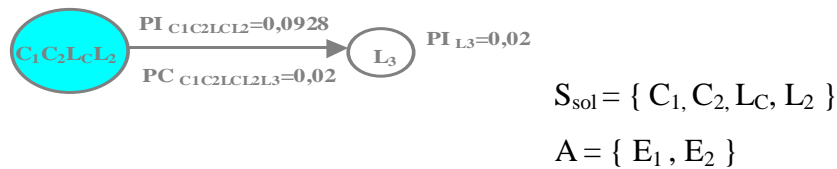


Figura C.1.18: Particionando os objetos em $C_1C_2L_cL_2$ e L_3

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho esquerdo é coberto pelos alarmes ativos: E_1 e E_2 . Portanto, o filho esquerdo é selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de particionamento descrita a seguir.

Solução final:

O conjunto solução final será $S_{sol} = \{ C_2 \}$ para o Sub-gerente C.

2. Aplicação utilizando o algoritmo *Divida e Conquiste Proposto*

Da mesma maneira que a topologia de rede física apresentada no início deste capítulo, pode-se dividir o Gerenciamento de Falhas em três domínios (Gerente A, Gerente B e Gerente C). Assim, será analisado cada domínio individualmente, quando uma falha ocorrer na rede.

Simulação de falha:

Supor igualmente que os objetos A_1 e C_2 emitiam uma mensagem notificação de falha.

2.1. Sub-gerente A

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC), compatíveis com as probabilidades da tabela C.1.1, adicionado o fator de cobertura (C) para este sub-domínio, conforme a seguir:

$C_{A1} = 0,7$	$C_{LA} = 0,8$	$C_{L2} = 0,8$
$C_{A2} = 0,6$	$C_{L1} = 0,9$	

Tabela C.2.1: Fator de Cobertura dos objetos gerenciados para o Sub-gerente A .

Adicionando as probabilidades ao grafo tem-se:

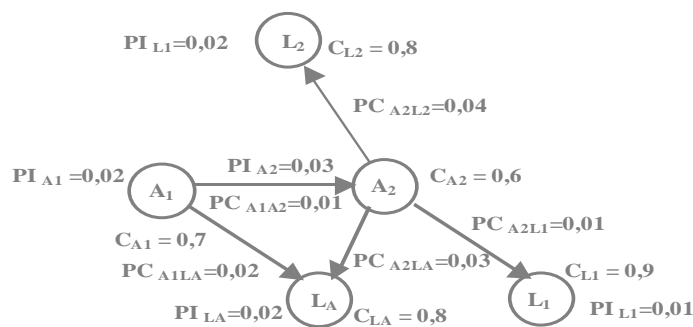


Figura C.2.1: Grafo de dependência adicionado as probabilidades de falhas

Fase de União

A fase de união utiliza quatro critérios em cada passo de sua execução com a finalidade de encontrar o objeto com maior probabilidade de ser uma falha primária, estes critérios são:

- 1) Unir os dois objetos com maior dependência condicional, caso existam mais de dois objetos que cumpram este critério, deve-se,
- 2) Selecionar a maior dependência que inclua o maior número de alarmes ativos, caso existam mais de dois objetos que cumpram este critério, deve-se,
- 3) Unir os dois objetos com maior dependência incondicional, caso existam mais de dois objetos que cumpram este critério, deve-se,
- 4) Unir dois objetos arbitrariamente.

Passo 1:

Dado o grafo fornecido inicial para o Sub-gerente A:

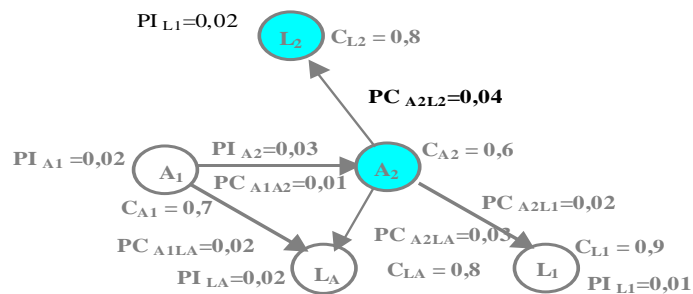


Figura C.2.2: União dos objetos A_2 e L_2

Aplicando-se o critério para a fase de união:

- 1) Unir os objetos com maior probabilidade condicional

Neste caso, existe uma probabilidade condicional com o valor de dependência igual a 0,04 (C_2 e L_2).

- 2) Coberto pelo maior número de alarmes ativos:

Este item não necessita ser verificado, pois o item anterior apresenta um única dependência.

- 3) Unir os objetos com maior probabilidade incondicional.

Não é necessário verificar este terceiro critério.

4) Unir dois objetos arbitrariamente.

Também não é necessário verificar este critério.

Cálculo da probabilidade incondicional da nova partição:

$$PU_{A_2L_2} = \frac{0,03 \times 0,6 + 0,04 \times 0,02 \times 0,6}{\text{objeto } A_2} + \frac{0,02 \times 0,8}{\text{objeto } L_2} = 0,0345 = PI_{A_2L_2}$$

Cálculo do fator de cobertura para nova partição:

$$\text{Sendo, } PU_{A_2} = 0,03 \times 0,6 + 0,04 \times 0,02 \times 0,6 = 0,0185 \text{ e}$$

$$PU_{L_2} = 0,02 \times 0,8 = 0,016$$

$$FCU_{A_2L_2} = (P_{A_2} / PU_{A_2L_2}) \times C_{A_2} + (P_{L_2} / PU_{A_2L_2}) \times C_{L_2}$$

$$FCU_{A_2L_2} = (0,0185 / 0,0345) \times 0,6 + (0,016 / 0,0345) \times 0,8 = 0,6928$$

Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.

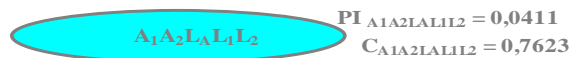


Figura C.2.3: Objeto final da fase de união

Ou seja, o domínio de gerenciamento do Sub-gerente A tem uma probabilidade de falha de 0,0411, e 0,7623 (76,23%) das falhas deste domínio são cobertos pelo sistema de gerenciamento.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

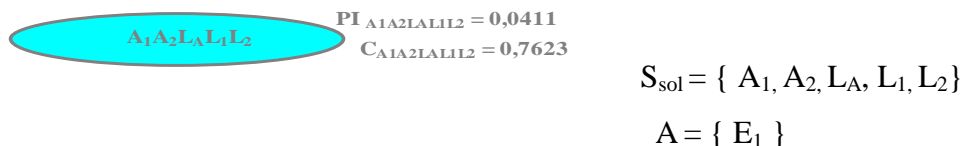


Figura C.2.4: Objeto obtido na fase de união

Fase de particionamento

Passo 1:

Particionar o objeto em:

- Filho direito: $A_1A_2L_AL_2$
- Filho esquerdo: L_1

Assim, obtém-se o seguinte grafo:

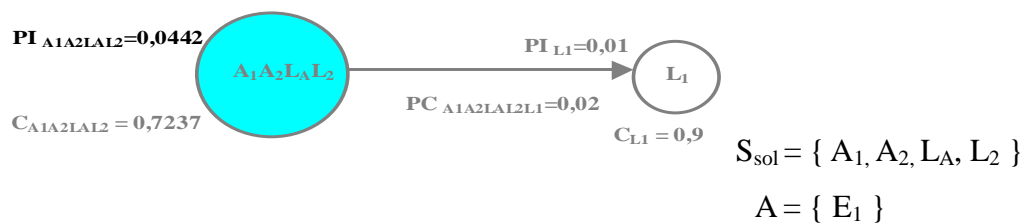


Figura C.2.5: Particionando os objetos em $A_1A_2L_AL_2$ e L_1

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho esquerdo é coberto pelo alarme ativo E_1 . O filho direito não é coberto por nenhum alarme ativo. Portanto, o filho esquerdo é selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se o conjunto solução descrito a seguir.

Solução final:

O conjunto solução final será composto pela união das soluções parciais $S_{sol1} = \{ A_1 \}$ e $S_{sol2} = \{ A_2 \}$, ou seja, a solução final para o Sub-gerente A será: $S_{sol} = \{ A_1, A_2 \}$

Determinação da seqüência de teste:

A fase de testes deverá ser realizados nos objetos A_1 e A_2 .

Neste caso, a seqüência de teste poderá ser baseada nas probabilidades de falha dos objetos:

$$PU_{A1} = \frac{0,02 \times 0,7}{PI_{A1}} + \frac{0,01 \times 0,03 \times 0,7}{PC_{A1A2}} + \frac{0,02 \times 0,02 \times 0,7}{PC_{A1LA}} = 0,0145$$

$$PU_{A2} = \frac{0,03 \times 0,6}{PI_{A2}} + \frac{0,04 \times 0,02 \times 0,6}{PC_{A2L2}} + \frac{0,02 \times 0,01 \times 0,6}{PC_{A2L1}} + \frac{0,04 \times 0,02 \times 0,6}{PC_{A2L2}} = 0,0189$$

A seqüência de testes deverá ser realizada na ordem decrescente das probabilidades de falhas, ou seja, deverá ser feito o teste no objeto A₂ e após no objeto A₁. Como nenhum destes objetos é fonte primária de falha, se a fase teste for realizada neste domínio, não será detectada nenhum elemento falho.

2. 2. Sub-gerente B

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC) e fator de cobertura (C) para este Sub-domínio B, compatíveis com as probabilidades da tabela C.1.4, adicionado o fator de cobertura (C) para este sub-domínio, conforme a seguir:

$C_{B1} = 0,7$	$C_{LB} = 0,8$	$C_{L13} = 0,9$
$C_{B2} = 0,6$	$C_{L1} = 0,9$	

Tabela C.2.2: Probabilidades de falhas dos objetos gerenciados para o Sub-gerente B.

Adicionando as probabilidades ao grafo tem-se:

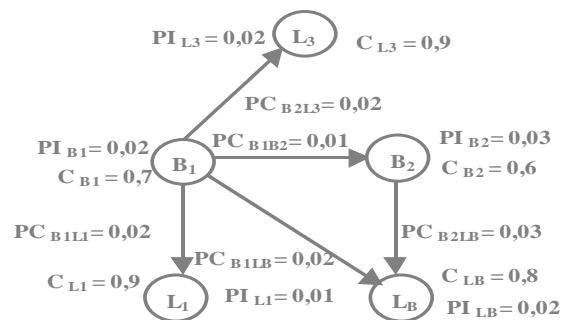


Figura C.2.6: Grafo de dependência adicionado as probabilidades de falhas

Fase de União

Passo 1:

Dado o grafo fornecido inicialmente para o Sub-gerente A:

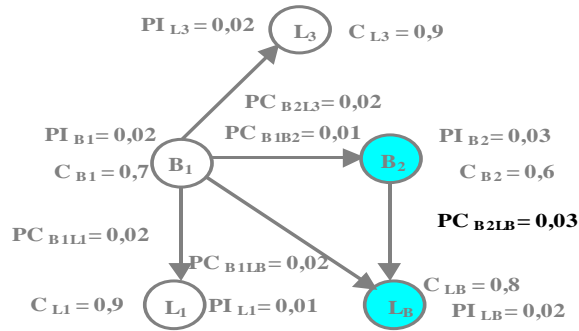


Figura C.2.7: União dos objetos B₂ e L_B

Aplicando-se o critério para a fase de união:

1) Unir os objetos com maior probabilidade condicional

Neste caso, a dependência B₂ e L_B tem o maior valor de 0,03.

Não é necessário verificar os demais critérios para este passo, pois existe somente uma dependência no critério anterior.

Cálculo da probabilidade incondicional da nova partição:

$$PU_{B_2L_B} = \underbrace{0,03 \times 0,6}_{\text{objeto } B_2} + \underbrace{0,03 \times 0,02 \times 0,6}_{\text{objeto } L_2} + \underbrace{0,02 \times 0,8}_{\text{objeto } L_2} = 0,0344 = PI_{B_2L_B}$$

Cálculo da probabilidade incondicional da nova partição:

Sendo, $PU_{B_2} = 0,03 \times 0,6 + 0,03 \times 0,02 \times 0,6 = 0,0184$, e

$$PU_{L_B} = 0,02 \times 0,8 = 0,016$$

$$FCU_{B_2L_B} = (P_{B_2} / PU_{B_2L_B}) \times C_{B_2} + (PU_{L_B} / PU_{B_2L_B}) \times C_{L_B}$$

$$FCU_{A_2C_{1L_2}} = (0,0184 / 0,0344) \times 0,6 + (0,016 / 0,0344) \times 0,8 = 0,6930$$

No grafo anterior observa-se dois arcos de dependência entre os objetos B₁ e B₂, e entre os objetos B₁ e L_B com os valores respectivos de 0,01 e 0,02, deve-se selecionar o maior valor de dependência para o novo objeto B₂L_B, ou seja, o valor 0,02. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.

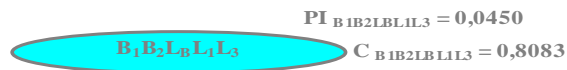


Figura C.2.8: Objeto final da fase de união

Ou seja, o domínio de gerenciamento do Sub-gerente B tem uma probabilidade de falha de 0,0450, e 0,8053 (80,53%) das falhas deste domínio são cobertos pelo sistema de gerenciamento.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

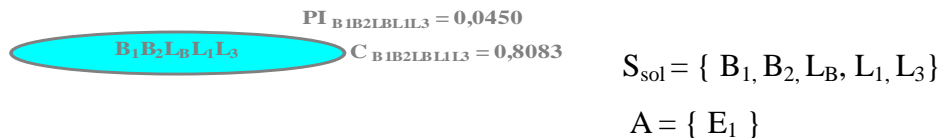


Figura C.2.9: Objeto obtido na fase de união

Passo 1:

Particionar o objeto em:

- Filho direito: $B_1B_2L_B L_1$
- Filho esquerdo: L_3

Assim, obtém-se o seguinte grafo:

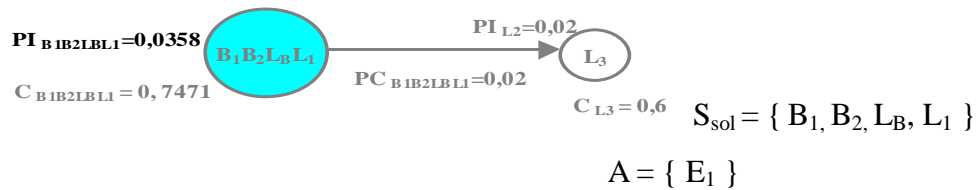


Figura C.2.10: Particionando os objetos em $B_1B_2L_B L_1$ e L_3

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelo alarme ativo E_1 .

O filho esquerdo é coberto pelo alarme ativo E_1 .

O filho esquerdo é selecionado, pois cumpre ambos os critérios. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de particionamento descrita a seguir.

Solução final:

O conjunto solução será igual a $S_{Sol} = \{ B_2 \}$, esta é a solução final para o Sub-gerente B.

Determinação da seqüência de teste:

A fase de testes deverá ser realizada no objetos B_2 . Como este objeto não é fonte primária de falha, se a fase teste for realizada para este objeto não será detectado nenhuma falha.

3.2. 3. Sub-gerente C

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falha incondicional (PI) e condicional (PC) e o fator de cobertura (C) para este Sub-domínio C, compatíveis com as probabilidades da tabela C.1.7, adicionado o fator de cobertura (C) para este sub-domínio, conforme a seguir:

$C_{C1} = 0,7$	$C_{LC} = 0,8$	$C_{L3} = 0,9$
$C_{C2} = 0,7$	$C_{L2} = 0,8$	

Tabela C.2.3: Probabilidades de falhas dos objetos gerenciados para o Sub-gerente C.

Adicionando as probabilidades ao grafo tem-se:

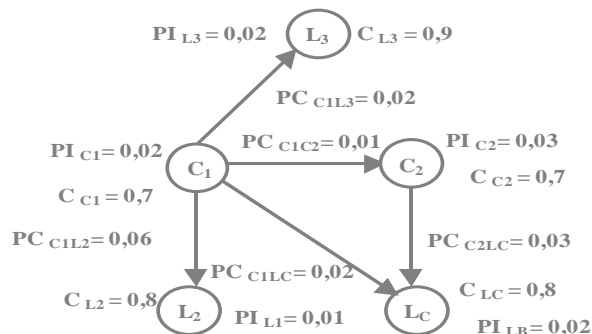


Figura C.2.11: Grafo de dependência adicionado as probabilidades de falhas

Fase de União

Passo 1:

Dado o grafo fornecido inicialmente para o Sub-gerente C:

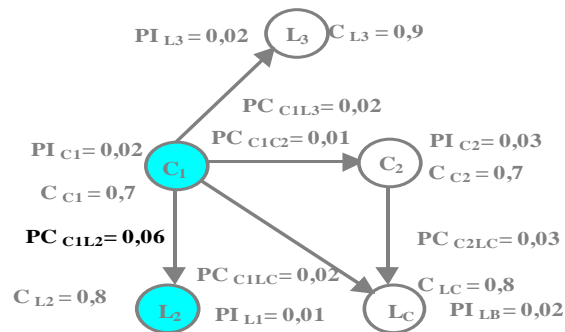


Figura C.2.13: União dos objetos C₁ e L₂

Aplicando-se o critério para a fase de união:

1) Unir os objetos com maior probabilidade condicional

Neste caso, a dependência entre os objetos C₁ e L₂ tem o maior valor de 0,06.

Os demais critérios não precisam ser verificados, pois o item anterior apresenta um único objeto selecionado.

Cálculo da probabilidade incondicional da nova partição:

$$PU_{C_1L_2} = \frac{0,02 \times 0,7 + 0,06 \times 0,02 \times 0,7}{\text{objeto } C_1 \quad \text{objeto } L_2} + \frac{0,02 \times 0,8}{\text{objeto } L_C} = 0,0308 = PI_{C_1L_2}$$

Cálculo da probabilidade incondicional da nova partição:

$$P_{C_1} = 0,02 \times 0,7 + 0,06 \times 0,02 \times 0,7 = 0,0148, \text{ e}$$

$$P_{L_2} = 0,02 \times 0,8 = 0,016$$

$$FCU_{C_1L_2} = (P_{C_1} / PU_{C_1L_2}) \times C_{C_1} + (P_{L_2} / PU_{C_1L_2}) \times C_{L_2}$$

$$FCU_{A_2C_1L_2} = (0,0148 / 0,0308) \times 0,7 + (0,016 / 0,0308) \times 0,8 = 0,7519$$

Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir.

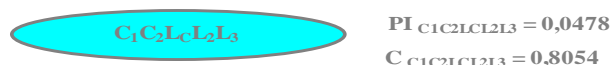


Figura C.2.14: Objeto final da fase de união

Ou seja, o domínio de gerenciamento do Sub-gerente C tem uma probabilidade de falha de 0,0478, e 0,8054 (80,54%) das falhas deste domínio são cobertos pelo sistema de gerenciamento.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

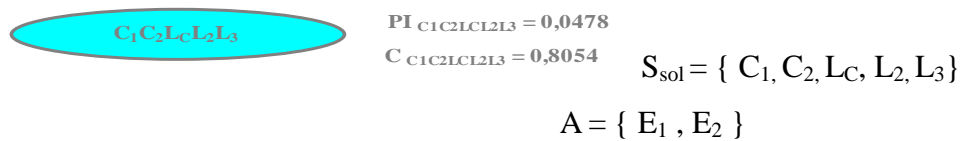


Figura C.2.15: Objeto obtido na fase de união

Fase de particionamento

Passo 1:

Particionar o objeto em:

- Filho direito: B₁B₂L_BL₁
- Filho esquerdo: L₃

Assim, obtém-se o seguinte grafo:

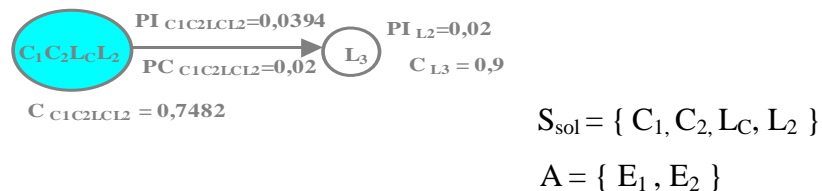


Figura C.2.16: Particionando os objetos em C₁C₂L_cL₂ e L₃

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho esquerdo apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho esquerdo é coberto pelos alarmes ativos: E₁ e E₂. Portanto, o filho esquerdo é selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de particionamento descrita a seguir.

Solução final:

O conjunto solução final será composto pelos conjuntos soluções parciais $S_{sol\ 1} = \{ C_2 \}$ e $S_{sol\ 2} = \{ C_1, L_2 \}$. Portanto, o conjunto solução $S_{sol} = \{ C_1, C_2, L_2 \}$ é a solução final para o Sub-gerente C.

Determinação da seqüência de teste:

A fase de testes deverá ser realizada em função do valor decrescente da probabilidade de falha. Assim, calcula-se a probabilidade de falha dos objetos selecionados para a fase de teste:

$$PU_{C1} = \frac{0,02 \times 0,7}{PI_{C1}} + \frac{0,02 \times 0,02 \times 0,7}{PC_{C1LC}} + \frac{0,06 \times 0,02 \times 0,7}{PC_{C1L2}} + \frac{0,02 \times 0,02 \times 0,7}{PC_{C1L3}} = 0,0154$$

$$PU_{C2} = \frac{0,03 \times 0,7}{PI_{C1}} + \frac{0,03 \times 0,02 \times 0,7}{PC_{C2LC}} = 0,0214$$

$$PU_{L2} = \frac{0,02 \times 0,7}{PI_{C1}} = 0,014$$

Portanto a seqüência de teste deverá ser realizada na seguinte ordem: C_2, C_1, L_2 . Como o objeto C_2 é fonte primária de falha, então, a falha será detectada no primeiro teste realizado, os demais objetos podem não necessitar ser testados.

ANEXO D

Arquitetura de Gerenciamento de Falhas Hierárquico

1. Aplicação utilizando o algoritmo *Divida e Conquiste*

Serão analisados primeiramente o nível 1 de gerenciamento e após o nível 2 do capítulo 7.3.

Nível 1 de Gerenciamento:

Localidade A

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falhas, compatíveis com a aplicação da tabela B.1.1, conforme a seguir:

$PI_{A1} = 0,02$	$PI_{A2} = 0,03$	$PC_{A1LA} = 0,02$
$PI_{LA} = 0,02$	$PC_{A1A2} = 0,01$	$PC_{A2LA} = 0,03$

Tabela D.1.1: Probabilidades de falhas na localidade A.

Adicionando as probabilidades ao grafo tem-se:

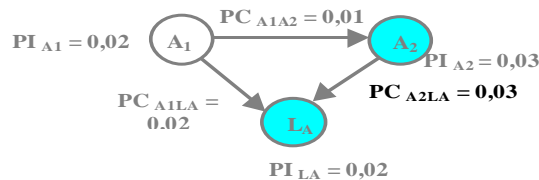


Figura D.1.1: Adicionando probabilidades de falha ao grafo do domínio A

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e os respectivos alarmes gerados:

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
A ₁	A ₂ e L _A	E ₁
A ₂	L _A	E ₂
L _A	-	E ₃

Tabela D.1.2: Objetos sinalizados e alarmes gerados.

Assim os alarmes são formados como a seguir:

$$E_1 = \{ A_1, A_2, L_A \}; E_2 = \{ A_2, L_A \}; E_3 = \{ L_A \}.$$

Relação de alarmes cobertos para cada objeto:

Objeto	Alarmes que cobrem o objeto
A ₁	E ₁
A ₂	E ₁ , E ₂
L _A	E ₁ , E ₂ , E ₃

Tabela D.1.3: Relação entre objetos e alarmes gerados

Algoritmo *Divida e Conquiste*

Ao final da fase de união do algoritmo *Divida e Conquiste* tem-se um único objeto com sua correspondente probabilidade incondicional, este objeto corresponde ao nó A do nível 2 de gerenciamento.

Fase de União

A fase de união utiliza dois critérios em cada passo de sua execução com a finalidade de encontrar o objeto com maior probabilidade de ser uma falha primária, conforme descrito no exemplo anterior:

Passo 1:

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Verifica-se que a maior dependência neste caso ocorre entre os objetos A₂ e L_A.

A probabilidade do novo objeto A₂L_A pode ser calculado como:

$$PU_{A_2L_A} = \frac{0,03 + 0,03 \times 0,02}{\text{Objeto } A_2} + \frac{0,02}{\text{Objeto } L_A} = 0,0506 = PI_{A_2L_A}$$

Pela figura D.1.1 verifica-se que existem dois arcos de probabilidade condicional a partir do objeto A₁ para os objetos A₂ e L_A. Selecionando-se a probabilidade condicional de maior valor, ou seja, 0,02.

Passo 2:

Assim obtém-se o seguinte grafo de dependência:

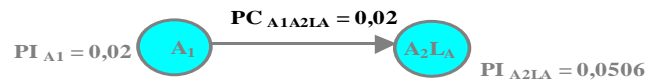


Figura D.1.2: União dos objetos A_1 e A_2L_A

Aplicando-se o critério para a fase de união:

1) Unir os objetos com maior probabilidade condicional

Como existe uma única dependência, esta é selecionado.

2) Selecionar a maior dependência que inclua o maior número de alarmes ativos.

Como existe uma única dependência, não é necessário verificar este segundo critério.

Cálculo da probabilidade incondicional da nova partição:

A probabilidade do novo objeto $A_1A_2L_A$ pode ser calculado como:

$$PU_{A_1A_2L_A} = \underbrace{0,02}_{\text{Objeto } A_2} + \underbrace{0,02 \times 0,0506}_{\text{Objeto } L_A} + 0,0506 = 0,0716 = PI_{A_1A_2L_A}$$

Assim, tem-se o seguinte objeto final para a localidade A:



Figura D.1.3: Objeto final da localidade A

A probabilidade deste objeto final será a probabilidade incondicional para a localidade A no grafo de dependência do nível 2 de gerenciamento.

Localidade B

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades, conforme a seguir:

$PI_{B1} = 0,02$	$PI_{B2} = 0,03$	$PC_{B1LB} = 0,02$
$PI_{LB} = 0,02$	$PC_{B1B2} = 0,01$	$PC_{B2LB} = 0,03$

Tabela D.1.4: Probabilidades de falhas para a localidade B.

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e os respectivos alarmes gerados:

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
B ₁	B ₂ e L _B	E ₁
B ₂	L _B	E ₂
L _B	-	E ₃

Tabela D.1.5: Relação entre objetos sinalizados e alarmes gerados.

Assim os alarmes são formados como a seguir:

$$E_1 = \{ B_1, B_2, L_B \}; E_2 = \{ B_2, L_B \}; E_3 = \{ L_B \}.$$

Relação de alarmes que cobrem cada objeto:

Objeto	Alarmes que cobrem o objeto
B ₁	E ₁
B ₂	E ₁ , E ₂
L _B	E ₁ , E ₂ , E ₃

Tabela D.1.6: Relação entre objeto gerenciado e alarmes gerados.

Adicionando as probabilidades ao grafo tem-se:

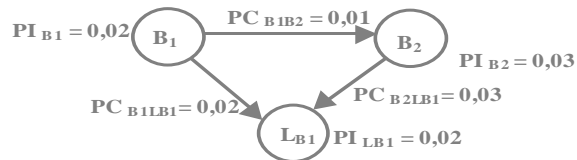


Figura D.1.4: Adicionando probabilidades de falha ao grafo do domínio B

Algoritmo *Divida e Conquiste*

Como a localidade B apresenta o mesmo modelo físico e lógico e as mesmas probabilidades condicionais e incondicionais, o objeto final da fase de união será análogo ao obtido para a localidade A, ou seja:



Figura D.1.5: Objeto final da localidade B

Da mesma forma que para a localidade A, a probabilidade deste objeto final será a probabilidade incondicional para a localidade B no grafo de dependência do nível 2 de gerenciamento.

Localidade C

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades, compatíveis com a tabela B.1.1, conforme a seguir:

$PI_{C_1} = 0,02$	$PI_{C_2} = 0,03$	$PC_{C_1L_C} = 0,02$
$PI_{L_C} = 0,02$	$PC_{C_1C_2} = 0,01$	$PC_{C_2L_C} = 0,03$

Tabela D.1.7: Probabilidades de falhas na localidade C.

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e os respectivos alarmes gerados:

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
C_1	C_2 e L_C	E_1
C_2	L_C	E_2
L_C	-	E_3

Tabela D.1.8: Relação entre objetos sinalizados e alarmes gerados.

Assim, os alarmes são formados como a seguir:

$$E_1 = \{ C_1, C_2, L_C \}; E_2 = \{ C_2, L_C \}; E_3 = \{ L_C \}.$$

Relação de alarmes que cobrem cada objeto:

Objeto	Alarmes que cobrem o objeto
C_1	E_1
C_2	E_1, E_2
L_C	E_1, E_2, E_3

Tabela D.1.9: Relação entre objeto gerenciado e alarmes gerados

Adicionando as probabilidades ao grafo tem-se:

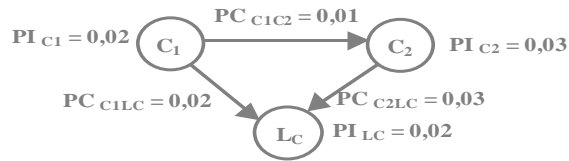


Figura D.1.6: Adicionando probabilidades de falha ao grafo do domínio C

Algoritmo *Divida e Conquiste*

Como a localidade C apresenta o mesmo modelo físico e lógico e as mesmas probabilidades condicionais e incondicionais, o objeto final da fase de união será análogo ao obtido para a localidade A e B, ou seja:



Figura D.1.7: Objeto final da localidade C

Nível 2 de Gerenciamento:

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades, estas probabilidades são compatíveis com as probabilidades da tabela B.1.1, conforme a seguir:

$PI_A = 0,0716$	$PC_{AC} = 0,06$
$PI_B = 0,0716$	$PC_{CB} = 0,01$
$PI_C = 0,0716$	$PC_{AB} = 0,04$
$PI_{L1} = 0,01$	$PC_{AL1} = 0,02$
$PI_{L2} = 0,02$	$PC_{CL2} = 0,06$
$PI_{L3} = 0,02$	$PC_{CL3} = 0,03$
$PI_{AL2} = 0,04$	$PC_{BL1} = 0,02$
$PI_{BL3} = 0,02$	

Tabela D.1.10: Probabilidades de falhas para o nível 2 de gerenciamento .

Adicionando as probabilidades ao grafo tem-se:

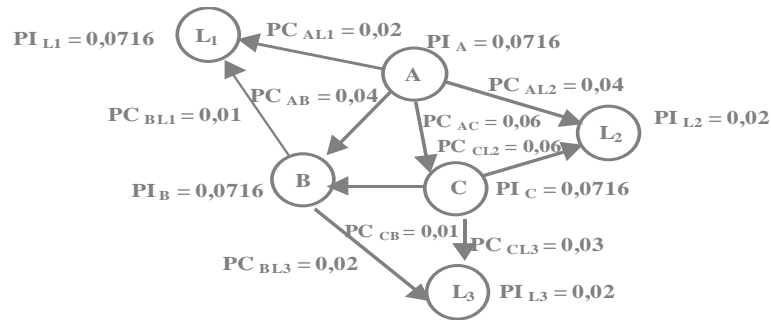


Figura D.1.8: Grafo de dependência adicionado as probabilidades de falhas

Tem-se para o exemplo em questão, a seguinte tabela relacionando: objeto sinalizado, objetos que o objeto sinalizado depende, e os respectivos alarmes gerados:

Objeto sinalizado	Objetos que o objeto sinalizado depende	Alarme gerado
A	B, C, L ₁ e L ₂	E ₁
B	L ₁ e L ₃	E ₂
C	B, L ₂ e L ₃	E ₃
L ₁	-	E ₄
L ₂	-	E ₅
L ₃	-	E ₆

Tabela D.1.11: Relação entre objetos gerenciados e alarmes gerados.

Assim o alarmes são formados como a seguir:

$E_1 = \{A, B, C, L_1, L_2\}$; $E_2 = \{B, L_1, L_3\}$; $E_3 = \{B, C, L_2, L_3\}$; $E_4 = \{L_1\}$; $E_5 = \{L_2\}$; $E_6 = \{L_3\}$.

Relação de alarmes cobertos para cada objeto:

Objeto	Alarmes que cobrem o objeto
A	E ₁
B	E ₁ , E ₂ , E ₃
C	E ₁ , E ₃
L ₁	E ₁ , E ₂ , E ₄
L ₂	E ₁ , E ₃ , E ₅
L ₃	E ₂ , E ₃ , E ₆

Tabela D.1.12: Relação entre objetos gerenciados e alarmes gerados.

Simulação de falha simples:

Supor igualmente a Arquitetura de Gerenciamento Centralizada, que os objetos A_1 e C_2 emitiam uma mensagem notificação de falha. Como consequência, o objeto de gerenciamento A_1 emite uma mensagem de falha ao Sub-gerente A e, o objeto de gerenciamento C_2 emite uma mensagem de falha ao Sub-gerente C. Ou seja, os Sub-gerentes A e C emitem uma mensagem de falha ao gerente de rede.

Nível 2 de Gerenciamento

No nível 2 de gerenciamento localiza-se o gerente de rede, ele tem a função de determinar em qual domínio existe a maior probabilidade de possuir um objeto que é fonte primária de falha. Para tanto, ele possui o grafo de dependência apresentado anteriormente (figura D.1.8) e a sinalização dos seguintes alarmes:

- O alarme E_1 é sinalizado devido a cobertura de falha no objeto A.
- Os alarmes E_1 e E_3 são sinalizados devido a cobertura da falha no objeto C.

Ou seja, são emitidos os alarmes E_1 e E_3 .

Simulação do algoritmo *Divida e Conquiste*

Partindo-se do grafo de dependência definido para este exemplo (figura D.1.8), e que foram sinalizados os alarmes E_1 e E_3 , pode-se aplicar os seguintes critérios da fase de união:

Fase de União:

Passo 1

Dado o grafo fornecido inicialmente para o nível 1 de gerenciamento:

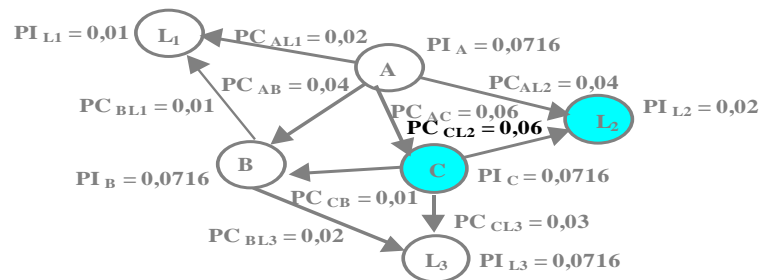


Figura D.1.9: União dos objetos C e L₂

Aplicando-se o critério para a fase de união:

- Unir os objetos com maior probabilidade condicional

Existem duas probabilidades condicionais com o mesmo valor de dependência igual a 0,06 (entre os objetos C e L₂, e entre os objetos A e L₂). Como ambas as dependências cumprem o critério, seleciona-se arbitrariamente a dependência entre os objetos C e L₂.

Cálculo da probabilidade incondicional da nova partição:

A probabilidade do novo objeto CL₂ pode ser calculado como:

$$PU_{CL_2} = \frac{0,0716}{\text{objeto C}} + \frac{0,06 \times 0,02}{\text{objeto L}_2} + \frac{0,02}{\text{objeto L}_2} = 0,0928 = PI_{CL_2}$$

Pela figura D.1.9, observa-se que existem dois arcos de probabilidade condicional a partir do objeto A para o novo objeto CL₂ com valores de 0,06 (do objeto A para o objeto C) e 0,04 (do objeto A para o objeto L₂). Assim, deverá ser selecionado apenas a maior dependência, ou seja, o valor 0,06. Os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir. Assim, obtém-se o seguinte objeto final, ou seja, o objeto D que representa toda a rede gerenciada:



Figura D.1.10: Objeto final ou D obtido na fase de união

Ou seja, a rede apresenta uma probabilidade de falha incondicional de 0,2753.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

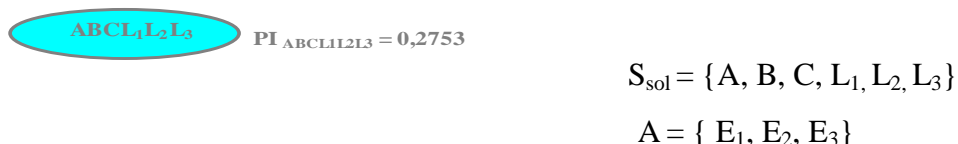


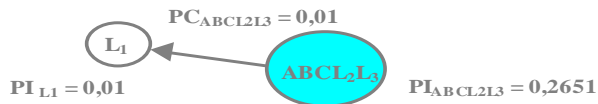
Figura D.1.11: Objeto obtido na fase de união

Passo 1:

Particionar o objeto em:

- Filho direito: L_1
- Filho esquerdo: $ABCL_2 L_3$

Assim, obtém-se o seguinte grafo:



$$S_{sol} = \{A, B, C, L_2, L_3\}$$

$$A = \{E_1, E_2, E_3\}$$

Figura D.1.12: Particionando os objetos em $ABCL_2L_3$ e L_1

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelos alarmes ativos: E_1 e E_3 . O filho esquerdo é coberto pelo alarme ativo E_1 . Portanto, o filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, sendo portanto o selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se a solução final descrita a seguir.

Solução final:

O conjunto solução final será o conjunto $S_{sol} = \{C\}$. Como o objeto C é a fonte primária de falha (suposição feita para o exemplo), a falha será detectada em seu domínio.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.



$$S_{sol} = \{C_1, C_2, L_C\}$$

$$A = \{E_1\}$$

Figura D.1.13: Objeto final da fase de união

Passo 1:

Particionar o objeto em:

- Filho direito: C_1
- Filho esquerdo: C_2L_C

Assim, obtém-se o seguinte grafo:

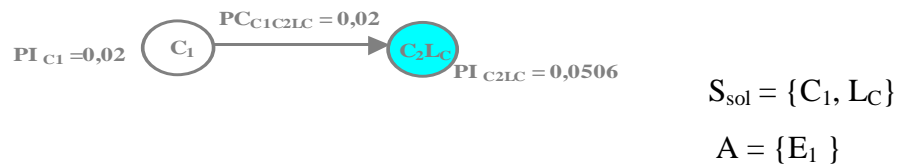


Figura D.1.14: Selecionando o filho C_2L_C

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

Ambos os filhos são cobertos pelo alarme ativo E_1 .

O filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, sendo portanto, o selecionado.

Passo 2:

Particionar o objeto em:

- Filho direito: C_2
- Filho esquerdo: L_C

Assim, obtém-se o seguinte grafo:

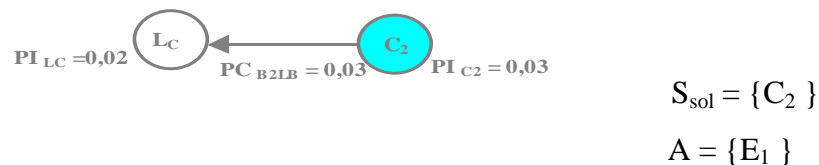


Figura D.1.15: Selecionando o filho C_2

Selecionar o filho que cumpre os seguintes critérios:

1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

2) Coberto por todos os alarmes ativos:

Ambos os filhos é coberto pelo alarme ativo E_1 . Portanto, o filho direito é o selecionado .

Assim, terá como solução final $S_{sol} = \{ C_2 \}$

Portanto, o Sub-gerente C deverá submeter o objeto C_2 a fase de teste, como este é a fonte primária de falha (suposição inicial), então, será encontrado o objeto falho no primeiro teste desta sub-rede.

2. Aplicação utilizando o algoritmo *Divida e Conquiste Proposto*

Nível 1 de Gerenciamento:

Localidade A

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falhas incondicional (PI), condicional (PC) compatíveis com as probabilidades da tabela D.1.1 e, seja adicionado os fatores de cobertura (C), conforme a seguir:

$C_{A1} = 0,7$	$C_{A2} = 0,6$	$C_{LA} = 0,8$
----------------	----------------	----------------

Tabela D.2.1: Probabilidades de falhas para a localidade A .

Adicionando as probabilidades ao grafo tem-se:

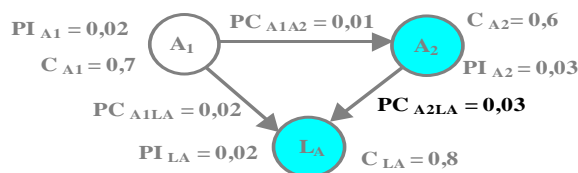


Figura D.2.1: Adicionando probabilidades de falha ao grafo do domínio A

Fase de União

Passo 1:

1) Unir os objetos com maior probabilidade condicional

Verifica-se que a maior dependência neste caso ocorre entre os objetos A_2 e L_A .

Não necessário verificar os demais critérios para este passo.

Cálculo da probabilidade incondicional da nova partição:

A probabilidade do novo objeto A_2L_A pode ser calculado como:

$$PU_{A_2L_A} = \frac{0,03 \times 0,6 + 0,03 \times 0,02 \times 0,6 + 0,02 \times 0,8}{\text{Objeto } A_2 \quad \text{Objeto } L_A} = 0,0344 = PI_{A_2L_A}$$

Cálculo do valor do fator de cobertura para a nova partição:

$$FCU_{A_2L_A} = (PU_{A_2} / PU_{A_2L_A}) \times C_{A_2} + (PU_{L_A} / PU_{A_2L_A}) \times C_{L_A}$$

sendo, $PU_{A_2} = 0,6 \times 0,03 + 0,03 \times 0,02 \times 0,6 = 0,0184$, e

$$PU_{L_A} = 0,8 \times 0,02 = 0,016$$

$$FCU_{A_2L_B} = (0,0184/0,0344) \times 0,6 + (0,016/ 0,0344) \times 0,8 = 0,6930 = C_{A_2L_B}$$

Pela figura D.2.1 verifica-se que existem dois arcos de probabilidade condicional a partir do objeto A_1 para os objetos A_2 e L_A ,

Selecionando-se a probabilidade condicional de maior valor, ou seja, 0,02.

Passo 2:

Assim obtém-se o seguinte grafo de dependência:

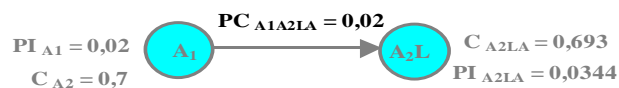


Figura D.2.2: União dos objetos A_1 e A_2L_A

Aplicando-se o critério para a fase de união:

1) Unir os objetos com maior probabilidade condicional

Como existe uma única dependência, esta é selecionado.

2) Selecionar a maior dependência que inclua o maior número de alarmes ativos.
 Como existe uma única dependência, não é necessário verificar este segundo critério.
 Cálculo da probabilidade incondicional da nova partição:

A probabilidade do novo objeto $A_1A_2L_A$ pode ser calculado como:

$$PU_{A_1A_2L_A} = \frac{0,02 \times 0,7 + 0,02 \times 0,0344 \times 0,7}{\text{Objeto } A_2} + \frac{0,0344 \times 0,693}{\text{Objeto } L_A} = 0,0383 = PI_{A_1A_2L_A}$$

Cálculo do valor do fator de cobertura para a nova partição:

$$FCU_{A_1A_2L_A} = (PU_{A_1} / PU_{A_1A_2L_A}) \times C_{A_1} + (PU_{A_2L_A} / PU_{A_1A_2L_A}) \times C_{A_2L_A}$$

sendo, $PU_{A_1} = 0,02 \times 0,7 + 0,02 \times 0,0344 \times 0,7 = 0,0145$, e

$$PU_{A_2L_A} = 0,0344 \times 0,693 = 0,0238 = PI_{A_2L_A}$$

$$FCU_{A_1A_2L_A} = (0,0145/0,0383) \times 0,7 + (0,0238 / 0,0383) \times 0,693 = 0,6957 = C_{A_1A_2L_A}$$

Assim objetos o seguinte objeto final para a localidade A:



Figura D.2.3: Objeto final da localidade A

A probabilidade deste objeto final será a probabilidade incondicional para a localidade A no grafo de dependência do nível de gerenciamento 2.

Localidade B

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falhas incondicional (PI), condicional (PC) compatíveis com as probabilidades da tabela D.1.4 e, seja adicionado os fatores de cobertura (C), conforme a seguir:

$C_{B_1} = 0,7$	$C_{B_2} = 0,6$	$C_{L_1} = 0,8$
-----------------	-----------------	-----------------

Tabela D.2.2: Probabilidades de falha para a localidade B.

Adicionando as probabilidades ao grafo tem-se:

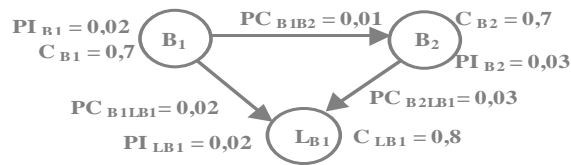


Figura D.2.3: Adicionando probabilidades de falha ao grafo do domínio B

Algoritmo *Divida e Conquiste*

Como a localidade B apresenta o mesmo modelo físico e lógico e as mesmas probabilidades condicionais e incondicionais, o objeto final da fase de união será idêntico ao obtido para a localidade A, ou seja:



Figura D.2.4: Objeto final da localidade B

Da mesma forma que para a localidade A, a probabilidade deste objeto final será a probabilidade incondicional para a localidade A no grafo de dependência do nível de gerenciamento 1.

Localidade C

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falhas incondicional (PI), condicional (PC) compatíveis com as probabilidades da tabela D.1.7 e, sejam adicionados os fatores de cobertura (C), conforme a seguir:

$C_{C1} = 0,7$	$C_{C2} = 0,7$	$C_{LC} = 0,8$
----------------	----------------	----------------

Tabela D.2.3: Probabilidades de falhas para o localidade C.

Adicionando as probabilidades ao grafo tem-se:

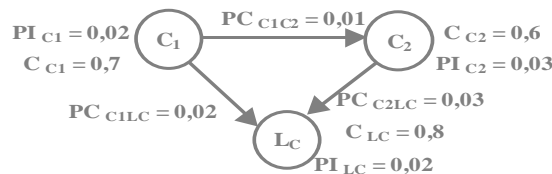


Figura D.2.5: Adicionando probabilidades de falha ao grafo do domínio C

Fase de União

Passo 1:

1) Unir os objetos com maior probabilidade condicional

Verifica-se que a maior dependência neste caso ocorre entre os objetos C_2 e L_C . Não é necessário verificar os demais critérios para este passo.

A probabilidade do novo objeto A_2L_A pode ser calculado como:

$$PU_{C_2L_C} = \frac{0,03 \times 0,7 + 0,03 \times 0,02 \times 0,7}{\text{Objeto } C_2} + \frac{0,02 \times 0,8}{\text{Objeto } L_C} = 0,0374 = PI_{C_2L_C}$$

Cálculo do valor do fator de cobertura para a nova partição:

$$FCU_{C_2L_C} = (PU_{C_2} / PU_{C_2L_C}) \times C_{C_2} + (PU_{L_C} / PU_{C_2L_C}) \times C_{L_C}$$

sendo, $PU_{C_2} = 0,7 \times 0,03 + 0,03 \times 0,02 \times 0,7 = 0,0214$, e

$$PU_{L_C} = 0,8 \times 0,02 = 0,016$$

$$FCU_{C_2L_C} = (0,0214/0,0374) \times 0,7 + (0,016/ 0,0374) \times 0,8 = 0,7428 = C_{C_2L_C}$$

Pela figura D.2.5 verifica-se que existem dois arcos de probabilidade condicional a partir do objeto C_1 para os objetos C_2 e L_C . Selecionando-se a probabilidade condicional de maior valor, ou seja, 0,02.

Passo 2:

Assim obtém-se o seguinte grafo de dependência:

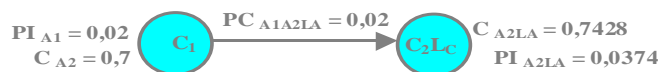


Figura D.2.6: União dos objetos C_1 e C_2L_C

Aplicando-se o critério para a fase de união:

1) Unir os objetos com maior probabilidade condicional

Como existe uma única dependência, esta é selecionado. Não é necessário verificar os demais critérios para este passo. A probabilidade do novo objeto $C_1C_2L_C$ pode ser calculado como:

$$PU_{C_1C_2L_C} = \frac{0,02 \times 0,7 + 0,02 \times 0,0374 \times 0,7}{\text{Objeto } C_1} + \frac{0,0374 \times 0,7428}{\text{Objeto } C_2L_C} = 0,0428 = PI_{C_1C_2L_C}$$

Cálculo do valor do fator de cobertura para a nova partição:

$$FCU_{C_1C_2LC} = (PU_{C_1} / PU_{C_1C_2LC}) \times C_{C_1} + (PU_{C_2LC} / PU_{C_1C_2LC}) \times C_{C_2LC}$$

sendo, $PU_{C_1} = 0,02 \times 0,7 + 0,02 \times 0,0344 \times 0,7 = 0,0150$, e

$$PU_{C_2LC} = 0,0374 \times 0,7428 = 0,0278$$

$$FCU_{C_1C_2LC} = (0,0150/0,0428) \times 0,7 + (0,0278 / 0,0428) \times 0,7428 = 0,7278$$

Assim objetos o seguinte objeto final para a localidade A:



Figura D.2.7: Objeto final da localidade C

Nível 1 de Gerenciamento:

Supor que os objetos de gerenciamento apresentem as seguintes probabilidades de falhas incondicional (PI), condicional (PC) compatíveis com as probabilidades da tabela D.1.10 e, seja adicionado os fatores de cobertura (C), conforme a seguir:

$C_{L1} = 0,9$	$C_{L2} = 0,8$	$C_{L3} = 0,9$
----------------	----------------	----------------

Tabela D.2.4 : Probabilidades de falhas no nível 1 de gerenciamento.

Adicionando as probabilidades ao grafo tem-se:

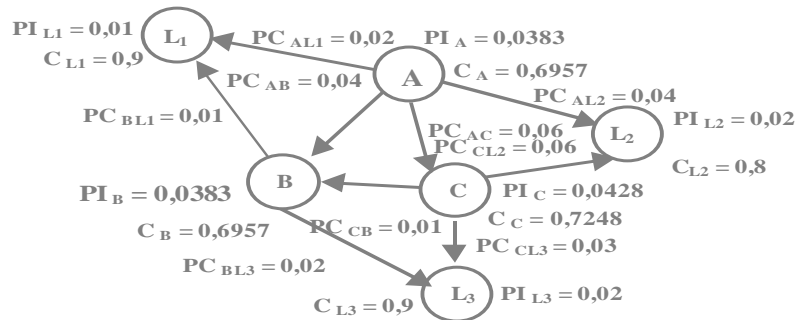


Figura D.2.8: Grafo de dependência adicionado as probabilidades de falhas

Supor igualmente uma falha no nó C_2 . Como consequência C_1 também falha, devido a dependência entre os objetos (veja figura D.2.8). Uma falha em C_1 provoca uma falha em A_2 (dependência entre domínios diferentes), e esta falha em A_2 provoque uma falha em A_1 (veja figura D.2.1).

Para a geração de alarmes, supor também que apenas as falhas em A_1 e C_2 foram cobertas, ou seja, foram apenas observadas falhas nestes objetos. Como consequência, o objeto de

gerenciamento A_1 emite uma mensagem de falha ao Sub-gerente A e o objeto de gerenciamento C_2 emite uma mensagem de falha ao Sub-gerente C. Os Sub-gerentes A e C emitem uma mensagem de falha ao Gerente de rede, a figura a seguir ilustra melhor este processo:

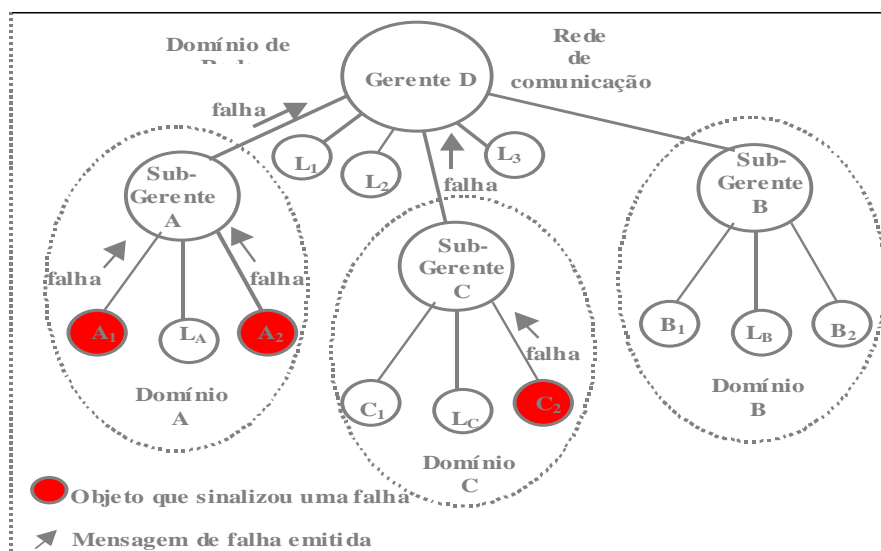


Figura D.2.9: Emissão de mensagens de falha pela rede

Pela figura D.2.9 observa-se que a arquitetura ou modelo lógico da rede hierárquica é em árvore.

Nível 1 de Gerenciamento

No nível 1 de gerenciamento localiza-se o gerente de rede, ele tem a função de determinar em qual domínio existe a maior probabilidade de possuir um objeto que é fonte primária de falha. Para tanto, ele possui o grafo de dependência apresentado anteriormente (figura D.2.8) e a sinalização dos seguintes alarmes determinado como a seguir:

- O alarme A_1 é sinalizado devido a cobertura do objeto A.
- Os alarmes E_1 e E_3 são sinalizados devido a cobertura da falha do objeto C.

Como consequência o gerente ou administrador de rede observará a sinalização dos alarmes E_1 e E_3 (alarmes ativos durante a ocorrência da falha), e a notificação de falha nas localidades A e C.

Fase de União:

Passo 1:

Dado o grafo fornecido inicialmente para o nível 0 de gerenciamento:

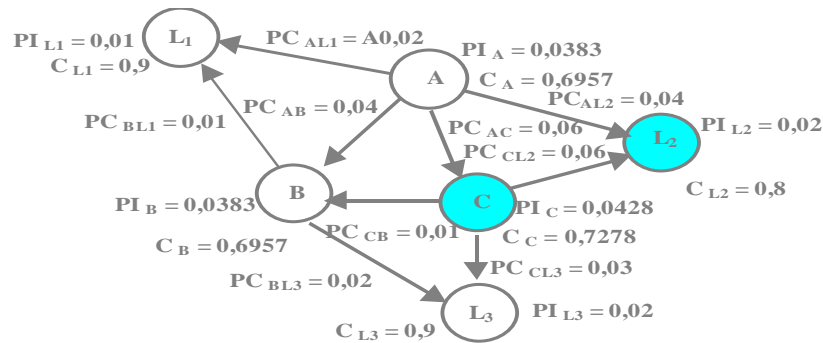


Figura D.2.10: União dos objetos C e L₂

Aplicando-se o critério para a fase de união:

1) Unir os objetos com maior probabilidade condicional

Igualmente ao exemplo anterior, existem duas probabilidades condicionais com o mesmo valor de dependência (entre os objetos A e L₂ e entre os objetos C e L₂) igual a 0,06.

2) Coberto pelo maior número de alarmes ativos:

Os objetos A e C são cobertos pelos seguintes alarmes: O objeto A é coberto pelo alarme E₁. O objeto C é coberto pelos alarmes ativos E₁ e E₃. Ou seja, a dependência entre os objetos A e C é coberta pelos alarmes ativos E₁ e E₃. Os objetos C e L₂ são cobertos pelos seguintes alarmes:

O objeto C é coberto pelo alarmes ativos E₁ e E₃. O objeto L₂ é coberto pelos alarmes ativos E₁ e E₃. Ou seja, a dependência entre os objetos C e L₂ é coberta pelos alarmes ativos E₁ e E₃. Verifica-se que ambos os critérios são cumpridos pelas duas dependências.

3) Unir os objetos com maior probabilidade incondicional

Os objetos C e L₂ apresentam as maiores probabilidades incondicionais.

4) Unir dois objetos arbitrariamente.

Não é necessário verificar este critério, pois o caso anterior apresenta uma dependência. A probabilidade do novo objeto CL₂ pode ser calculado como:

$$PU_{CL2} = \frac{0,7278 \times 0,0428 + 0,06 \times 0,02 \times 0,7278 + 0,8 \times 0,02}{\text{objeto C} \quad \text{objeto L}_2} = 0,0480 = PI_{CL2}$$

Cálculo do valor do fator de cobertura para a nova partição:

$$FCU_{CL2} = (PU_C / PU_{CL2}) \times C_C + (PU_{L2} / PU_{CL2}) \times C_{L2}$$

sendo, $PU_C = 0,7278 \times 0,0428 + 0,06 \times 0,02 \times 0,7278 = 0,0320$, e

$$PU_{L2} = 0,8 \times 0,02 = 0,016$$

$$FCU_{L2} = (0,0320 / 0,0480) \times 0,7278 + (0,016 / 0,0480) \times 0,8 = 0,7519 = C_{L2}$$

Pela figura D.2.10, observa-se que existem dois arcos de probabilidade condicional a partir do objeto A para o novo objeto C L₂ com valores de 0,06 (do objeto A para o objeto C) e 0,04 (do objeto A para o objeto L₂). Assim, deverá ser selecionado apenas a maior dependência, ou seja, o valor 0,06. Os demais passos são aplicados sucessivamente até obter-se o objeto final para a fase de união a seguir. Assim, obtém-se o seguinte objeto final, ou seja, o objeto D:



Figura D.2.11: Objeto final ou D obtido na fase de união

Ou seja, a rede apresenta uma probabilidade de falha incondicional de 0,0640 e com um fator de cobertura de 0,7813 (78,13 %) das falhas surgidas na rede.

Fase de particionamento

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.

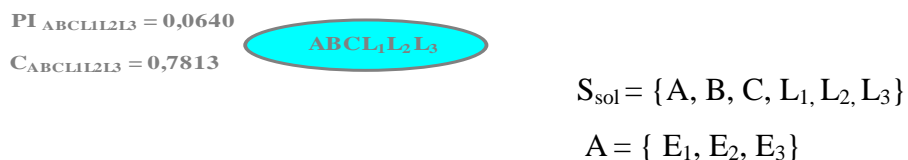


Figura D.2.12: Objeto obtido na fase de união

Passo 1:

Particionar o objeto em:

- Filho direito: L_1
- Filho esquerdo: $ABC L_2 L_3$

Assim, obtém-se o seguinte grafo:

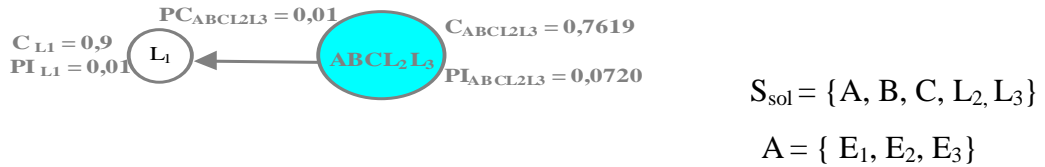


Figura D.2.13: Particionando os objetos em $ABC L_2 L_3$ e L_1

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

O filho direito é coberto pelos alarmes ativos E_1 e E_3 . O filho esquerdo é coberto pelos alarmes ativos E_1 e E_2 . Portanto, o filho direito é coberto por todos os alarmes e tem a maior probabilidade incondicional, sendo portanto o selecionado. Assim, os demais passos são aplicados sucessivamente até obter-se a solução final a seguir.

Solução final:

O conjunto solução final será $S_{sol} = \{C\}$.

Determinação da seqüência de teste:

A fase de testes deverá ser realizados nos objetos C. Neste caso, como existe um único objeto selecionado, não é necessário determinar a seqüência de teste. Como o objeto C é a fonte primária de falha (suposição feita para o exemplo), a falha será detectada no primeiro teste. A fase de testes pode ser realizada pelos sub-gerentes através de funções de gerenciamento automatizadas, no qual os sub-gerentes podem além de detectar a falhas e realizar testes,

também isolar o componente falho e redirecionar automaticamente o serviço executado pelo componente falho para outro componente de rede.

Nível 1 de Gerenciamento:

A partir da análise em função do algoritmo *Divida e Conquiste* no nível 2 de gerenciamento, o gerente deste nível de gerenciamento envia uma mensagem para os domínio selecionado pelo algoritmo *Divida e Conquiste* (domínio C) para realizar a mesma análise em suas sub-redes, ou seja, também realizarem o algoritmo *Divida e Conquiste* em seus grafos de dependência, pois estas sub-redes apresentam a maior probabilidade de serem uma falha primária. A figura a seguir ilustra tal processo:

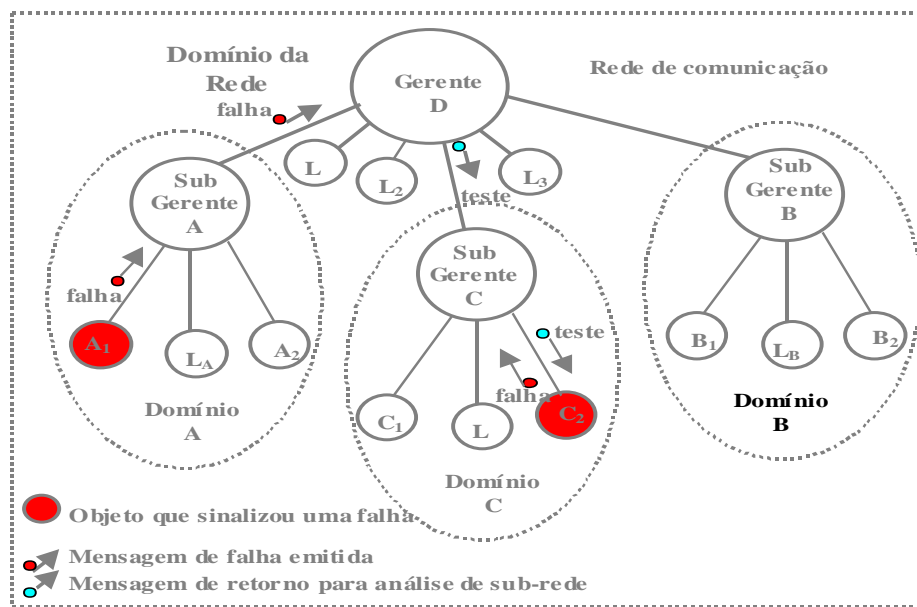


Figura D.2.14: Emissão de mensagens de falha e teste pela rede

Portanto, a sub-redes C devem ser analisada primeiro através do algoritmo *Divida e Conquiste* presente em seu sub-gerente, ou seja, o Sub-gerente C.

Localidade C

Similarmente a aplicação do algoritmo *Divida e Conquiste* no nível 1 de gerenciamento determinou que um objeto no domínio C poderia ser uma falha primária.

Fase de União:

A fase de união foi realizada na primeira parte do exemplo, portanto sendo utilizado para os próxima fase do algoritmo.

Fase de particionamento:

Dado o objeto final da fase de união apresentado a seguir, pode-se aplicar os dois critérios da fase de particionamento.



Figura D.2.15: Objeto final da fase de união

Passo 1:

Particionar o objeto em:

- Filho direito: C_1
- Filho esquerdo: $C_2 L_C$

Assim, obtém-se o seguinte grafo:

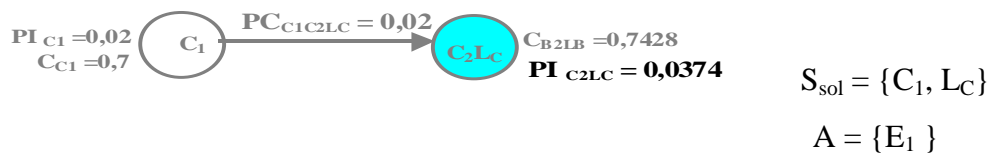


Figura D.2.16: Selecionando o filho C_2L_C

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

Ambos os filhos é coberto pelo alarme ativo E_1 . Portanto, o filho direito é o selecionado .

Passo 2:

Particionar o objeto em:

- Filho direito: C_2
- Filho esquerdo: L_C

Assim, obtém-se o seguinte grafo:

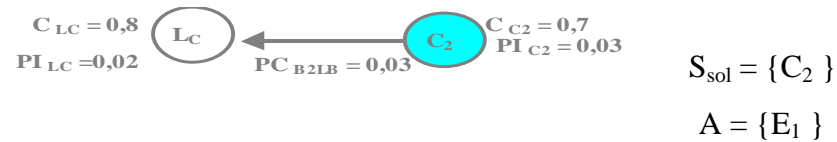


Figura D.2.17: Selecionando o filho C_2

Selecionar o filho que cumpre os seguintes critérios:

- 1) Apresenta a maior probabilidade incondicional

Neste caso, o filho direito apresenta a maior probabilidade incondicional.

- 2) Coberto por todos os alarmes ativos:

Ambos os filhos é coberto pelo alarme ativo E_1 . Portanto, o filho direito é o selecionado .

Assim, terá como solução final $S_{sol} = \{C_2\}$.

Portanto, o Sub-gerente C deverá submeter o objeto C_2 a fase de teste, como este é fonte primária de falha (suposição inicial), então, será encontrado o objeto falho no primeiro teste desta sub-rede.

ANEXO E

Simulações para o algoritmo *Divida e Conquiste proposto*

1. Todos os objetos com iguais probabilidades condicional, incondicional e fator de cobertura, sendo obtido a seguinte tabela:

1.1. Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E ₁	A ₂ ,C ₁
E ₂	L ₁
E ₃	L ₃
E ₄	B ₂
E ₅	C ₂
E ₆	C ₂
E ₇	L _A
E ₈	L _B
E ₉	L _C
E ₁₀	L ₁
E ₁₁	L ₂
E ₁₂	L ₃
E ₁ ,E ₂	A ₂ ,C ₁
E ₁ ,E ₂ ,E ₃	L ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄	L ₁ ,B ₁ ,A ₂ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	B ₁ ,B ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	B ₁ ,L _C ,B ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	L _A ,L _B ,L _C ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	B ₁ ,L _B ,L _A ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	L _B ,L _C ,B ₁ ,L _A ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _A ,L _C ,L _B ,L ₁ ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	L ₁ ,L _A ,L _C ,B ₁ ,L _B ,L ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L ₁ ,L _A ,L _B ,L _C ,A ₁ ,L ₃ ,L ₂ ,A ₂

1.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	C ₁ ,A ₂
E ₂	B ₁
E ₃	L ₃
E ₄	B ₂
E ₅	B ₁ ,L ₁
E ₆	L _C
E ₇	L _A
E ₈	L _B
E ₉	L _C
E ₁₀	L ₁
E ₁₁	L ₂
E ₁₂	L ₃
E ₁ ,E ₂	C ₁ ,A ₂
E ₁ ,E ₂ ,E ₃	B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄	B ₁ ,L _B ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	B ₁ ,L _B ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	L _B ,B ₁ ,C ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	L _B ,L _A ,B ₁ ,C ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	B ₁ ,L _B ,L _A ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	L _A ,B ₁ ,L _B ,L _C ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _A ,L _B ,L _C ,B ₁ ,L ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	B ₁ ,L _C ,L _A ,L ₁ ,L _B ,L ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L ₃ ,B ₁ ,L ₁ ,L _A ,L _C ,L _B ,L ₂ ,A ₂

2. Simulação com o objeto A_1 com probabilidade incondicional (p_i) maior que os outros objetos, sendo os outros objetos com probabilidades iguais aos da dissertação, sendo obtido a seguinte tabela:

2.1. Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E_1	A_2
E_2	B_1, L_B
E_3	L_3
E_4	L_B
E_5	C_2
E_6	L_C
E_7	A_1, L_A
E_8	L_B
E_9	L_C
E_{10}	L_1
E_{11}	L_2
E_{12}	L_3
E_1, E_2	B_1, A_2
E_1, E_2, E_3	L_1, A_2
E_1, E_2, E_3, E_4	B_1, L_B, L_1, A_2
E_1, E_2, E_3, E_4, E_5	B_1, A_2, B_2
$E_1, E_2, E_3, E_4, E_5, E_6$	L_C, B_1, L_B, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7$	L_A, B_1, L_C, A_2, B_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$	L_B, L_A, B_1, L_B, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9$	L_C, L_A, B_1, L_B, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}$	$L_A, L_B, L_1, B_1, L_C, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}$	$L_A, L_C, L_1, L_B, B_1, L_2, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}$	$L_1, L_B, L_C, B_1, L_A, L_3, L_2, A_2$

2.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	A ₁
E ₂	A ₁ ,A ₂
E ₃	L ₂ ,L ₃
E ₄	B ₁ ,B ₂
E ₅	B ₁ ,B ₂
E ₆	L ₂ ,C ₂
E ₇	A ₁ ,L _A
E ₈	A ₁ ,L ₃ ,L _B
E ₉	L ₃ ,L _C
E ₁₀	L ₁ ,C ₁
E ₁₁	C ₂ ,L ₂
E ₁₂	L ₃ ,L _B
E ₁ ,E ₂	A ₁ ,A ₂
E ₁ ,E ₂ ,E ₃	A ₁ ,L _A ,L ₁
E ₁ ,E ₂ ,E ₃ ,E ₄	L _A ,L ₁ ,L _B
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	L ₃ ,B ₁ ,A ₂ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	L _C ,B ₁ ,A ₂ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	L _A ,L _C ,B ₁ ,A ₂ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	A ₁ ,L _A ,L _B ,L ₃ ,B ₁ ,C ₂ ,L ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	A ₁ ,L _C ,L _A ,L _B ,B ₁ ,L ₃ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _B ,L ₁ ,L _A ,L _C ,B ₁ ,C ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	A ₁ ,L _C ,L ₁ ,L _A ,L _B ,B ₂ ,C ₁ ,A ₂ ,L ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L _A ,L _B ,B ₁ ,L ₁ ,L _C ,L ₃ ,L ₂ ,C ₂ ,A ₂

3. Simulação com o objeto C_1 com probabilidade incondicional (p_i) maior que os outros objetos, sendo os outros objetos com probabilidades iguais aos da dissertação, sendo obtido a seguinte tabela:

3.1. Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E_1	C_1, A_2
E_2	L_A
E_3	C_1, B_1
E_4	L_B
E_5	C_1
E_6	C_1, L_C
E_7	C_1, L_A
E_8	L_B
E_9	C_1, L_C
E_{10}	L_1
E_{11}	C_1, L_2
E_{12}	L_3
E_1, E_2	L_1, A_2
E_1, E_2, E_3	L_1, A_2
E_1, E_2, E_3, E_4	L_1, B_1, A_2, B_2
E_1, E_2, E_3, E_4, E_5	B_1, A_2, B_2
$E_1, E_2, E_3, E_4, E_5, E_6$	B_1, L_B, L_C, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7$	B_1, L_B, L_A, L_C, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$	L_A, L_B, L_C, A_1, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9$	L_B, B_1, L_C, L_A, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}$	$L_A, L_C, L_B, L_1, B_1, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}$	$L_1, L_A, L_C, B_1, L_B, L_2, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}$	$L_1, L_A, L_B, L_C, B_1, L_3, L_2, A_2$

3.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	C ₁ ,A ₂
E ₂	C ₁
E ₃	C ₁ ,B ₁
E ₄	C ₁ ,B ₂
E ₅	C ₁
E ₆	C ₁ ,C ₂
E ₇	B ₁ ,L _A
E ₈	C ₁ ,L _B
E ₉	C ₁ ,L _C
E ₁₀	C ₁ ,L ₁
E ₁₁	C ₁ ,L ₂
E ₁₂	C ₁ ,L ₃
E ₁ ,E ₂	C ₁ ,A ₂
E ₁ ,E ₂ ,E ₃	C ₁ ,L ₃ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄	C ₁ ,B ₁ ,B ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	B ₁ ,L _B ,L ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	C ₁ ,L _B ,L ₃ ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	B ₁ ,L _A ,C ₂ ,B ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	C ₁ ,L _A ,L _C ,L _B ,L ₃ ,A ₂ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	C ₁ ,L _C ,L _B ,L _A ,L ₁ ,C ₂ ,B ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	C ₁ ,L _B ,L _C ,B ₁ ,L ₁ ,L _A ,A ₂ ,B ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	C ₁ ,L _B ,L _A ,B ₁ ,L ₁ ,L _C ,B ₂ ,A ₂ ,L ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L ₁ ,L _A ,L _B ,L _C ,B ₁ ,L ₃ ,L ₂ ,A ₂

4. Simulação com o objeto L_B com probabilidade incondicional (p_i) maior que os outros objetos, sendo os outros objetos com probabilidades iguais aos da dissertação, sendo obtido a seguinte tabela:

4.1. Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E_1	L_B, C_1
E_2	L_2
E_3	L_B
E_4	L_B
E_5	L_2
E_6	L_C
E_7	L_A
E_8	L_B, B_2
E_9	L_C
E_{10}	L_1, B_1
E_{11}	C_1, L_2
E_{12}	L_3, C_1
E_1, E_2	L_A, C_1
E_1, E_2, E_3	L_B, B_1, L_C
E_1, E_2, E_3, E_4	L_B, B_2
E_1, E_2, E_3, E_4, E_5	L_B, B_1, L_C
$E_1, E_2, E_3, E_4, E_5, E_6$	L_B, B_1, L_C, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7$	L_B, L_A, B_1, A_2, C_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$	L_B, L_C, B_1, L_1, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9$	L_B, B_1, L_C, L_A, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}$	$L_B, L_A, L_C, L_1, B_1, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}$	$L_B, L_A, L_1, B_1, L_C, A_2, L_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}$	$L_B, B_1, L_1, L_3, L_C, L_A, A_2, L_2$

4.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	L _B ,A ₂
E ₂	L _B ,B ₁
E ₃	L _B
E ₄	L _B
E ₅	L _B ,B ₁
E ₆	L _B ,C ₂
E ₇	L ₁ ,L _A
E ₈	L _B
E ₉	L _B ,L _C
E ₁₀	L _B ,L ₁ ,C ₁
E ₁₁	L _B ,L ₂
E ₁₂	L _B ,L ₃
E ₁ ,E ₂	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	L _B ,L _C ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	L _B ,B ₁ ,L _A ,L _C ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	L _B ,L _A ,B ₁ ,L _C ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	L _B ,B ₁ ,L _A ,L _C ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _B ,B ₁ ,L _A ,L ₁ ,L _C ,C ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	L _B ,L _A ,B ₁ ,L ₁ ,L _C ,C ₂ ,A ₂ ,L ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L _B ,L _A ,L ₁ ,L ₃ ,B ₁ ,L _C ,A ₂ ,L ₂

5. Simulação para o algoritmo *Divida e Conquiste* e proposto com o objeto A_1 apresentando umas de suas probabilidades condicionais (p_{ij}) maior que os outros objetos (selecionando a maior probabilidade condicional entre os objetos A_1 e A_2), sendo os outros objetos com probabilidade condicional, incondicional e fator de cobertura iguais, sendo obtido a seguinte tabela:

5.1. Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E_1	A_2
E_2	A_2
E_3	A_2
E_4	L_B, A_2
E_5	L_B, B_1, A_2
E_6	L_C, A_2, L_2
E_7	A_2
E_8	L_B, A_2
E_9	L_C, A_2, L_2
E_{10}	A_2
E_{11}	L_C, A_2, L_2
E_{12}	L_C, A_2, L_2
E_1, E_2	A_2
E_1, E_2, E_3	A_2
E_1, E_2, E_3, E_4	L_B, B_1, A_2
E_1, E_2, E_3, E_4, E_5	L_B, B_1, A_2
$E_1, E_2, E_3, E_4, E_5, E_6$	$L_B, B_1, L_C, A_2, L_2, C_1$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7$	B_1, L_C, L_B, A_2, C_1
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$	$L_B, B_1, L_C, A_2, L_2, C_1$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9$	$L_B, B_1, L_C, A_2, L_2, C_1$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}$	B_1, L_C, L_B, A_2, B_1
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}$	B_1, L_C, L_B, A_2, C_1
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}$	B_1, L_C, L_B, A_2, C_1

5.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	A ₂
E ₂	L ₂
E ₃	A ₂
E ₄	L _B
E ₅	B ₁ ,B ₂
E ₆	C ₂
E ₇	A ₂
E ₈	L _B
E ₉	L _C
E ₁₀	L ₁
E ₁₁	A ₂ ,L ₂
E ₁₂	L ₃
E ₁ ,E ₂	A ₂
E ₁ ,E ₂ ,E ₃	A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄	B ₁ ,A ₂ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	B ₁ ,L _B ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	B ₁ ,L _B ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	B ₁ ,L _B ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	B ₁ ,L _B ,A ₂

6. Simulação para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto* com o objeto C_1 apresentando umas de suas probabilidades condicionais (p_{ij}) maior que os outros objetos (selecionando a maior probabilidade condicional entre os objetos C_1 e C_2), sendo os outros objetos com probabilidade condicional, incondicional e fator de cobertura iguais.

6.1. Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E_1	L_1, L_B, A_2
E_2	L_B, B_1
E_3	L_B
E_4	L_B
E_5	L_B, B_1
E_6	L_B, C_2, L_2
E_7	L_A, L_B
E_8	L_B
E_9	L_B, C_2, L_2
E_{10}	L_1, L_B
E_{11}	L_B, C_2, L_2
E_{12}	L_B, C_2, L_2
E_1, E_2	L_1, L_B, B_1, A_2
E_1, E_2, E_3	L_1, L_B, B_1, A_2
E_1, E_2, E_3, E_4	L_1, L_B, B_1, A_2
E_1, E_2, E_3, E_4, E_5	L_1, L_B, B_1, A_2
$E_1, E_2, E_3, E_4, E_5, E_6$	$L_B, B_1, L_1, C_2, C_1, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7$	$L_B, L_1, L_A, B_1, C_2, C_1, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$	$L_A, L_B, L_1, B_1, C_2, C_1, A_2, L_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9$	$L_A, L_B, L_1, B_1, C_2, B_1, A_2, L_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}$	$L_B, L_1, L_A, B_1, C_2, C_1, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}$	$L_B, L_1, L_A, B_1, C_2, C_1, A_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}$	$L_B, L_1, L_A, B_1, C_2, C_1, A_2$

6.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	C ₂ ,A ₂
E ₂	C ₂ ,C ₁
E ₃	A ₁ ,C ₂
E ₄	L _B
E ₅	C ₂
E ₆	C ₂
E ₇	L _A ,C ₂
E ₈	L ₁ ,L _B
E ₉	L _C ,C ₂
E ₁₀	L ₃ ,L ₁
E ₁₁	C ₂ ,L ₂
E ₁₂	L ₃ ,C ₂
E ₁ ,E ₂	C ₂ ,C ₁ ,A ₂
E ₁ ,E ₂ ,E ₃	L ₃ ,C ₂ ,A ₂ ,C ₁
E ₁ ,E ₂ ,E ₃ ,E ₄	L ₃ ,L _B ,C ₂ ,C ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	B ₁ ,C ₂ ,C ₁ ,B ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	L ₁ ,L _A ,C ₂ ,A ₂ ,B ₂ ,C ₁
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	L _A ,L _B ,B ₁ ,C ₂ ,A ₂ ,C ₁
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	L _C ,L ₁ ,L _B ,B ₁ ,C ₂ ,C ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _A ,L _C ,L ₁ ,L _B ,B ₁ ,C ₂ ,A ₂ ,C ₁ ,B ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	L _B ,L _A ,B ₁ ,L ₁ ,L _C ,C ₂ ,B ₂ ,C ₁ ,L ₂ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L _A ,L ₃ ,L _C ,L _B ,L ₁ ,C ₂ ,A ₂ ,L ₂ ,B ₂ ,C ₁

7. Simulação para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto* com o objeto L_B apresentando umas de suas probabilidades condicionais (p_{ij}) maior que os outros objetos (selecionando a maior probabilidade condicional entre os objetos B_1 e L_B), sendo os outros objetos com probabilidade condicional, incondicional e fator de cobertura iguais.

7.1 Algoritmo *Divida e Conquiste*

Alarmes ativos	Seqüência de objetos falhos
E_1	A_2, C_1
E_2	L_2
E_3	L_B
E_4	L_B
E_5	L_2
E_6	L_C
E_7	L_A
E_8	L_B
E_9	L_C
E_{10}	L_1
E_{11}	L_2
E_{12}	L_B, L_3
E_1, E_2	A_2, C_1
E_1, E_2, E_3	L_1, A_2
E_1, E_2, E_3, E_4	L_B, B_1, A_2
E_1, E_2, E_3, E_4, E_5	L_B, B_1, A_2
$E_1, E_2, E_3, E_4, E_5, E_6$	L_B, B_1, L_C, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7$	L_B, B_1, L_C, L_A, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$	L_B, B_1, L_C, L_A, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9$	L_B, B_1, L_C, L_A, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}$	L_B, B_1, L_C, L_1, A_2
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}$	$L_B, L_A, L_1, B_1, L_C, L_2$
$E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_{10}, E_{11}, E_{12}$	$L_B, B_1, L_1, L_3, L_C, L_A, A_2, L_2$

7.2. Algoritmo *Divida e Conquiste Proposto*

Alarmes ativos	Seqüência de objetos falhos
E ₁	L _B ,A ₂
E ₂	L _B ,B ₁
E ₃	L _B
E ₄	L _B
E ₅	L _B ,B ₁
E ₆	L _B ,L _C
E ₇	L _A
E ₈	L _B
E ₉	L _B ,L _C
E ₁₀	L _B ,L ₁
E ₁₁	L _B ,L ₂
E ₁₂	L _B ,L ₃
E ₁ ,E ₂	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅	L _B ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆	L _B ,B ₁ ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇	L _B ,B ₁ ,L _A ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈	L _B ,B ₁ ,L _A ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉	L _B ,L _A ,L _C ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀	L _B ,L _C ,L _A ,L ₁ ,B ₁ ,A ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁	L _B ,L _C ,B ₁ ,L _A ,L ₁ ,L ₂ ,A ₂ ,C ₂
E ₁ ,E ₂ ,E ₃ ,E ₄ ,E ₅ ,E ₆ ,E ₇ ,E ₈ ,E ₉ ,E ₁₀ ,E ₁₁ ,E ₁₂	L _B ,L ₁ ,L _C ,B ₁ ,L ₃ ,L _A ,C ₂ ,L ₂ ,A ₂

8. Simulação para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto* variando o fator de cobertura do objeto A_1 (variando o fator de cobertura de 0% a 100%), supondo os alarmes ativos de E_1 a E_6 , sendo os outros objetos com probabilidade condicional, incondicional e fator de cobertura iguais ao apresentado no capítulo 7 (Algoritmos de localização de falhas aplicados as arquiteturas de gerenciamento).

Fator de Cobertura	Seqüência de objetos falhos
0%	C_2, B_2, A_1, C_1
10%	C_2, B_2, A_2, C_1, A_1
20%	C_2, B_2, A_2, A_1, C_1
30%	A_1, C_2, B_2, A_2, C_1
40%	A_1, C_2, B_2, A_2, C_1
50%	A_1, C_2, B_2, A_2, C_1
60%	A_1, C_2, B_2, A_2, C_1
70%	A_1, C_2, B_2, A_2, C_1
80%	A_1, C_2, B_2, A_2, C_1
90%	A_1, C_2, B_2, A_2, C_1
100%	A_1, C_2, B_2, A_2, C_1

9. Simulação para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto* variando o fator de cobertura do objeto C_1 (variando o fator de cobertura de 0% a 100%), supondo os alarmes ativos E_1 a E_5 , sendo os outros objetos com probabilidade condicional, incondicional e fator de cobertura iguais ao apresentado no capítulo 7 (Algoritmos de localização de falhas aplicados as arquiteturas de gerenciamento).

Fator de Cobertura	Seqüência de objetos falhos
0%	B_2, A_2, L_2, C_1
10%	A_2, B_2, L_2, C_1
20%	A_2, B_2, L_C, C_1
30%	B_2, A_2, L_2, C_1
40%	B_2, A_2, L_2, C_1
50%	B_2, A_2, L_C, C_1
60%	B_2, A_2, C_1
70%	A_2, B_2, C_1
80%	A_2, B_2, C_1
90%	C_1, B_2, A_2
100%	C_1, B_2, A_2, C_1

10. Simulação para o algoritmo *Divida e Conquiste* e *Divida e Conquiste Proposto* variando o fator de cobertura do objeto L_B (variando o fator de cobertura de 0% a 100%), supondo os alarmes ativos de E_1 a E_8 , sendo os outros objetos com probabilidade condicional incondicional e fator de cobertura iguais ao apresentado no capítulo 7 (Algoritmos de localização de falhas aplicados as arquiteturas de gerenciamento).

Fator de Cobertura	Seqüência de objetos falhos
0%	C_2, B_2, A_2, L_c, C_1
10%	C_2, B_2, A_2, L_c, C_1
20%	C_2, B_2, A_2, L_c, C_1
30%	C_2, B_2, A_2, L_c, C_1
40%	C_2, B_2, A_2, L_c, C_1
50%	C_2, B_2, A_2, L_c, C_1
60%	$C_2, B_2, A_2, L_c, C_1, L_B$
70%	$C_2, B_2, A_2, L_c, C_1, L_B$
80%	$C_2, A_2, B_2, L_B, L_c, C_1$
90%	$C_2, A_2, B_2, L_B, L_c, C_1$
100%	C_2, L_B, A_2, L_c, C_1

Referências Bibliográficas

[AIDAROUS-96] AIDAROUS, S.; PLEVYAK, T. **Telecommunications Network Management**, IEEE Series on Network Management, 1996, p. 1-58.

[BOULOUTAS-91] BOLOUTAS, A. T.; HART, G. W.; SCHWARTZ M. **Two Extensions of the Viterbi Algorithm**, IEEE Transactions on Information Theory, vol. 37, n° 2, Mar. 1991.

[BOULOUTAS-95] BOLOUTAS, A. T.; KATZELA, I.; CALO, S. B. **Centralized vs Distributed Fault Localization**, ISINM '95, Santa Barbara, May 1995.

[BOULOUTAS-96] BOLOUTAS, A. T.; KATZELA, I. **Comparison of Distributed Fault Identification Schemes in Communication**, IBM Corp., T.J. Watson Research Center, 1996.

[CAMARGO-94] CAMARGO, J. B. Jr. **Apostila do Curso Confiabilidade de Sistemas – PCS 37**, Escola Politécnica, Universidade de São Paulo.

[CARVALHO-92] CARVALHO, T.C.M.B. **Gerenciamento de redes: Uma abordagem de Sistemas abertos**, Makron Books do Brasil , 1992, p. 1-7, 35-73.

[CARVALHO-98] CARVALHO, T.C.M.B. **Apostila do Curso Gerenciamento de Redes de Computadores– PCS 5742**, Escola Politécnica, Universidade de São Paulo.

[CRONK-98] CRONCK, R. N.; CALLAHAN, P. H.; BERNSTEIN, L. **Rule-Based Expert Systems for Network Mangement and Operations: An Introduction**, IEEE Network, 1998, p. 7-21.

[FAULT MANAGMENT-98] FAULT MANAGMENT, **DataPro Information Services**, GartnerGroup, 5 Aug. 1996. [31]

[GOLDSZMIDT-98] GOLDSZMIDT, G.; YEMINI, Y. **Delegated for Network Management**, IEEE Communications Magazine, 1998, p. 66-70. [32]

[HIGH AVAILABILITY: AN OVERVIEW-98] HIGH AVAILABILITY: AN OVERVIEW, **DataPro Information Services**, GartnerGroup, 21 May 1998.

[JAKOBSON-93] JAKOBSON, G.; WEISSMAN, M. D. **Alarm Correlation**, IEEE Network, 1993, p. 52-59.

[JALOTE-96] JALOTE, P. **Fault Tolerant Systems**, Department Science and Engineering Indian Institute of Tecnology, Kanpur, Prentice Hall, New Jersey, 1994, p. 115-139.

[KATZELA-93] KATZELA, I.; SCHWARTZ, M. **Schemes for Fault Identification in Communication Networks**, Center for Telecommunication Research, Columbia University, 1993.

[KEHL-93] KEHL, W.; HOPFMÜLLER H. **Model-Based Reasoning for Mangement of Telecommunication Networks**, IEEE Simposium Management Networks, 1993, p. 13-17.

[LEINWAND-93] LEINWAND, A.; FANG, K. **Netowork Management: A Practical Perspective**, Addison-Wesley, 1993, p. 1-17, 19-35.

[LEMOS-97] LEMOS, M. A. **Engenharia do conhecimento Aplicada ao Domínio de Gerenciamento de Falhas em Redes de Comunicação: Uma abordagem baseada em modelo**, São Paulo, 1997, Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo.

[MANSFIELD-93] MANSFIELD, G.; JAYANTHI, K.; HIGUCHI, K.; NEMOTO, Y.; NOGUCHI, S. **The MIKB Model for Intelligent Network Management**, Advance Intelligent Communication Systems Laboratories, IEEE Network, 1993, p. 1210-1214.

[MEIRA-96] MEIRA, D. M.; NOGUEIRA, J. M. S. **The SIS Project: A Distributed Platform for the Integration**, Departamento de ciência da computação da Universidade federal de Belo Horizonte, IEEE Network, 1996, p. 175-185. [30]

[MEIRA-97] MEIRA, D.M., **Um Modelo Para Correlação de Alarmes em Redes de Telecomunicações**, Tese (Doutorado) pelo Instituto de Ciências Exatas da UFMG, Nov. 1997.

[MEIRA e NOGUEIRA-97] MEIRA, D. M.; [NOGUEIRA] J. M. S., **Um Modelo Geral de Redes Telecomunicações para Aplicações de Gerenciamento de Falhas**, XV Simpósio Brasileiro de Redes de Telecomunicações, 8 Set. 1997.

[NETWORK MANAGMENT FUNCTIONS-98] NETWORK MANAGMENT FUNCTIONS, **DataPro Information Services**, GartnerGroup, 26 May 1998.

[NIGATE-95] NIGATE, Y. A. **Element Correlation using Rule and Object Based Techniques**, AT&T - Bell Laboratories, 1995.

[NOGUEIRA-97] NOGUEIRA, J. M. S. ; MEIRA D. M. **Métodos e Algoritmos para Correlação de Alarmes em Redes de Telecomunicações**, XV Simpósio Brasileiro de Redes de Computadores, 1997, páginas 79-97, 1997.

[PERFORMANCE MANAGEMENT-98] PERFORMANCE MANAGEMENT, **DataPro Information Services**, GartnerGroup, 6 May 1998.

[RAMAN-98] RAMAN L. **OSI Systems and Network Management**, IEEE Communications Magazine, p. 46- 53, 1998.

[RECOMMENDATION M.3010-92] RECOMMENDATION M.3010, **Principles for a Telecommunications Management Network, Maintenance: Telecommunications Management Network**, ITU (International Telecommunication Union), Oct. 92.

[SIEGL-94] SIEGL, M. R. **What is Network Management ?**, University of Tecnology, Vienna, 1994.

[SIEGL-95] SIEGL, M. R., TRAUSMUTH, G. **Hierarchical Network Management: A Concept and its Prototype in SNMPv2**, University of Tecnology, Vienna, Austria, , p. 121-129.

[SLOMAN-96] M. **Network and Distributed Systems Management**, Addison-Wesley, Imperial College of Science, Tecnology & Medicine University of London, 1996, p. 3-64, 131-164, 303-344.

[STALLINGS-93] STALLINGS, W. **SNMP, SNMPv2 and CMIP, The Practical Guide to Network Management Company**, Addison-Wesley, 1993, p. 5-61.

[SUTTER-96] SUTTER, M. T.; ZELDIN P. E. **Designing Expert Sustersms for Real-Time Diagnostics of Self-Correcting Networks**, IEEE Network, 1996, p. 43-51.

[TEWARY-98] TEWARY, K.; HARROLD, M. J. **An Approach to Fault Modeling and Fault Seeding using the Program Dependence Graph**, <http://www.cis.ohio-state.edu/~harrold/abstracts/fault.html>, 1998.

[VASSILA-95] VASSILA, A.; KNIGHT, G. J. **Introducing Active Managed Objects for Effective and Autonomous Distributed Management**, University College London, UK, 1995. [33]

[WEINER-95] WEINER, A. J.; THURMAN D. A.; MITCHELL, C. M. **Appying Case-based reasoning to aid fault management in supervisory control**, IEEE International conference in System, 1995.