

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PROPOSTA DE MODELO PARA CÁLCULO DE
DISPONIBILIDADE EM REDES BASEADO NA
DECOMPOSIÇÃO DE ESPAÇO DE ESTADOS**

MAXIMIRA CARLOTA CORRÊA

ORIENTADOR: HONÓRIO ASSIS FILHO CRISPIM

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

**PUBLICAÇÃO: PPGENE.DM – 058/08
BRASÍLIA – DF: AGOSTO - 2008**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PROPOSTA DE MODELO PARA CÁLCULO DE
DISPONIBILIDADE EM REDES BASEADO NA
DECOMPOSIÇÃO DE ESPAÇO DE ESTADOS**

MAXIMIRA CARLOTA CORRÊA

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
ENGENHARIA ELÉTRICA**

APROVADA POR:

**Prof. Honório Assis Filho Crispim, Dr. (ENE-UnB)
(Orientador)**

**Prof. Georges Amvame Nzé, Dr. (ENE-UnB)
(Examinador Interno)**

**Prof. Flavio Elias Gomes de Deus, Dr. (ENE-UnB)
(Examinador Interno)**

BRASÍLIA, 05 DE AGOSTO DE 2008

FICHA CATALOGRÁFICA

CORRÊA, MAXIMIRA CARLOTA

Proposta de modelo para cálculo de disponibilidade em redes baseado na decomposição de espaço de estados [Distrito Federal] 2008.

(ENE/FT/UnB, Mestre, Engenharia Elétrica, Comunicação(2008).Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Redes de Computadores

2. Disponibilidade de Redes

3. Algoritmo

4. Planejamento de Redes

I. ENC/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CORRÊA, M. C. (2008). Proposta de modelo para cálculo de disponibilidade em redes baseado na decomposição de espaço de estados. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM - 058/08, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 104p.

CESSÃO DE DIREITOS

AUTOR: Maximira Carlota Corrêa

TÍTULO: Proposta de modelo para cálculo de disponibilidade em redes baseado na decomposição de espaço de estados

GRAU: Mestre ANO: 2008

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Maximira Carlota Corrêa

QE: 32 – Conjunto F – Casa 48 – Guará II

71.0 65-061 - Brasília – DF – Brasil.

Maximira.carlota@uol.com.br

DEDICATÓRIA

Dedico esta dissertação a Djehuty, Deus Egípcio dos Escribas.

E a Aset, Senhora da Magia, Rainha de todos os mundos.

A meu companheiro Dylan Siegel.

E a meus filhos Cedric e Darla.

AGRADECIMENTOS

Agradeço ao meu orientador Honório Assis Filho Crispim e meu co-orientador Paulo Ubiratan Alves Ferreira, pelo imenso auxílio que me prestaram e pela imensa paciência que demonstraram comigo durante a duração deste trabalho.

Ao professor José Marcos Câmara Brito, do Inatel, pela ajuda em uma questão crítica.

Aos colegas de trabalho na Brasil Telecom, pelo apoio e ajuda e por assumirem minhas demandas durante as fases críticas deste trabalho. Principalmente a Elias Marques Cotrim, pelas relevantes observações.

A meus superiores, Renato Costa Pereira e Sérgio Medeiros de Sousa, pela compreensão e apoio.

A nossos diretores pela oportunidade proporcionada ao me escolherem para participar deste programa e por patrociná-lo.

Ao “Brother” Emerson de Jesus Duarte, coordenador de rede do extinto Centro Nacional de Redes e Serviços, por ser um exemplo para mim em todos os aspectos, como profissional, como chefe, como pessoa.

Aos membros da Tradição Caminhos das Sombras e em especial aos membros do Labirinto do Dragão, pela paciência que demonstraram durante o estresse do período e pelo apoio constante.

A minha mãe, por ficar com meus filhos enquanto eu assistia às aulas e depois, enquanto eu trabalhava nesta dissertação.

A meus filhos, Cedric e Darla, que ficaram sem a mãe vezes sem conta enquanto eu assistia às aulas e mesmo quando estava em casa, mas só via a tela do computador. Obrigada, queridos, por entender quando a mãe de vocês não podia ser interrompida.

A meus Deuses, por toda a ajuda que me deram e por sentir sempre sua presença ao meu lado.

E especialmente a meu companheiro, Dylan Siegel, pelo apoio, paciência e compreensão. Sem você eu não teria conseguido. Cada linha deste trabalho reflete seu amor e incentivo incondicionais. Obrigada, meu amor!

RESUMO

PROPOSTA DE MODELO PARA CÁLCULO DE DISPONIBILIDADE EM REDES BASEADO NA DECOMPOSIÇÃO DE ESPAÇO DE ESTADOS

Autora: Maximira Carlota Corrêa de Carvalho

Orientador: Honório Assis Filho Crispim

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Agosto de 2008

Nos últimos anos observou-se um aumento considerável no uso de redes de comunicação de dados. Quanto mais pessoas fazem uso das redes de computadores e quanto maior o número de serviços que estão sendo prestados nestas redes, mais é necessário garantir que a rede esteja disponível para o uso de seus clientes. Este trabalho visa propor um método para avaliação e cálculo da disponibilidade em redes IP, de forma que as prestadoras de serviço de comunicação de dados disponham de uma ferramenta segura de análise da variação da disponibilidade da rede quando a mesma sofre alterações e/ou falhas. Este trabalho foi dedicado às redes IP e seu foco será o cálculo da disponibilidade entre dois pontos A e B quaisquer. A revisão bibliográfica, além de contextualizar o assunto de redes de computadores, irá focar na teoria da decomposição de espaço de estados. Depois será descrito o algoritmo proposto e sua implementação em linguagem Perl. Então serão apresentados os testes realizados nos trechos selecionados das redes em estudo. Na conclusão deste trabalho foi validado o método proposto para o cálculo de disponibilidade em redes IP.

ABSTRACT

PROPOSAL OF A MODEL FOR AVAILABILITY CALCULUS IN IP NETWORKS BASED ON THE THEORY OF SPACE STATE DECOMPOSITION.

Author: Maximira Carlota Corrêa de Carvalho

Supervisor: Honório Assis Filho Crispim

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Agosto de 2008

In recent years there was a considerable growth in computer networks applications. The more people use computer networks and the greatest the number of services these networks provide the greatest the necessity to make sure the network is available for its clients to use. This work comes to provide the means to evaluate e calculate the availability of IP networks in order to offer the communication service provides a secure tool for the analysis of the network availability variation when the network is altered or fail. This work will be dedicated to the analysis of IP networks and its scope is the evaluation of the availability between two given points A and B. The bibliography review will contextualize the matter of computer networks, with focus in the Theory of State Space Decomposition. Then the algorithm will be described with its implementation in Perl programming language. Last, the tests performed in selected parts of the network will be presented. At the conclusion of this work the provided method for availability calculus in IP networks is valid was considered valid.

LISTA DE ABREVIATURAS E SIGLAS

ATM	<i>Asynchronous Transfer Mode</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
DNS	<i>Domain Name System</i>
GNU	<i>GNU is Not Unix</i>
HTTP	<i>Hipertext Transfer Protocol</i>
IP	<i>Internet Protocol</i>
LAN	<i>Local Area Network</i>
MAN	<i>Metropolitan Área network</i>
MTBF	<i>Mean Time Between Failures</i>
MTTF	<i>Mean Time do Failure</i>
MTTR	<i>Mean Time to Repair</i>
PDA	<i>Personal Digital Assistant</i>
PERL	<i>Practical Extration Report Language</i>
SAN	<i>Storage Area Network</i>
SLA	<i>Service Level Agreement</i>
SMDS	<i>Switched Multi-megabit Data Services</i>
SNMP	<i>Simple Network Management Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
VPN	<i>Virtual Private Network</i>
WAN	<i>Wide Area Network</i>
xDSL	<i>Digital Subscriber Line</i>

SUMÁRIO

1- O PROBLEMA	1
1.1 - INTRODUÇÃO	1
1.2 - DEFINIÇÃO DO PROBLEMA	1
1.3 - OBJETIVOS DO ESTUDO	2
2- REVISÃO DA LITERATURA	3
2.1 - PRELIMINARES	3
2.2 - REDES DE COMPUTADORES	4
2.2.1 - Definição	4
2.2.2 - Classificação das redes segundo área de cobertura e utilização	4
2.2.3 - Classificação das redes segundo a topologia	7
2.2.4 - Classificação das redes segundo o meio de transmissão	11
2.3 - HIERARQUIZAÇÃO DE REDES DE COMPUTADORES	12
2.4 - FALHAS EM REDES DE COMPUTADORES	13
2.5 - MODELAGEM DE REDES SOB A FORMA DE GRAFOS	14
2.6 - MODELAGEM MATEMÁTICA DE UMA REDE	17
2.7 - VETORES ASSOCIATIVOS	19
2.8 - PERL	20
2.9 - DISPONIBILIDADE EM REDES	21
2.10 - DECOMPOSIÇÃO DE ESPAÇO DE ESTADOS	28
3 - METODOLOGIA	45
3.1 - APRESENTAÇÃO	45
3.2 - UNIVERSO	45
3.3 - AMOSTRA	45
3.4 - HIPÓTESES	46
4 - IMPLEMENTAÇÃO	47
5 - TESTES	54
5.1 - DESCRIÇÃO DOS TESTES REALIZADOS	54
5.2 - TESTES E DESCRIÇÃO DOS RESULTADOS	55
5.2.1 - Rede 1	55
5.2.2 - Rede 2	59
5.2.3 - Rede 3	66
5.2.4 - Rede 4	71
5.2.5 - Rede 5	75
5.2.6 - Rede 6	79
5.2.7 - Análise de desempenho do algoritmo	82
6 - CONCLUSÕES E RECOMENDAÇÕES	85
6.1 - CONCLUSÕES GERAIS	85
6.2 - RECOMENDAÇÕES	86
7 - REFERÊNCIAS BIBLIOGRÁFICAS	87

LISTA DE TABELAS

Tabela 2.1 - Matriz de Adjacência	18
Tabela 2.2 - Matriz de Incidência.....	18
Tabela 2.3 - Matriz Exemplo.....	19
Tabela 2.4 - Disponibilidade e tempo de Serviço.....	22
Tabela 5.1 - Entrada de dados para as alterações na rede 1.....	57
Tabela 5.2 - Entrada dos dados para as redes 2A e 2B.....	61
Tabela 5.3 - Entrada dos dados das redes 2C e 2D.....	64
Tabela 5.4 - Entrada dos dados para as alterações da rede 4.....	73
Tabela 5.5 - Entrada dos dados para a rede 5 e variações.....	76
Tabela 5.6 - Resultados para a rede 6.....	81
Tabela 5.7 - Tempo de Execução do Algoritmo.....	82

LISTA DE FIGURAS

Figura 2.1: Redes em anel	8
Figura 2.2: Redes em estrela	9
Figura 2.3: Redes em barramento.....	10
Figura 2.4: Enlace dedicado	10
Figura 2.5: VPN.....	10
Figura 2.6: Hierarquização de redes IP	12
Figura 2.7: Comunicação entre as redes IP das prestadoras de serviço	13
Figura 2.8: Um grafo simples.....	15
Figura 2.9: Grafo com enlaces e arcos.....	16
Figura 2.10: Rede Simples.....	18
Figura 2.11: Parâmetros de recuperação de redes após uma falha.....	23
Figura 2.12: Rede exemplo para cálculo de disponibilidade.....	25
Figura 2.13: Rede em malha fechada.....	27
Figura 2.14: Rede em malha fechada	30
Figura 2.15: Árvore de Decomposição de Espaço de Estados	33
Figura 2.16: Rede em série	35
Figura 2.17: Rede em paralelo.....	36
Figura 2.18: Rede em série	37
Figura 2.19: Rede em paralelo.....	38
Figura 2.20: Redes diversas.....	38
Figura 4.1: Rede exemplo 1.....	49
Figura 5.1: Rede de teste 1	55
Figura 5.1.2: Rede de teste 1	57
Figura 5.1.3: Resultados para a Rede_1A	58
Figura 5.1.4: Resultados para a Rede_1B	58
Figura 5.2: Rede de teste 2	59
Figura 5.2.2: Rede de teste 2 – alterações A e B.....	61
Figura 5.2.5: Rede de teste 2 – alterações C e D.....	63
Figura 5.2.6: Resultado obtido para a rede_2C	64
Figura 5.2.7: Resultado obtido para a rede_2D.....	65
Figura 5.3: Rede de teste 3	66
Figura 5.3.1: Resultado do teste para a rede 3 entre os pontos S e T.....	67
Figura 5.3.2: Resultado do teste para a rede 3 entre os pontos B e J	68
Figura 5.3.3: Rede de teste 3 sem os roteadores e e f.....	69
Figura 5.3.4: Resultado do teste para a alteração da rede 3 entre os pontos S e T.....	70
Figura 5.3.5: Resultado do teste para a alteração da rede 3 entre os pontos B e J	70
Figura 5.4: Rede de teste 4	71
Figura 5.4.1: Resultado do teste para a rede 4 entre os pontos S e T.....	72
Figura 5.4.2: Rede de teste 4 com novos enlaces	73
Figura 5.4.3: Resultado do teste para a inclusão do novo enlace entre os pontos S e B	74
Figura 5.4.4: Resultado do teste para a inclusão do novo enlace entre os pontos S e E	75
Figura 5.5: Rede de teste 5 e variações	76
Figura 5.5.1: Resultado do teste para a rede 5.....	77
Figura 5.5.2: Resultado do teste para a rede 5 com a primeira alteração.....	78
Figura 5.5.3: Resultado do teste para a rede 5 com a segunda alteração.....	79
Figura 5.6: Rede de teste 6	80
Figura 5.7: Gráfico do tempo de execução do algoritmo	83
Figura 5.8: Gráfico de complexidade do algoritmo – notação “O Grande”	84

1 - O PROBLEMA

1.1 Introdução

Dados dois pontos A e B em uma rede IP (*Internet protocol*), não foi possível localizar uma ferramenta de conhecimento amplo que permita calcular de forma rápida e eficiente a disponibilidade da rede entre tais pontos.

Com um software dessa natureza, as equipes de planejamento e projeto se beneficiariam de um instrumento que permitisse avaliar como as alterações propostas para a rede afetariam a disponibilidade entre esses pontos.

Além disto, os profissionais de operação se beneficiariam de um mecanismo que permitisse avaliar o impacto de uma janela de manutenção em determinados seguimentos da rede na disponibilidade da mesma, de forma a garantir o cumprimento dos parâmetros de SLA entre o provedor de rede e seus clientes. Hoje, este impacto é apenas estimado.

1.2 Definição do Problema

A criação de um algoritmo para estimativa da disponibilidade entre dois pontos A e B de uma rede IP apresenta uma grande complexidade: a modelagem matemática da rede de forma que o algoritmo não seja dependente da rede em estudo e possa ser utilizado em qualquer malha IP.

Como uma solução para o problema, este trabalho propõe a aplicação da Teoria de Decomposição de Espaço de Estados, normalmente utilizada para definir a probabilidade de haver conexão entre dois nós A e B, para o cálculo da disponibilidade da rede entre os mesmos dois nós.

Após os argumentos apresentados, faz-se a seguinte pergunta, que constituirá no problema de pesquisa: **pode-se utilizar a decomposição de espaço de estados para cálculo de disponibilidade em redes IP?**

1.3 Objetivos do estudo

Objetivo Geral

- Verificar a adequação da modelagem baseada na decomposição de espaço de estados para cálculo de disponibilidade em redes IP.

Objetivos Específicos

- Apresentar a decomposição de espaço de estados como uma opção para o cálculo de disponibilidade em redes IP;
- Pesquisar as possíveis formas de implementar um algoritmo baseado na decomposição de espaço de estados;
- Propor um algoritmo e sua implementação para cálculo da disponibilidade em redes IP;
- Analisar os resultados obtidos com a utilização do algoritmo.;

2 - REVISÃO DA LITERATURA

2.1 - PRELIMINARES

Nesta revisão de literatura serão abordados os conceitos considerados básicos para a compreensão dos assuntos contidos no relatório escrito da pesquisa.

Assim, apresentam-se abordagens concisas sobre redes de computadores, o cenário atual da comunicação de dados, disponibilidade e confiabilidade em redes e a descrição da teoria da decomposição de espaço de estados.

2.2 - REDES DE COMPUTADORES

2.2.1 - Definição

Uma rede de computadores consiste em dois ou mais computadores e outros dispositivos ligados entre si e compartilhando recursos e dados. Antes do advento das redes de computadores, a troca de informações entre os diversos dispositivos era feita por operadores humanos, que carregavam as informações manualmente nos computadores.

Segundo Tanenbaum [TANENBAUM, 2003] uma rede pode ser definida por seu tamanho, topologia, meio físico e protocolo utilizado.

2.2.2 - Classificação das redes segundo área de cobertura e utilização

LAN (*Local Area Network*, ou Rede Local): são redes privadas contidas em um único edifício ou campus universitário [TANENBAUM, 2003]. Redes locais são basicamente um grupo de computadores interligados. Eles podem estar interligados por *switches* ou *hubs*. De modo geral uma rede local possui um ou mais roteadores que garante a comunicação com outras redes ou com a Internet. Hoje em dia temos também LANs baseadas em tecnologia *wireless*, onde um ou mais pontos de acesso (*access points*) garantem a interligação de grupos de computadores. O grande desafio atual da tecnologia sem fio (*wireless*) é garantir a segurança dos dados por ela trafegados, impedindo que uma pessoa não autorizada tenha acesso às informações de uma empresa, por exemplo.

MAN (*Metropolitan Area Network*): é uma rede que abrange uma área maior, de modo geral uma cidade ou parte da cidade. O exemplo mais conhecido de uma MAN é a rede de televisão a cabo [TANENBAUM, 2003]. Antigamente as MANs eram baseadas em protocolos de comunicação de longa distância, como SMDS (*Switched Multi-megabit Data Services*), ATM (*Asynchronous Transfer Mode*) e *Frame Relay*. Hoje se observa o crescimento do número de redes metropolitanas baseadas na família de protocolos Ethernet [TANENBAUM, 2003].

WAN (*Wide Area Network*, ou rede de longa distância): esta rede integra equipamentos em diversas localizações geográficas, envolvendo inclusive diversos países e continentes como a Internet. Uma WAN pode servir como meio de transporte para interconectar várias LANs e MANs. Em geral, as WANs contêm conjuntos de servidores, que formam sub-redes. Essas sub-redes têm a função de transportar os dados entre os computadores ou dispositivos de rede.

As WANs tornaram-se necessárias devido ao crescimento das empresas, onde as LANs e MANs não eram mais suficientes para atender a demanda de informações, pois era necessária uma forma de passar informação de uma empresa para outra de forma rápida e eficiente. Para atender a esta necessidade surgiram as WANs, que conectam redes dentro de uma vasta área geográfica, permitindo comunicação a grande distância.

A implementação de uma WAN cada vez mais demanda um bom planejamento por parte das empresas e administradores de redes. A forma de acesso a Internet, maior rede WAN existente, que mais vem crescendo recentemente é o acesso através de banda larga.

Segundo pesquisa realizada no ano de 2006 pela IDC Brasil o crescimento foi de 40,1%, tal percentual representa 1.6 milhão de novas conexões o que totaliza 5,7 milhões de usuários no território nacional. Em resumo, no período de seis anos (2001 a 2006), a banda larga cresceu 1.639% no Brasil [IDC].

A tecnologia mais utilizada no acesso banda larga é o xDSL que equivale a 78,2% das conexões banda larga existentes no país. O avanço de novas tecnologias no mercado ainda possibilitou ao consumidor brasileiro uma diminuição do valor de acesso a banda larga [IDC].

A concorrência, em especial entre as operadoras de TV a cabo e as de telefonia, pela preferência do consumidor resultou em uma queda de preço de aproximadamente 8%. Tal diminuição ainda possibilitou a alteração da velocidade já utilizada pelos assinantes. Preços menores foram os principais responsáveis pela opção dos consumidores por provedores que ofereciam maior velocidade. Os acessos superiores a 1 Mbps saltaram de 2% do mercado em dezembro de 2005 para 22% no mesmo período de 2006. As velocidades acima de 512 Kbps representaram 37% do mercado [IDC].

Recentemente, tem-se observado o aumento do número de um novo tipo de rede, as PANs (*Personal Area Networks* ou Rede de Área Pessoal) que são redes pequenas, normalmente formadas por um, dois ou três dispositivos, muito próximos uns dos outros, não mais do que uma dezena de metros. Em sua maioria elas utilizam o *Bluetooth* para a comunicação entre os dispositivos. Por exemplo, um computador portátil conectando-se a um outro e este a uma impressora.

Bluetooth é uma tecnologia de baixo custo para a comunicação sem fio entre dispositivos eletrônicos a pequenas distâncias. O nome *Bluetooth* é uma homenagem ao rei da Dinamarca e Noruega Harald Blåtand - em inglês Harold Bluetooth que é conhecido por unificar as tribos norueguesas, suecas e dinamarquesas. Da mesma forma, o protocolo procura unir diferentes tecnologias, como telefones móveis e computadores. É usado para comunicação entre pequenos dispositivos de uso pessoal, como PDAs, telefones celulares de nova geração, computadores portáteis, controles de video-games (*Play-Station 3*). Contudo, pode também ser utilizado para a comunicação de periféricos, como impressoras e scanners.

Dispositivos *Bluetooth* operam na faixa ISM (*Industrial, Scientific, Medical*) centrada em 2,45 GHz que era formalmente reservada para alguns grupos de usuários profissionais. Dispositivos *Bluetooth* comunicam-se entre si e formam uma rede denominada "piconet", na qual podem existir até oito dispositivos interligados, sendo um deles o mestre e os outros dispositivos escravos. Uma rede formada por diversos dispositivos mestres (com um número máximo de 10) pode ser obtida para maximizar o número de conexões. [BLUETOOTH]

Para um uso específico existem as redes SAN (*storage area network* ou área de armazenamento em rede). Uma rede SAN é uma rede projetada para agrupar dispositivos de armazenamento de dados de computador. As SANs são mais comuns nas grandes empresas em que se faz necessária a utilização de dispositivos de armazenamento de grande porte. Elas são utilizadas quando a empresa possui um grupo de servidores cujo volume de dados necessita de armazenamento externo, e não apenas nos discos internos dos servidores. São chamadas SANs as redes cujo propósito principal é a transferência de dados entre computadores e dispositivos de armazenamento e um sistema de armazenamento formado por dispositivos de armazenamento, computadores e/ou aplicações, e todo um controle via software, comunicando-se através de uma rede de computadores [TANENBAUM, 2003].

Uma SAN consiste em uma infra-estrutura de comunicação que provê conexões físicas com uma camada de gerenciamento, que organiza as conexões, os dispositivos de armazenamento e os computadores, tornando a transferência de dados robusta e segura. As SANs são diferenciadas de outras formas de armazenamento em rede pelo método de acesso em baixo nível que eles apresentam.

Em uma rede de armazenamento, o servidor envia pedidos por blocos específicos ou segmentos de dados de específicos discos. Esse método é conhecido como armazenamento de (*block storage*). O dispositivo age similarmente a um disco interno, acessando o bloco específico e enviando a resposta através da rede. [SNIA]

As SANs permitem que seja feito o backup dos dados dos servidores de forma segura, eficiente e eficaz.

2.2.3 - Classificação das redes segundo a topologia

A topologia de rede em anel consiste em estações conectadas através de um circuito fechado, em série, formando um anel. Redes em anel são capazes de transmitir e receber dados em qualquer direção. As configurações mais usuais, no entanto, são unidirecionais que possibilitam tanto o uso de equipamentos mais simples quanto os protocolos de comunicação que asseguram a entrega da mensagem corretamente e em seqüência ao destino, pois sendo unidirecionais evita-se o problema do roteamento. Nesta topologia cada estação está conectada a apenas duas outras estações, quando todas estão ativas [TANNENBAUM, 2003].

Uma desvantagem é que se, por acaso, apenas uma das máquinas falhar, toda a rede pode ser comprometida. Um exemplo de rede em anel é a tecnologia *Token Ring* da IBM, na qual existe a passagem de uma ficha (*token*) para as estações que quiserem transmitir seus dados [TANNENBAUM, 2003]. Uma estação deve executar a função de mestre, controlando a rede para evitar que a esta fique sem *token* ou tenha mais de um token ativo em um dado momento.

A Figura 2.1 apresenta tal topologia.

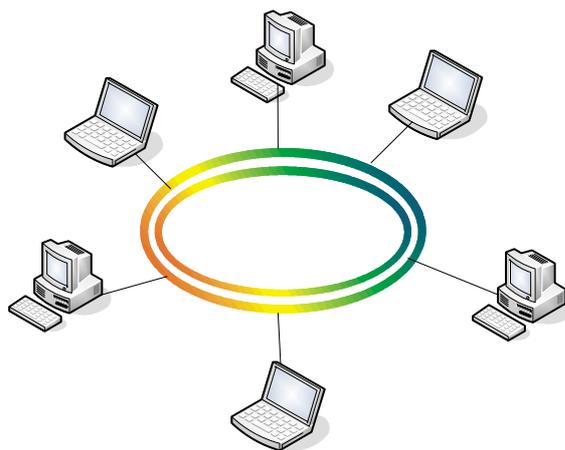


Figura 2.1: Redes em anel

Na topologia de rede em estrela, toda informação enviada de um nó para outro deverá obrigatoriamente passar pelo ponto central, ou concentrador, tornando o processo muito mais eficaz, já que os dados não irão passar por todas as estações. A estação central inteligente deve conectar cada estação da rede e distribuir o tráfego para que uma estação não receba, indevidamente, dados destinados às outras.

As redes em estrela, que são as mais comuns hoje em dia, utilizam cabos de par trançado e um *switch* como ponto central da rede. Existem também redes estrela com conexão passiva (similar ao barramento), na qual o elemento central nada mais é do que uma peça mecânica que atrela os “braços” entre si, não interferindo no sinal que flui por todos os nós, da mesma forma que o faria em redes com topologia barramento.

Contudo, este tipo de conexão passiva é mais comum em redes ponto-a-ponto lineares, sendo muito pouco utilizado já que os dispositivos concentradores (*HUBs*, Multiportas, Pontes e outros) não apresentam um custo tão elevado se levarmos em consideração as vantagens que são oferecidas. A Figura 2.2 apresenta a topologia em estrela.

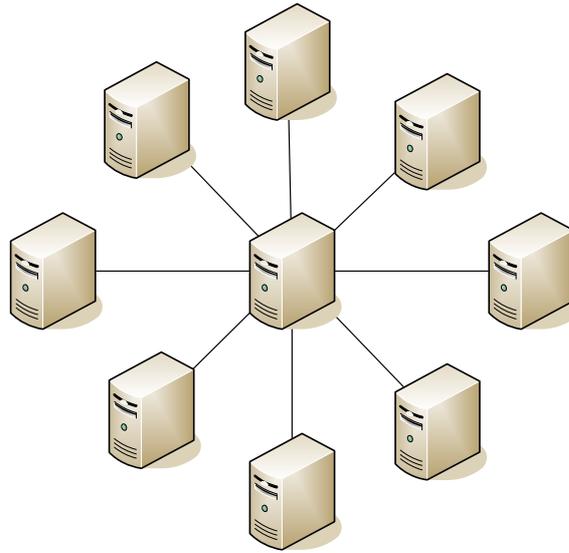


Figura 2.2: Redes em estrela

Uma rede em barramento (*bus*) é uma arquitetura de rede na qual um grupo de dispositivos é conectado através de um meio de comunicação partilhado, chamada barramento. Uma rede em barramento é a forma mais simples de se conectar múltiplos dispositivos, mas podem apresentar problemas quando mais de um dispositivo deseja transmitir simultaneamente.

Os sistemas que utilizam a topologia em barramento normalmente possuem algum mecanismo para detecção de colisões. Uma colisão ocorre quando mais de um dispositivo tenta transmitir ao mesmo tempo. Os mais conhecidos métodos de lidar com colisões são o CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), usado nas redes Ethernet e o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) utilizado nas redes *wireless* [TANNEMBAUM, 2003].

Em redes maiores é utilizada a topologia de árvore, onde temos vários hubs interligados entre si por *switches* ou roteadores. Em inglês é usado também o termo *Star Bus*, ou estrela em barramento, já que a topologia mistura características das topologias de estrela e barramento, Figura 2.3.

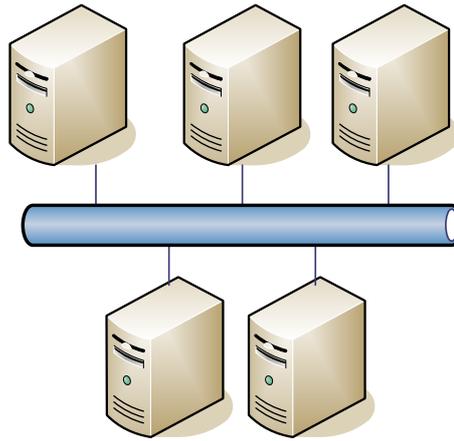


Figura 2.3: Redes em barramento

As redes ponto-a-ponto têm por função interligar apenas dois equipamentos. Isto pode ser feito através de uma linha de transmissão dedicada entre os dois pontos (Figura 2.4) ou através de um circuito virtual, interligando esses dois pontos de forma que, mesmo que dados passem por outras estações que estiverem conectadas, apenas a receptora poderá reconhecê-los. Esses caminhos virtuais podem ser estabelecidos sobre uma rede de transmissão de células como a ATM ou até mesmo através da Internet, com a implementação de uma VPN (*Virtual Private Network*), como mostrado na figura 2.5.



Figura 2.4: Enlace dedicado

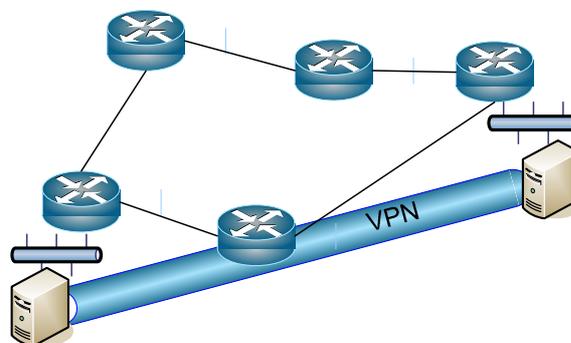


Figura 2.5: VPN

Quando se estabelece uma ligação ponto-a-ponto entre duas estações através de uma rede, isso pode ser feito por comutação de circuito ou por comutação de pacotes.

Na comutação de circuito estabelece-se uma ligação entre os dois pontos e à partir daí, tudo funciona como se houvesse uma linha dedicada entre os mesmos. Na comutação de pacotes as mensagens/pacotes são enviadas através da rede quando for possível. Desta forma, pode-se estabelecer um circuito virtual: a cada ligação é atribuído um percurso fixo, analogamente ao que se passa na rede telefônica. A idéia é evitar a escolha de um novo caminho para cada um dos pacotes. Quando uma ligação é estabelecida, o caminho da origem até o destino é memorizado no estabelecimento. Cada pacote contém o número de circuito virtual [TANNEMBAUM, 2003]

Uma outra forma é a comunicação por datagramas, onde não existem caminhos definidos. Os pacotes são enviados pelo melhor percurso no momento, independentes uns dos outros. Cada pacote contém a indicação de destino e da fonte e uma forma de identificação que permite ao destino montá-los na seqüência correta para recuperação da mensagem.

2.2.4 - Classificação das redes segundo o meio de transmissão

Rede por cabo é um tipo que se caracteriza pela adoção de cabos como meio de comunicação. As redes por cabo podem ser alimentadas tanto por cabos metálicos quanto por fibras ópticas. Atualmente, a fibra ótica garante maior imunidade a interferências eletromagnéticas, confiabilidade e maior velocidade de transmissão. Com a redução dos preços das fibras e dos componentes de conexão com as mesmas nos últimos anos, a fibra ótica sedimentou-se com o meio de transmissão preferido das redes atuais [OLIFER, 2008].

Rede sem fios (também chamada rede *wireless*) refere-se a um agrupamento de computadores (e outros dispositivos) em rede, interligados sem o uso de cabos, mas sim através de ondas de rádio ou outras formas de ondas eletromagnéticas.

2.3 - HIERARQUIZAÇÃO DE REDES DE COMPUTADORES

Para facilitar a operação, manutenção, os processos de planejamento das expansões, otimizações e a engenharia de tráfego, as prestadoras de serviços de telecomunicações organizam suas redes de forma hierárquica, conforme mostrado na Figura 2.6.

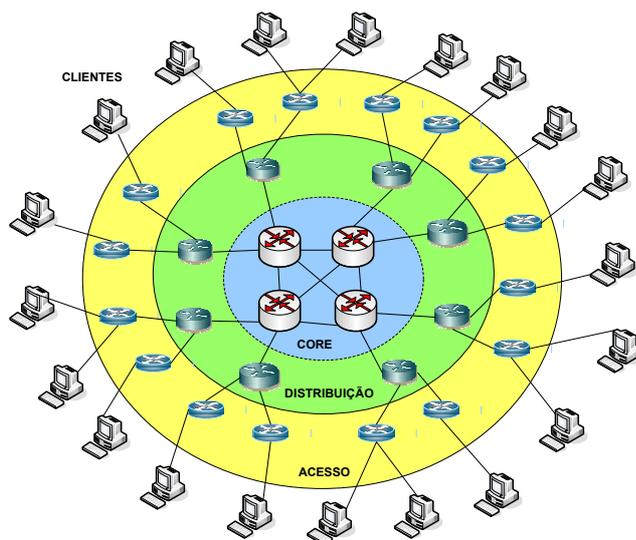


Figura 2.6: Hierarquização de redes IP

A comunicação entre as diversas prestadoras faz-se através de roteadores normalmente denominados Roteadores de Borda, que promovem a troca de tráfego entre os *backbones* das operadoras. Nestes roteadores normalmente rodam-se protocolos de roteamento especialmente designados para troca de tráfego entre AS (*Autonomous Systems*). Uma estrutura completa de comunicação entre prestadoras é mostrada na Figura 2.7.

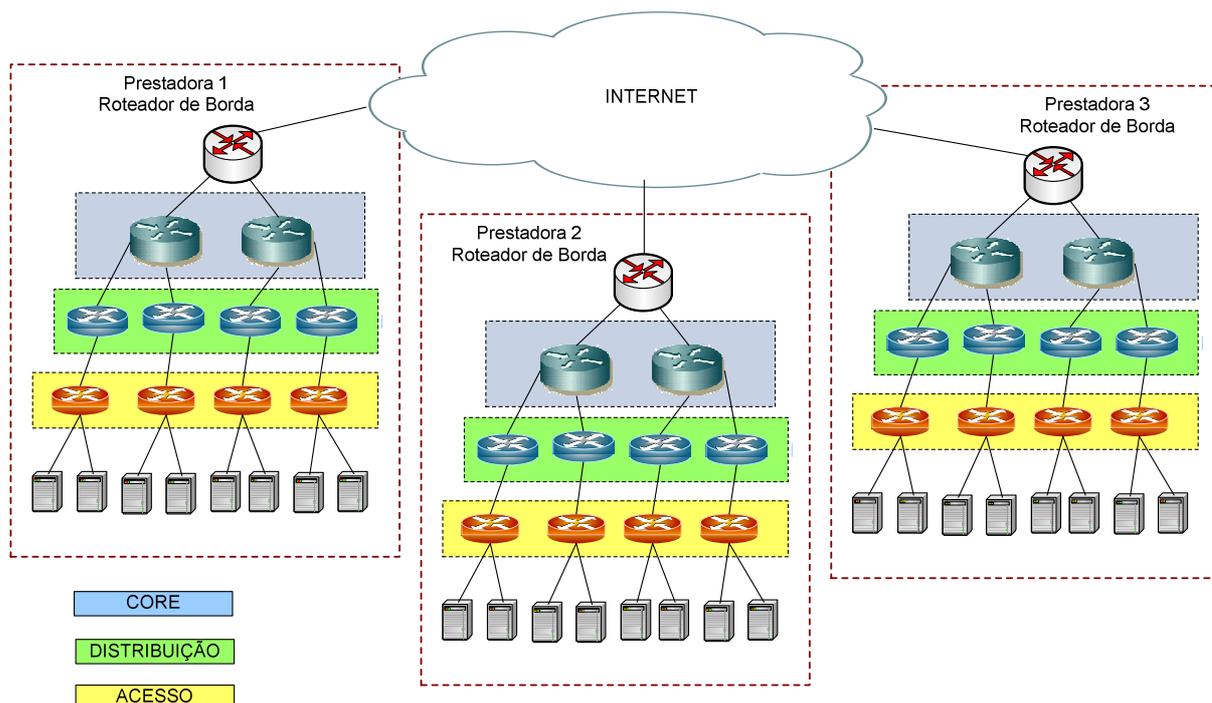


Figura 2.7: Comunicação entre as redes IP das prestadoras de serviço

2.4 - FALHAS EM REDES DE COMPUTADORES

Nos últimos anos pôde-se perceber um violento crescimento no tráfego de mensagens eletrônicas. Para acompanhar este crescimento, surge a demanda por computadores e redes mais confiáveis, com processamento mais rápido para garantir tempos de resposta cada vez menores.

Na medida em que a Internet e seus usos evoluem, é necessário que as redes evoluam no mesmo ritmo. Além disto, o número e a frequência das falhas de componentes crescem na mesma taxa [HUI, 2005]. Para garantir o acesso dos recursos e aplicativos é necessário projetar redes capazes de sustentar suas funções apesar das falhas em seus componentes [HUI, 2005].

Falhas em redes podem surgir de diferentes causas. Considerando exemplos de falhas de software e controle, podem ser citadas as falhas no protocolo de roteamento, por exemplo: não reconhecer uma rota alternativa causando perda de uma parte dos dados. Além disto, tem-se também a possibilidade de haver congestionamento de trechos da rede causando descarte de pacotes.

Além das falhas de software e controle, existem as falhas de topologia e componentes. Falhas topológicas podem ser divididas em aleatórias ou não aleatórias [HUI, 2005]. Sabotagens em redes não são considerados eventos aleatórios, enquanto que eventos climáticos, como tempestades, inundações são aleatórios.

Em síntese, catástrofes naturais podem afetar uma determinada região e causar dano a um grande número de componentes simultaneamente, enquanto ataques maliciosos normalmente possuem como alvo um determinado serviço ao invés de um determinado dispositivo.

Também deve-se considerar as falhas causadas por erro do operador, que correspondem a mais de 75% dos casos. E por fim temos as falhas dos componentes propriamente ditos, falhas de hardware e software que podem afetar um determinado dispositivo ou enlace [HUI, 2005].

2.5 - MODELAGEM DE REDES SOB A FORMA DE GRAFOS

Uma forma de modelar a rede para a realização de estudos matemáticos e computacionais é sob a forma de grafos.

Segundo Kershenbaum [KERSHENBAUM, 1993], um grafo G é definido por um grupo de vértices V e um grupo de arestas (*edges*) E . Os vértices são normalmente chamados nós e as arestas são também chamados enlaces e representam os caminhos, físicos e/ou virtuais que interligam os nós.

O grafo então pode ser definido como [KERSHENBAUM, 1993]:

$$G = (V, E) \quad (2.1)$$

A Figura 2.8 mostra um exemplo de um Grafo.

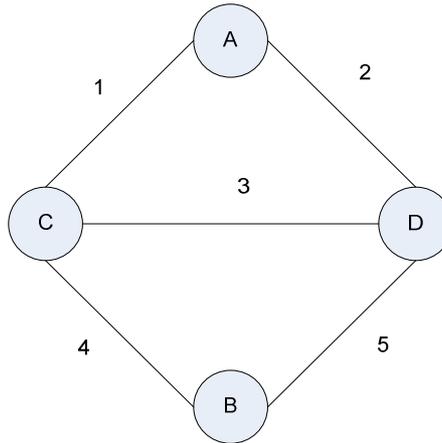


Figura 2.8: Um grafo simples.

O vetor V , com os vértices da rede, poderia em teoria ser infinito, mas em termos práticos ele é representado como abaixo:

$$V = \{v_i \mid i = 1, 2, 3, 4, \dots, N\} \quad (2.2)$$

Onde N é o número de nós da rede.

Da mesma forma o vetor das arestas E pode ser dado por:

$$E = \{e_j \mid j = 1, 2, 3, 4, \dots, M\} \quad (2.3)$$

Onde M é o número de arestas ou enlaces da rede.

Um enlace e_j corresponde a uma conexão entre dois nós. Supondo um enlace entre os vértices i e k Isso pode ser representado da seguinte forma:

$$e_j = (v_i, v_k) \quad (2.4)$$

Ou, de forma simplificada:

$$e_j = (i, k) \quad (2.5)$$

Um enlace que tem um nó como sua origem ou destino é chamado de um incidente neste nó. Nós são considerados adjacentes se existe um enlace entre eles.

Se o enlace for direcional, ou seja, se a direção do tráfego for importante, ele será chamado de arco e será representado por:

$$a_j = (v_i, v_k) \quad (2.6)$$

Onde a_j é o arco j de uma dada rede.

Ou, de forma simplificada:

$$a_j = (i, k) \quad (2.7)$$

A diferença entre enlaces e arcos pode vir a ser importante no tratamento de uma determinada rede. Graficamente, enlaces são linhas interligando os nós. Arcos são linhas com uma seta indicando a direção do tráfego.

A Figura 2.9 mostra um grafo com enlaces e arcos.

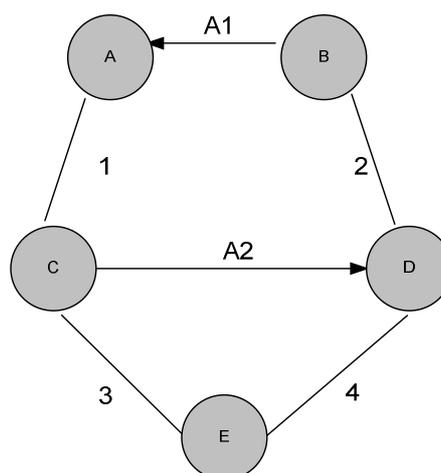


Figura 2.9: Grafo com enlaces e arcos.

Um grafo formado apenas por enlaces é chamado de não-orientado, um grafo formado por arcos é chamado de orientado.

Quando existe mais de um enlace ligando dois nós, estes enlances estão em paralelo e estes enlances são ditos múltiplos. Um enlace que ligue um nó a ele mesmo é chamado de *loop*.

Um grafo simples é aquele que não apresenta enlances múltiplos ou *loops*.

Neste trabalho serão utilizados grafos não-orientados, mas alguns deles possuem enlances múltiplos.

Um caminho em uma rede é uma seqüência de enlances que começa em um nó *S* e termina em um nó *T*, onde *S* e *T* são dois nós quaisquer da rede.

Um caminho que sai de um nó, passa por vários outros e retorna a este nó é chamado de ciclo. Por exemplo, na figura 2.8 os enlances 1,2 e 3 formam um ciclo.

Um grafo é chamado de conectado se existe pelo menos um caminho entre todos os pares de nós do grafo.

Uma árvore é um grafo sem ciclos. Uma árvore geradora (*Spanning tree*) é um grafo conectado sem ciclos.

2.6 - MODELAGEM MATEMÁTICA DE UMA REDE

Um grafo pode ser considerado como a representação cognitiva de uma rede, mas para que a rede possa ser analisada por algoritmos computacionais ela deve ser modelada matematicamente.

Uma das formas consagradas pela literatura [KERSHENBAUM, 1993] é a representação da rede através de uma matriz de adjacência e uma matriz de incidência. Para isso, cada enlace e cada nó é associado a um número ou letra.

Seja a rede mostrada na Figura 2.10:

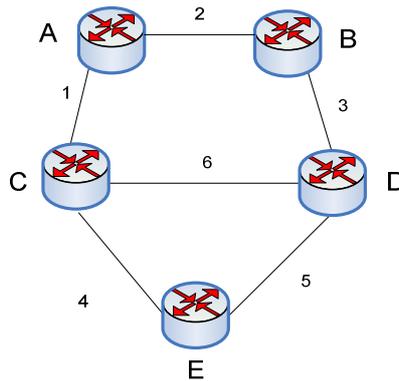


Figura 2.10: Rede Simples

A matriz de adjacência para esta rede é mostrada abaixo:

Tabela 2.1 - Matriz de Adjacência

$$\begin{bmatrix}
 & A & B & C & D & E \\
 A & 0 & 1 & 1 & 0 & 0 \\
 B & 1 & 0 & 0 & 1 & 0 \\
 C & 1 & 0 & 0 & 1 & 1 \\
 D & 0 & 1 & 1 & 0 & 1 \\
 E & 0 & 0 & 1 & 1 & 0
 \end{bmatrix}$$

A matriz de incidência para esta rede é mostrada abaixo:

Tabela 2.2 - Matriz de Incidência

$$\begin{bmatrix}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
 A & 1 & 1 & 0 & 0 & 0 & 0 \\
 B & 0 & 1 & 1 & 0 & 0 & 0 \\
 C & 1 & 0 & 0 & 1 & 0 & 1 \\
 D & 0 & 0 & 1 & 0 & 1 & 1 \\
 E & 0 & 0 & 0 & 1 & 1 & 0
 \end{bmatrix}$$

Tanto a matriz de adjacência quanto a matriz de incidência são matrizes esparsas. Uma matriz é chamada esparsa quando a maioria de suas entradas é constituída por zeros.

A matriz de adjacência possui exatamente dois “1s” para cada enlace da rede, logo ela será tão esparsa quanto a própria rede. A matriz de incidência, entretanto possui exatamente dois 1s em cada coluna e logo, será sempre esparsa [KERSENBAUM, 1993].

Neste trabalho optou-se pela modelagem da rede por vetores associativos de forma a otimizar o uso dos recursos de memória dos computadores que serão utilizados em sua aplicação. Os vetores associativos são uma funcionalidade da linguagem *Perl*.

2.7 - VETORES ASSOCIATIVOS

Suponha o seguinte trecho de código, escrito em *Perl*:

```
@cidade = ("Brasília", "Goiânia", "Cuiabá");  
Print $cidade[2];
```

O resultado obtido seria “Cuiabá” uma vez que os vetores normalmente começam a contar seus valores em zero (0).

Então na matriz M abaixo:

Tabela 2.3 - Matriz Exemplo

	1	2	3	4	5	6
A	1	1	0	0	0	0
B	0	1	1	0	0	0
C	1	0	0	1	0	1
D	0	0	1	0	1	1
E	0	0	0	1	1	0

O elemento M[3,2] teria valor 1.

Vetores associativos são listas de valores indexados por *strings*. Quando for necessário referir-se a uma elemento de um vetor associativo, deve-se fornecer uma *string*, que é chamada de chave. Seja o seguinte vetor:

```
%cidade = ("DF", "Brasília", "GO", "Goiânia", "MT", "Cuiabá");
```

```
Para cada $estado ( "DF", "GO", "MT") faça {  
    print "A capital de “, $estado, “ é “, $cidade{$estado};  
}
```

A vantagem do uso dos vetores associativos é a otimização do uso de memória para o armazenamento dos dados e a facilidade do acesso aos mesmos. Uma vez que os dados são

referenciados por *strings*, não é mais necessário guardar a posição numérica de um dado elemento, nem a realização de *loops* dentro de *loops* para referências matriciais.

Os vetores associativos são uma característica da linguagem *Perl* e foram a razão pela qual a *Perl* foi escolhida para o desenvolvimento do algoritmo deste trabalho.

2.8 - PERL

Perl significa *Practical Extration Report Language* e foi desenvolvida por Larry Wall para ser uma forma eficiente de extrair dados de relatórios. A *Perl* é um produto *GNU*, ou seja é gratuita e é uma linguagem interpretada.

Apesar de ter sido desenvolvida para a plataforma *Unix*, a *Perl* roda perfeitamente bem em outras plataformas.

Os *scripts* em *Perl* podem ser considerados a junção das funções do *awk*, *sed*, *Shell scripts* e a linguagem *C*. *Awk* é uma linguagem com muitas funcionalidades, excelente na manipulação de strings e arquivo texto, muito útil pra usar na linha de comando e em *Shell scripts* [ZAGO].

Devido à sua portabilidade, ela roda em múltiplas plataformas sem a necessidade de compilação, e devido às inúmeras funcionalidades e possibilidades que a linguagem oferece, *Perl* tem se tornado cada vez mais popular entre os programadores, particularmente aqueles acostumados a trabalhar com *Shell scripts* ou com *C*. Atualmente a *Perl* é vista por muitos como a linguagem ideal para o desenvolvimento de *scripts* para servidores Web [FOGHLÚ, 1997].

A *Perl* possui uma série de funções *built-in* para manipulação de dados e processos e é uma linguagem interpretada.

Um compilador recebe a linguagem de um programa e gera um arquivo executável. Cada código executável é criado para funcionar em uma plataforma específica. A vantagem de um executável é que ele pode ser copiado e instalado em outros computadores, mantendo o código original em segredo.

Um interpretador examina a listagem do programa e vai executando os comandos um a um neste mesmo momento. Não há necessidade de compilação, uma vez que um programa esteja escrito ele pode ser imediatamente executado.

A *Perl* é *especial* neste sentido porque ela executa ambas as funções. Ela compila um código de programa em um programa executável antes de executá-lo, mas não gera um executável separado, ela simplesmente o salva na memória e executa imediatamente.

Com isso ela combina a ciclo de desenvolvimento rápido de uma linguagem interpretada com a execução eficiente de um código compilado [PERL].

2.9 - DISPONIBILIDADE EM REDES

Disponibilidade é a habilidade de um serviço ou componente de infra-estrutura de TI (Tecnologia da Informação) de exercer sua função necessária em um dado momento ou por um determinado intervalo de tempo. Em termos de acordo de nível de serviço – SLA (*Service Level Agreement*), a disponibilidade pode ser definida como a proporção de horas acordadas para um serviço em que o cliente pode acessar o serviço [ITIL, 2008].

Existem duas formas de se calcular a disponibilidade de uma rede: o método percentual e o método de número de defeitos por milhão. O método percentual visa determinar ou acordar o número de minutos em que uma rede permaneceu ou deve permanecer em operação durante o período de um ano. Quando um contrato é assinado com SLA de 99,999% isso significa que a rede deverá estar em operação durante 99,999% do tempo.

Uma vez que o ano possui 365 dias, 24 horas por dia e 60 minutos por hora, calcula-se que o ano tem 525.600 minutos, isso sem contar os anos bissextos, que possuem um dia extra. A maneira utilizada para levar em consideração os anos bissextos, que só ocorrem a cada quatro anos é adicionar um quarto de um dia a cada ano. Isto resulta em 525.960 minutos por ano [OGGERINO, 2001].

Pode-se observar o tempo que um dispositivo pode permanecer indisponível de acordo com o percentual de disponibilidade definido ou acordado através da Tabela 2.4, com cálculos feitos pela autora.

Tabela 2.4 - Disponibilidade e tempo de Serviço

Percentual de disponibilidade	Tempo em operação (em minutos) por ano	Tempo em falha (em minutos) por ano	Tempo em falha por ano
99%	520700,4	5259,6	3,5 dias
99,9%	525434	525,96	8,5 horas
99,99%	525907,4	52,596	1 hora
99,999%	525954,7	5,2596	5 minutos
99,9999%	525959,5	0,52596	32 segundos

Além do número de minutos por ano em que um dispositivo funciona corretamente, um outro fator de interesse é a confiabilidade anual. Confiabilidade anual é o número de vezes por ano em que um dispositivo falha [OGGERINO, 2001].

A segunda forma de determinar a disponibilidade é o método dos defeitos por milhão (DPM). Neste método analisa-se o número de falhas ocorridas em um milhão de horas de funcionamento de um dado dispositivo ou de uma dada rede. É comum utilizar este método em grandes redes existentes. [OGGERINO, 2001]

O cálculo da disponibilidade de um componente individual é feito através da equação (2.8).

$$D = \frac{MTBF}{MTBF + MTTR} \quad (2.8)$$

O parâmetro MTBF (*Mean Time Between Failures*) descreve o número de horas entre as falhas de um determinado dispositivo. O parâmetro MTTF (*Mean Time do Failure*) descreve o número de horas entre colocar um dispositivo em serviço até que ele falhe novamente e o MTTR (*Mean Time to Repair*) – descreve o tempo necessário para a recuperação de um dispositivo em caso de falha. A relação entre estes parâmetros pode ser vista na Figura 2.11.

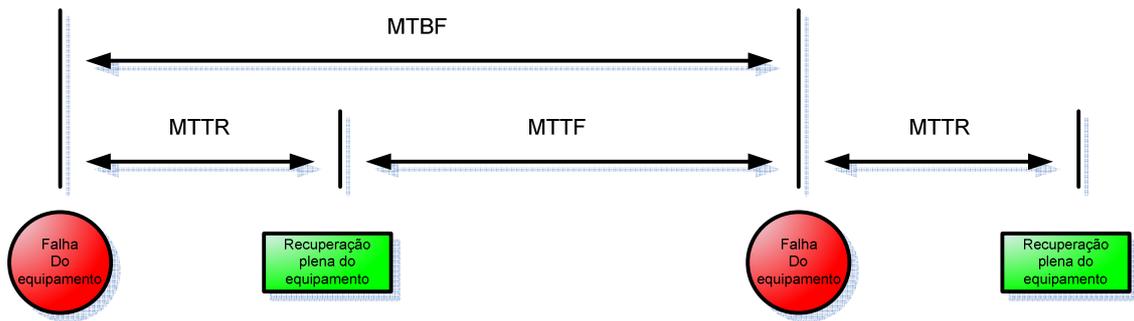


Figura 2.11: Parâmetros de recuperação de redes após uma falha.

O MTTR inclui as fases de detecção, diagnóstico e reparo de um dispositivo. Na maioria dos casos a fase de diagnóstico é a mais crítica, pois um diagnóstico errado pode levar ao aumento do número de horas necessárias para a correção de um dado problema.

De modo geral, os valores dos parâmetros MTBF e MTTR podem ser obtidos dos fabricantes dos dispositivos, mas é sempre correto utilizar os valores computados a partir dos dados de monitoração dos equipamentos.

Para calcular a disponibilidade de múltiplos componentes utiliza-se a decomposição destes componentes em caminhos série/paralelo. Considerando-se um caminho em série formado por n componentes, a disponibilidade do caminho será dada pela equação (2.9).

$$D = \prod_{i=1}^n D_i \quad (2.9)$$

Onde $D_i =$ Disponibilidade individual do componente i e n é o número total de componentes em série.

Em um sistema serial, se um componente falha, todo o sistema falha.

Em um sistema em paralelo, dois ou mais componentes são combinados de forma que o sistema funcionará se qualquer um dos componentes individuais estiver funcionando. No caso de uma rede, por exemplo, quando houver caminhos redundantes, estes caminhos estarão em paralelo. No caso de um sistema em paralelo a disponibilidade do sistema será dada pela equação (2.10).

$$D = 1 - \left[\prod_{i=1}^n (1 - D_i) \right] \quad (2.10)$$

Para dois dispositivos i e j em paralelo a disponibilidade do sistema será dada pela equação (2.11).

$$D = D_i + D_j - D_i \times D_j \quad (2.11)$$

A grande maioria dos sistemas e das redes é formada por combinações em série e paralelo. Em geral, pode-se estimar a disponibilidade de um sistema efetuando as combinações série e paralelo dentro deste sistema.

No caso de uma rede, para se estimar a disponibilidade é preciso definir o escopo do problema. Existem muitas possibilidades: é possível estimar a disponibilidade da rede como um todo; a disponibilidade de um determinado serviço; a disponibilidade fim a fim para dois nós quaisquer; a disponibilidade de um determinado trecho ou enlace; a disponibilidade de um determinado dispositivo ou grupos de dispositivos; de um caminho ou grupos de caminhos na rede e muitas outras combinações.

Para exemplificar o cenário de cálculo da disponibilidade para um determinado serviço ou sistema na rede, considera-se que é necessário calcular a disponibilidade para o serviço DNS (*Domain Name System*) da rede mostrada na Figura 2.12.

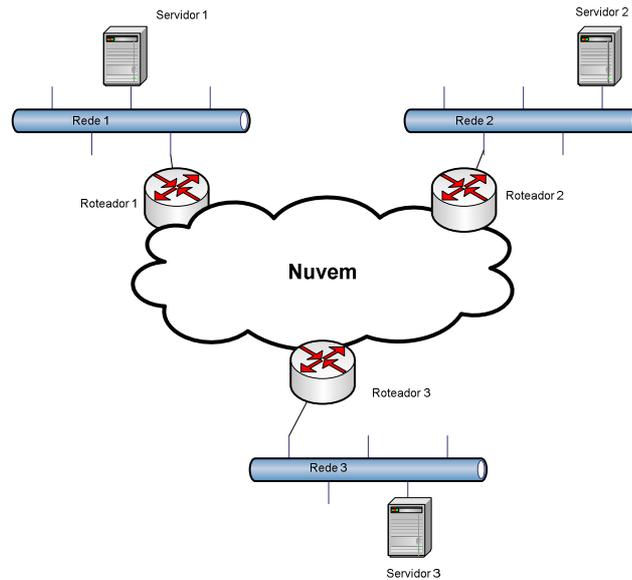


Figura 2.12: Rede exemplo para cálculo de disponibilidade

Os servidores 1, 2 e 3 são servidores de DNS. Todos os dispositivos da rede que necessitam do serviço DNS possuem os três servidores cadastrados para garantia de redundância.

Segundo a linha de raciocínio da divisão de um sistema em subsistemas série e paralelo, pode-se verificar que este serviço possui três sistemas. O sistema formado pelo roteador 1, rede 1 e servidor 1; o sistema formado pelo roteador 2, rede 2 e servidor 2 e o sistema formado pelo roteador 3, rede 3 e servidor 3.

No caso de cada um desses sistemas, da forma como foram considerados, uma falha em qualquer um dos componentes torna o sistema indisponível. Desta forma, os componentes destes sistemas estão em série. Entretanto, pode-se observar que se um dos sistemas falhar, o serviço DNS não está indisponível, então os sistemas 1, 2 e 3 podem ser considerados em paralelo.

Para ilustrar isso foram atribuídos valores arbitrários de disponibilidade para os componentes dos três sistemas.

Dados:

$$D_{roteador\ 1} = 99,9\%; D_{rede\ 1} = 99,93\%; D_{servidor\ 1} = 99,3\%$$

$$D_{roteador\ 2} = 99,98\%; D_{rede\ 2} = 99,3\%; D_{servidor\ 2} = 99,93\%$$

$$D_{roteador\ 3} = 99,9\%; D_{rede\ 3} = 99,95\%; D_{servidor\ 3} = 99,2\%$$

Sistema 1:

$$D_{sistema\ 1} = D_{roteador\ 1} \times D_{rede\ 1} \times D_{servidor\ 1} = 0,999 \times 0,9993 \times 0,993 = 0,9913 = 99,13\%$$

Sistema 2:

$$D_{sistema\ 2} = D_{roteador\ 2} \times D_{rede\ 2} \times D_{servidor\ 2} = 0,9998 \times 0,993 \times 0,9993 = 0,9921 = 99,21\%$$

Sistema 3:

$$D_{sistema\ 3} = D_{roteador\ 3} \times D_{rede\ 3} \times D_{servidor\ 3} = 0,999 \times 0,9995 \times 0,992 = 0,9905 = 99,05\%$$

Disponibilidade do serviço:

$$D_{DNS} = 1 - \left[(1 - D_{sistema\ 1}) \times (1 - D_{sistema\ 2}) \times (1 - D_{sistema\ 3}) \right]$$

$$D_{DNS} = 1 - \left[(1 - 0,9913) \times (1 - 0,9921) \times (1 - 0,9905) \right]$$

$$D_{DNS} = 1 - [0,0087 \times 0,0079 \times 0,0095] = 0,99999 = 99,999\%$$

Para o cálculo da disponibilidade fim a fim: tomam-se dois nós *A* e *B* da rede e calcula-se a disponibilidade da rede traçando um caminho entre esses dois nós. Isto é útil, por exemplo, quando um cliente contrata vários acessos em várias cidades diferentes e deseja saber a disponibilidade da rede para a comunicação entre sua matriz e uma de suas filiais.

Supondo a rede mostrada na Figura 2.13, cujos valores de disponibilidade dos enlaces e roteadores estão indicados.

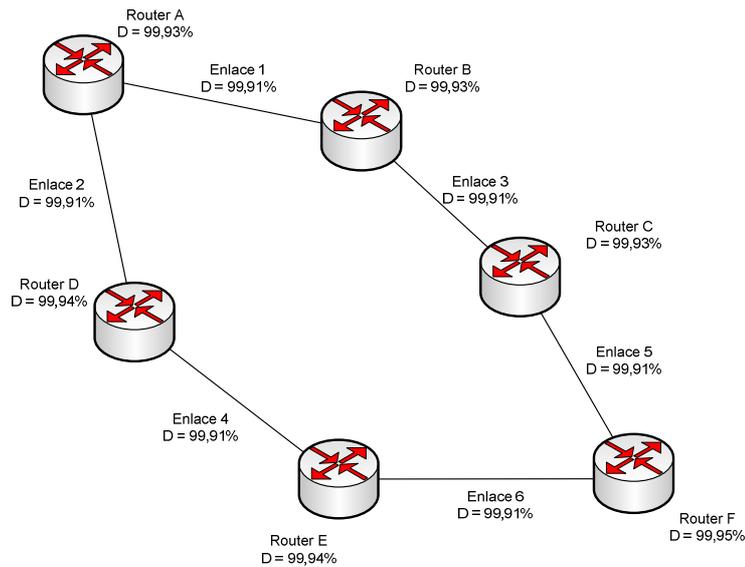


Figura 2.13: Rede em malha fechada.

Considerando que é necessário avaliar a disponibilidade da rede entre os roteadores *A* e *F*. O cálculo será feito como se segue:

1 – Redução do caminho serial formado pelos enlaces 1,3 e 5 e pelos roteadores B e C a um único componente cuja disponibilidade chamada de D_x .

$$D_x = D_1 \times D_3 \times D_5 \times D_B \times D_C = 0,9991 \times 0,9991 \times 0,9991 \times 0,9993 \times 0,9993 = 0,9959 = 99,59\%$$

2 – Redução do caminho serial formado pelos enlaces 2,4 e 6 e pelos roteadores D e E a um único componente cuja disponibilidade chamada de D_y .

$$D_y = D_2 \times D_4 \times D_6 \times D_D \times D_E = 0,9991 \times 0,9991 \times 0,9991 \times 0,9994 \times 0,9994 = 0,9961 = 99,61\%$$

3 – Redução do caminho serial formado pelos componentes virtuais cujas disponibilidades são D_x e D_y e que estão ligados em paralelo.

$$D_{xy} = D_x + D_y - D_x \times D_y = 0,9959 + 0,9961 - (0,9959 \times 0,9961) = 0,99998 = 99,998\%$$

3 – Cálculo da disponibilidade do caminho entre os roteadores A e F, incluindo ambos no cálculo.

$$D_{AF} = D_A \times D_{xy} \times D_F = 0,9993 \times 0,99998 \times 0,9995 = 0,99878 = 99,878\%$$

Este trabalho está baseado no cálculo da disponibilidade fim a fim entre dois nós predeterminados da rede.

Quando uma rede pode ser decomposta em subsistemas série e paralelo, o cálculo da disponibilidade entre dois nós A e B quaisquer é relativamente simples, como foi visto no exemplo acima. Contudo, quando a rede em questão possui topologia em malha, tal decomposição torna-se viável apenas por aproximações. Este trabalho visa propor um método para o cálculo da disponibilidade fim a fim de uma rede, de uma forma que possa considerar tanto uma rede que pode ser decomposta em sistemas série e paralelo quanto uma rede em malha.

2.10 - DECOMPOSIÇÃO DE ESPAÇO DE ESTADOS

Como foi dito na introdução, a criação de um algoritmo para estimativa da disponibilidade entre dois pontos A e B de uma rede IP mostra-se complexo devido à necessidade de se modelar matematicamente a rede de forma que o algoritmo não seja dependente da rede em estudo e possa ser utilizado em qualquer rede IP em malha.

Como uma solução para o problema, este trabalho propõe a aplicação da Teoria de Decomposição de Espaço de Estados, normalmente utilizada para definir a probabilidade de haver conexão entre dois nós A e B, para o cálculo da disponibilidade da rede entre os mesmos dois nós.

Segundo Ball [BALL, 1992], a confiabilidade de uma rede é dada pela probabilidade de haver conexão entre esses dois pontos quaisquer desta rede.

Uma aproximação para encontrar a confiabilidade de uma rede em malha fechada é decompor o espaço de probabilidades, considerando que cada enlace e cada nó da rede esteja operando ou esteja em falha. Não há como um componente estar em falha e operando ao mesmo tempo, logo estes eventos são mutuamente excludentes.

Então, segundo Kershenbaum [KERSHENBAUM, 1993], se N é uma rede com C componentes, a probabilidade da rede estar conectada P_W é dada pela equação (2.12):

$$P_W = \sum_{k=0}^N N_w(k) p^k (1-p)^{C-k} \quad (2.12)$$

Onde p^k é a probabilidade do componente k estar funcionando, $N_w(k)$ é o número de estados em que a rede está funcionando com k componentes funcionando e $(C-k)$ componentes em falha.

E para cada componente k da rede N , a probabilidade da rede estar operando $P_W(N)$ é dada pela equação (2.13):

$$P_W(N) = P(k) P_w(N|k) + (1 - P(k)) P_w(N - k) \quad (2.13)$$

Onde $P(k)$ é a probabilidade do componente k estar funcionando; $(1 - p(k))$ é a probabilidade do componente k estar em falha; $(N|k)$ é a rede N com k funcionando e $(N-k)$ é a rede N com k em falha.

Uma vez que os eventos k operando ou k em falha são mutuamente excludentes, um deles é sempre verdade. Então a probabilidade de que a rede N esteja operando pode ser decomposta em situações em que k está operando e situações em que k está em falha.

A equação (2.13) pode ser usada recursivamente para todos os componentes da rede. Supondo que o objetivo seja encontrar a probabilidade de que haja conectividade entre dois componentes S e T da rede. Esta rede estará funcionando se houver um caminho de componentes funcionando entre S e T e estará em falha se houver um corte entre eles. Então, uma vez que seja determinado que existe um caminho ou um corte na rede, a recursividade pode ser interrompida [KERSHENBAUM, 1993].

Supondo a rede mostrada na Figura 2.14:

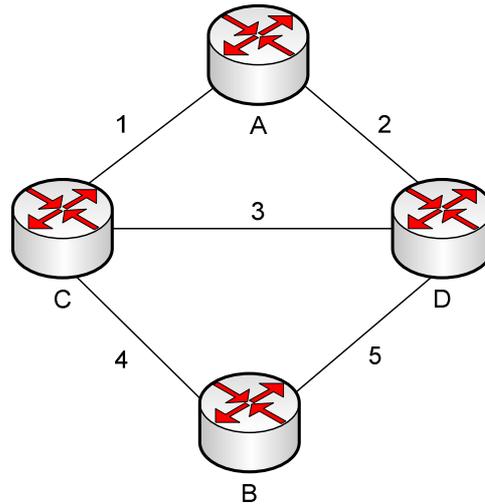


Figura 2.14: Rede em malha fechada

Deve-se calcular a probabilidade de comunicação entre A e B. Para simplificar, supõe-se que os nós não falham e a probabilidade de falha nos enlaces é dada por P_i onde i é o número do enlace. A probabilidade de falha no enlace i será dada por $(1 - P_i)$.

Partindo-se do nó A consideramos que todos os enlaces estão em estado aberto, ou seja, podem estar operando ou em falha.

Será utilizada a seguinte nomenclatura:

Evento iK = enlace i operando

Evento iF = enlace i em Falha

A partir do nó A podemos considerar que:

Evento $1K$ = enlace 1 operando = probabilidade P_1

Evento $1F$ = enlace 1 em falha = probabilidade $(1 - P_1)$

Nenhum dos dois eventos garante a conectividade A->B, mas nenhum deles garante que essa conectividade não é possível, então ambos os eventos são considerados abertos.

Se ambos os eventos estão abertos, a análise segue para o enlace seguinte:

Evento 1K2K = enlace 1 operando, enlace 2 operando = probabilidade $P_1 \times P_2$

Evento 1K2F = enlace 1 operando, enlace 2 operando = probabilidade $P_1 \times (1 - P_2)$

Evento 1F2K = enlace 1 em falha, enlace 2 operando = probabilidade $(1 - P_1) \times P_2$

Evento 1F2F = enlace 1 em falha, enlace 2 em falha = probabilidade $(1 - P_1) \times (1 - P_2)$ (-)

O Evento 1F2F inviabiliza a conectividade entre *A* e *B*, este evento é marcado com (-) e não entra no cálculo da probabilidade de comunicação entre *A* e *B*.

Seguindo a análise com os demais enlaces:

Evento 1K2K3K = $P_1 \times P_2 \times P_3$

Evento 1K2K3F = $P_1 \times P_2 \times (1 - P_3)$

Evento 1K2F3K = $P_1 \times (1 - P_2) \times P_3$

Evento 1K2F3F = $P_1 \times (1 - P_2) \times (1 - P_3)$

Evento 1F2K3K = $(1 - P_1) \times P_2 \times P_3$

Evento 1F2K3F = $(1 - P_1) \times P_2 \times (1 - P_3)$

Todos os eventos acima estão abertos, ou seja não garantem a comunicação entre *A* e *B* e nem a inviabilizam.

Seguindo com a análise:

Evento 1K2K3K4K = $P_1 \times P_2 \times P_3 \times P_4$ (+)

Evento 1K2K3K4F = $P_1 \times P_2 \times P_3 \times (1 - P_4)$

Evento 1K2K3F4K = $P_1 \times P_2 \times (1 - P_3) \times P_4$ (+)

Evento 1K2K3F4F = $P_1 \times P_2 \times (1 - P_3) \times (1 - P_4)$

Evento 1K2F3K4K = $P_1 \times (1 - P_2) \times P_3 \times P_4$ (+)

$$\text{Evento 1K2F3K4F} = P_1 \times (1 - P_2) \times P_3 \times (1 - P_4)$$

$$\text{Evento 1K2F3F4K} = P_1 \times (1 - P_2) \times (1 - P_3) \times P_4 (+)$$

$$\text{Evento 1K2F3F4F} = P_1 \times (1 - P_2) \times (1 - P_3) \times (1 - P_4) (-)$$

$$\text{Evento 1F2K3K4K} = (1 - P_1) \times P_2 \times P_3 \times P_4 (+)$$

$$\text{Evento 1F2K3K4F} = (1 - P_1) \times P_2 \times P_3 \times (1 - P_4)$$

$$\text{Evento 1F2K3F4K} = (1 - P_1) \times P_2 \times (1 - P_3) \times P_4$$

$$\text{Evento 1F2K3F4F} = (1 - P_1) \times P_2 \times (1 - P_3) \times (1 - P_4)$$

Os eventos marcados com (+) garantem a conectividade entre A e B e serão levados em conta no cálculo da probabilidade de conectividade entre A e B. Os eventos marcados com (-) inviabilizam a conectividade entre A e B e devem ser desconsiderados. Como ainda existem eventos abertos, segue-se o procedimento:

$$\text{Evento 1K2K3K4F5K} = P_1 \times P_2 \times P_3 \times (1 - P_4) \times P_5 (+)$$

$$\text{Evento 1K2K3K4F5F} = P_1 \times P_2 \times P_3 \times (1 - P_4) \times (1 - P_5) (-)$$

$$\text{Evento 1K2K3F4F5K} = P_1 \times P_2 \times (1 - P_3) \times (1 - P_4) \times P_5 (+)$$

$$\text{Evento 1K2K3F4F5F} = P_1 \times P_2 \times (1 - P_3) \times (1 - P_4) \times (1 - P_5) (-)$$

$$\text{Evento 1K2F3K4F5K} = P_1 \times (1 - P_2) \times P_3 \times (1 - P_4) \times P_5 (+)$$

$$\text{Evento 1K2F3K4F5F} = P_1 \times (1 - P_2) \times P_3 \times (1 - P_4) \times (1 - P_5) (-)$$

$$\text{Evento 1F2K3K4F5K} = (1 - P_1) \times P_2 \times P_3 \times (1 - P_4) \times P_5 (+)$$

$$\text{Evento 1F2K3K4F5F} = (1 - P_1) \times P_2 \times P_3 \times (1 - P_4) \times (1 - P_5) (-)$$

$$\text{Evento 1F2K3F4K5K} = (1 - P_1) \times P_2 \times (1 - P_3) \times P_4 \times P_5 (+)$$

$$\text{Evento 1F2K3F4K5F} = (1 - P_1) \times P_2 \times (1 - P_3) \times P_4 \times (1 - P_5) (-)$$

$$\text{Evento 1F2K3F4F5K} = (1 - P_1) \times P_2 \times (1 - P_3) \times (1 - P_4) \times P_5 (+)$$

$$\text{Evento 1F2K3F4F5F} = (1 - P_1) \times P_2 \times (1 - P_3) \times (1 - P_4) \times (1 - P_5) (-)$$

Como não existem mais eventos abertos, podemos computar a probabilidade de conexão entre A e B somando os valores obtidos nos eventos marcados com (+).

A árvore de eventos para esta rede pode ser vista na Figura 2.15:

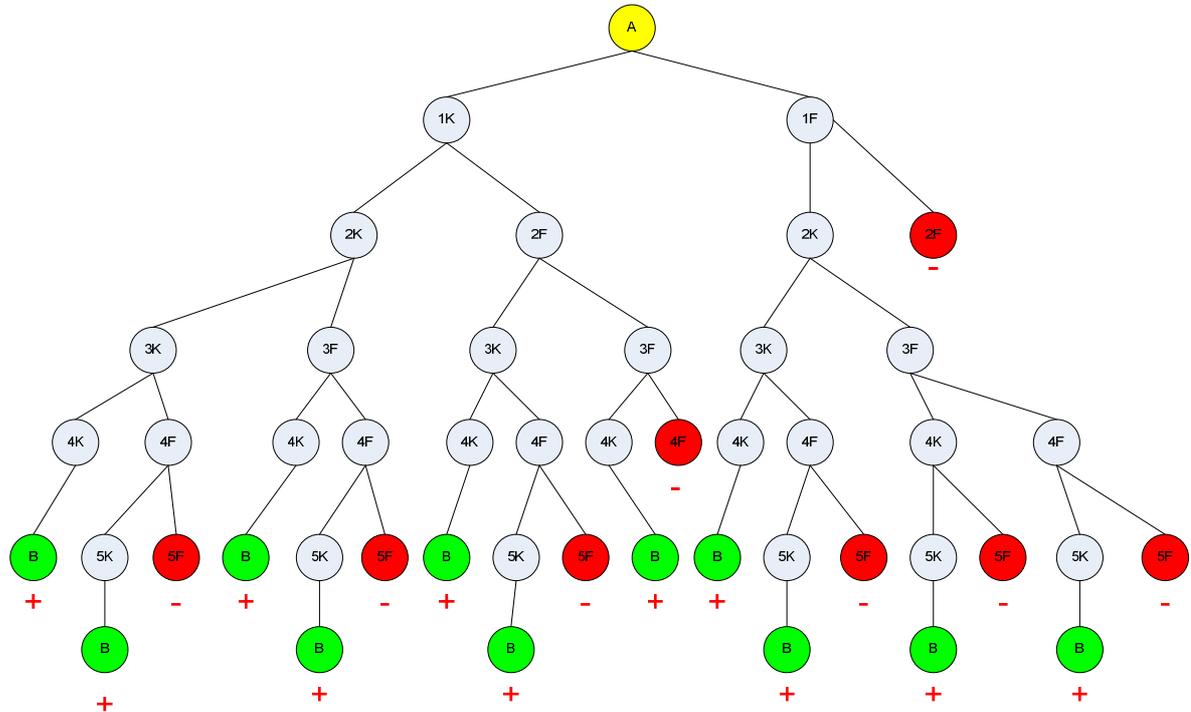


Figura 2.15: Árvore de Decomposição de Espaço de Estados

Considerando os seguintes valores numéricos para as probabilidades dos enlaces estarem ativos:

$$P_1 = 0,99$$

$$P_2 = 0,98$$

$$P_3 = 0,99$$

$$P_4 = 0,97$$

$$P_5 = 0,96$$

Colocando esses valores nos eventos marcados com (+) serão obtidos:

$$\text{Evento } 1K2K3K4K = P_1 \times P_2 \times P_3 \times P_4 = 931,68 * 10^{-3}$$

$$\text{Evento 1K2K3F4K} = P_1 \times P_2 \times (1 - P_3) \times P_4 = 9,4109 * 10^{-3}$$

$$\text{Evento 1K2F3K4K} = P_1 \times (1 - P_2) \times P_3 \times P_4 = 19,014 * 10^{-3}$$

$$\text{Evento 1K2F3F4K} = P_1 \times (1 - P_2) \times (1 - P_3) \times P_4 = 192,06 * 10^{-6}$$

$$\text{Evento 1F2K3K4K} = (1 - P_1) \times P_2 \times P_3 \times P_4 = 9,4109 * 10^{-3}$$

$$\text{Evento 1K2K3K4F5K} = P_1 \times P_2 \times P_3 \times (1 - P_4) \times P_5 = 922,08 * 10^{-6}$$

$$\text{Evento 1K2K3F4F5K} = P_1 \times P_2 \times (1 - P_3) \times (1 - P_4) \times P_5 = 279,42 * 10^{-6}$$

$$\text{Evento 1K2F3K4F5K} = P_1 \times (1 - P_2) \times P_3 \times (1 - P_4) \times P_5 = 564,54 * 10^{-6}$$

$$\text{Evento 1F2K3K4F5K} = (1 - P_1) \times P_2 \times P_3 \times (1 - P_4) \times P_5 = 279,42 * 10^{-6}$$

$$\text{Evento 1F2K3F4K5K} = (1 - P_1) \times P_2 \times (1 - P_3) \times P_4 \times P_5 = 91,258 * 10^{-6}$$

$$\text{Evento 1F2K3F4F5K} = (1 - P_1) \times P_2 \times (1 - P_3) \times (1 - P_4) \times P_5 = 2,8224 * 10^{-6}$$

A probabilidade de haver conectividade entre A e B será dada pela soma dos eventos acima relacionados.

$$P_{AB} = 971,85 * 10^{-3} = 0,97185 = 97,185\%$$

Para verificar a possibilidade do uso da decomposição de estados para a criação de um algoritmo recursivo visando o cálculo da disponibilidade de uma rede entre dois componentes quaisquer foram realizados cálculos comparativos de disponibilidade utilizando um simples agrupamento série/paralelo como o que será obtido através da decomposição de espaço de estados.

Seja:

P_i = Probabilidade do componente i estar funcionando

P_j = Probabilidade do componente j estar funcionando

Q_i = probabilidade do componente i não estar funcionando = $1 - P_i$

Q_j = probabilidade do componente j não estar funcionando = $1 - P_j$

Dados:

MTBF – *Mean Time Between Failures*

MTTF – *Mean Time do Failure*

MTTR – *Mean Time to Repair*

A equação (2.14) apresenta a probabilidade da facilidade i estar funcionando.

$$P_i = 1 - \frac{MTTR}{MTBF} \quad (2.14)$$

A probabilidade de duas facilidades independentes estarem funcionando simultaneamente é dada pela equação (2.15).

$$P_{ij} = P_i \times P_j \quad (2.15)$$

A probabilidade de ao menos uma facilidade estar funcionando é dada pela expressão (2.16)

$$P_{iouj} = P_i \times P_j + P_i \times Q_j + Q_i \times P_j = P_i + P_j - P_i \times P_j \quad (2.16)$$

Considere o diagrama da Figura 2.16, formado pelos nós A, B e C e pelos enlaces 1 e 2. Considere que os nós não falham e a probabilidade dos enlaces estarem funcionando é dada por P_1 e P_2 . A probabilidade de haver conectividade de um extremo a outro nesta rede é dada por:

$$P_{AC} = P_1 \times P_2 \quad (2.17)$$

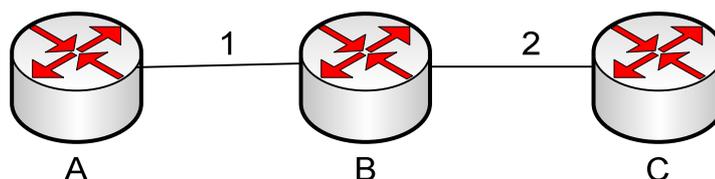


Figura 2.16: Rede em série

Já que só haverá conectividade se ambos os enlaces estiverem funcionando simultaneamente.

Pela análise da rede apresentada na Figura 2.17, formada pelos nós A e B e pelos enlaces 1 e 2, pode-se observar que haverá conectividade de um extremo ao outro da rede se qualquer um dos enlaces estiver funcionando. Logo, a probabilidade de haver conectividade nesta rede será dada por:

$$P_{AC} = P_1 \times P_2 + P_1 \times (1 - P_2) + P_2 \times (1 - P_1)$$

$$P_{AC} = P_1 \times P_2 + P_1 - P_1 \times P_2 + P_2 - P_1 \times P_2$$

$$P_{AC} = P_1 + P_2 - P_1 \times P_2 \quad (2.18)$$

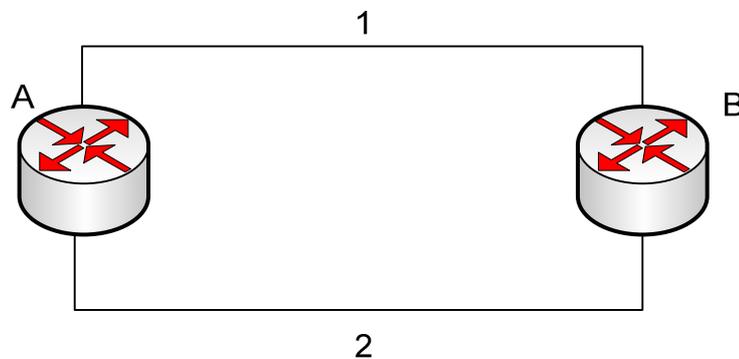


Figura 2.17: Rede em paralelo

Seja:

D_i = disponibilidade do componente i

D_j = disponibilidade do componente j

Disponibilidade de um determinado componente, equação (2.19).

$$D_i = \frac{MTBF}{MTBF + MTTR} \quad (2.19)$$

A disponibilidade de um sistema é calculada modelando-se o sistema como uma interconexão de partes em série e paralelo [OGGERINO, 2001]. As regras para a colocação de determinados componentes em série ou em paralelo são as seguintes:

- Se a falha de um componente torna a combinação inoperante, então os componentes são considerados em série.

- Se a falha de um componente faz com que sua função seja assumida por outro componente, eles são considerados em paralelo.

E a disponibilidade de dois componentes, 1 e 2 em série é dada pela equação (2.20).

$$D_t = D_1 \times D_2 \quad (2.20)$$

Se os dois componentes estiverem em paralelo a disponibilidade será dada por:

$$D_T = D_1 \times D_2 + D_1(1 - D_2) + D_2(1 - D_1)$$

$$D_t = D_1 \times D_2 + D_1 - D_1 \times D_2 + D_2 - D_1 \times D_2$$

$$D_t = D_1 + D_2 - D_1 \times D_2 \quad (2.21)$$

Considerando a rede da Figura 2.18. Imagine que os nós possuem 100% de disponibilidade e a disponibilidade de cada enlace é dada por D1 e D2.

A disponibilidade da rede será dada pela equação (2.22).

$$D_t = D_1 \times D_2 \quad (2.22)$$

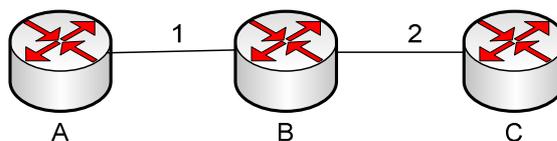


Figura 2.18: Rede em série

Considerando agora a rede apresentada na Figura 2.19.

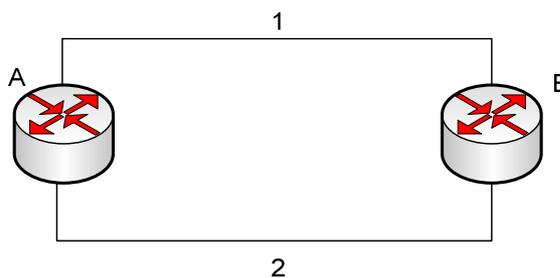


Figura 2.19: Rede em paralelo

A disponibilidade da rede será dada pela equação (2.23)

$$D_t = D_1 + D_2 - D_1 \times D_2 \quad (2.23)$$

Comparando esses resultados com os resultados obtidos acima para probabilidade, percebe-se que os cálculos de disponibilidade em série e paralelo são realizados da mesma forma que os cálculos de probabilidade de comunicação em série e paralelo.

Supondo as redes 1, 2 e 3 mostradas na figura 2.20.

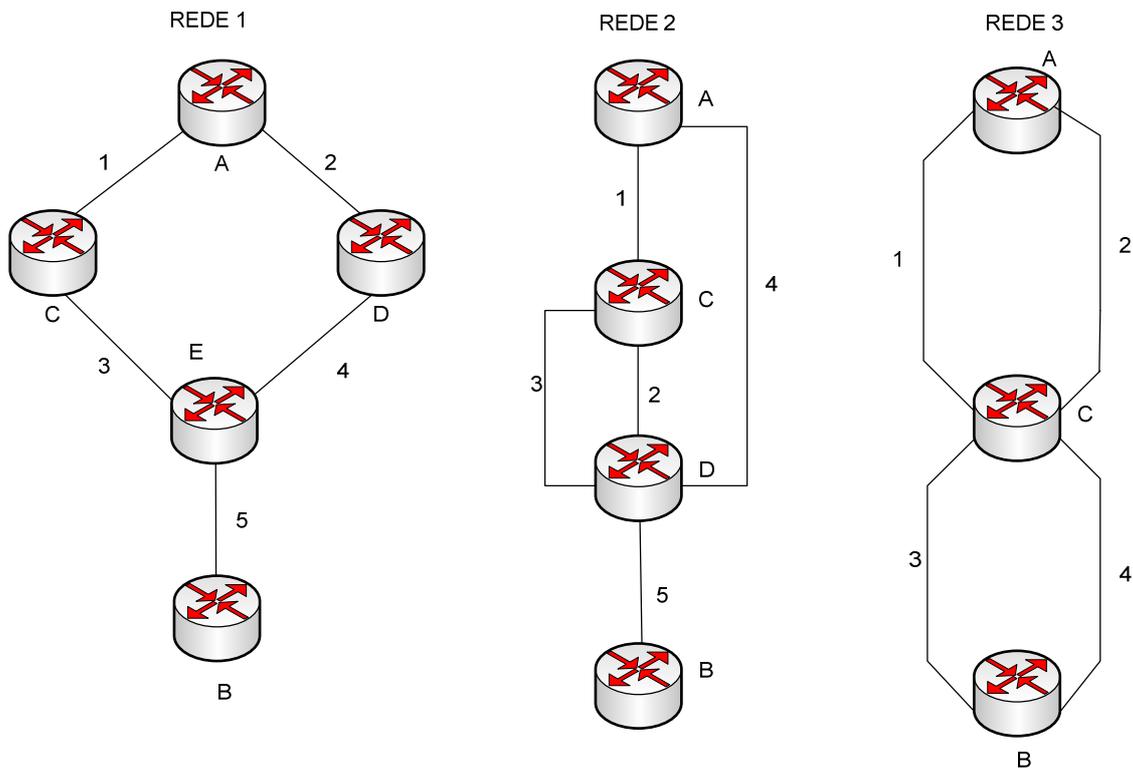


Figura 2.20: Redes diversas

Visando provar que a decomposição de espaço de estados pode ser utilizada para estimativa da disponibilidade em redes, foram comparados os resultados obtidos com simples agrupamento série/paralelo com os resultados obtidos pelo uso do algoritmo.

Para simplificação dos cálculos supôs-se que os nós não falham, ou seja possuem disponibilidade igual a 1.

Rede 1:

Sejam:

D1 = disponibilidade do enlace 1

D2 = disponibilidade do enlace 2

D3 = disponibilidade do enlace 3

D4 = disponibilidade do enlace 4

D5 = disponibilidade do enlace 5

Cálculo da disponibilidade da rede com agrupamento série paralelo:

Os enlaces 1 e 3 estão em série:

$$D_{13} = D_1 \times D_3$$

Os enlaces 2 e 4 estão em série:

$$D_{24} = D_2 \times D_4$$

O caminho formado pelos enlaces 1 e 3 está em paralelo com o caminho formado pelos enlaces 2 e 4

$$D_{AE} = D_1 \times D_3 + D_2 \times D_4 - D_1 \times D_2 \times D_3 \times D_4$$

Em série com o enlace 5:

$$D_{AB} = (D_1 \times D_3 + D_2 \times D_4 - D_1 \times D_2 \times D_3 \times D_4) \times D_5$$

$$D_{AB} = D_1 \times D_3 \times D_5 + D_2 \times D_4 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 \quad (2.24)$$

Cálculo pela decomposição de espaço de estados:

Seja:

1F = evento que significa enlace 1 em falha

1K = evento que significa enlace 1 operando

Os eventos que garantem a comunicação entre A e B são:

$$1K2K3K4K5K = D_1 \times D_2 \times D_3 \times D_4 \times D_5$$

$$1K2K3K4F5K = D_1 \times D_2 \times D_3 \times (1 - D_4) \times D_5$$

$$1K2K3F4K5K = D_1 \times D_2 \times (1 - D_3) \times D_4 \times D_5$$

$$1K2F3K4K5K = D_1 \times (1 - D_2) \times D_3 \times D_4 \times D_5$$

$$1K2F3K4F5K = D_1 \times (1 - D_2) \times D_3 \times (1 - D_4) \times D_5$$

$$1F2K3K4K5K = (1 - D_1) \times D_2 \times D_3 \times D_4 \times D_5$$

$$1F2K3F4K5K = (1 - D_1) \times D_2 \times (1 - D_3) \times D_4 \times D_5$$

$$D_{AB} = D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times (1 - D_4) \times D_5 + D_1 \times D_2 \times (1 - D_3) \times D_4 \times D_5 +$$

$$D_1 \times (1 - D_2) \times D_3 \times D_4 \times D_5 + D_1 \times (1 - D_2) \times D_3 \times (1 - D_4) \times D_5 + (1 - D_1) \times D_2 \times D_3 \times D_4 \times D_5 +$$
$$(1 - D_1) \times D_2 \times (1 - D_3) \times D_4 \times D_5$$

$$D_{AB} = D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_4 \times D_5 -$$
$$D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_3 \times D_4 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_3 \times D_5 -$$
$$D_1 \times D_3 \times D_4 \times D_5 - D_1 \times D_2 \times D_3 \times D_5 + D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_2 \times D_3 \times D_4 \times D_5 -$$
$$D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_2 \times D_4 \times D_5 - D_2 \times D_3 \times D_4 \times D_5 - D_1 \times D_2 \times D_4 \times D_5 +$$
$$D_1 \times D_2 \times D_3 \times D_4 \times D_5$$

Após reduzir os termos:

$$D_{AB} = D_1 \times D_3 \times D_5 + D_2 \times D_4 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 \quad (2.25)$$

Comparando-se as expressões obtidas em (2.24) e (2.25), verifica-se que foi obtido o mesmo resultado.

Rede 2

Sejam:

D1 = disponibilidade do enlace 1

D2 = disponibilidade do enlace 2

D3 = disponibilidade do enlace 3

D4 = disponibilidade do enlace 4

D5 = disponibilidade do enlace 5

Cálculo da disponibilidade da rede com agrupamento série paralelo:

Paralelo entre os enlaces 2 e 3:

$$D_{CD} = D_3 + D_2 - D_2 \times D_3$$

Em série com o enlace 1:

$$D_{AD}(1) = D_1 \times D_2 + D_1 \times D_3 - D_1 \times D_2 \times D_3$$

Formando o paralelo com o enlace 4:

$$D_{AD} = D_1 \times D_2 + D_1 \times D_3 - D_1 \times D_2 \times D_3 + D_4 - D_1 \times D_2 \times D_4 - D_1 \times D_3 \times D_4 + D_1 \times D_2 \times D_3 \times D_4$$

$$D_{AB} = D_1 \times D_2 \times D_5 + D_1 \times D_3 \times D_5 - D_1 \times D_2 \times D_3 \times D_5 + D_4 \times D_5 - D_1 \times D_2 \times D_4 \times D_5 - D_1 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times D_4 \times D_5 \quad (2.26)$$

Cálculo pela decomposição de espaço de estados:

Seja:

1F = evento que significa enlace 1 em falha

1K = evento que significa enlace 1 operando

Os eventos que garantem a comunicação entre A e B são:

$$1K2K3K4K5K = D_1 \times D_2 \times D_3 \times D_4 \times D_5$$

$$1K2K3K4F5K = D_1 \times D_2 \times D_3 \times (1 - D_4) \times D_5$$

$$1K2K3F4K5K = D_1 \times D_2 \times (1 - D_3) \times D_4 \times D_5$$

$$1K2K3F4F5K = D_1 \times D_2 \times (1 - D_3) \times (1 - D_4) \times D_5$$

$$1K2F3K4K5K = D_1 \times (1 - D_2) \times D_3 \times D_4 \times D_5$$

$$1K2F3K4F5K = D_1 \times (1 - D_2) \times D_3 \times (1 - D_4) \times D_5$$

$$1K2F3F4K5K = D_1 \times (1 - D_2) \times (1 - D_3) \times D_4 \times D_5$$

$$1F2K3K4K5K = (1 - D_1) \times D_2 \times D_3 \times D_4 \times D_5$$

$$1F2K3F4K5K = (1 - D_1) \times D_2 \times (1 - D_3) \times D_4 \times D_5$$

$$1F2F3K4K5K = (1 - D_1) \times (1 - D_2) \times D_3 \times D_4 \times D_5$$

$$1F2F3F4K5K = (1 - D_1) \times (1 - D_2) \times (1 - D_3) \times D_4 \times D_5$$

$$\begin{aligned} D_{AB} = & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times (1 - D_4) \times D_5 + D_1 \times D_2 \times (1 - D_3) \times D_4 \times D_5 + \\ & D_1 \times D_2 \times (1 - D_3) \times (1 - D_4) \times D_5 + D_1 \times (1 - D_2) \times D_3 \times D_4 \times D_5 + D_1 \times (1 - D_2) \times D_3 \times (1 - D_4) \times D_5 + \\ & D_1 \times (1 - D_2) \times (1 - D_3) \times D_4 \times D_5 + (1 - D_1) \times D_2 \times D_3 \times D_4 \times D_5 + (1 - D_1) \times D_2 \times (1 - D_3) \times D_4 \times D_5 + \\ & (1 - D_1) \times (1 - D_2) \times D_3 \times D_4 \times D_5 + (1 - D_1) \times (1 - D_2) \times (1 - D_3) \times D_4 \times D_5 \end{aligned}$$

$$\begin{aligned} D_{AB} = & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_4 \times D_5 - \\ & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_4 - D_1 \times D_2 \times D_3 \times D_5 - D_1 \times D_2 \times D_4 \times D_5 + \\ & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_3 \times D_4 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_3 \times D_5 - \\ & D_1 \times D_2 \times D_3 \times D_5 - D_1 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_1 \times D_4 \times D_5 - \\ & D_1 \times D_2 \times D_4 \times D_5 - D_1 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_2 \times D_3 \times D_4 \times D_5 - \\ & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_2 \times D_4 \times D_5 - D_1 \times D_2 \times D_4 \times D_5 - D_2 \times D_3 \times D_4 \times D_5 + \\ & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_3 \times D_4 \times D_5 - D_1 \times D_3 \times D_4 \times D_5 - D_2 \times D_3 \times D_4 \times D_5 + \\ & D_1 \times D_2 \times D_3 \times D_4 \times D_5 + D_4 \times D_5 - D_1 \times D_4 \times D_5 - D_2 \times D_4 \times D_5 + D_1 \times D_2 \times D_4 \times D_5 - \\ & D_3 \times D_4 \times D_5 + D_1 \times D_3 \times D_4 \times D_5 + D_2 \times D_3 \times D_4 \times D_5 - D_1 \times D_2 \times D_3 \times D_4 \times D_5 \end{aligned}$$

$$D_{AB} = D_1 \times D_2 \times D_5 + D_1 \times D_3 \times D_5 - D_1 \times D_2 \times D_3 \times D_5 + D_4 \times D_5 - D_1 \times D_2 \times D_4 \times D_5 - D_1 \times D_3 \times D_4 \times D_5 + D_1 \times D_2 \times D_3 \times D_4 \times D_5 \quad (2.27)$$

Observa-se que os resultados obtidos em (2.26) e (2.27) são idênticos.

Rede 3

Sejam:

D1 = disponibilidade do enlace 1

D2 = disponibilidade do enlace 2

D3 = disponibilidade do enlace 3

D4 = disponibilidade do enlace 4

Cálculo da disponibilidade da rede com agrupamento série paralelo:

$$D_{AC} = D_1 + D_2 - D_1 \times D_2$$

$$D_{CB} = D_3 + D_4 - D_3 \times D_4$$

$$D_{AB} = (D_1 + D_2 - D_1 \times D_2) \times (D_3 + D_4 - D_3 \times D_4)$$

$$D_{AB} = D_1 \times D_3 + D_1 \times D_4 - D_1 \times D_3 \times D_4 + D_2 \times D_3 + D_2 \times D_4 - D_2 \times D_3 \times D_4 - D_1 \times D_2 \times D_3 - D_1 \times D_2 \times D_4 + D_1 \times D_2 \times D_3 \times D_4$$

(2.28)

Cálculo pela decomposição de espaço de estados:

Seja:

1F = evento que significa enlace 1 em falha

1K = evento que significa enlace 1 operando

Os eventos que garantem a comunicação entre A e B são:

$$1K2K3K = D_1 \times D_2 \times D_3$$

$$1K2K3F4K = D_1 \times D_2 \times (1 - D_3) \times D_4$$

$$1K2F3K = D_1 \times (1 - D_2) \times D_3$$

$$1K2F3F4K = D_1 \times (1 - D_2) \times (1 - D_3) \times D_4$$

$$1F2K3K = (1 - D_1) \times D_2 \times D_3$$

$$1F2K3F4K = (1 - D_1) \times D_2 \times (1 - D_3) \times D_4$$

$$D_{AB} = D_1 \times D_2 \times D_3 + D_1 \times D_2 \times (1 - D_3) \times D_4 + D_1 \times (1 - D_2) \times D_3 +$$

$$D_1 \times (1 - D_2) \times (1 - D_3) \times D_4 + (1 - D_1) \times D_2 \times D_3 + (1 - D_1) \times D_2 \times (1 - D_3) \times D_4$$

$$D_{AB} = D_1 \times D_2 \times D_3 + D_1 \times D_2 \times D_4 - D_1 \times D_2 \times D_3 \times D_4 + D_1 \times D_3 - D_1 \times D_2 \times D_3 +$$

$$D_1 \times D_4 - D_1 \times D_2 \times D_4 - D_1 \times D_3 \times D_4 + D_1 \times D_2 \times D_3 \times D_4 + D_2 \times D_3 - D_1 \times D_2 \times D_3 +$$

$$D_2 \times D_4 - D_2 \times D_3 \times D_4 - D_1 \times D_2 \times D_4 + D_1 \times D_2 \times D_3 \times D_4$$

$$D_{AB} = D_1 \times D_3 + D_1 \times D_4 - D_1 \times D_3 \times D_4 + D_2 \times D_3 + D_2 \times D_4 - D_2 \times D_3 \times D_4 - D_1 \times D_2 \times D_3 -$$

$$D_1 \times D_2 \times D_4 + D_1 \times D_2 \times D_3 \times D_4$$

(2.29)

Comparando-se (2.28) e (2.29) observa-se que foram obtidos os mesmos resultados.

Nas três redes observadas o resultado obtido pelo simples agrupamento série/paralelo foi o mesmo obtido pela decomposição de espaço de estados. Extrapolando este resultado para qualquer rede em malha pode-se perceber que para cálculos de disponibilidade é possível utilizar as mesmas regras de redução de grafos que são utilizadas no cálculo da probabilidade de haver conectividade entre dois nós da rede.

Então, para a modelagem proposta para o cálculo de disponibilidade de uma rede em malha, estarão sendo utilizadas as técnicas de redução de grafos da teoria de confiabilidade em redes em malha, mas estarão sendo utilizados os valores de disponibilidade arbitrários. Na prática, para o uso do algoritmo proposto poderão ser usados valores de disponibilidade obtidos à partir dos dados históricos dos equipamentos.

3 - METODOLOGIA

3.1 - APRESENTAÇÃO

Este estudo teve como foco a criação e implementação de um algoritmo para cálculo de disponibilidade em redes baseado na teoria da decomposição de espaço de estados.

Uma vez criado o algoritmo, o mesmo foi testado em diversos seguimentos de rede para sua validação.

3.2 - UNIVERSO

Nosso universo de testes se divide em duas categorias: rede externa e rede interna.

Por rede externa considerou-se a rede IP que é utilizada para prestação de serviços a clientes.

Por rede interna, considerou-se a intranet da empresa, cujo funcionamento dentro dos padrões de normalidade é de fundamental importância na prestação dos serviços pela rede externa.

3.3 - AMOSTRA

O processo de amostragem será intencional. Serão escolhidos trechos específicos da rede de forma a testar determinadas propriedades do algoritmo e exemplificar a atuação do mesmo em situações de alteração nos trechos propostos.

3.4 - HIPÓTESES

1. A modelagem por decomposição de espaço de estados é adequada para cálculo de disponibilidade em redes de computadores;
2. A modelagem por decomposição de espaço de estados não é adequada para cálculo de disponibilidade em redes de computadores;
3. A modelagem por decomposição de espaço de estados é adequada para cálculo de disponibilidade em redes de computadores e a implementação pelo algoritmo que é o cerne deste trabalho provou ser adequada;
4. A modelagem por decomposição de espaço de estados é adequada para cálculo de disponibilidade em redes de computadores, mas a implementação não provou ser adequada;

4 - IMPLEMENTAÇÃO

4.1 - DESCRIÇÃO DETALHADA DO ALGORITMO

A chamada do algoritmo pede como parâmetros o nome do arquivo de texto que possui dos dados da rede que será analisada, o nó que será considerado como ponta *A* e o nó que será considerado como ponta *B*.

O algoritmo proposto inicia sua execução pela leitura dos dados da rede em um arquivo de texto como pode ser visto no pseu-código abaixo:

```
ler arquivo de dados da rede
ler identificação da ponta A
ler identificação da ponta B
```

Os dados devem ser inseridos da seguinte forma:

Roteadores:

Nome do roteador; disponibilidade; estado

Enlaces:

Palavra link; identificação do enlace; origem; destino, disponibilidade do enlace; estado

O algoritmo então realiza a leitura dos dados e os armazena em vetores associativos conforme o pseudo-código abaixo:

```
enquanto houver dados de rede faça {
    pular linhas vazias
    pular linhas de comentário
    retirar o último caracter na linha
    se o dado lido for um enlace faça {
        separar os dados contidos na linha em campos
        armazenar o roteador de origem do enlace
        armazenar o roteador de destino do enlace
        armazenar o valor da disponibilidade histórica do enlace
        contar o número de enlaces
        armazenar o estado daquele enlace
    } fim da leitura dos dados do enlace
    se o dado não for um enlace, é um roteador então faça {
        separar os dados contidos na linha em campos
        atribuir o valor da disponibilidade do roteador
        contar o numero de roteadores
        atribuir o estado do roteador
    } fim leitura dos dados do roteador
} fim do enquanto }
```

}

Uma vez lidos os dados, o algoritmo monta a matriz de conexão para cada roteador, conforme o pseudo-código abaixo:

```
para cada roteador faça {
    imprimir a identificação do roteador
    para cada enlace faça {
        se o roteador for a origem do enlace faça {
            armazenar o destino deste enlace em uma variavel
            imprimir o enlace e o destino do enlace
        } fim se
        se o roteador for o destino do enlace faça {
            armazenar a origem do enlace em uma variável
            imprimir o enlace e a origem do enlace
        } fim se
    } fim para cada enlace
} fim para cada roteador
```

A matriz de conexões possui o seguinte formato:

para todos os roteadores da rede teremos uma linha como abaixo:

```
{ roteador_A: link_AB, roteador_B; link_AC, roteador_C}
{ roteador_B: link_AB, roteador_A; link_BD, roteador_D}
```

onde o *link_AB* é o enlace que leva do *roteador_A* ao *roteador_B*, *link_AC* é o enlace que leva do *roteador_A* ao *roteador_C* e assim por diante.

Em cada linha, ela possui um dos roteadores da rede seguido de pares formados pelos enlaces que saem ou chegam deste roteador e dos roteadores que estão na ponta desses enlaces. Por exemplo, seja a rede mostrada na Figura 4.1

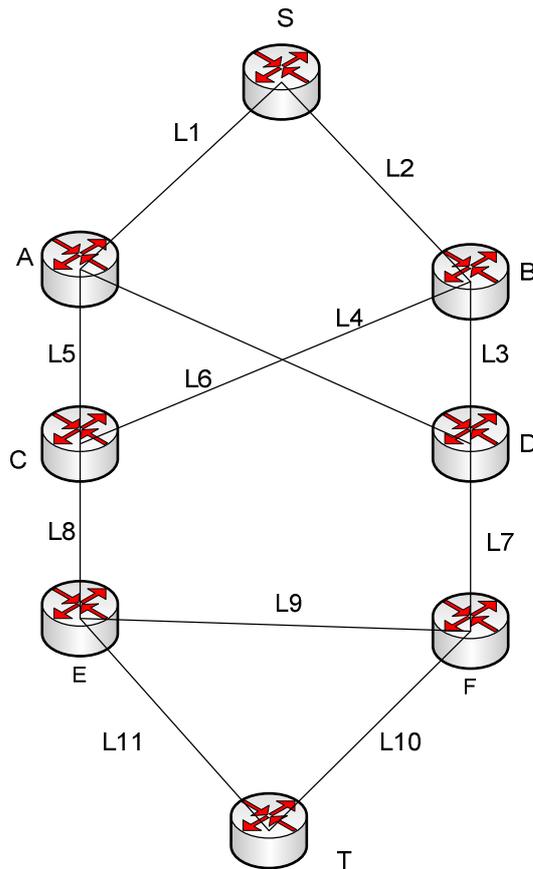


Figura 4.1: Rede exemplo 1

A matriz de conexões para esta rede será:

F:L10,T;L7,D;L9,E;
 A:L5,C;L6,D;L1,S;
 S:L1,A;L2,B;
 T:L11,E;L10,F;
 D:L6,A;L7,F;L3,B;
 C:L5,A;L4,B;L8,E;
 E:L11,T;L8,C;L9,F;
 B:L4,C;L2,S;L3,D;

O algoritmo então inicializa a variável $\$soma_disp$, que irá armazenar o valor da disponibilidade da rede em análise entre os pontos A e B e calcula os valores de duas outras variáveis $\$total_routers$ e $\$total_links$, que serão utilizadas para gerar as combinações possíveis de estados para todos os roteadores e enlaces da rede.

Conforme anteriormente descrito, a decomposição de espaço de estados é uma análise que considera a rede como sendo o resultado da combinação dos estados de cada um de seus componentes e o efeito que cada estado tem na existência ou não de conectividade entre os pontos A e B .

Para este cálculo é preciso considerar cada nó e enlace da rede em seus estados de funcionando (estado = 1) ou fora de serviço (estado = 0) e verificar para cada estado se há ou não conectividade entre os pontos *A* e *B*.

A forma encontrada para resolver este desafio foi a utilização da base 2. Em uma rede com 5 enlaces e 4 roteadores, tem-se:

$2^5 = 32$ combinações possíveis de estados para os 5 enlaces e $2^4 = 16$ combinações possíveis de estados para os roteadores.

Calcula-se então o número de combinações possíveis para os roteadores e os enlaces. As duas variáveis serão convertidas para número binário e o valor resultante será atribuído ao vetor de estados dos roteadores e dos enlaces. Com este procedimento pode-se garantir que cada situação da rede será testada conforme pode ser visto no trecho abaixo.

```
inicio do loop de controle dos estados dos roteadores {
  loop de controle dos estados dos roteadores, loop decrescente, do valor total a zero
  converter para binário o numero total de estados dos roteadores e armazenar em uma string.
  se o numero atual de combinações gerar um vetor menor do que 32 bits
  usar rotina simplificada de conversão
  se o numero atual de combinações gerar um vetor maior do que 32 bits
  usar rotina ampliada de conversão que parte o vetor em segmentos de 32 bits depois os une
  para cada roteador faça{
    atribuir o vetor recebido acima como vetor de estados dos roteadores
  } fim para
  Se o roteador ponta_A receber estado 0, voltar ao inicio do loop para tentar novo vetor de estados
  se o roteador ponta_B receber estado 0, voltar ao inicio do loop para tentar novo vetor de estados
  inicio do loop de controle dos estados dos enlaces {
    loop de controle dos estados dos enlaces, loop decrescente, do valor total a zero
    converter para binário o numero total de estados dos enlaces e armazenar em uma string.
    se o numero atual de combinações gerar um vetor menor do que 32 bits
      usar rotina simplificada de conversão
    se o numero atual de combinações gerar um vetor maior do que 32 bits
      usar rotina ampliada de conversão que parte o vetor em segmentos de 32 bits depois os une
    para cada enlace faça{
      atribuir o vetor recebido acima como vetor de estados dos roteadores
    } fim para
    testar existência de conexão entre ponta A e ponta B através da rotina testa_conecta
    Se todos os enlaces estiverem com estado 1 e não houver conexão entre ponta_A e ponta_B faça {
      voltar ao inicio do loop de estado dos roteadores
    } fim se
    Se houver conexão entre ponta_A e ponta_B faça {
      inicializa com 1 o valor do produto da disponibilidade prod_disp
      para cada roteador faça {
        se o estado do roteador for 1 {
          multiplica o produto de disponibilidade pela disponibilidade do
roteador
```

```

    }
    se o estado do roteador for 0 {
        multiplica o produto de disponibilidade por (1 - (disponibilidade do
roteador))
    }
} fim para cada roteador
para cada enlace faça {
    se o estado do enlace for 1 {
        multiplica o produto de disponibilidade pela disponibilidade do enlace
    }
    se o estado do enlace for 0 {
        multiplica o produto de disponibilidade por (1 - (disponibilidade do
enlace))
    }
} fim para cada enlace
somar a disponibilidade soma_disp com o produto obtido prod_disp
} fim se
} fim loop de estados dos enlaces

} fim loop de estados dos roteadores

```

A algoritmo se inicia com o loop que testa todas as combinações de estados dos roteadores e dos enlaces. O loop recebe um label *SITUA_ROUTER* que será utilizado para o retorno em determinadas situações que serão descritas abaixo.

É necessário testar se o número de combinações de estados ultrapassa o valor que gera 32 bits devido à rotina de conversão para números binários, Se o número de combinações tiver mais do que 32 bits será necessário dividir o número em frações de 32 bits para a correta conversão. E isso será feito na rotina *dec2bin_gd* que está mostrada no algoritmo. Se o número de combinações tiver menos do que 32 bits utiliza-se a rotina *dec2bin* que também será mostrada no algoritmo.

O número binário convertido será transformado em uma *string* e esta gera os valores 0 ou 1 que serão armazenados nos vetores de estados dos roteadores e enlaces.

Para cada estado possível dos roteadores, testa-se todas as combinações possíveis de estados de enlaces, para verificar se há conectividade entre os pontos *A* e *B* e calcular a disponibilidade da rede.

Para otimizar o algoritmo e reduzir o número de iterações foram feitas as seguintes considerações:

1. O objetivo do algoritmo é calcular a disponibilidade entre os pontos *A* e *B*, não tem sentido realizar este cálculo se um dos dois pontos estiver fora de serviço. Então, se qualquer situação em que a ponta *A* ou a ponta *B* receber estado 0 será descartada e o algoritmo volta à posição de rótulo *SITUA_ROUTER* a fim de encontrar um novo vetor de estado para os roteadores.
2. Para cada vetor de estados dos roteadores, testa-se em primeiro lugar a situação em que todos os enlaces estão ativos. Se não houver conectividade entre os pontos *A* e *B* com todos os enlaces ativos para um dado vetor de estado dos roteadores é porque um ou mais roteadores cujo estado foi atribuído como fora de serviço ou 0 era(m) crítico(s) no caminho entre *A* e *B*. Neste caso não tem sentido testar mais nenhuma situação dos enlaces, pois não haverá conectividade entre os pontos *A* e *B*. neste caso, o algoritmo volta mais uma vez ao rótulo *SITUA_ROUTER* para testar um novo vetor de estado dos roteadores.

Se houver conectividade entre os pontos *A* e *B* para um dado vetor de estados dos enlaces e dos roteadores, é calculada a disponibilidade da rede neste estado considerando que, para cada roteador ou enlace que estava fora de serviço, estado 0, utiliza-se o valor $(1 - \text{disponibilidade})$ do enlace ou roteador. Assim pode-se considerar o efeito que cada componente da rede causa na disponibilidade total da rede entre os pontos *A* e *B*.

Ao final, a disponibilidade da rede será obtida considerando a soma de todas as possibilidades em que temos conectividade entre os pontos *A* e *B*. Ou seja, cada caminho que garante a conectividade entre os pontos *A* e *B* concorre para a disponibilidade final da rede entre esses pontos.

O algoritmo se encerra então imprimindo a hora e o valor encontrado da disponibilidade para aquela rede entre os pontos *A* e *B* determinados.

A rotina que testa a conectividade da rede entre os pontos *A* e *B* é mostrada no pseudo-código abaixo

```
Início da rotina de testes de conexões
  Inicialização das variáveis
  Inicia-se na ponta_A colocando-o como o roteador sob teste
```

```

Loop que testa que chegou até a ponta_B {
  inicializar a variável que mostra se achou ou não a ponta_B
  início do teste de caminho {
    verifica se o enlace que sai do roteador sob teste já foi testado, se foi faça {
      se o enlace já foi testado, volta-se ao início para escolher um enlace que
      ainda não tenha sido testado
    } fim se
    se o enlace que sai do roteador sob teste estiver ativo e o roteador na outra ponta deste
    enlace estiver ativo faça {
      o roteador que está sendo testado é colocado como anterior
      altera a variável que informa que este enlace já foi testado
      atribui-se ao último roteador o roteador sob teste
      o próximo roteador a ser testado será o roteador no fim do enlace
      altera a variável que mostra que um caminho válido foi encontrado
      adiciona 1 à variável de controle de nós no caminho que está sendo testado
    } fim se
    se vc encontrou um caminho válido e o estado do próximo enlace é ativo faça {
      verifica-se se o próximo roteador será a ponta_B se fim faça {
        deixe a rotina retornando 1
      }
    } fim se
  } fim do teste do caminho
  se não foi encontrado um caminho válido faça {
    se o roteador sob teste é a ponta_A faça {
      não há conexão entre A e B, sai da rotina retornando 0
    } fim se
    retorne a um roteador anterior para procurar um outro caminho válido
    decresce o contador de passos naquele caminho.
  } fim se
  se o próximo roteador for a ponta_B faça {
    sai da rotina retornando 1
  } fim se
} Fim do loop que testa se há caminho da ponta_A à ponta_B

```

Fim da rotina de testes de conexões

A estrutura da rotina é simples, ela começa sempre na ponta A e procura entre os enlaces que saem ou chegam no roteador da ponta A um enlace que esteja com estado 1 e cujo roteador para o qual este enlace vai também esteja funcionando. Este novo roteador passa a ser o próximo a ser verificado e o algoritmo continua até chegar à ponta B.

Cada roteador pelo qual a rotina passa é armazenado em um vetor de anteriores e o enlace pelo qual a rotina passou é marcado como testado.

Se o algoritmo não encontra um caminho possível a partir de um determinado roteador ele retorna ao roteador anterior para procurar uma nova possibilidade.

Se um determinado enlace já tiver sido testado uma vez, a rotina procura um novo enlace para testar.

Se a rotina retorna à ponta *A* sem ter encontrado um caminho possível, é sinal de que não há conectividade entre os pontos *A* e *B*.

Se ao chegar em um determinado roteador, a rotina percebe que há um enlace direto para a ponta *B* e que este enlace está funcionando ela retorna ao programa principal informando que há conectividade. O mesmo se dá se ela alcança a ponta *B*.

O algoritmo inteiro, com suas rotinas e o programa principal é mostrado no Anexo A.

5 - TESTES

5.1 - DESCRIÇÃO DOS TESTES REALIZADOS

As redes que foram testadas são segmentos de redes especificamente escolhidos para verificar as propriedades do algoritmo. Em todas elas, foi considerado que a informação desejada é a estimativa de disponibilidade entre os nós *S* e *T*.

Cada rede foi testada da seguinte forma:

- Foi feito um teste com a rede em sua topologia original.
- Foram simuladas alterações na topologia e a rede foi novamente testada para verificar o efeito da alteração da topologia sobre a disponibilidade da rede entre os nós *S* e *T*.
- Para cada topologia de cada rede foram coletados os dados de tempo de execução do algoritmo.

O algoritmo pode ser utilizado da seguinte forma:

- calcular a disponibilidade real da rede entre dois pontos considerando-se a possibilidade de falhas em nós e enlaces.
- retirar ou acrescentar um enlace ou nó na rede e calcular a disponibilidade entre dois pontos da rede para a nova situação, comparando-a com o valor anteriormente obtido.
- observar ao longo do tempo como os dados históricos de disponibilidade dos equipamentos e enlaces afetam a disponibilidade da rede como um todo.

Uma vez que o objetivo deste trabalho é prover As profissionais de planejamento e projeto de redes uma ferramenta que permita a avaliação de possíveis alterações da rede na disponibilidade da mesma entre dois pontos quaisquer, os testes foram feitos utilizando um computador similar às estações de trabalho dos referidos profissionais.

Dados do equipamento utilizado nos testes:

- Estação de Trabalho: Intel ® Pentium ® M processador 1.6 GHz, 1 GB de RAM
- Sistema Operacional: Microsoft Windows XP Profissional Versão 2002, service Pack 2
- Interpretador da linguagem: Active Perl 5.8.8.820 [BROWN, 2001].

5.2 - TESTES E DESCRIÇÃO DOS RESULTADOS

5.2.1 - Rede 1

A primeira rede a ser testada é mostrada na Figura 5.1. Para este teste consideramos inicialmente que todos os enlaces e nós possuem disponibilidade individual e histórica de 0,9.

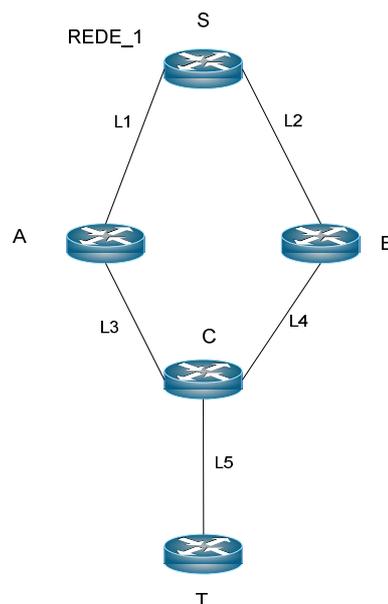


Figura 5.1: Rede de teste 1

Texto de entrada dos dados da rede no algoritmo:

Rede_1.txt

S;0.9;1
A;0.9;1
B;0.9;1
C;0.9;1
T;0.9;1
link;L1;S;A;0.9;1
link;L2;S;B;0.9;1
link;L3;A;C;0.9;1
link;L4;B;C;0.9;1
link;L5;C;T;0.9;1

O resultado do algoritmo é mostrado da figura 5.1.1:

```
Inicio: 13:18

Matriz de conexoes:

A:L3,C;L1,S;
S:L2,B;L1,A;
T:L5,C;
C:L5,T;L3,A;L4,B;
B:L2,S;L4,C;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.6079153599

Fim: 13:18

Tempo transcorrido em segundos desde o inicio da execucao: 0.06
```

Figura 5.1.1 – Resultado obtido para a rede 1

Supondo que a rede foi então alterada para as situações mostradas na figura 5.1.2. A *Rede_1A* foi criada inserindo-se um enlace entre os roteadores *A* e *B*. A rede original apresentava um caminho crítico entre os roteadores *C* e *T*, a *Rede_1B* foi criada inserindo-se um enlace em paralelo ao existente entre os roteadores *C* e *T*.

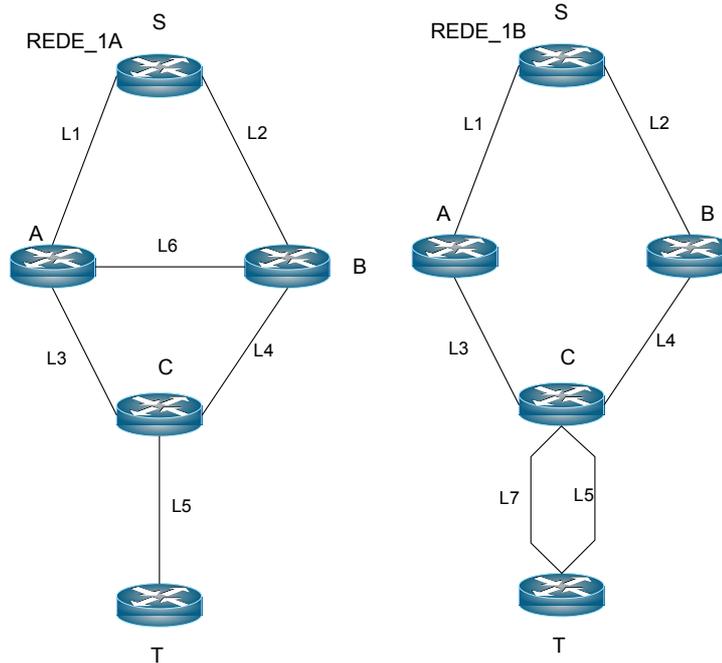


Figura 5.1.2: Rede de teste 1

Texto de entrada dos dados das redes no algoritmo, conforme tabela 5.1:

Tabela 5.1 - Entrada de dados para as alterações na rede 1.

Rede_1A.txt	Rede_1B.txt
S;0.9;1	S;0.9;1
A;0.9;1	A;0.9;1
B;0.9;1	B;0.9;1
C;0.9;1	C;0.9;1
T;0.9;1	T;0.9;1
link;L1;S;A;0.9;1	link;L1;S;A;0.9;1
link;L2;S;B;0.9;1	link;L2;S;B;0.9;1
link;L3;A;C;0.9;1	link;L3;A;C;0.9;1
link;L4;B;C;0.9;1	link;L4;B;C;0.9;1
link;L5;C;T;0.9;1	link;L5;C;T;0.9;1
link;L6;A;B;0.9;1	link;L7;C;T;0.9;1

A resultado do algoritmo para a rede_1A é mostrado na figura 5.1.3.

```
Inicio: 13:18

Matriz de conexoes:

A:L6,B;L3,C;L1,S;
S:L2,B;L1,A;
T:L5,C;
C:L5,T;L3,A;L4,B;
B:L2,S;L6,A;L4,C;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.61566376968

Fim: 13:18

Tempo transcorrido em segundos desde o inicio da execucao: 0.03
```

Figura 5.1.3 – Resultados para a Rede_1A

O resultado do algoritmo para a rede_1B é mostrado na figura 5.1.4:

```
Inicio: 13:18

Matriz de conexoes:

A:L3,C;L1,S;
S:L2,B;L1,A;
T:L5,C;L7,C;
C:L5,T;L3,A;L7,T;L4,B;
B:L2,S;L4,C;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.668706895889999

Fim: 13:18

Tempo transcorrido em segundos desde o inicio da execucao: 0.04
```

Figura 5.1.4 – Resultados para a Rede_1B

Observa-se que o acréscimo do enlace entre os roteadores *A* e *B* aumenta a disponibilidade da rede em 1 ponto percentual, mas o acréscimo de um enlace em paralelo entre os roteadores *C* e *T* causa um aumento de 6 pontos percentuais.

5.2.2 - Rede 2

A segunda rede a ser testada é mostrada na Figura 5.2

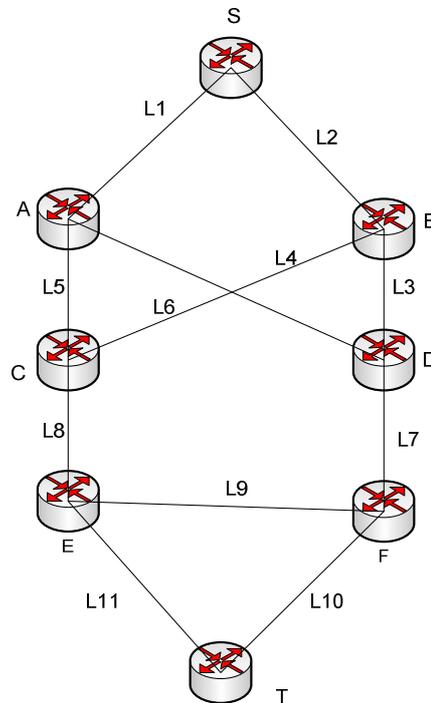


Figura 5.2: Rede de teste 2

Texto de entrada dos dados da rede no algoritmo:

Rede_2.txt

```
S;0.99;1
A;0.98;1
B;0.98;1
C;0.97;1
D;0.99;1
E;0.98;1
F;0.97;1
T;0.97;1
link;L1;S;A;0.99;1
link;L2;S;B;0.99;1
link;L3;B;D;0.99;1
link;L4;B;C;0.99;1
link;L5;A;C;0.99;1
link;L6;A;D;0.99;1
link;L7;D;F;0.99;1
link;L8;C;E;0.99;1
link;L9;E;F;0.99;1
link;L10;F;T;0.99;1
link;L11;E;T;0.99;1
```

Resultado do algoritmo, mostrado na Figura 5.2.1.

```
Inicio: 13:16

Matriz de conexoes:

F:L10,T;L7,D;L9,E;
A:L5,C;L6,D;L1,S;
S:L1,A;L2,B;
T:L11,E;L10,F;
D:L6,A;L7,F;L3,B;
C:L5,A;L4,B;L8,E;
E:L11,T;L8,C;L9,F;
B:L4,C;L2,S;L3,D;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.955728033561116

Fim: 13:16

Tempo transcorrido em segundos desde o inicio da execucao: 2.723
```

Figura 5.2.1 – Resultados para a Rede_2

Foram feitas as alterações na rede mostradas na figura 5.2.2. Para formar a *rede_2A* foi retirado o enlace entre os roteadores *E* e *F*. Para esta alteração não se espera uma variação significativa no valor da disponibilidade entre os pontos *S* e *T*. A *rede_2B* foi criada pela inclusão do enlace *L12* entre os roteadores *C* e *F* da rede original. Espera-se com esta alteração um pequeno aumento na disponibilidade da rede entre os pontos *S* e *T*.

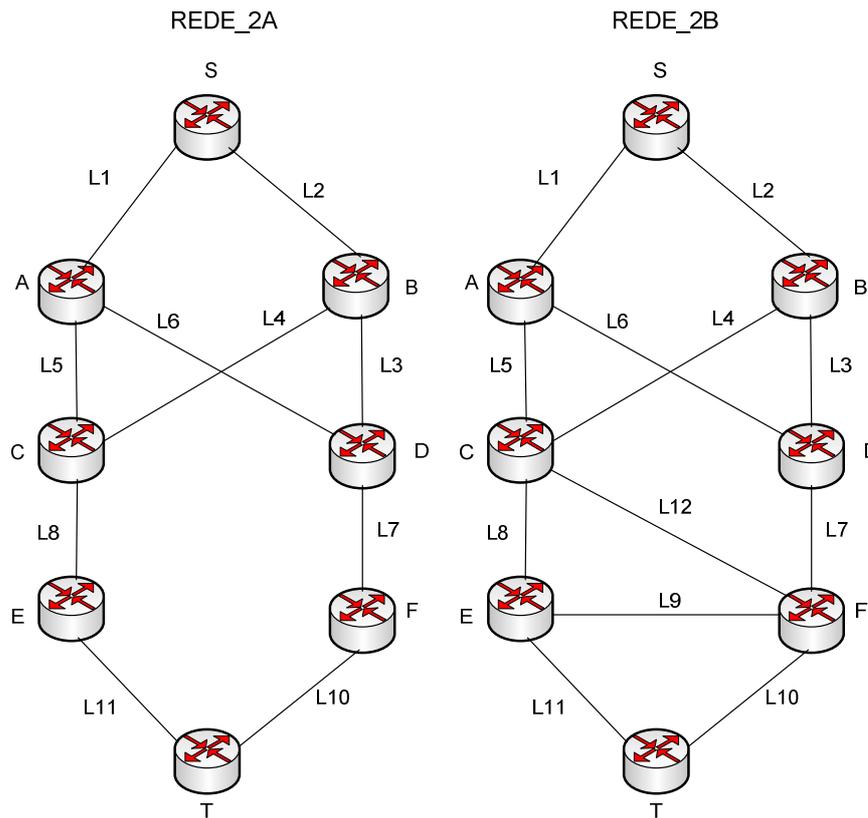


Figura 5.2.2: Rede de teste 2 – alterações A e B

Texto de entrada dos dados das redes 2A e 2B no algoritmo é mostrado na Tabela 5.2

Tabela 5.2 - Entrada dos dados para as redes 2A e 2B

Rede_2A.txt	Rede_2B.txt
S;0.99;1	S;0.99;1
A;0.98;1	A;0.98;1
B;0.98;1	B;0.98;1
C;0.97;1	C;0.97;1
D;0.99;1	D;0.99;1
E;0.98;1	E;0.98;1
F;0.97;1	F;0.97;1
T;0.97;1	T;0.97;1
link;L1;S;A;0.99;1	link;L1;S;A;0.99;1
link;L2;S;B;0.99;1	link;L2;S;B;0.99;1
link;L3;B;D;0.99;1	link;L3;B;D;0.99;1
link;L4;B;C;0.99;1	link;L4;B;C;0.99;1
link;L5;A;C;0.99;1	link;L5;A;C;0.99;1
link;L6;A;D;0.99;1	link;L6;A;D;0.99;1
link;L7;D;F;0.99;1	link;L7;D;F;0.99;1
link;L8;C;E;0.99;1	link;L8;C;E;0.99;1
link;L10;F;T;0.99;1	link;L9;E;F;0.99;1
link;L11;E;T;0.99;1	link;L10;F;T;0.99;1
	link;L11;E;T;0.99;1
	link;L12;C;F;0.99;1

E foram obtidos os resultados mostrados na Figura 5.2.3 para a rede_2A e 5.2.4 para a rede_2B.

Inicio: 13:16

Matriz de conexoes:

F:L10,T;L7,D;
A:L5,C;L6,D;L1,S;
S:L1,A;L2,B;
T:L11,E;L10,F;
D:L6,A;L7,F;L3,B;
C:L5,A;L4,B;L8,E;
E:L11,T;L8,C;
B:L4,C;L2,S;L3,D;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.955046995741571

Fim: 13:16

Tempo transcorrido em segundos desde o inicio da execucao: 1.412

Figura 5.2.3 – Resultados para a Rede_2A

Inicio: 13:16

Matriz de conexoes:

F:L10,T;L12,C;L7,D;L9,E;
A:L5,C;L6,D;L1,S;
S:L1,A;L2,B;
T:L11,E;L10,F;
D:L6,A;L7,F;L3,B;
C:L5,A;L12,F;L4,B;L8,E;
E:L11,T;L8,C;L9,F;
B:L4,C;L2,S;L3,D;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.956267577247304

Fim: 13:16

Tempo transcorrido em segundos desde o inicio da execucao: 6.499

Figura 5.2.4 – Resultados para a Rede_2B

Como era esperado a retirada no enlace *L9* entre os roteadores *E* e *F* provoca uma queda da disponibilidade da rede entre os pontos *S* e *T* de menos de 0,1 ponto percentual. Já o acréscimo do enlace *L12* provoca um aumento de 0,1 ponto percentual.

Em seguida foram feitas as alterações mostradas na Figura 5.2.5. Para formar a *rede_2C* foi retirado o enlace *L8* entre os roteadores *C* e *E*. Ao fazer isso o roteador *F* torna-se um componente crítico da rede e espera-se com uma queda na disponibilidade entre os pontos *S* e *T*.

A *rede_2D* foi criada com a retirada também do enlace *L12*. Neste cenário criou-se um caminho crítico formado pelos roteadores *D* e *F* e pelo enlace *L7* e isso deve provocar uma queda ainda mais significativa na disponibilidade da rede entre os pontos *S* e *T*.

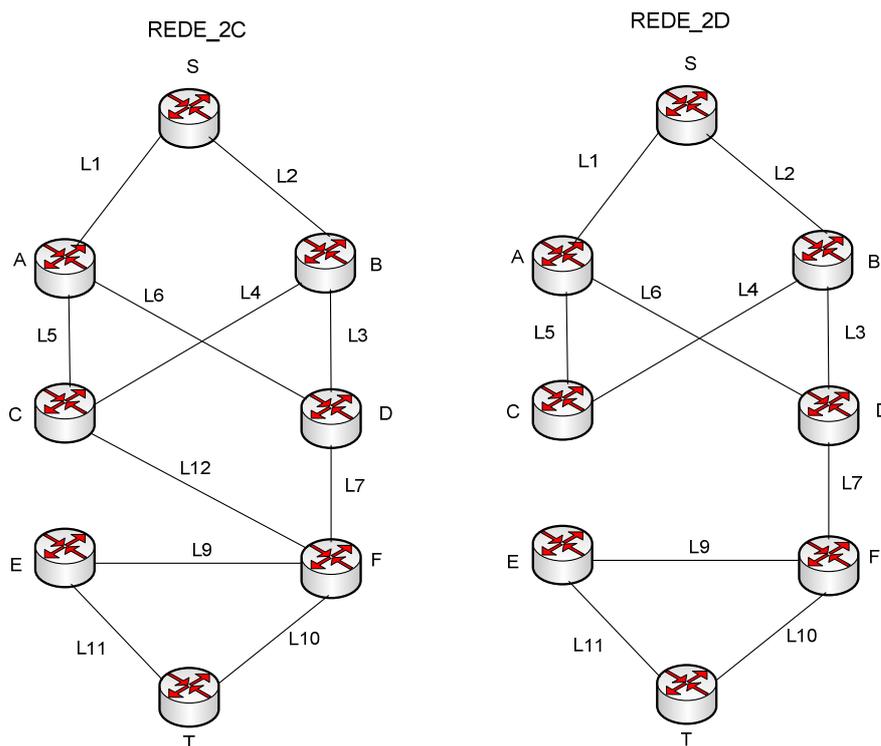


Figura 5.2.5: Rede de teste 2 – alterações C e D

O texto de entrada dos dados das redes 2C e 2D no algoritmo é mostrado na Tabela 5.3.

Tabela 5.3 - Entrada dos dados das redes 2C e 2D

Rede_2C.txt	Rede_2D.txt
S;0.99;1	S;0.99;1
A;0.98;1	A;0.98;1
B;0.98;1	B;0.98;1
C;0.97;1	C;0.97;1
D;0.99;1	D;0.99;1
E;0.98;1	E;0.98;1
F;0.97;1	F;0.97;1
T;0.97;1	T;0.97;1
link;L1;S;A;0.99;1	link;L1;S;A;0.99;1
link;L2;S;B;0.99;1	link;L2;S;B;0.99;1
link;L3;B;D;0.99;1	link;L3;B;D;0.99;1
link;L4;B;C;0.99;1	link;L4;B;C;0.99;1
link;L5;A;C;0.99;1	link;L5;A;C;0.99;1
link;L6;A;D;0.99;1	link;L6;A;D;0.99;1
link;L7;D;F;0.99;1	link;L7;D;F;0.99;1
link;L9;E;F;0.99;1	link;L9;E;F;0.99;1
link;L10;F;T;0.99;1	link;L10;F;T;0.99;1
link;L11;E;T;0.99;1	link;L11;E;T;0.99;1
link;L12;C;F;0.99;1	

O resultado da alteração que criou a rede_2C é mostrado na Figura 5.2.6.

<p>Inicio: 13:16</p> <p>Matriz de conexoes:</p> <p>F:L10,T;L12,C;L7,D;L9,E; A:L5,C;L6,D;L1,S; S:L1,A;L2,B; T:L11,E;L10,F; D:L6,A;L7,F;L3,B; C:L5,A;L12,F;L4,B; E:L11,T;L9,F; B:L4,C;L2,S;L3,D;</p> <p>A Disponibilidade da rede entre os pontos S e T totaliza: 0.929442223649303</p> <p>Fim: 13:17</p> <p>Tempo transcorrido em segundos desde o inicio da execucao: 2.383</p>
--

Figura 5.2.6: Resultado obtido para a rede_2C

O resultado da alteração que criou a rede_2D é mostrado na figura 5.2.7.

```
Inicio: 13:17

Matriz de conexoes:

F:L10,T;L7,D;L9,E;
A:L5,C;L6,D;L1,S;
S:L1,A;L2,B;
T:L11,E;L10,F;
D:L6,A;L7,F;L3,B;
C:L5,A;L4,B;
E:L11,T;L9,F;
B:L4,C;L2,S;L3,D;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.91133300769366

Fim: 13:17

Tempo transcorrido em segundos desde o inicio da execucao: 0.861
```

Figura 5.2.7: Resultado obtido para a rede_2C

Observa-se que a retirada do enlace *L8*, *rede_2C*, deixa a rede totalmente dependente do roteador *F* para prover um caminho entre *S* e *T* e por isso há uma queda de 3 pontos percentuais na disponibilidade da mesma.

A retirada do enlace *L12*, piora a situação, já que deixa a rede dependente do enlace *L7* e do roteador *F* para prover o caminho entre *S* e *T*. A disponibilidade da rede entre os pontos definidos cai então 4 pontos percentuais em comparação com a rede original.

5.2.3 - Rede 3

A rede seguinte a ser testada é mostrada na Figura 5.3

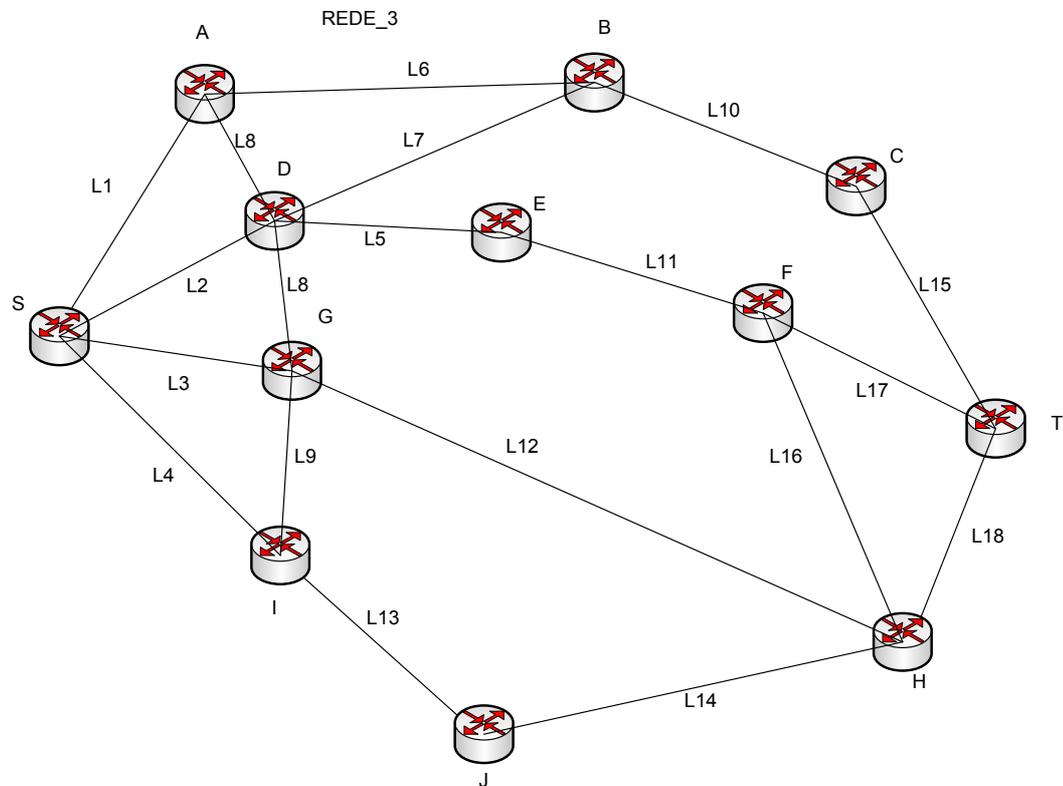


Figura 5.3: Rede de teste 3

Texto de entrada dos dados da rede no algoritmo:

Rede_3.txt

```
S;0.99;1
A;0.98;1
B;0.99;1
C;0.99;1
D;0.99;1
E;0.99;1
F;0.99;1
G;0.99;1
H;0.98;1
I;0.99;1
J;0.99;1
T;0.99;1
link;L1;S;A;0.99;1
link;L2;S;D;0.99;1
```

link;L3;S;G;0.99;1
link;L4;S;I;0.99;1
link;L5;D;E;0.99;1
link;L6;A;B;0.98;1
link;L7;B;D;0.99;1
link;L8;D;G;0.99;1
link;L9;G;I;0.99;1
link;L10;B;C;0.97;1
link;L11;E;F;0.99;1
link;L12;G;H;0.99;1
link;L13;I;J;0.99;1
link;L14;J;H;0.99;1
link;L15;C;T;0.99;1
link;L16;F;H;0.99;1
link;L17;F;T;0.99;1
link;L18;H;T;0.99;1

O resultado da disponibilidade entre os pontos S e T é mostrado na figura 5.3.1:

Inicio: 21:47

Matriz de conexoes:

F:L11,E;L17,T;L16,H;
A:L1,S;L6,B;
S:L4,I;L1,A;L2,D;L3,G;
J:L14,H;L13,I;
T:L18,H;L17,F;L15,C;
E:L5,D;L11,F;
B:L6,A;L10,C;L7,D;
H:L18,T;L14,J;L12,G;L16,F;
C:L15,T;L10,B;
D:L5,E;L2,S;L8,G;L7,B;
I:L4,S;L13,J;L9,G;
G:L8,D;L12,H;L3,S;L9,I;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.97992664984866

Fim: 01:37

Tempo transcorrido em segundos desde o inicio da execucao: 13504.889

Figura 5.3.1: Resultado do teste para a rede 3 entre os pontos S e T

A mesma rede foi avaliada para a disponibilidade entre os roteadores *B* e *J* a título de comparação e os resultados são mostrados na figura 5.3.2.

Inicio: 04:30

Matriz de conexoes:

F:L11,E;L17,T;L16,H;
A:L1,S;L6,B;
S:L4,I;L1,A;L2,D;L3,G;
J:L14,H;L13,I;
T:L18,H;L17,F;L15,C;
E:L5,D;L11,F;
B:L6,A;L10,C;L7,D;
H:L18,T;L14,J;L12,G;L16,F;
C:L15,T;L10,B;
D:L5,E;L2,S;L8,G;L7,B;
I:L4,S;L13,J;L9,G;
G:L8,D;L12,H;L3,S;L9,I;

A Disponibilidade da rede entre os pontos B e J totaliza: 0.979354639367756

Fim: 07:22

Tempo transcorrido em segundos desde o inicio da execucao: 10166.378

Figura 5.3.2: Resultado do teste para a rede 3 entre os pontos B e J

Suponha que os roteadores *E* e *F* tenham que ser desligados para uma substituição como mostra a Figura 5.3.3. Espera-se uma redução na disponibilidade da rede entre os pontos *S* e *T*.

Texto de entrada dos dados da rede no algoritmo:

Rede_3A.txt

S;0.99;1
A;0.98;1
B;0.99;1
C;0.99;1
D;0.99;1
G;0.99;1
H;0.98;1
I;0.99;1
J;0.99;1
T;0.99;1
link;L1;S;A;0.99;1
link;L2;S;D;0.99;1
link;L3;S;G;0.99;1
link;L4;S;I;0.99;1

link;L6;A;B;0.98;1
 link;L7;B;D;0.99;1
 link;L8;D;G;0.99;1
 link;L9;G;I;0.99;1
 link;L10;B;C;0.97;1
 link;L12;G;H;0.99;1
 link;L13;I;J;0.99;1
 link;L14;J;H;0.99;1
 link;L15;C;T;0.99;1
 link;L18;H;T;0.99;1

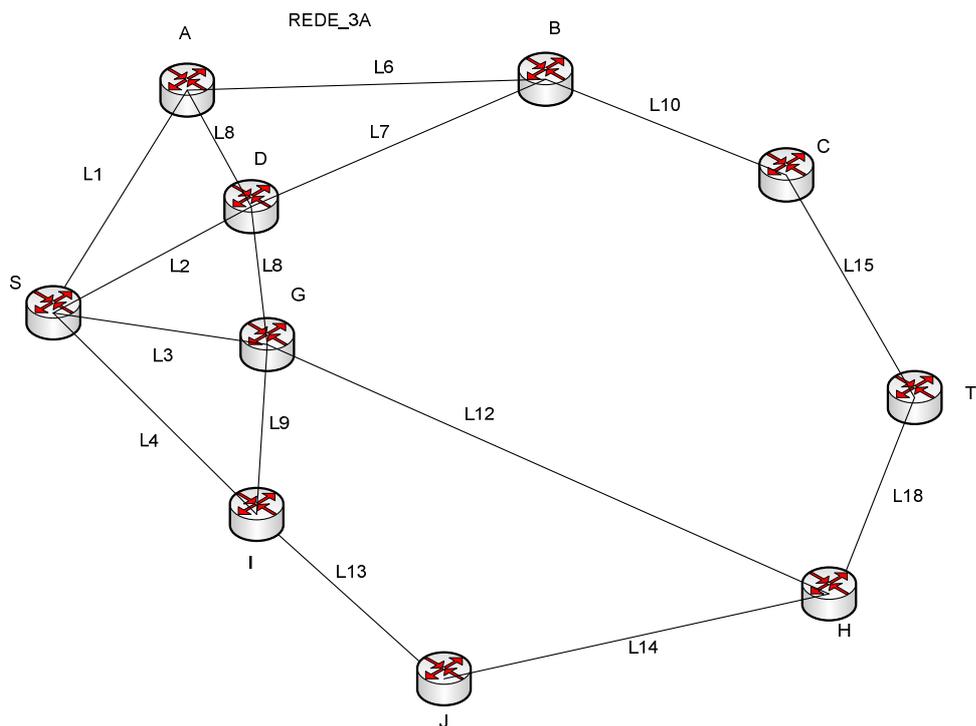


Figura 5.3.3: Rede de teste 3 sem os roteadores e e f

O resultado obtido para a disponibilidade da rede entre os roteadores S e T é mostrado na figura 5.3.4.

Inicio: 20:02

Matriz de conexoes:

H:L18,T;L12,G;L14,J;
A:L6,B;L1,S;
S:L4,I;L1,A;L2,D;L3,G;
J:L14,H;L13,I;
T:L18,H;L15,C;
D:L7,B;L2,S;L8,G;
C:L10,B;L15,T;
I:L4,S;L13,J;L9,G;
G:L12,H;L3,S;L8,D;L9,I;
B:L10,C;L6,A;L7,D;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.959493822884796

Fim: 20:05

Tempo transcorrido em segundos desde o inicio da execucao: 146.761

Figura 5.3.4: Resultado do teste para a alteraao da rede 3 entre os pontos S e T

E observa-se, na Figura 5.3.5, tambem o resultado da alteraao considerando a disponibilidade entre os roteadores B e J.

Inicio: 20:06

Matriz de conexoes:

H:L18,T;L12,G;L14,J;
A:L6,B;L1,S;
S:L4,I;L1,A;L2,D;L3,G;
J:L14,H;L13,I;
T:L18,H;L15,C;
D:L7,B;L2,S;L8,G;
C:L10,B;L15,T;
I:L4,S;L13,J;L9,G;
G:L12,H;L3,S;L8,D;L9,I;
B:L10,C;L6,A;L7,D;

A Disponibilidade da rede entre os pontos B e J totaliza: 0.978763527074638

Fim: 20:08

Tempo transcorrido em segundos desde o inicio da execucao: 118.74

Figura 5.3.5: Resultado do teste para a alteraao da rede 3 entre os pontos B e J

Como esperado a retirada dos roteadores E e F da rede, junto com seus enlaces, causou uma queda de 2 pontos percentuais na disponibilidade da rede entre os pontos S e T, mas entre os pontos B e J, a queda foi de apenas 0,1 ponto percentual.

Com este teste percebe-se que o efeito da retirada de roteadores na rede é mais pronunciado quando estes estão no caminho entre os pontos sob análise.

5.2.4 - Rede 4

A rede seguinte a ser testada é mostrada na figura 5.4

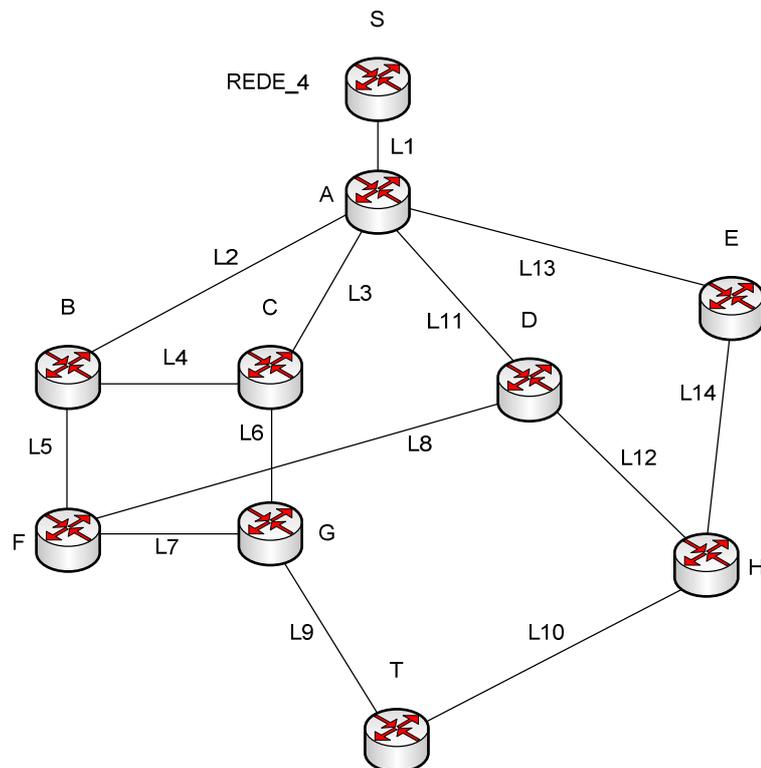


Figura 5.4: Rede de teste 4

Texto de entrada dos dados da rede no algoritmo:

Rede_4.txt

```
S;0.99;1  
A;0.98;1  
B;0.98;1  
C;0.97;1  
D;0.99;1  
E;0.98;1
```

F;0.97;1
G;0.97;1
H;0.96;1
T;0.97;1
link;L1;S;A;0.99;1
link;L2;A;B;0.99;1
link;L3;A;C;0.99;1
link;L4;B;C;0.99;1
link;L5;B;F;0.99;1
link;L6;C;G;0.99;1
link;L7;F;G;0.99;1
link;L8;D;F;0.99;1
link;L9;G;T;0.99;1
link;L10;H;T;0.99;1
link;L11;A;D;0.99;1
link;L12;D;H;0.99;1
link;L13;A;E;0.99;1
link;L14;E;H;0.99;1

O resultado obtido para a disponibilidade da rede entre os roteadores S e T é mostrado na figura 5.4.1.

Inicio: 14:47

Matriz de conexoes:

F:L5,B;L7,G;L8,D;
A:L11,D;L1,S;L2,B;L13,E;L3,C;
S:L1,A;
T:L10,H;L9,G;
E:L14,H;L13,A;
B:L5,F;L4,C;L2,A;
H:L10,T;L12,D;L14,E;
C:L6,G;L4,B;L3,A;
D:L11,A;L12,H;L8,F;
G:L6,C;L7,F;L9,T;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.929362480999582

Fim: 14:49

Tempo transcorrido em segundos desde o inicio da execucao: 80.856

Figura 5.4.1: Resultado do teste para a rede 4 entre os pontos S e T

Observa-se que uma falha no roteador A inviabiliza a comunicação entre S e T. Os profissionais de planejamento de rede decidem então incluir um novo enlace. Mas qual será a melhor opção, um enlace entre S e B ou um enlace entre S e E?

Ambas as possibilidades são mostradas na figura 5.4.2

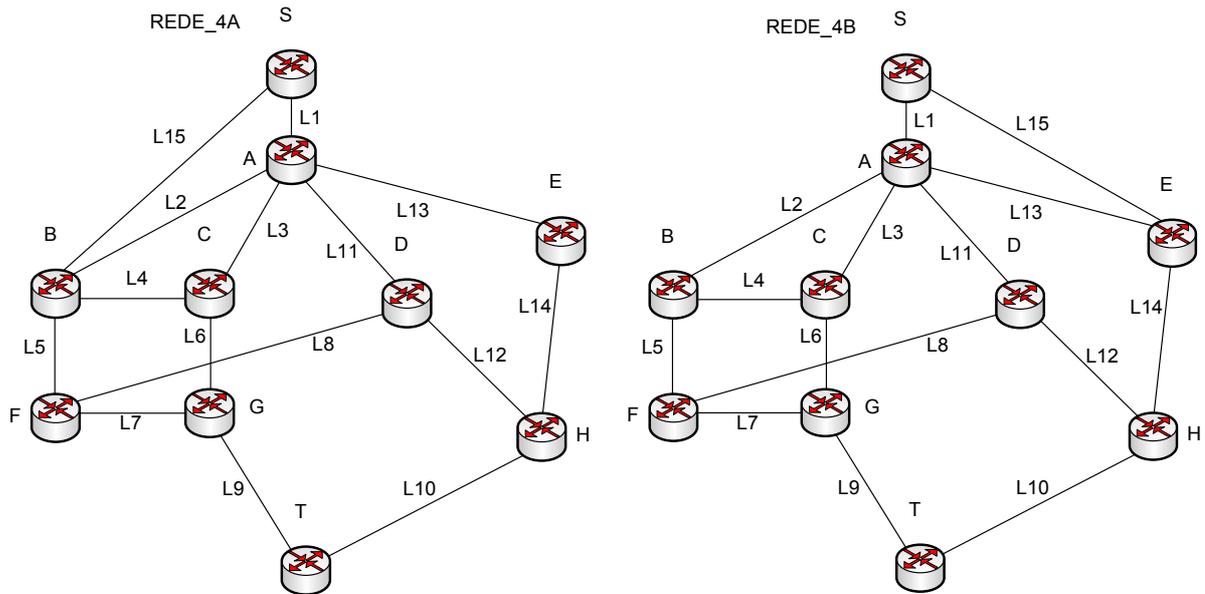


Figura 5.4.2: Rede de teste 4 com novos enlaces

Ambas as possibilidades são simuladas e comparam-se os resultados obtidos.

A entrada de dados no algoritmo é mostrada na tabela 5.4:

Tabela 5.4 - Entrada dos dados para as alterações da rede 4.

Rede_4A.txt	Rede_4B.txt
S;0.99;1	S;0.99;1
A;0.98;1	A;0.98;1
B;0.98;1	B;0.98;1
C;0.97;1	C;0.97;1
D;0.99;1	D;0.99;1
E;0.98;1	E;0.98;1
F;0.97;1	F;0.97;1
G;0.97;1	G;0.97;1
H;0.96;1	H;0.96;1
T;0.97;1	T;0.97;1
link;L1;S;A;0.99;1	link;L1;S;A;0.99;1
link;L2;A;B;0.99;1	link;L2;A;B;0.99;1
link;L3;A;C;0.99;1	link;L3;A;C;0.99;1
link;L4;B;C;0.99;1	link;L4;B;C;0.99;1

link;L5;B;F;0.99;1	link;L5;B;F;0.99;1
link;L6;C;G;0.99;1	link;L6;C;G;0.99;1
link;L7;F;G;0.99;1	link;L7;F;G;0.99;1
link;L8;D;F;0.99;1	link;L8;D;F;0.99;1
link;L9;G;T;0.99;1	link;L9;G;T;0.99;1
link;L10;H;T;0.99;1	link;L10;H;T;0.99;1
link;L11;A;D;0.99;1	link;L11;A;D;0.99;1
link;L12;D;H;0.99;1	link;L12;D;H;0.99;1
link;L13;A;E;0.99;1	link;L13;A;E;0.99;1
link;L14;E;H;0.99;1	link;L14;E;H;0.99;1
link;L15;S;B;0.99;1	link;L15;S;E;0.99;1

E foram obtidos os resultados mostrados na Figura 5.4.3 para a inclusão do novo enlace entre *S* e *B* e na Figura 5.4.4 para a inclusão do novo enlace entre os pontos *S* e *E*.

Início: 11:48

Matriz de conexoes:

F:L5,B;L7,G;L8,D;
A:L11,D;L1,S;L2,B;L13,E;L3,C;
S:L1,A;L15,B;
T:L10,H;L9,G;
E:L14,H;L13,A;
B:L5,F;L4,C;L2,A;L15,S;
H:L10,T;L12,D;L14,E;
C:L6,G;L4,B;L3,A;
D:L11,A;L12,H;L8,F;
G:L6,C;L7,F;L9,T;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.956737468915672

Fim: 11:52

Tempo transcorrido em segundos desde o inicio da execucao: 255.547

Figura 5.4.3: Resultado do teste para a inclusão do novo enlace entre os pontos S e B

Inicio: 12:04

Matriz de conexoes:

F:L5,B;L7,G;L8,D;
A:L11,D;L1,S;L2,B;L13,E;L3,C;
S:L1,A;L15,E;
T:L10,H;L9,G;
E:L14,H;L13,A;L15,S;
B:L5,F;L4,C;L2,A;
H:L10,T;L12,D;L14,E;
C:L6,G;L4,B;L3,A;
D:L11,A;L12,H;L8,F;
G:L6,C;L7,F;L9,T;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.956542825532017

Fim: 12:08

Tempo transcorrido em segundos desde o inicio da execucao: 250.37

Figura 5.4.4: Resultado do teste para a inclusão do novo enlace entre os pontos S e E

Com esses resultados observa-se que com o acréscimo de um novo enlace na rede para o roteador S, a disponibilidade da rede entre S e T sobe 3 pontos percentuais, mas não há uma diferença significativa entre interligar este enlace ao roteador B ou ao roteador E.

5.2.5 - Rede 5

A rede seguinte a ser testada é mostrada na figura 5.5, abaixo. Considera-se primeiro a rede em sua forma original, depois acrescenta-se o roteador C e enlaces para os roteadores A, D, E e F. Depois acrescenta-se o roteador H e enlaces para os roteadores E, F, G e T

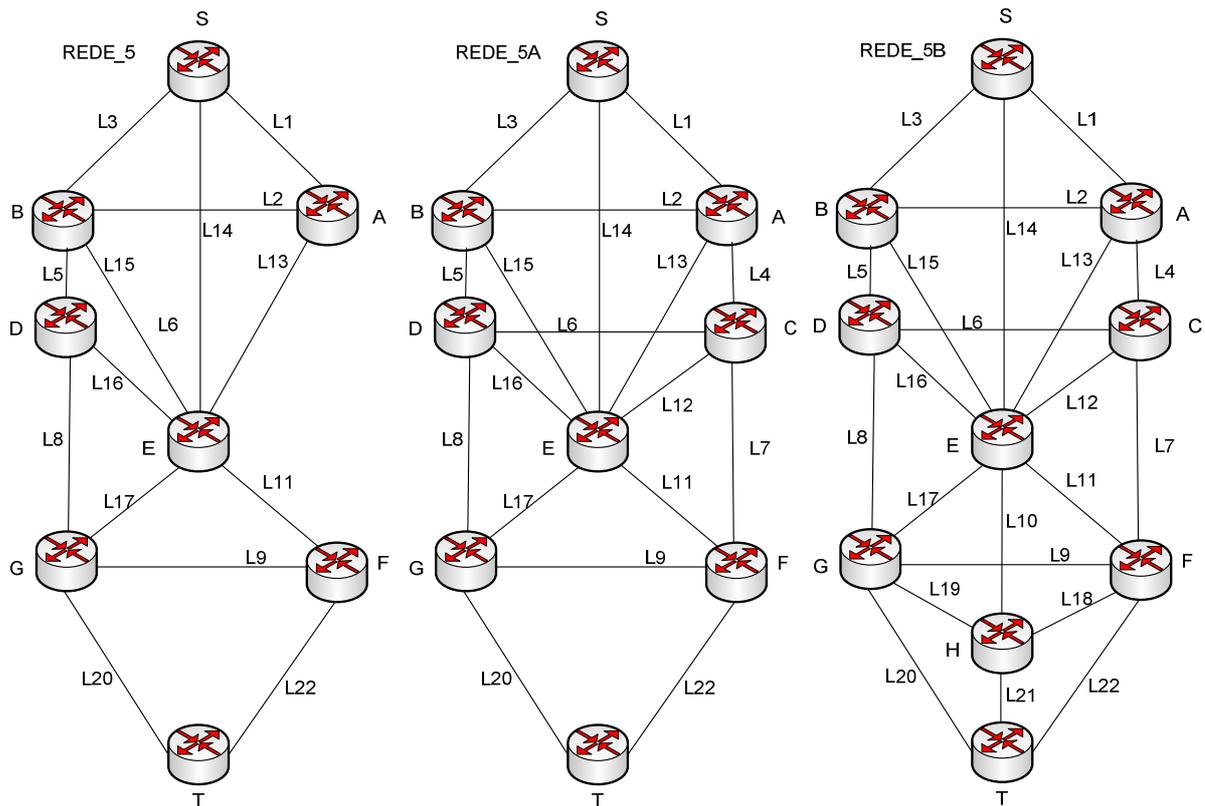


Figura 5.5: Rede de teste 5 e variações

A entrada de dados no algoritmo para a rede e as variações A e B é mostrada na tabela 5.4.

Tabela 5.5 - Entrada dos dados para a rede 5 e variações

Rede_5.txt	Rede_5A.txt	Rede_5B.txt
S;0.99;1	S;0.99;1	S;0.99;1
A;0.98;1	A;0.98;1	A;0.98;1
B;0.99;1	B;0.99;1	B;0.99;1
D;0.97;1	C;0.98;1	C;0.98;1
E;0.98;1	D;0.97;1	D;0.97;1
F;0.98;1	E;0.98;1	E;0.98;1
G;0.99;1	F;0.98;1	F;0.98;1
T;0.98;1	G;0.99;1	G;0.99;1
link;L1;S;A;0.99;1	T;0.98;1	H;0.99;1
link;L2;A;B;0.99;1	link;L1;S;A;0.99;1	T;0.98;1
link;L3;S;B;0.99;1	link;L2;A;B;0.99;1	link;L1;S;A;0.99;1
link;L5;B;D;0.99;1	link;L3;S;B;0.99;1	link;L2;A;B;0.99;1
link;L8;D;G;0.99;1	link;L4;A;C;0.99;1	link;L3;S;B;0.99;1
link;L9;F;G;0.99;1	link;L5;B;D;0.99;1	link;L4;A;C;0.99;1
link;L11;E;F;0.99;1	link;L6;C;D;0.99;1	link;L5;B;D;0.99;1
link;L13;E;E;0.99;1	link;L7;C;F;0.99;1	link;L6;C;D;0.99;1
link;L14;S;E;0.99;1	link;L8;D;G;0.99;1	link;L7;C;F;0.99;1
link;L15;B;E;0.99;1	link;L9;F;G;0.99;1	link;L8;D;G;0.99;1
link;L16;D;E;0.99;1	link;L11;E;F;0.99;1	link;L9;F;G;0.99;1

link;L17;E;G;0.99;1 link;L20;G;T;0.99;1 link;L22;F;T;0.99;1	link;L12;C;E;0.99;1 link;L13;E;E;0.99;1 link;L14;S;E;0.99;1 link;L15;B;E;0.99;1 link;L16;D;E;0.99;1 link;L17;E;G;0.99;1 link;L20;G;T;0.99;1 link;L22;F;T;0.99;1	link;L10;E;H;0.99;1 link;L11;E;F;0.99;1 link;L12;C;E;0.99;1 link;L13;E;E;0.99;1 link;L14;S;E;0.99;1 link;L15;B;E;0.99;1 link;L16;D;E;0.99;1 link;L17;E;G;0.99;1 link;L18;F;H;0.99;1 link;L19;G;H;0.99;1 link;L20;G;T;0.99;1 link;L21;H;T;0.99;1 link;L22;F;T;0.99;1
---	--	---

E foram obtidos os resultados mostrados na Figura 5.5.1 para a rede original.

<p>Inicio: 14:38</p> <p>Matriz de conexoes:</p> <p>F:L11,E;L9,G;L22,T; A:L1,S;L2,B; S:L1,A;L14,E;L3,B; T:L20,G;L22,F; D:L5,B;L16,E;L8,G; G:L20,T;L17,E;L8,D;L9,F; E:L11,F;L17,G;L16,D;L14,S;L13,E;L15,B; B:L5,D;L2,A;L3,S;L15,E;</p> <p>A Disponibilidade da rede entre os pontos S e T totaliza: 0.968089629021296</p> <p>Fim: 14:38</p> <p>Tempo transcorrido em segundos desde o inicio da execucao: 41.209</p>

Figura 5.5.1: Resultado do teste para a rede 5.

Então acrescentou-se o roteador *C* e os enlaces para os roteadores *A*, *D*, *E* e *F*. E o resultado obtido é mostrado na figura 5.5.2.

```
Inicio: 14:39

Matriz de conexoes:

F:L11,E;L7,C;L9,G;L22,T;
A:L4,C;L1,S;L2,B;
S:L1,A;L14,E;L3,B;
T:L20,G;L22,F;
D:L5,B;L8,G;L6,C;L16,E;
C:L4,A;L6,D;L12,E;L7,F;
G:L17,E;L8,D;L20,T;L9,F;
E:L11,F;L17,G;L14,S;L15,B;L12,C;L16,D;L13,E;
B:L5,D;L2,A;L15,E;L3,S;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.969513459154703

Fim: 15:09

Tempo transcorrido em segundos desde o inicio da execucao: 1765.348
```

Figura 5.5.2: Resultado do teste para a rede 5 com a primeira alteração.

Observa-se que a alteração causou apenas um aumento de 0.1 pontos percentuais na disponibilidade da rede, uma vez que a mesma já possuía caminhos redundantes entre *S* e *T*.

Com a inclusão do roteador *H* e enlaces para os roteadores *E*, *F*, *G* e *T* foram obtidos os resultados mostrados na figura 5.5.3 e, conforme esperado, o aumento da disponibilidade não foi significativo, apenas um aumento de 0,01 pontos percentuais.

Observando o tempo de execução do arquivo, percebe-se que uma rede da magnitude da rede final obtida, levou o tempo de execução de 41 segundos na rede original para 21 horas, 32 minutos e 50 segundos na rede completa. Estes valores foram obtidos em uma estação de trabalho do porte citado na metodologia e, durante a execução do algoritmo, a estação permaneceu em uso com vários aplicativos abertos. Não houve redução da capacidade de trabalho da estação.

Início: 20:22

Matriz de conexões:

F:L18,H;L11,E;L7,C;L9,G;L22,T;
A:L4,C;L1,S;L2,B;
S:L1,A;L14,E;L3,B;
T:L20,G;L21,H;L22,F;
E:L11,F;L17,G;L14,S;L15,B;L10,H;L12,C;L16,D;L13,E;
B:L5,D;L2,A;L15,E;L3,S;
H:L18,F;L19,G;L10,E;L21,T;
C:L4,A;L6,D;L12,E;L7,F;
D:L5,B;L8,G;L6,C;L16,E;
G:L19,H;L17,E;L8,D;L20,T;L9,F;

A Disponibilidade da rede entre os pontos S e T totaliza: 0.970084781909714

Fim: 18:18

Tempo transcorrido em segundos desde o início da execução: 77570.631

Figura 5.5.3: Resultado do teste para a rede 5 com a primeira alteração.

5.2.6 - Rede 6

A última rede a ser testada é mostrada na figura 5.6. Com ela será analisado o comportamento do algoritmo para várias duplas de pontos sob análise. O objetivo deste teste foi mostrar que o tempo de execução do algoritmo não depende dos pontos escolhidos para análise, ou seja, para uma mesma rede, a análise pode ser feita entre quaisquer dois pontos sem variação significativa no tempo de execução.

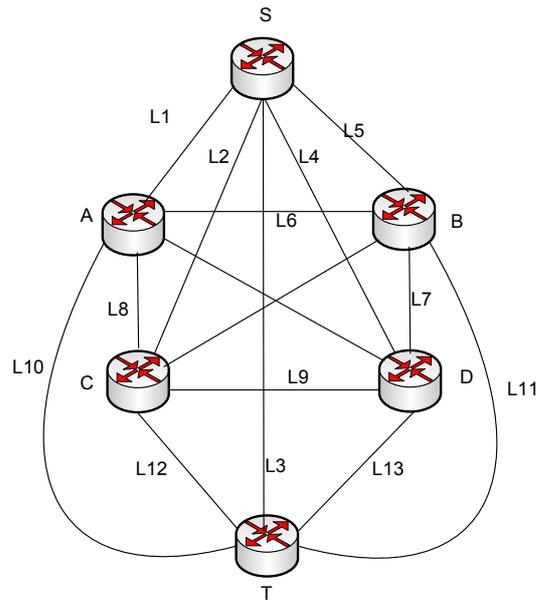


Figura 5.6: Rede de teste 6

Entrada dos dados no algoritmo:

Rede_6.txt

```

S;0.99;1
A;0.98;1
B;0.98;1
C;0.97;1
D;0.99;1
T;0.97;1
link;L1;S;A;0.99;1
link;L2;S;C;0.99;1
link;L3;S;T;0.99;1
link;L4;S;D;0.99;1
link;L5;S;B;0.99;1
link;L6;A;B;0.98;1
link;L7;B;D;0.99;1
link;L8;A;C;0.99;1
link;L9;C;D;0.99;1
link;L10;A;T;0.99;1
link;L11;B;T;0.97;1
link;L12;C;T;0.99;1
link;L13;D;T;0.99;1
  
```

Começando com a análise normal de conectividade entre os pontos S e T, foi avaliada a disponibilidade entre os pontos A e D; S e C e depois B e T. Os resultados obtidos e seus respectivos tempos de execução são mostrados na tabela

Tabela 5.6 - Resultados para a rede 6

Pontos sob análise	Disponibilidade	Tempo de execução (s)
S e T	0.960299981355456	9.002
A e D	0.970197976110835	9.113
S e C	0.960299765250431	9.353
B e T	0.950599537034541	8.672

O tempo médio de execução foi de 9.035 segundos com desvio padrão de 0,282 segundos.

5.2.7 - Análise de desempenho do algoritmo

A tabela 5.1 abaixo mostra a relação entre o tempo de execução do algoritmo em segundos em função dos parâmetros da rede.

Tabela 5.7 - Tempo de Execução do Algoritmo

	routers	links	routers + enlaces	routers X enlaces	tempo de execução
rede_1	5	5	10	25	0,06
rede_1A	5	6	11	30	0,03
rede_1B	5	6	11	30	0,04
rede_2	8	11	19	88	2,723
rede_2A	8	10	18	80	1,412
rede_2B	8	12	20	96	6,499
rede_2C	8	11	19	88	2,383
rede_2D	8	10	18	80	0,861
rede_3	12	18	30	216	13540,889
rede_3 BJ	12	18	30	216	10166,378
rede_3A	10	14	24	140	146,761
rede_3A BJ	10	14	24	140	118,74
rede_4	10	14	24	140	80,856
rede_4A	10	15	25	150	255,547
rede_4B	10	15	25	150	250,37
rede_5	8	14	22	112	41,209
rede_5A	9	18	27	162	1765,348
rede_5B	10	22	32	220	77570,631
rede_6 ST	6	13	19	78	9,002
rede_6 AD	6	13	19	78	9,113
rede_6 SC	6	13	19	78	9,353
rede_6 BT	6	13	19	78	8,672

Observa-se que o tempo de execução do algoritmo, executado em uma estação de trabalho, aumenta para várias horas quando ele é executado em grandes redes, como pode ser visto no gráfico abaixo, figura 5.7:

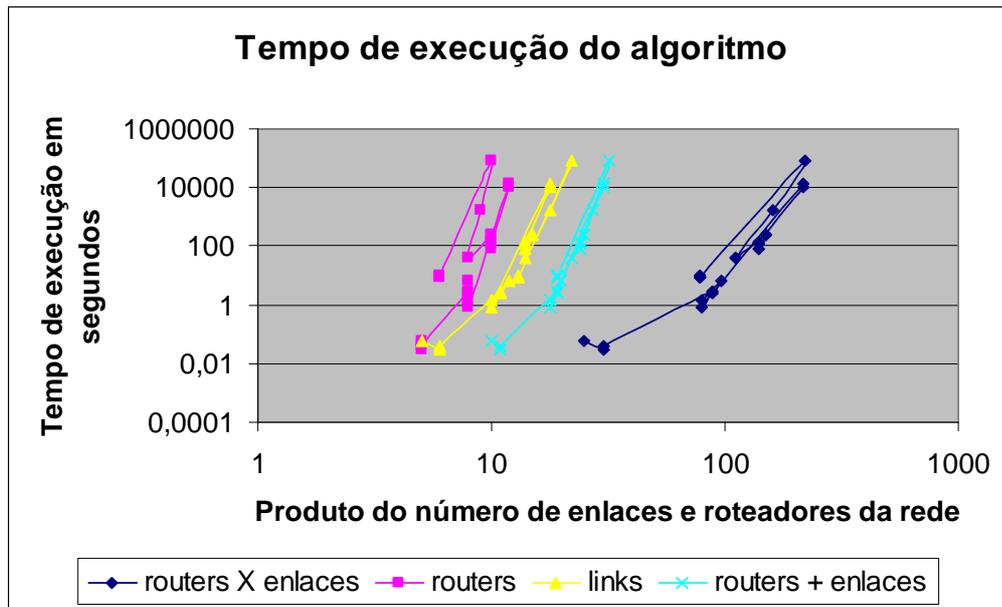


Figura 5.7: Gráfico do tempo de execução do algoritmo

Um dos métodos para a análise da complexidade de um algoritmo é a notação “.O Grande”.

Definição: uma função $g(N)$ diz-se ser $O(f(N))$ se existem constantes c_0 e N_0 tais que $g(N) < c_0 \times f(N)$ para qualquer $N > N_0$ [FREITAS, 2005].

Seja: N = número de roteadores

M = número de enlaces.

No caso do algoritmo presente sua complexidade é dada por:

$$g(N) = O(2^N \times 2^M) = O(2^{N+M}) \quad (5.1)$$

Pois o algoritmo testa todas as combinações possíveis de situações de roteadores e enlaces.

Logo a curva de complexidade do algoritmo pode ser vista na Figura 5.8 abaixo:

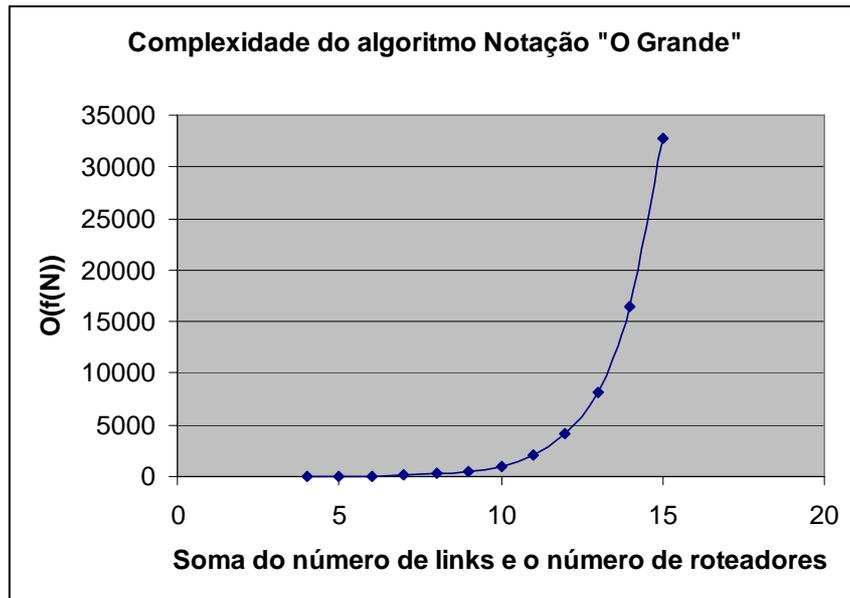


Figura 5.8: Gráfico de complexidade do algoritmo – notação “O Grande”

6 - CONCLUSÕES E RECOMENDAÇÕES

6.1 - CONCLUSÕES GERAIS

Após implementar o algoritmo e testá-lo em várias redes, pode-se chegar à conclusão de que, de acordo com as hipóteses apresentadas no capítulo 3, a hipótese 3 é que melhor exprime a situação:

A modelagem por decomposição de espaço de estados é adequada para cálculo de disponibilidade em redes de computadores, e a implementação pelo algoritmo que é o cerne deste trabalho provou ser adequada. Entretanto, para ser utilizado em estações de trabalho ele deve ser aplicado apenas para redes de pequeno porte. Em computadores de maior capacidade, ele pode ser executado para qualquer tamanho de rede. Para que ele possa ser executado em estações de trabalho para redes maiores sugerem-se as seguintes modificações no algoritmo:

1 – Acrescentar uma rotina capaz de aprender a reconhecer enlaces e roteadores críticos, de forma que, após a percepção de que a falha nestes equipamentos inviabiliza a comunicação entre os pontos sob análise, o algoritmo não mais necessite testar a conectividade entre os dois pontos caso algum deles esteja em falha, eliminando iterações desnecessárias.

2 – Buscar a implementação dos grafos através da estrutura de ponteiros dinâmicos e não de matriz de conexão conforme pode ser observada na biblioteca GTL (Graph Template Library).

6.2 - RECOMENDAÇÕES

Como recomendações para trabalhos futuros sugerem-se:

1. Implementar as rotinas de otimização citadas nas conclusões acima.
2. Implementar o algoritmo de forma que ele possa ser executado via internet através de uma *webpage*. Utilizando Java script será possível inclusive que o operador possa desenhar sua rede online e o algoritmo automaticamente criaria o arquivo de entrada de dados.
3. Alterar o algoritmo de forma a considerar parâmetros como custo e velocidade dos enlaces na análise da disponibilidade.
4. Alterar o algoritmo de forma a levar em consideração o caminho mínimo entre dois pontos dados.
5. Alterar o algoritmo de forma a levar em consideração protocolos de roteamento.
6. Alterar o algoritmo de forma a considerar enlaces unidirecionais.

7 - REFERÊNCIAS BIBLIOGRÁFICAS

- ARAUJO, Everton C. Algoritmos – Fundamento e Prática. Visual Books, 2007.
- ASHLOCK , Daniel. Evolutionary Computation for Modeling and Optimization. Springer Science+Business Media, Inc. October 2005
- BALL, Michael O.; COLBOURN, Charles J.; PROVAN, J. Scott. Network Reliability. Technical Research Report. College Park, University of Mariland. June 1992.
- BERTSEKAS, Dimitri; GALLAGER, Robert G. Data Networks, Second Edition. Prentice-Hall, Inc. 1992.
- BLUETOOTH, The Official Bluetooth technology Info Site. Disponível em:
<<http://www.bluetooth.com>> Acesso em Janeiro de 2007.
- BROWN, Martin. Active Perl – Guia do Programador. Ciência Moderna, 2001.
- DENGIZ, Berna; ALTIPARMAK, Fulya; Smith, Alice E. Efficient Optimization of All-Terminal Reliable Networks, Using as Evolutionary Approach. IEEE Transactions on Reliability, Vol. 47, No. 1, March 1997.
- FARIA, João pascal. Análise de Complexidade de Algoritmos. FEUP/LEEC, 2001/2002. Disponível em:<http://paginas.fe.up.pt/~jpf/teach/AED/slides3.pdf> Acesso em Julho de 2008.
- FREITAS, Ana Teresa. Algoritmos e complexidade. INESC-ID/IT, 2005. Disponível em:
<<http://kdbio.inesc-id.pt/~atf/GFB/PDFs/Analise.pdf>> Acesso em Julho de 2008.
- FOGHLÚ, Míchaél. Perl 5 – Guia de referência rápida, Editora Campus, 1997.
- GTL (1999). G<T,L> Research. Disponível em: < <http://www.infosun.fim.uni-passau.de/GTL>> Acesso em Janeiro de 2008.
- GOLDSCHMIDT, O. ISP Backbone Traffic Inference Methods to Support Traffic Engineering . In Internet Statistics and Metrics Analysis (ISMA) Workshop, San Diego, CA, December 2000. Disponível em:
<http://www.caida.org/workshops/isma/0012/talks/olivier/sld001.htm> Acesso em Dezembro de 2007.
- HUI, Kin-Ping. Network Reliability Estimation, University of Adelaide, 2005.

- IDC. Disponível em: <<http://www.idc.com>> Acesso em Janeiro de 2007.
- ITIL – IT Infrastructure Library. 2003
- JAN, Rong-Hong. Design of Reliable Networks. National Chiao Tung University.
- KERSHENBAUM, Aaron. Telecommunications Network Design Algorithms, McGraw-Hill International Editions, 1993.
- OGGERINO, Chris. High Availability Network Fundamentals, Cisco Press, 2001.
- OLIFER, Natalia; OLIFER, Victor; UCHOA, Elvira Maria Antunes. Redes de Computadores, LTC, 2008.
- PERL. The Source for Perl. Disponível em: <<http://www.perl.com>> Acesso em Fevereiro de 2008.
- QUIGLEY, Ellie. Perl by Example, 2ª edição, Prentice Hall PTR, 1998
- SAHNI, Sartaj. Data Structures, Algorithms and Applications in C++. Silicon Press, 2004
- SNIA. Advancing Storage and Information Technology. Disponível em: <http://www.snia.org> Acesso em Junho de 2008.
- TANNENBAUM, Andrew S. Redes de computadores, 4ª edição, Editora Campus, 2003.
- WIRTH, Niklaus, Algoritmos e estruturas de dados. LTC, 1989
- YOO, Y.B., A Comparison of Algorithms for Terminal Pair Reliability. IEEE Transactions on Reliability, 1998.
- ZAGO. Scripts – FAQ, Tutoriais, Documentações e indicações. Disponível em: <<http://www.zago.eti.br>> Acessado em junho de 2008.
- ZOMORODIAN, Afra J. Topology for Computing. Cambridge Monographs on Applied and Computational Mathematics. January, 2005.

ANEXOS

A – ALGORITMO PARA CÁLCULO DE DISPONIBILIDADE EM REDES IP BASEADO NA DECOMPOSIÇÃO DE ESPAÇO DE ESTADOS

```

# algoritmo que lê os dados dos roteadores e enlaces e transforma na matriz
# de conexões dos roteadores. Esta matriz possui o seguinte formato:
# { roteador link_A roteador_A link_B roteador_B }
# { roteador_A link_A roteador link_C roteador_C }
# onde o link_A é o link que leva do roteador ao roteador_A e o link_B é o
# link que leva do roteador ao roteador_B
$| = 1;
#leitura dos dados
$nome_arquivo = @ARGV[0];
$ponta_A = @ARGV[1];
$ponta_B = @ARGV[2];
unlink "saida_$nome_arquivo";
open(REDE, $nome_arquivo) or die "Arquivo de rede -erro: $! \n";
open(SAIDA, ">>saida_$nome_arquivo") || die "Impossivel criar: saida\n";
# Le informacoes do arquivo da rede
undef %status_link;
undef %status_router;
while (<REDE>) {
    next if /^\/s*!\/;
    next if /^\/\n/;
    chomp;
    if (/^[Ll][Ii][Nn][Kk]/) {
        (undef,$link, $origem, $destino, $disponibilidade,$estado) = split(/;/);
        $roteador_origem{$link} = $origem;
        $roteador_destino{$link} = $destino;
        $disponibilidade{$link} = $disponibilidade;
        $num_links++;
        $status_link{$link} = $estado;
    } else {
        ($roteador, $disponibilidade, $estado) = split(/;/);
        $disponibilidade{$roteador} = $disponibilidade;
        $num_routers++;
        $status_router{$roteador} = $estado;
    }
}
$data = `time /T`;
print SAIDA "Inicio: ", $data, "\n";
close REDE;
print "Li o arquivo \n";
# criação da matrix de conexões
print "\nMontando a matriz de conexoes:\n";
print SAIDA "Matriz de conexoes:\n";
print "\nExecutando aguarde ... \n";
foreach $roteador (keys %status_router) {
    print SAIDA "\n", $roteador, ":";
    foreach $link (keys %status_link) {
        if ($roteador_origem{$link} eq $roteador) {
            $matriz{$roteador}{$link} = $roteador_destino{$link};
            print SAIDA $link, ",", $matriz{$roteador}{$link}, ";";
        } elsif ($roteador_destino{$link} eq $roteador) {
            $matriz{$roteador}{$link} = $roteador_origem{$link};
            print SAIDA $link, ",", $matriz{$roteador}{$link}, ";";
        }
    }
}

```

```

    }
}
$soma_disp = 0;
undef $link;
undef $roteador;
$total_routers = (2**$num_routers) -1;
$total_links = (2**$num_links) -1;
#print "\n\nExecutando, aguarde ";
#print "\n\nPercentual a executar:\n";
SITUA_ROUTER: for ($igl=$total_routers; $igl>=1; $igl--) {
    $percentual = 100 * $igl/$total_routers;
    print "\r", sprintf("%.3f", $percentual), "%  ";
    if ($igl < 4294967296){
        @binstr = dec2bin($igl);
    } else {
        @binstr = dec2bin_gd($igl);
    }
    foreach $node (keys %status_router){
        $status_router{$node} = chop (@binstr);
    }
    if ($status_router{$ponta_A} eq '0') {
        next;
    }
    if ($status_router{$ponta_B} eq '0') {
        next;
    }
    for ($jgl=$total_links; $jgl>=1; $jgl--) {
        if ($jgl < 4294967296){
            @binstr = dec2bin($jgl);
        } else {
            @binstr = dec2bin_gd($jgl);
        }
        foreach $link (keys %status_link){
            $status_link{$link} = chop (@binstr);
        }
        $conecta = testa_conecta();
        if (($jgl == $total_links) && ($conecta == 0)) {
            next SITUA_ROUTER;
        }
        if ($conecta == 1) {
            # print SAIDA keys (%status_router), ",", values(%status_router), "\n";
            # print SAIDA keys (%status_link), ",", values(%status_link), "\n";
            $prod_disp = 1;
            foreach $node (keys %status_router){
                if ($status_router{$node} ==1){
                    $prod_disp = $prod_disp * $disponibilidade{$node};
                } else {
                    $prod_disp = $prod_disp * (1 - $disponibilidade{$node});
                }
            }
            foreach $linx (keys %status_link){
                if ($status_link{$linx} ==1){
                    $prod_disp = $prod_disp * $disponibilidade{$linx};
                } else {
                    $prod_disp = $prod_disp * (1 - $disponibilidade{$linx});
                }
            }
            $soma_disp += $prod_disp;
        }
    }
}

```

```

}
$data = `time /T`;
print SAIDA "\n\nA Disponibilidade da rede entre os pontos ", $ponta_A , " e ", $ponta_B, " totaliza: ",
$soma_disp, "\n";
print "\n\nA Disponibilidade da rede entre os pontos ", $ponta_A , " e ", $ponta_B, " totaliza: ", $soma_disp, "\n";
print SAIDA "\nFim: ", $data, "\n";
($user,undef,undef,undef) = times;
print "\nTempo transcorrido em segundos desde o inicio da execucao: ", $user, "\n";
print SAIDA "\nTempo transcorrido em segundos desde o inicio da execucao: ", $user, "\n";
close SAIDA;

sub dec2bin {
    my @str = unpack("B32", pack("N", shift));
    return @str;
}

sub dec2bin_gd {
    my $str = "";
    my $num = shift;
    do {
        my $low = $num & 0xFFFFFFFF; # parte de 32 bits menos significativa do numero
        $str = unpack("B32", pack("N", $low)) . $str;
        $num = int($num/(2**32)); # part mais significativa do numero
    } while ($num);

    return $str;
}

sub testa_conecta {
    my $proximo = $ponta_A;
    my $aux = 1;
    undef @anterior;
    undef %testado;
    undef $ultimo;
    while ($proximo ne $ponta_B) {
        my $achei = 0;
        TESTA_LINK: for $link_sub (keys %{$matriz{$proximo}}) {
            if (defined($testado{$link_sub})) {
                next TESTA_LINK;
            }
            if ($achei == 0) {
                for (my $j=$aux; $j> 0; $j--) {
                    next TESTA_LINK if ($matriz{$proximo}{$link_sub} eq
$anterior[$j]);
                }
                if (($status_link{$link_sub} == 1) &&
($status_router{$matriz{$proximo}{$link_sub}} == 1)){
                    $anterior[$aux] = $proximo;
                    $testado{$link_sub} = 1;
                    $ultimo = $proximo;
                    $proximo = $matriz{$proximo}{$link_sub};
                    $achei = 1;
                    $aux ++;
                }
            }
            if (($achei == 1) && ($status_link{$link_sub} eq '1')) {
                if ($matriz{$ultimo}{$link_sub} eq $ponta_B) {
                    $proximo = $matriz{$ultimo}{$link_sub};
                    return 1;
                }
            }
        }
    }
}

```

```
        }
    }
}
if ($achei == 0){
    if (($proximo eq $ponta_A) || ($aux == 0)) {
        return 0;
    }
    $proximo = $anterior[$aux];
    $aux --;
}
if ($proximo eq $ponta_B) {
    return 1;
}
}
}
```